2024美团技术年货

CODE A BETTER LIFE

— 年度合集 —

美团 美团

序言

新春将近，一年一度的美团技术年货也如约而至！

路虽远，行则将至，梦虽遥，追则可及。2024 年，是美团技术博客走过的第 11 个年头，我们没有恢弘的叙事，只是年复年、日复日的默默坚持。截止目前，美团技术团队微信公众号累计发布了 600 多篇技术文章，感谢大家一路同行，见证我们的成长。

值蛇年春节到来之际，我们精选过去一年公众号 30 多篇技术文章和科研论文，整理制作成一本 600 多页的电子书，作为一份特别的新年礼物，献给每一位热爱技术的你。

这本电子书内容覆盖算法、工程、测试、安全、数据等多个技术领域，希望能为你的工作和学习带来一些启发与帮助。也欢迎你将这份电子书分享给更多志同道合、积极向上的同事和朋友，一起携手共进，砥砺前行。

在新的一年里，愿大家乘风破浪，勇往直前，历尽千帆，梦想终将实现！

目录

基本功 | 一文讲清多线程和多线程同步

多线程编程是现代软件开发中的一项关键技术，在多线程编程中，开发者可以将复杂的任务分解为多个独立的线程，使其并行执行，从而充分利用多核处理器的优势。然而，多线程编程也带来了挑战，例如线程同步、死锁和竞态条件等问题。本篇文章将深入探讨多线程编程的基本概念（原子操作、CAS、Lock-free、内存屏障、伪共享、乱序执行等）、常见模式和最佳实践。通过具体的代码示例，希望能够帮助大家掌握多线程编程的核心技术，并在实际开发中应用这些知识，提升软件的性能和稳定性。

1 多线程

1.1 线程的概念

十多年前，主流观点主张在可能的情况下优先选择多进程而非多线程。如今，多线程编程已经成为编程领域的事实标准。多线程技术在很大程度上改善了程序的性能和响应能力，使其能够更加高效地利用系统资源，这不仅归功于多核处理器的普及和软硬件技术的进步，还归功于开发者对多线程编程的深入理解和技术创新。

那么什么是线程呢？线程是一个执行上下文，它包含诸多状态数据：每个线程有自己的执行流、调用栈、错误码、信号掩码、私有数据。Linux 内核用任务（Task）表示一个执行流。

1.1.1 执行流

一个任务里被依次执行的指令会形成一个指令序列（IP 寄存器值的历史记录），这个

指令序列就是一个指令流，每个线程会有自己的执行流。考虑下面的代码（本文代码块为 C++）：

```
int calc(int a, int b, char op) {
  int c = 0;
  if (op == '+')
    c = a + b;
  else if (op == '-')
    c = a - b;
  else if (op == '*')
    c = a * b;
  else if (op == '/')
    c = a / b;
  else
    printf("invalid operation\n");
  return c;
}
```

calc 函数被编译成汇编指令，一行 C 代码对应一个或多个汇编指令，在一个线程里执行 calc，那么这些机器指令会被依次执行。但是，被执行的指令序列跟代码顺序可能不完全一致，代码中的分支、跳转等语句，以及编译器对指令重排、处理器乱序执行会影响指令的真正执行顺序。

1.1.2 逻辑线程 vs 硬件线程

线程可以进一步区分为逻辑线程和硬件线程。

逻辑线程

程序上的线程是一个逻辑上的概念，也叫任务、软线程、逻辑线程。线程的执行逻辑由代码描述，比如编写一个函数实现对一个整型数组的元素求和：

```
int sum(int a[], int n) {
    int x = 0;
    for (int i = 0; i < n; ++i)
        x += a[i];
    return x;
}
```

这个函数的逻辑很简单，它没有再调用其他函数（更复杂的功能逻辑可以在函数里调

用其他函数)。我们可以在一个线程里调用这个函数对某数组求和；也可以把 sum 设置为某线程的入口函数，每个线程都会有一个入口函数，线程从入口函数开始执行。sum 函数描述了逻辑，即要做什么以及怎么做，偏设计；但它没有描述物质，即没有描述这个事情由谁做，事情最终需要派发到实体去完成。

硬件线程

与逻辑线程对应的是硬件线程，这是逻辑线程被执行的物质基础。

芯片设计领域，一个硬件线程通常指为执行指令序列而配套的硬件单元，一个 CPU 可能有多个核心，然后核心还可能支持超线程，1 个核心的 2 个超线程复用一些硬件。从软件的视角来看，无须区分是真正的 Core 和超出来的 VCore，基本上可以认为是 2 个独立的执行单元，每个执行单元是一个逻辑 CPU，从软件的视角看 CPU 只需关注逻辑 CPU。一个软件线程由哪个 CPU/ 核心去执行，以及何时执行，不归应用程序员管，它由操作系统决定，操作系统中的调度系统负责此项工作。

1.2 线程、核心、函数的关系

线程入口函数是线程执行的起点，线程从入口函数开始、一个指令接着一个指令执行，中间它可能会调用其他函数，那么它的控制流就转到了被调用的函数继续执行，被调用函数里还可以继续调用其他函数，这样便形成一个函数调用链。

前面的数组求和例子，如果数组特别大，则哪怕是一个简单的循环累加也可能耗费很长的时间，可以把这个整型数组分成多个小数组，或者表示成二维数组(数组的数组)，每个线程负责一个小数组的求和，多个线程并发执行，最后再累加结果。

所以，为了提升处理速度，可以让多个线程在不同数据区段上执行相同(或相似)的计算逻辑，同样的处理逻辑可以有多个执行实例(线程)，这对应对数据拆分线程。当然，也可以为两个线程指定不同的入口函数，让各线程执行不同的计算逻辑，这对应对逻辑拆分线程。

我们用一个例子来阐述线程、核心和函数之间的关系，假设有遛狗、扫地两类工作要做：

- 遛狗就是为狗系上绳子然后牵着它在小区里溜达一圈，这句话就描述了遛狗的逻辑，即对应到函数定义，它是一个对应到设计的静态的概念。
- 每项工作，最终需要人去做，人就对应到硬件：CPU/Core/VCore，是任务被完成的物质基础。

那什么对应软件线程？ 任务拆分。

一个例子

假设现在有 2 条狗需要遛、3 个房间需要打扫。可以把遛狗拆成 2 个任务，一个任务是遛小狗，另一个任务是遛大狗；打扫房间拆分为 3 个任务，3 个房间对应 3 个任务，执行这样的拆分策略后，将会产生 2+3=5 个任务。但如果只有 2 个人，2 个人无法同时做 5 件事，让某人在某时干某事由调度系统负责。

如果张三在遛小狗，那就对应一个线程被执行，李四在扫房间 A，则表示另一个线程在执行中，可见线程是一个动态的概念。

软件线程不会一直处于执行中，原因是多方面的。上述例子是因为人手不够，所以遛大狗的任务还处于等待被执行的状态，其他的原因包括中断、抢占、条件依赖等。比如李四扫地过程中接到一个电话，他需要去处理更紧急的事情（接电话），则扫地这个事情被挂起，李四打完电话后继续扫地，则这个线程会被继续执行。

如果只有 1 个人，则上述 5 个任务依然可以被依次或交错完成，所以多线程是一个编程模型，多线程并不一定需要多 CPU 多 Core，单 CPU 单 Core 系统依然可以运行多线程程序（虽然最大化利用多 CPU 多 Core 的处理能力是多线程程序设计的一个重要目标）。1 个人无法同时做多件事，单 CPU/ 单 Core 也不可以，操作系统通过时间分片技术应对远多于 CPU/Core 数的多任务执行的挑战。也可以把有些任务只分配给某些人去完成，这对应到 CPU 亲和性和绑核。

1.3 程序、进程、线程、协程

进程和线程是操作系统领域的两个重要概念，两者既有区别又有联系。

1.3.1 可执行程序

C/C++ 源文件经过编译器（编译 + 链接）处理后，会产生可执行程序文件，不同系统有不同格式，比如 Linux 系统的 ELF 格式、Windows 系统的 EXE 格式，可执行程序文件是一个静态的概念。

1.3.2 进程是什么

可执行程序在操作系统上的一次执行对应一个进程，进程是一个动态的概念：进程是执行中的程序。同一份可执行文件执行多次，会产生多个进程，这跟一个类可以创建多个实例一样。进程是资源分配的基本单位。

1.3.3 线程是什么

一个进程内的多个线程代表着多个执行流，这些线程以并发模式独立执行。操作系统中，被调度执行的最小单位是线程而非进程。进程是通过共享存储空间对用户呈现的逻辑概念，同一进程内的多个线程共享地址空间和文件描述符，共享地址空间意味着进程的代码（函数）区域、全局变量、堆、栈都被进程内的多线程共享。

1.3.4 进程和线程的关系

先看看 linus 的论述，在 1996 年的一封邮件里，Linus 详细阐述了他对进程和线程关系的深刻洞见，他在邮件里写道：

- 把进程和线程区分为不同的实体是背着历史包袱的传统做法，没有必要做这样的区分，甚至这样的思考方式是一个主要错误。
- 进程和线程都是一回事：一个执行上下文（context of execution），简称为 COE，其状态包括：
  - CPU 状态（寄存器等）
  - MMU 状态（页映射）
  - 权限状态（uid、gid 等）
  - 各种通信状态（打开的文件、信号处理器等）
- 传统观念认为：进程和线程的主要区别是线程有 CPU 状态（可能还包括其他

最小必要状态），而其他上下文来自进程；然而，这种区分法并不正确，这是一种愚蠢的自我设限。

- Linux 内核认为根本没有所谓的进程和线程的概念，只有 COE（Linux 称之为任务），不同的 COE 可以相互共享一些状态，通过此类共享向上构建起进程和线程的概念。
- 从实现来看，Linux 下的线程目前是 LWP 实现，线程就是轻量级进程，所有的线程都当作进程来实现，因此线程和进程都是用 task_struct 来描述的。这一点通过 /proc 文件系统也能看出端倪，线程和进程拥有比较平等的地位。对于多线程来说，原本的进程称为主线程，它们在一起组成一个线程组。
- 简言之，内核不要基于进程 / 线程的概念做设计，而应该围绕 COE 的思考方式去做设计，然后，通过暴露有限的接口给用户去满足 pthreads 库的要求。

1.3.5 协程

用户态的多执行流，上下文切换成本比线程更低，微信用协程改造后台系统后，获得了更大吞吐能力和更高稳定性。如今，协程库也进了 C++ 20 新标准。

1.4 为什么需要多线程

1.4.1 什么是多线程

一个进程内多个线程并发执行的情况就叫多线程，每个线程是一个独立的执行流，多线程是一种编程模型，它与处理器无关、跟设计有关。

需要多线程的原因包括：

- **并行计算**：充分利用多核，提升整体吞吐，加快执行速度。
- **后台任务处理**：将后台线程和主线程分离，在特定场景它是不可或缺的，如：响应式用户界面、实时系统等。

我们用 2 个例子作说明。

1.4.2 通过多线程并发提升处理能力

假设你要编写一个程序，用于统计一批文本文件的单词出现次数，程序的输入是文件名列表，输出一个单词到次数的映射。

```
// 类型别名：单词到次数的映射
using word2count = std::map<std::string, unsigned int>;

// 合并“单词到次数映射列表”
word2count merge(const std::vector<word2count>& w2c_list) {/*todo*/}

// 统计一个文件里单词出现次数（单词到次数的映射）
word2count word_count_a_file(const std::string& file) {/*todo*/}

// 统计一批文本文件的单词出现次数
word2count word_count_files(const std::vector<std::string>& files) {
    std::vector<word2count> w2c_list;
    for (auto &file : files) {
        w2c_list.push_back(word_count_a_file(file));
    }
    return merge(w2c_list);
}

int main(int argc, char* argv[]) {
    std::vector<std::string> files;
    for (int i = 1; i < argc; ++i) {
        files.push_back(argv[i]);
    }
    auto w2c = word_count_files(files);
    return 0;
}
```

这是一个单线程程序，word_count_files 函数在主线程里被 main 函数调用。如果文件不多、又或者文件不大，那么运行这个程序，很快就会得到统计结果，否则，可能要等一段长的时间才能返回结果。

重新审视这个程序会发现：函数 word_count_a_file 接受一个文件名，吐出从该文件计算出的局部结果，它不依赖于其他外部数据和逻辑，可以并发执行，所以，可以为每个文件启动一个单独的线程去运行 word_count_a_file，等到所有线程都执行完，再合并得到最终结果。

实际上，为每个文件启动一个线程未必合适，因为如果有数万个小文件，那么启动数

万个线程，每个线程运行很短暂的时间，大量时间将耗费在线程创建和销毁上，一个改进的设计：

- 开启一个线程池，线程数等于 Core 数或二倍 Core 数（策略）。
- 每个工作线程尝试去文件列表（文件列表需要用锁保护起来）里取一个文件。
    - 成功，统计这个文件的单词出现次数。
    - 失败，该工作线程就退出。
- 待所有工作线程退出后，在主线程里合并结果。

这样的多线程程序能加快处理速度，前面数组求和可以采用相似的处理，如果程序运行在多 CPU 多 Core 的机器上，就能充分利用多 CPU 多 Core 硬件优势，多线程加速执行是多线程的一个显而易见的主要目的，此其一。

1.4.3 通过多线程改变程序编写方式

其二，有些场景会有阻塞的调用，如果不用多线程，那么代码不好编写。

比如某程序在执行密集计算的同时，需要监控标准输入（键盘），如果键盘有输入，那么读取输入并解析执行，但如果获取键盘输入的调用是阻塞的，而此时键盘没有输入到来，那么其他逻辑将得不到机会执行。

代码看起来会像下面这样子：

```
// 从键盘接收输入，经解释后，会构建一个 Command 对象返回
Command command = getCommandFromStdInput();
// 执行命令
command.run();
```

针对这种情况，我们通常会开启一个单独的线程去接收输入，而用另外的线程去处理其他计算逻辑，避免处理输入阻塞其他逻辑处理，这也是多线程的典型应用，它改变了程序的编写方式，此其二。

1.5 线程相关概念

1.5.1 时间分片

CPU 先执行线程 A 一段时间，然后再执行线程 B 一段时间，然后再执行线程 A 一段时间，CPU 时间被切分成短的时间片、分给不同线程执行的策略就是 CPU 时间分片。时间分片是对调度策略的一个极度简化，实际上操作系统的调度策略非常精细，要比简单的时间分片复杂的多。如果一秒钟被分成大量的非常短的时间片，比如 100 个 10 毫秒的时间片，10 毫秒对人的感官而言太短了，以致于用户觉察不到延迟，仿佛计算机被该用户的任务所独占（实际上并不是），操作系统通过进程的抽象获得了这种任务独占 CPU 的效果（另一个抽象是进程通过虚拟内存独占存储）。

1.5.2 上下文切换

把当前正在 CPU 上运行的任务迁走，并挑选一个新任务到 CPU 上执行的过程叫调度，任务调度的过程会发生上下文切换（context swap），即保存当前 CPU 上正在运行的线程状态，并恢复将要被执行的线程的状态，这项工作由操作系统完成，需要占用 CPU 时间（sys time）。

1.5.3 线程安全函数与可重入

一个进程可以有多个线程在同时运行，这些线程可能同时执行一个函数，如果多线程并发执行的结果和单线程依次执行的结果是一样的，那么就是线程安全的，反之就不是线程安全的。

不访问共享数据，共享数据包括全局变量、static local 变量、类成员变量，只操作参数、无副作用的函数是线程安全函数，线程安全函数可多线程重入。每个线程有独立的栈，而函数参数保存在寄存器或栈上，局部变量在栈上，所以只操作参数和局部变量的函数被多线程并发调用不存在数据竞争。

C 标准库有很多编程接口都是非线程安全的，比如时间操作 / 转换相关的接口：ctime()/gmtime()/localtime()，c 标准通过提供带 _r 后缀的线程安全版本，比如：

```
char* ctime_r(const time* clock, char* buf);
```

这些接口的线程安全版本，一般都需要传递一个额外的 char * buf 参数，这样的话，函数会操作这块 buf，而不是基于 static 共享数据，从而做到符合线程安全的要求。

1.5.4 线程私有数据

因为全局变量（包括模块内的 static 变量）是进程内的所有线程共享的，但有时应用程序设计中需要提供线程私有的全局变量，这个变量仅在函数被执行的线程中有效，但却可以跨多个函数被访问。

比如在程序里可能需要每个线程维护一个链表，而会使用相同的函数来操作这个链表，最简单的方法就是使用同名而不同变量地址的线程相关数据结构。这样的数据结构可以由 Posix 线程库维护，成为线程私有数据 (Thread-specific Data，或称为 TSD)。

Posix 有线程私有数据相关接口，而 C/C++ 等语言提供 thread_local 关键字，在语言层面直接提供支持。

1.5.5 阻塞和非阻塞

一个线程对应一个执行流，正常情况下，指令序列会被依次执行，计算逻辑会往前推进。但如果因为某种原因，一个线程的执行逻辑不能继续往前走，那么我们就说线程被阻塞住了。就像下班回家，但走到家门口发现没带钥匙，只能在门口徘徊，任由时间流逝，而不能进入房间。

线程阻塞的原因有很多种，比如：

- 线程因为 acquire 某个锁而被操作系统挂起，如果 acquire 睡眠锁失败，线程会让出 CPU，操作系统会调度另一个可运行线程到该 CPU 上执行，被调度走的线程会被加入等待队列，进入睡眠状态。
- 线程调用了某个阻塞系统调用而等待，比如从没有数据到来的套接字上读数据，从空的消息队列里读消息。

- 线程在循环里紧凑的执行测试 & 设置指令并一直没有成功，虽然线程还在 CPU 上执行，但它只是忙等（相当于白白浪费 CPU），后面的指令没法执行，逻辑同样无法推进。

如果某个系统调用或者编程接口有可能导致线程阻塞，那么便被称之为阻塞系统调用；与之对应的是非阻塞调用，调用非阻塞的函数不会陷入阻塞，如果请求的资源不能得到满足，它会立即返回并通过返回值或错误码报告原因，调用的地方可以选择重试或者返回。

2 多线程同步

前面讲了多线程相关的基础知识，现在进入第二个话题，多线程同步。

2.1 什么是多线程同步

同一进程内的多个线程会共享数据，对共享数据的并发访问会出现 Race Condition，这个词的官方翻译是竞争条件，但 condition 翻译成条件令人困惑，特别是对初学者而言，它不够清晰明了，翻译软件显示 condition 有状况、状态的含义，可能翻译成竞争状况更直白。

多线程同步是指：

- 协调多个线程对共享数据的访问，避免出现数据不一致的情况。
- 协调各个事件的发生顺序，使多线程在某个点交汇并按预期步骤往前推进，比如某线程需要等另一个线程完成某项工作才能开展该线程的下一步工作。

要掌握多线程同步，需先理解为什么需要多线程同步、哪些情况需要同步。

2.2 为什么需要同步

理解为什么要同步（Why）是多线程编程的关键，它甚至比掌握多线程同步机制（How）本身更加重要。识别什么地方需要同步是编写多线程程序的难点，只有准确识别需要保护的数据、需要同步的点，再配合系统或语言提供的合适的同步机制，才

能编写安全高效的多线程程序。

下面通过几个例子解释为什么需要同步。

示例 1

有 1 个长度为 256 的字符数组 msg 用于保存消息，函数 read_msg() 和 write_msg() 分别用于 msg 的读和写：

```
// example 1
char msg[256] = "this is old msg";

char* read_msg() {
    return msg;
}

void write_msg(char new_msg[], size_t len) {
    memcpy(msg, new_msg, std::min(len, sizeof(msg)));
}

void thread1() {
    char new_msg[256] = "this is new msg, it's too looooooong";
    write_msg(new_msg, sizeof(new_msg));
}

void thread2() {
    printf("msg=%s\n", read_msg());
}
```

如果线程 1 调用 write_msg()，线程 2 调用 read_msg()，并发操作，不加保护。因为 msg 的长度是 256 字节，完成长达 256 字节的写入需要多个内存周期，在线程 1 写入新消息期间，线程 2 可能读到不一致的数据。即可能读到“this is new msg”，而后半段内容“it’s very…”线程 1 还没来得及写入，它不是完整的新消息。

在这个例子中，不一致表现为数据不完整。

示例 2

比如对于二叉搜索树（BST）的节点，一个结构体有 3 个成分：

- 一个指向父节点的指针

- 一个指向左子树的指针
- 一个指向右子树的指针

```
// example 2
struct Node {
  struct Node *parent;
      struct Node *left_child, *right_child;
};
```

这 3 个成分是有关联的，将节点加入 BST，要设置这 3 个指针域，从 BST 删除该节点，要修改该节点的父、左孩子节点、右孩子节点的指针域。对多个指针域的修改，不能在一个指令周期完成，如果完成了一个成分的写入，还没来得修改其他成分，就有可能被其他线程读到了，但此时节点的有些指针域还没有设置好，通过指针域去取数可能会出错。

示例 3

考虑两个线程对同一个整型变量做自增，变量的初始值是 0，我们预期 2 个线程完成自增后变量的值为 2。

```
// example 3
int x = 0; // 初始值为 0
void thread1() { ++x; }
void thread2() { ++x; }
```

简单的自增操作，包括三步：

- **加载**：从内存中读取变量 x 的值存放到寄存器
- **更新**：在寄存器里完成自增
- **保存**：把位于寄存器中的 x 的新值写入内存

两个线程并发执行 ++x，让我们看看真实情况是什么样的：

1. 如果 2 个线程，先后执行自增，在时间上完成错开。无论是 1 先 2 后，或是 2 先 1 后，那么 x 的最终值是 2，符合预期。但多线程并发并不能确保对一个变量的访问在时间上完全错开。

2. 如果时间上没有完全错开，假设线程 1 在 core1 上执行，线程 2 在 core2 上执行，那么，一个可能的执行过程如下：

- 首先，线程 1 把 x 读到 core1 的寄存器，线程 2 也把 x 的值加载到 core2 的寄存器，此时，存放在两个 core 的寄存器中 x 的副本都是 0。
- 然后，线程 1 完成自增，更新寄存器里 x 的值的副本（0 变 1），线程 2 也完成自增，更新寄存器里 x 的值的副本（0 变 1）。
- 再然后，线程 1 将更新后的新值 1 写入变量 x 的内存位置。
- 最后，线程 2 将更新后的新值 1 写入同一内存位置，变量 x 的最终值是 1，不符合预期。

线程 1 和线程 2 在同一个 core 上交错执行，也有可能出现同样的问题，这个问题跟硬件结构无关。之所以会出现不符合预期的情况，主要是因为“加载 + 更新 + 保存”这 3 个步骤不能在一个内存周期内完成。多个线程对同一变量并发读写，不加同步的话会出现数据不一致。

在这个例子中，不一致表现为 x 的终值既可能为 1 也可能为 2。

示例 4

用 C++ 类模板实现一个队列：

```
// example 4
template <typename T>
class Queue {
    static const unsigned int CAPACITY = 100;
    T elements[CAPACITY];
    int num = 0, head = 0, tail = -1;
public:
    // 入队
    bool push(const T& element) {
        if (num == CAPACITY) return false;
        tail = (++tail) % CAPACITY;
        elements[tail] = element;
        ++num;
        return true;
    }
    // 出队
```

```
    void pop() {
        assert(!empty());
        head = (++head) % CAPACITY;
        --num;
    }
    // 判空
    bool empty() const {
        return num == 0;
    }
    // 访队首
    const T& front() const {
        assert(!empty());
        return elements[head];
    }
};
```

代码解释：

- T elements[] 保存数据；2 个游标，分别用于记录队首 head 和队尾 tail 的位置（下标）。
- push() 接口，先移动 tail 游标，再把元素添加到队尾。
- pop() 接口，移动 head 游标，弹出队首元素（逻辑上弹出）。
- front() 接口，返回队首元素的引用。
- front()、pop() 先做断言，调用 pop()/front() 的客户代码需确保队列非空。

假设现在有一个 Queue<int> 实例 q，因为直接调用 pop 可能 assert 失败，我们封装一个 try_pop()，代码如下：

```
Queue<int> q;
void try_pop() {
    if (!q.empty()) {
        q.pop();
    }
}
```

如果多个线程调用 try_pop()，会有问题，为什么?

原因：判空 + 出队这 2 个操作，不能在一个指令周期内完成。如果线程 1 在判断队列非空后，线程 2 穿插进来，判空也为伪，这样就有可能 2 个线程竞争弹出唯一的元素。

多线程环境下，读变量然后基于值做进一步操作，这样的逻辑如果不加保护就会出错，这是由数据使用方式引入的问题。

示例 5

再看一个简单的，简单的对 int32_t 多线程读写。

```
// example 5
int32_t data[8] = {1,2,3,4,5,6,7,8};

struct Foo {
    int32_t get() const { return x; }
    void set(int32_t x) { this->x = x; }
    int32_t x;
} foo;

void thread_write1() {
    for (;;) { for (auto v : data) { foo.set(v); } }
}

void thread_write2() {
    for (;;) { for (auto v : data) { foo.set(v); } }
}

void thread_read() {
    for (;;) { printf("%d", foo.get()); }
}
```

2 个写线程 1 个读线程，写线程在无限循环里用 data 里的元素值设置 foo 对象的 x 成分，读线程简单的打印 foo 对象的 x 值。程序一直跑下去，最后打印出来的数据，会出现除 data 初始化值外的数据吗?

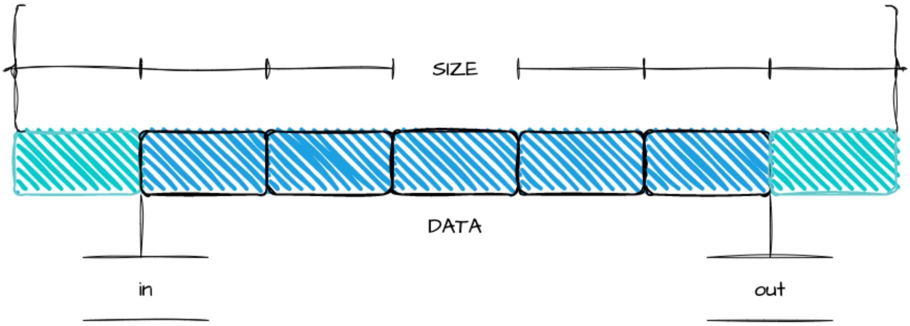
Foo::get 的实现有问题吗？如果有问题？是什么问题?

示例 6

看一个用数组实现 FIFO 队列的程序，一个线程写 put()，一个线程读 get()。

```
// example 6
#include <iostream>
#include <algorithm>
```

```
// 用数组实现的环型队列
class FIFO {
    static const unsigned int CAPACITY = 1024;  // 容量：需要满足是2^N

    unsigned char buffer[CAPACITY];             // 保存数据的缓冲区
    unsigned int in = 0;                        // 写入位置
    unsigned int out = 0;                       // 读取位置

    unsigned int free_space() const { return CAPACITY - in + out; }
public:
    // 返回实际写入的数据长度(<= len)，返回小于len时对应空闲空间不足
    unsigned int put(unsigned char* src, unsigned int len) {
        // 计算实际可写入数据长度(<=len)
        len = std::min(len, free_space());

        // 计算从in位置到buffer结尾有多少空闲空间
        unsigned int l = std::min(len, CAPACITY - (in & (CAPACITY - 1)));
        // 1. 把数据放入buffer的in开始的缓冲区，最多到buffer结尾
        memcpy(buffer + (in & (CAPACITY - 1)), src, l);
        // 2. 把数据放入buffer开头(如果上一步还没有放完)，len - l为0代表上一步
完成数据写入
        memcpy(buffer, src + l, len - l);

        in += len; // 修改in位置，累加，到达uint32_max后溢出回绕
        return len;
    }

    // 返回实际读取的数据长度(<= len)，返回小于len时对应buffer数据不够
    unsigned int get(unsigned char *dst, unsigned int len) {
        // 计算实际可读取的数据长度
        len = std::min(len, in - out);

        unsigned int l = std::min(len, CAPACITY - (out & (CAPACITY - 1)));
        // 1. 从out位置开始拷贝数据到dst，最多拷贝到buffer结尾
        memcpy(dst, buffer + (out & (CAPACITY - 1)), l);
        // 2. 从buffer开头继续拷贝数据(如果上一步还没拷贝完)，len - l为0代表上
一步完成数据获取
        memcpy(dst + l, buffer, len - l);

        out += len; // 修改out，累加，到达uint32_max后溢出回绕
        return len;
    }
};
```



kfifo

环型队列只是逻辑上的概念，因为采用了数组作为数据结构，所以实际物理存储上并非环型。

- put() 用于往队列里放数据，参数 src+len 描述了待放入的数据信息。
- get() 用于从队列取数据，参数 dst+len 描述了要把数据读到哪里、以及读多少字节。
- capacity 精心选择为 2 的 n 次方，可以得到 3 个好处：
    - 非常技巧性的利用了无符号整型溢出回绕，便于处理对 in 和 out 移动
    - 便于计算长度，通过按位与操作 & 而不必除余
    - 搜索 kfifo 获得更详细的解释
- in 和 out 是 2 个游标：
    - in 用来指向新写入数据的存放位置，写入的时候，只需要简单增加 in。
    - out 用来指示从 buffer 的什么位置读取数据的，读取的时候，也只需简单增加 out。
    - in 和 out 在操作上之所以能单调增加，得益于上述 capacity 的巧妙选择。
- 为了简化，队列容量被限制为 1024 字节，不支持扩容，这不影响多线程的讨论。

写的时候，先写入数据再移动 in 游标；读的时候，先拷贝数据，再移动 out 游标；in 游标移动后，消费者才获得 get 到新放入数据的机会。

直觉告诉我们 2 个线程不加同步的并发读写，会有问题，但真有问题吗？如果有，到底有什么问题？怎么解决？

2.3 保护什么

多线程程序里，我们要保护的是数据而非代码，虽然 Java 等语言里有临界代码、sync 方法，但最终要保护的还是代码访问的数据。

2.4 串行化

如果有一个线程正在访问某共享（临界）资源，那么在它结束访问之前，其他线程不能执行访问同一资源的代码（访问临界资源的代码叫临界代码），其他线程想要访问同一资源，则它必须等待，直到那个线程访问完成，它才能获得访问的机会，现实中有很多这样的例子。比如高速公路上的汽车过检查站，假设检查站只有一个车道，则无论高速路上有多少车道，过检查站的时候只能一辆车接着一辆车，从单一车道鱼贯而入。

对多线程访问共享资源施加此种约束就叫串行化。

2.5 原子操作和原子变量

针对前面的两个线程对同一整型变量自增的问题，如果“load、update、store”这 3 个步骤是不可分割的整体，即自增操作 ++x 满足原子性，上面的程序便不会有问题。

因为这样的话，2 个线程并发执行 ++x，只会有 2 个结果：

- 线程 a ++x，然后线程 b ++x，结果是 2。
- 线程 b ++x，然后线程 a ++x，结果是 2。

除此之外，不会出现第三种情况，线程 a、b 孰先孰后，取决于线程调度，但不影响最终结果。

Linux 操作系统和 C/C++ 编程语言都提供了整型原子变量，原子变量的自增、自减等操作都是原子的，操作是原子性的，意味着它是一个不可细分的操作整体，原子变量的用户观察它，只能看到未完成和已完成 2 种状态，看不到半完成状态。

如何保证原子性是实现层面的问题，应用程序员只需要从逻辑上理解原子性，并能恰当的使用它就行了。原子变量非常适用于计数、产生序列号这样的应用场景。

2.6 锁

前面举了很多例子，阐述多线程不加同步并发访问数据会引起什么问题，下面讲解用锁如何做同步。

2.6.1 互斥锁

针对线程 1 write_msg() + 线程 2 read_msg() 的问题，如果能让线程 1 write_msg() 的过程中，线程 2 不能 read_msg()，那就不会有问题。这个要求，其实就是要让多个线程互斥访问共享资源。

互斥锁就是能满足上述要求的同步机制，互斥是排他的意思，它可以确保在同一时间，只能有一个线程对那个共享资源进行访问。

互斥锁有且只有 2 种状态：

- 已加锁（locked）状态
- 未加锁（unlocked）状态

互斥锁提供加锁和解锁两个接口：

- **加锁（acquire）**：当互斥锁处于未加锁状态时，则加锁成功（把锁设置为已加锁状态），并返回；当互斥锁处于已加锁状态时，那么试图对它加锁的线程会被阻塞，直到该互斥量被解锁。
- **解锁（release）**：通过把锁设置为未加锁状态释放锁，其他因为申请加锁而陷入等待的线程，将获得执行机会。如果有多个等待线程，只有一个会获得锁而继续执行。

我们为某个共享资源配置一个互斥锁，使用互斥锁做线程同步，那么所有线程对该资源的访问，都需要遵从“加锁、访问、解锁”的三步：

```
DataType shared_resource;
Mutex shared_resource_mutex;

void shared_resource_visitor1() {
    // step1: 加锁
    shared_resource_mutex.lock();
    // step2: operate shared_resouce
    // operation1
    // step3: 解锁
    shared_resource_mutex.unlock();
}

void shared_resource_visitor2() {
    // step1: 加锁
    shared_resource_mutex.lock();
    // step2: operate shared_resouce
    // operation2
    // step3: 解锁
    shared_resource_mutex.unlock();
}
```

shared_resource_visitor1() 和 shared_resource_visitor2() 代表对共享资源的不同操作，多个线程可能调用同一个操作函数，也可能调用不同的操作函数。

假设线程 1 执行 shared_resource_visitor1()，该函数在访问数据之前，申请加锁，如果互斥锁已经被其他线程加锁，则调用该函数的线程会阻塞在加锁操作上，直到其他线程访问完数据，释放（解）锁，阻塞在加锁操作的线程 1 才会被唤醒，并尝试加锁：

- 如果没有其他线程申请该锁，那么线程 1 加锁成功，获得了对资源的访问权，完成操作后，释放锁。
- 如果其他线程也在申请该锁，那么：
    - 如果其他线程抢到了锁，那么线程 1 继续阻塞。
    - 如果线程 1 抢到了该锁，那么线程 1 将访问资源，再释放锁，其他竞争该锁的线程得以有机会继续执行。

如果不能承受加锁失败而陷入阻塞的代价，可以调用互斥量的 try_lock() 接口，它在加锁失败后会立即返回。

注意：在访问资源前申请锁访问后释放锁，是一个编程契约，通过遵守契约而获得数据一致性的保障，它并非一种硬性的限制，即如果别的线程遵从三步曲，而另一个线程不遵从这种约定，代码能通过编译且程序能运行，但结果可能是错的。

2.6.2 读写锁

读写锁跟互斥锁类似，也是申请锁的时候，如果不能得到满足则阻塞，但读写锁跟互斥锁也有不同，读写锁有 3 个状态：

- 已加读锁状态
- 已加写锁状态
- 未加锁状态

对应 3 个状态，读写锁有 3 个接口：加读锁，加写锁，解锁：

- 加读锁：如果读写锁处于已加写锁状态，则申请锁的线程阻塞；否则把锁设置为已加读锁状态并成功返回。
- 加写锁：如果读写锁处于未加锁状态，则把锁设置为已加写锁状态并成功返回；否则阻塞。
- 解锁：把锁设置为未加锁状态后返回。

读写锁提升了线程的并行度，可以提升吞吐。它可以让多个读线程同时读共享资源，而写线程访问共享资源的时候，其他线程不能执行，所以，读写锁适合对共享资源访问“读大于写”的场合。读写锁也叫“共享互斥锁”，多个读线程可以并发访问同一资源，这对应共享的概念，而写线程是互斥的，写线程访问资源的时候，其他线程无论读写，都不可以进入临界代码区。

考虑一个场景：如果有线程 1、2、3 共享资源 x，读写锁 rwlock 保护资源，线程 1 读访问某资源，然后线程 2 以写的形式访问同一资源 x，因为 rwlock 已经被加了读锁，所以线程 2 被阻塞，然后过了一段时间，线程 3 也读访问资源 x，这时候线程 3 可以继续执行，因为读是共享的，然后线程 1 读访问完成，线程 3 继续访问，过了一段时间，在线程 3 访问完成前，线程 1 又申请读资源，那么它还是会获得访问权，但

是写资源的线程 2 会一直被阻塞。

为了避免共享的读线程饿死写线程，通常读写锁的实现，会给写线程优先权，当然这处决于读写锁的实现，作为读写锁的使用方，理解它的语义和使用场景就够了。

2.6.3 自旋锁

自旋锁（Spinlock）的接口跟互斥量差不多，但实现原理不同。线程在 acquire 自旋锁失败的时候，它不会主动让出 CPU 从而进入睡眠状态，而是会忙等，它会紧凑的执行测试和设置 (Test-And-Set) 指令，直到 TAS 成功，否则就一直占着 CPU 做 TAS。

自旋锁对使用场景有一些期待，它期待 acquire 自旋锁成功后很快会 release 锁，线程运行临界区代码的时间很短，访问共享资源的逻辑简单，这样的话，别的 acquire 自旋锁的线程只需要忙等很短的时间就能获得自旋锁，从而避免被调度走陷入睡眠，它假设自旋的成本比调度的低，它不愿耗费时间在线程调度上（线程调度需要保存和恢复上下文需要耗费 CPU）。

内核态线程很容易满足这些条件，因为运行在内核态的中断处理函数里可以通过关闭调度，从而避免 CPU 被抢占，而且有些内核态线程调用的处理函数不能睡眠，只能使用自旋锁。

而运行在用户态的应用程序，则推荐使用互斥锁等睡眠锁。因为运行在用户态应用程序，虽然很容易满足临界区代码简短，但持有锁时间依然可能很长。在分时共享的多任务系统上、当用户态线程的时间配额耗尽，或者在支持抢占式的系统上、有更高优先级的任务就绪，那么持有自旋锁的线程就会被系统调度走，这样持有锁的过程就有可能很长，而忙等自旋锁的其他线程就会白白消耗 CPU 资源，这样的话，就跟自旋锁的理念相背。

Linux 系统优化过后的 mutex 实现，在加锁的时候会先做有限次数的自旋，只有有限次自旋失败后，才会进入睡眠让出 CPU，所以，实际使用中，它的性能也足够好。此外，自旋锁必须在多 CPU 或者多 Core 架构下，试想如果只有一个核，那么它执

行自旋逻辑的时候，别的线程没有办法运行，也就没有机会释放锁。

2.6.4 锁的粒度

合理设置锁的粒度，粒度太大会降低性能，太小会增加代码编写复杂度。

2.6.5 锁的范围

锁的范围要尽量小，最小化持有锁的时间。

2.6.6 死锁

程序出现死锁有两种典型原因：

ABBA 锁

假设程序中有 2 个资源 X 和 Y，分别被锁 A 和 B 保护，线程 1 持有锁 A 后，想要访问资源 Y，而访问资源 Y 之前需要申请锁 B，而如果线程 2 正持有锁 B，并想要访问资源 X，为了访问资源 X，所以线程 2 需要申请锁 A。线程 1 和线程 2 分别持有锁 A 和 B，并都希望申请对方持有的锁，因为线程申请对方持有的锁，得不到满足，所以便会陷入等待，也就没有机会释放自己持有的锁，对方执行流也就没有办法继续前进，导致相持不下，无限互等，进而死锁。

上述的情况似乎很明显，但如果代码量很大，有时候，这种死锁的逻辑不会这么浅显，它被复杂的调用逻辑所掩盖，但抽茧剥丝，最根本的逻辑就是上面描述的那样。这种情况叫 ABBA 锁，既某个线程持有 A 锁申请 B 锁，而另一个线程持有 B 锁申请 A 锁。这种情况可以通过 try lock 实现，尝试获取锁，如果不成功，则释放自己持有的锁，而不一根筋下去。另一种解法就是锁排序，对 A/B 两把锁的加锁操作，都遵从同样的顺序（比如先 A 后 B），也能避免死锁。

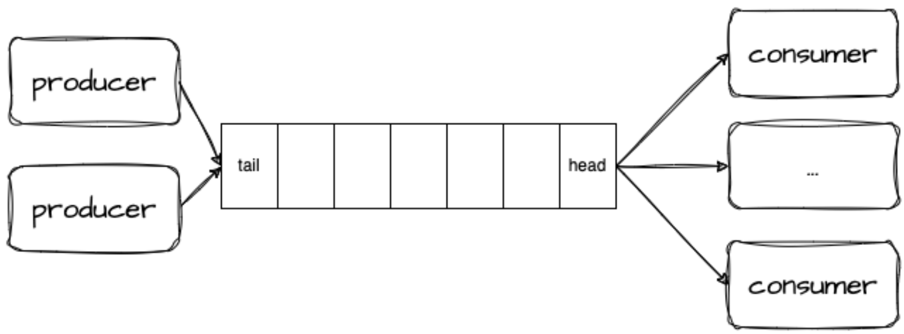
自死锁

对于不支持重复加锁的锁，如果线程持有某个锁，而后又再次申请锁，因为该锁已经被自己持有，再次申请锁必然得不到满足，从而导致死锁。

2.7 条件变量

条件变量常用于生产者消费者模式，需配合互斥量使用。

假设你要编写一个网络处理程序，I/O 线程从套接字接收字节流，反序列化后产生一个个消息（自定义协议），然后投递到一个消息队列，一组工作线程负责从消息队列取出并处理消息。这是典型的生产者 – 消费者模式，I/O 线程生产消息（往队列 put），Work 线程消费消息（从队列 get），I/O 线程和 Work 线程并发访问消息队列，显然，消息队列是竞争资源，需要同步。



proceduer-consumer

可以给队列配置互斥锁，put 和 get 操作前都先加锁，操作完成再解锁。代码差不多是这样的：

```
void io_thread() {
    while (1) {
        Msg* msg = read_msg_from_socket();
        msg_queue_mutex.lock();
        msg_queue.put(msg);
        msg_queue_mutex.unlock();
    }
}

void work_thread() {
    while (1) {
        msg_queue_mutex.lock();
        Msg* msg = msg_queue.get();
        msg_queue_mutex.unlock();
```

```
        if (msg != nullptr) {
            process(msg);
        }
    }
}
```

work 线程组的每个线程都忙于检查消息队列是否有消息，如果有消息就取一个出来，然后处理消息，如果没有消息就在循环里不停检查，这样的话，即使负载很轻，但 work_thread 还是会消耗大量的 CPU 时间。

我们当然可以在两次查询之间加入短暂的 sleep，从而让出 cpu，但是这个睡眠的时间设置为多少合适呢？设置长了的话，会出现消息到来得不到及时处理（延迟上升）；设置太短了，还是无辜消耗了 CPU 资源，这种不断问询的方式在编程上叫轮询。

轮询行为逻辑上，相当于你在等一个投递到楼下小邮局的包裹，你下楼查验没有之后就上楼回房间，然后又下楼查验，你不停的上下楼查验，其实大可不必如此，何不等包裹到达以后，让门卫打电话通知你去取呢？

条件变量提供了一种类似通知 notify 的机制，它让两类线程能够在一个点交汇。条件变量能够让线程等待某个条件发生，条件本身受互斥锁保护，因此条件变量必须搭配互斥锁使用，锁保护条件，线程在改变条件前先获得锁，然后改变条件状态，再解锁，最后发出通知，等待条件的睡眠中的线程在被唤醒前，必须先获得锁，再判断条件状态，如果条件不成立，则继续转入睡眠并释放锁。

对应到上面的例子，工作线程等待的条件是消息队列有消息（非空），用条件变量改写上面的代码：

```
void io_thread() {
    while (1) {
        Msg* msg = read_msg_from_socket();
        {
            std::lock_guard<std::mutex> lock(msg_queue_mutex);
            msg_queue.push_back(msg);
        }
        msg_queue_not_empty.notify_all();
    }
}
```

```
void work_thread() {
    while (1) {
        Msg* msg = nullptr;
        {
            std::unique_lock<std::mutex> lock(msg_queue_mutex);
            msg_queue_not_empty.wait(lock, []{ return !msg_queue.empty(); });
            msg = msg_queue.get();
        }
        process(msg);
    }
}
```

std::lock_guard 是互斥量的一个 RAII 包装类，std::unique_lock 除了会在析构函数自动解锁外，还支持主动 unlock()。

生产者在往 msg_queue 投递消息的时候，需要对 msg_queue 加锁，通知 work 线程的代码可以放在解锁之后，等待 msg_queue_not_empty 条件必须受 msg_queue_mutex 保护，wait 的第二个参数是一个 lambda 表达式，因为会有多个 work 线程被唤醒，线程被唤醒后，会重新获得锁，检查条件，如果不成立，则再次睡眠。条件变量的使用需要非常谨慎，否则容易出现不能唤醒的情况。

C 语言的条件变量、Posix 条件变量的编程接口跟 C++ 的类似，概念上是一致的，故在此不展开介绍。

2.8 lock-free 和无锁数据结构

2.8.1 锁同步的问题

线程同步分为阻塞型同步和非阻塞型同步。

- 互斥量、信号、条件变量这些系统提供的机制都属于阻塞型同步，在争用资源的时候，会导致调用线程阻塞。
- 非阻塞型同步是指在无锁的情况下，通过某种算法和技术手段实现不用阻塞而同步。

锁是阻塞同步机制，阻塞同步机制的缺陷是可能挂起你的程序，如果持有锁的线程崩

溃或者 hang 住，则锁永远得不到释放，而其他线程则将陷入无限等待；另外，它也可能导致优先级倒转等问题。所以，我们需要 lock-free 这类非阻塞的同步机制。

2.8.2 什么是 lock-free

lock-free 没有锁同步的问题，所有线程无阻碍的执行原子指令，而不是等待。比如一个线程读 atomic 类型变量，一个线程写 atomic 变量，它们没有任何等待，硬件原子指令确保不会出现数据不一致，写入数据不会出现半完成，读取数据也不会读一半。

那到底什么是 lock-free ? 有人说 lock-free 就是不使用 mutex / semaphores 之类的无锁（lock-Less）编程，这句话严格来说并不对。

我们先看一下 wiki 对 Lock-free 的描述 :

> Lock-freedom allows individual threads to starve but guarantees system-wide throughput. An algorithm is lock-free if, when the program threads are run for a sufficiently long time, at least one of the threads makes progress (for some sensible definition of progress). All wait-free algorithms are lock-free. In particular, if one thread is suspended, then a lock-free algorithm guarantees that the remaining threads can still make progress. Hence, if two threads can contend for the same mutex lock or spinlock, then the algorithm is not lock-free. (If we suspend one thread that holds the lock, then the second thread will block.)

翻译一下：

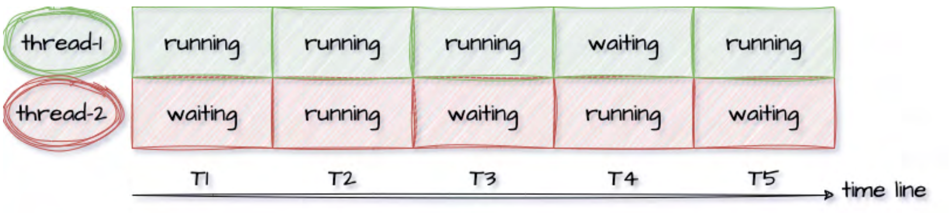
- 第 1 段：lock-free 允许单个线程饥饿但保证系统级吞吐。如果一个程序线程执行足够长的时间，那么至少一个线程会往前推进，那么这个算法就是 lock-free 的。
- 第 2 段：尤其是，如果一个线程被暂停，lock-free 算法保证其他线程依然能够往前推进。

第 1 段给 lock-free 下定义，第 2 段则是对 lock-free 作解释：如果 2 个线程竞争同一个互斥锁或者自旋锁，那它就不是 lock-free 的；因为如果暂停（Hang）持有锁的线程，那么另一个线程会被阻塞。

wiki 的这段描述很抽象，它不够直观，稍微再解释一下：lock-free 描述的是代码逻辑的属性，不使用锁的代码，大部分具有这种属性。大家经常会混淆这 lock-free 和无锁这 2 个概念。其实，lock-free 是对代码（算法）性质的描述，是属性；而无锁是说代码如何实现，是手段。

lock-free 的关键描述是：如果一个线程被暂停，那么其他线程应能继续前进，它需要有系统级（system-wide）的吞吐。

如图，两个线程在时间线上，至少有一个线程处于 running 状态。



Lock-free

我们从反面举例来看，假设我们要借助锁实现一个无锁队列，我们可以直接使用线程不安全的 std::queue + std::mutex 来做：

```
template <typename T>
class Queue {
public:
    void push(const T& t) {
        q_mutex.lock();
        q.push(t);
        q_mutex.unlock();
    }
private:
    std::queue<T> q;
    std::mutex q_mutex;
};
```

如果有线程 A/B/C 同时执行 push 方法，最先进入的线程 A 获得互斥锁。线程 B 和 C 因为获取不到互斥锁而陷入等待。这个时候，线程 A 如果因为某个原因（如出现异常，或者等待某个资源）而被永久挂起，那么同样执行 push 的线程 B/C 将被永久挂起，系统整体（system-wide）没法推进，而这显然不符合 lock-free 的要求。因此：所有基于锁（包括 spinlock）的并发实现，都不是 lock-free 的。

因为它们都会遇到同样的问题：即如果永久暂停当前占有锁的线程 / 进程的执行，将会阻塞其他线程 / 进程的执行。而对照 lock-free 的描述，它允许部分 process（理解为执行流）饿死但必须保证整体逻辑的持续前进，基于锁的并发显然是违背 lock-free 要求的。

2.8.3 CAS loop 实现 Lock-free

Lock-Free 同步主要依靠 CPU 提供的 read-modify-write 原语，著名的“比较和交换”CAS（Compare And Swap）在 X86 机器上是通过 cmpxchg 系列指令实现的原子操作，CAS 逻辑上用代码表达是这样的：

```
bool CAS(T* ptr, T expect_value, T new_value) {
   if (*ptr != expect_value) {
      return false;
   }
   *ptr = new_value;
   return true;
}
```

CAS 接受 3 个参数：

- 内存地址
- 期望值，通常传第一个参数所指内存地址中的旧值
- 新值

逻辑描述：CAS 比较内存地址中的值和期望值，如果不相同就返回失败，如果相同就将新值写入内存并返回成功。

当然这个 C 函数描述的只是 CAS 的逻辑，这个函数操作不是原子的，因为它可以划

分成几个步骤：读取内存值、判断、写入新值，各步骤之间是可以插入其他操作的。不过前面讲了，原子指令相当于把这些步骤打包，它可能是通过 lock; cmpxchg 指令实现的，但那是实现细节，程序员更应该注重在逻辑上理解它的行为。

通过 CAS 实现 Lock-free 的代码通常借助循环，代码如下：

```
do {
    T expect_value = *ptr;
} while (!CAS(ptr, expect_value, new_value));
```

1. 创建共享数据的本地副本：expect_value。
2. 根据需要修改本地副本，从 ptr 指向的共享数据里 load 后赋值给 expect_value。
3. 检查共享的数据跟本地副本是否相等，如果相等，则把新值复制到共享数据。

第三步是关键，虽然 CAS 是原子的，但加载 expect_value 跟 CAS 这 2 个步骤，并不是原子的。所以，我们需要借助循环，如果 ptr 内存位置的值没有变（*ptr \== expect_value），那就存入新值返回成功；否则说明加载 expect_value 后，ptr 指向的内存位置被其他线程修改了，这时候就返回失败，重新加载 expect_value，重试，直到成功为止。

CAS loop 支持多线程并发写，这个特点太有用了，因为多线程同步，很多时候都面临多写的问题，我们可以基于 CAS 实现 Fetch-and-add(FAA) 算法，它看起来像这样：

```
T faa(T& t) {
    T temp = t;
    while (!compare_and_swap(x, temp, temp + 1));
}
```

第一步加载共享数据的值到 temp，第二步比较 + 存入新值，直到成功。

2.8.4 无锁数据结构：lock-free stack

无锁数据结构是通过非阻塞算法而非锁保护共享数据，非阻塞算法保证竞争共享资源

的线程，不会因为互斥而让它们的执行无限期暂停；无阻塞算法是 lock-free 的，因为无论如何调度都能确保有系统级的进度。wiki 定义如下：

> A non-blocking algorithm ensures that threads competing for a shared resource do not have their execution indefinitely postponed by mutual exclusion. A non-blocking algorithm is lock-free if there is guaranteed system-wide progress regardless of scheduling.

下面是 C++ atomic compare_exchange_weak() 实现的一个 lock-free 堆栈（from CppReference）：

```
template <typename T>
struct node {
    T data;
    node* next;
    node(const T& data) : data(data), next(nullptr) {}
};

template <typename T>
class stack {
    std::atomic<node<T>*> head;
public:
    void push(const T& data) {
      node<T>* new_node = new node<T>(data);
      new_node->next = head.load(std::memory_order_relaxed);
      while (!head.compare_exchange_weak(new_node->next, new_node,
                                          std::memory_order_release,
                                          std::memory_order_relaxed));
    }
};
```

代码解析：

- 节点（node）保存 T 类型的数据 data，并且持有指向下一个节点的指针。
- std::atomic<node*> 类型表明 atomic 里放置的是 Node 的指针，而非 Node 本身，因为指针在 64 位系统上是 8 字节，等于机器字长，再长没法保证原子性。
- stack 类包含 head 成员，head 是一个指向头结点的指针，头结点指针相当于堆顶指针，刚开始没有节点，head 为 NULL。

- push 函数里，先根据 data 值创建新节点，然后要把它放到堆顶。
- 因为是用链表实现的栈，所以，如果新节点要成为新的堆顶（相当于新节点作为新的头结点插入），那么新节点的 next 域要指向原来的头结点，并让 head 指向新节点。
- new_node->next = head.load 把新节点的 next 域指向原头结点，然后 head.compare_exchange_weak(new_node->next, new_node)，让 head 指向新节点。
- C++ atomic 的 compare_exchange_weak() 跟上述的 CAS 稍有不同，head.load() 不等于 new_node->next 的时候，它会把 head.load() 的值重新加载到 new_node->next。
- 所以，在加载 head 值和 CAS 之间，如果其他线程调用 push 操作，改变了 head 的值，那没有关系，该线程的本次 cas 失败，下次重试便可以了。
- 多个线程同时 push 时，任一线程在任意步骤阻塞 / 挂起，其他线程都会继续执行并最终返回，无非就是多执行几次 while 循环。

这样的行为逻辑显然符合 lock-free 的定义，注意用 CAS+Loop 实现自旋锁不符合 lock-free 的定义，注意区分。

2.9 程序序：Program Order

对单线程程序而言，代码会一行行顺序执行，就像我们编写的程序的顺序那样。比如：

```
a = 1;
b = 2;
```

会先执行 a=1 再执行 b=2，从程序角度看到的代码行依次执行叫程序序，我们在此基础上构建软件，并以此作为讨论的基础。

2.10 内存序：Memory Order

与程序序相对应的内存序，是指从某个角度观察到的对于内存的读和写所真正发生

的顺序。内存操作顺序并不唯一，在一个包含 core0 和 core1 的 CPU 中，core0 和 core1 有着各自的内存操作顺序，这两个内存操作顺序不一定相同。从包含多个 Core 的 CPU 的视角看到的全局内存操作顺序跟单 core 视角看到的内存操作顺序亦不同，而这种不同，对于有些程序逻辑而言，是不可接受的，例如：

程序序要求 a = 1 在 b = 2 之前执行，但内存操作顺序可能并非如此，对 a 赋值 1 并不确保发生在对 b 赋值 2 之前，这是因为：

- 如果编译器认为对 b 赋值没有依赖对 a 赋值，那它完全可能在编译期调整编译后的汇编指令顺序。
- 即使编译器不做调整，到了执行期，也有可能对 b 的赋值先于对 a 赋值执行。

虽然对一个 Core 而言，如上所述，这个 Core 观察到的内存操作顺序不一定符合程序序，但内存操作序和程序序必定产生相同的结果，无论在单 Core 上对 a、b 的赋值哪个先发生，结果上都是 a 被赋值为 1、b 被赋值为 2，如果单核上乱序执行会影响结果，那编译器的指令重排和 CPU 乱序执行便不会发生，硬件会提供这项保证。

但多核系统，硬件不提供这样的保证，多线程程序中，每个线程所工作的 Core 观察到的不同内存操作序，以及这些顺序与全局内存序的差异，常常导致多线程同步失败，所以，需要有同步机制确保内存序与程序序的一致，内存屏障（Memory Barrier）的引入，就是为了解决这个问题，它让不同的 Core 之间，以及 Core 与全局内存序达成一致。

2.11 乱序执行：Out-of-order Execution

乱序执行会引起内存顺序跟程序顺序不同，乱序执行的原因是多方面的，比如编译器指令重排、超标量指令流水线、预测执行、Cache-Miss 等。内存操作顺序无法精确匹配程序顺序，这有可能带来混乱，既然有副作用，那为什么还需要乱序执行呢？答案是为了性能。

我们先看看没有乱序执行之前，早期的有序处理器（In-order Processors）是怎么

处理指令的?

- 指令获取，从代码节内存区域加载指令到 I-Cache
- 译码
- 如果指令操作数可用（例如操作数位于寄存器中），则分发指令到对应功能模块中；如果操作数不可用，通常是需要从内存加载，则处理器会 stall，一直等到它们就绪，直到数据被加载到 Cache 或拷贝进寄存器
- 指令被功能单元执行
- 功能单元将结果写回寄存器或内存位置

乱序处理器（Out-of-order Processors）又是怎么处理指令的呢?

- 指令获取，从代码节内存区域加载指令到 I-Cache
- 译码
- 分发指令到指令队列
- 指令在指令队列中等待，一旦操作数就绪，指令就离开指令队列，那怕它之前的指令未被执行（乱序）
- 指令被派往功能单元并被执行
- 执行结果放入队列（Store Buffer），而不是直接写入 Cache
- 只有更早请求执行的指令结果写入 cache 后，指令执行结果才写入 Cache，通过对指令结果排序写入 cache，使得执行看起来是有序的

指令乱序执行是结果，但原因并非只有 CPU 的乱序执行，而是由两种因素导致：

- **编译期**：指令重排（编译器），编译器会为了性能而对指令重排，源码上先后的两行，被编译器编译后，可能调换指令顺序，但编译器会基于一套规则做指令重排，有明显依赖的指令不会被随意重排，指令重排不能破坏程序逻辑。
- **运行期**：乱序执行（CPU），CPU 的超标量流水线、以及预测执行、Cache-Miss 等都有可能导致指令乱序执行，也就是说，后面的指令有可能先于前面的指令执行。

2.12 Store Buffer

为什么需要 Store Buffer？

考虑下面的代码：

```
void set_a() {
    a = 1;
}
```

- 假设运行在 core0 上的 set_a() 对整型变量 a 赋值 1，计算机通常不会直接写穿通到内存，而是会在 Cache 中修改对应 Cache Line
- 如果 Core0 的 Cache 里没有 a，赋值操作（store）会造成 Cache Miss
- Core0 会 stall 在等待 Cache 就绪（从内存加载变量 a 到对应的 Cache Line），但 Stall 会损害 CPU 性能，相当于 CPU 在这里停顿，白白浪费着宝贵的 CPU 时间
- 有了 Store Buffer，当变量在 Cache 中没有就位的时候，就先 Buffer 住这个 Store 操作，而 Store 操作一旦进入 Store Buffer，core 便认为自己 Store 完成，当随后 Cache 就位，store 会自动写入对应 Cache。

所以，我们需要 Store Buffer，每个 Core 都有独立的 Store Buffer，每个 Core 都访问私有的 Store Buffer，Store Buffer 帮助 CPU 遮掩了 Store 操作带来的延迟。

Store Buffer 会带来什么问题?

```
a = 1;
b = 2;
assert(a == 1);
```

上面的代码，断言 a==1 的时候，需要读（load）变量 a 的值，而如果 a 在被赋值前就在 Cache 中，就会从 Cache 中读到 a 的旧值（可能是 1 之外的其他值），所以断言就可能失败。但这样的结果显然是不能接受的，它违背了最直观的程序顺序性。

问题出在变量 a 除保存在内存外，还有 2 份拷贝：一份在 Store Buffer 里，一份在

Cache 里；如果不考虑这 2 份拷贝的关系，就会出现数据不一致。那怎么修复这个问题呢?

可以通过在 Core Load 数据的时候，先检查 Store Buffer 中是否有悬而未决的 a 的新值，如果有，则取新值；否则从 cache 取 a 的副本。这种技术在多级流水线 CPU 设计的时候就经常使用，叫 Store Forwarding。有了 Store Buffer Forwarding，就能确保单核程序的执行遵从程序顺序性，但多核还是有问题，让我们考查下面的程序：

多核内存序问题

```
int a = 0; // 被CPU1 Cache
int b = 0; // 被CPU0 Cache

// CPU0 执行
void x() {
    a = 1;
    b = 2;
}

// CPU1 执行
void y() {
    while (b == 0);
    assert(a == 1);
}
```

假设 a 和 b 都被初始化为 0；CPU0 执行 x() 函数，CPU1 执行 y() 函数；变量 a 在 CPU1 的 local Cache 里，变量 b 在 CPU0 的 local Cache 里。

- CPU0 执行 a = 1 的时候，因为 a 不在 CPU0 的 local cache，CPU0 会把 a 的新值 1 写入 Store Buffer 里，并发送 Read Invalidate 消息给其他 CPU。
- CPU1 执行 while (b == 0)，因为 b 不在 CPU1 的 local cache 里，CPU1 会发送 Read 消息去其他 CPU 获取 b 的值。
- CPU0 执行 b = 2，因为 b 在 CPU0 的 local Cache，所以直接更新 local cache 中 b 的副本。
- CPU0 收到 CPU1 发来的 read 消息，把 b 的新值 2 发送给 CPU1；同时存放 b 的 Cache Line 的状态被设置为 Shared，以反应 b 同时被 CPU0 和

CPU1 cache 住的事实。

- CPU1 收到 b 的新值 2 后结束循环，继续执行 assert(a == 1)，因为此时 local Cache 中的 a 值为 0，所以断言失败。
- CPU1 收到 CPU0 发来的 Read Invalidate 后，更新 a 的值为 1，但为时已晚，程序在上一步已经崩了（assert 失败）。

怎么办？答案留到内存屏障一节揭晓。

2.13 Invalidate Queue

为什么需要 Invalidate Queue？

当一个变量加载到多个 core 的 Cache，则这个 Cache Line 处于 Shared 状态，如果 Core1 要修改这个变量，则需要通过发送核间消息 Invalidate 来通知其他 Core 把对应的 Cache Line 置为 Invalid，当其他 Core 都 Invalid 这个 CacheLine 后，则本 Core 获得该变量的独占权，这个时候就可以修改它了。

收到 Invalidate 消息的 core 需要回 Invalidate ACK，一个个 core 都这样 ACK，等所有 core 都回复完，Core1 才能修改它，这样 CPU 就白白浪费。

事实上，其他核在收到 Invalidate 消息后，会把 Invalidate 消息缓存到 Invalidate Queue，并立即回复 ACK，真正 Invalidate 动作可以延后再做，这样一方面因为 Core 可以快速返回别的 Core 发出的 Invalidate 请求，不会导致发生 Invalidate 请求的 Core 不必要的 Stall，另一方面也提供了进一步优化可能，比如在一个 CacheLine 里的多个变量的 Invalidate 可以攒一次做了。

但写 Store Buffer 的方式其实是 Write Invalidate，它并非立即写入内存，如果其他核此时从内存读数，则有可能不一致。

2.14 内存屏障

那有没有方法确保对 a 的赋值一定先于对 b 的赋值呢？有，内存屏障被用来提供这个保障。

内存屏障（Memory Barrier），也称内存栅栏、屏障指令等，是一类同步屏障指令，是 CPU 或编译器在对内存随机访问的操作中的一个同步点，同步点之前的所有读写操作都执行后，才可以开始执行此点之后的操作。语义上，内存屏障之前的所有写操作都要写入内存；内存屏障之后的读操作都可以获得同步屏障之前的写操作的结果。

内存屏障，其实就是提供一种机制，确保代码里顺序写下的多行，会按照书写的顺序，被存入内存，主要是解决 Store Buffer 引入导致的写入内存间隙的问题。

```
void x() {
    a = 1;
    wmb();
    b = 2;
}
```

像上面那样在 a=1 和 b=2 之间插入一条内存屏障语句，就能确保 a=1 先于 b=2 生效，从而解决了内存乱序访问问题，那插入的这句 smp_mb()，到底会干什么呢?

回忆前面的流程，CPU0 在执行完 a = 1 之后，执行 smp_mb() 操作，这时候，它会给 Store Buffer 里的所有数据项做一个标记（marked），然后继续执行 b = 2，但这时候虽然 b 在自己的 cache 里，但由于 store buffer 里有 marked 条目，所以，CPU0 不会修改 cache 中的 b，而是把它写入 Store Buffer；所以 CPU0 收到 Read 消息后，会把 b 的 0 值发给 CPU1，所以继续在 while (b) 自旋。

简而言之，Core 执行到 write memory barrier（wmb）的时候，如果 Store Buffer 还有悬而未决的 store 操作，则都会被 mark 上，直到被标注的 Store 操作进入内存后，后续的 Store 操作才能被执行，因此 wmb 保障了 barrier 前后操作的顺序，它不关心 barrier 前的多个操作的内存序，以及 barrier 后的多个操作的内存序，是否与 Global Memory Order 一致。

```
a = 1;
b = 2;
wmb();
c = 3;
d = 4;
```

wmb() 保证“a=1;b=2”发生在“c=3;d = 4”之前，不保证 a = 1 和 b = 2 的内存序，也不保证 c = 3 和 d = 4 的内部序。

Invalidate Queue 的引入的问题

就像引入 Store Buffer 会影响 Store 的内存一致性，Invalidate Queue 的引入会影响 Load 的内存一致性。因为 Invalidate queue 会缓存其他核发过来的消息，比如 Invalidate 某个数据的消息被 delay 处置，导致 core 在 Cache Line 中命中这个数据，而这个 Cache Line 本应该被 Invalidate 消息标记无效。如何解决这个问题呢?

一种思路是硬件确保每次 load 数据的时候，需要确保 Invalidate Queue 被清空，这样可以保证 load 操作的强顺序

软件的思路，就是仿照 wmb() 的定义，加入 rmb() 约束。rmb() 给我们的 invalidate queue 加上标记。当一个 load 操作发生的时候，之前的 rmb() 所有标记的 invalidate 命令必须全部执行完成，然后才可以让随后的 load 发生。这样，我们就在 rmb() 前后保证了 load 观察到的顺序等同于 global memory order

所以，我们可以像下面这样修改代码：

```
a = 1;
wmb();
b = 2;
while(b != 2) {};
rmb();
assert(a == 1);
```

系统对内存屏障的支持

gcc 编译器在遇到内嵌汇编语句 asm volatile(“” ::: “memory”)，将以此作为一条内存屏障，重排序内存操作，即此语句之前的各种编译优化将不会持续到此语句之后。

Linux 内核提供函数 barrier() 用于让编译器保证其之前的内存访问先于其之后的完成。

```
#define barrier() __asm__ __volatile__("" ::: "memory")
```

CPU 内存屏障：

- 通用 barrier，保证读写操作有序，mb() 和 smp_mb()
- 写操作 barrier，仅保证写操作有序，wmb() 和 smp_wmb()
- 读操作 barrier，仅保证读操作有序，rmb() 和 smp_rmb()

小结

- 为了提高处理器的性能，SMP 中引入了 store buffer(以及对应实现 store buffer forwarding) 和 invalidate queue。
- store buffer 的引入导致 core 上的 store 顺序可能不匹配于 global memory 的顺序，对此，我们需要使用 wmb() 来解决。
- invalidate queue 的存在导致 core 上观察到的 load 顺序可能与 global memory order 不一致，对此，我们需要使用 rmb() 来解决。
- 由于 wmb() 和 rmb() 分别只单独作用于 store buffer 和 invalidate queue，因此这两个 memory barrier 共同保证了 store/load 的顺序。

3 伪共享

多个线程同时读写同一个 Cache Line 中的变量、导致 CPU Cache 频繁失效，从而使得程序性能下降的现象称为**伪共享**（False Sharing）。

```
const size_t shm_size = 16*1024*1024; //16M
static char shm[shm_size];
std::atomic<size_t> shm_offset{0};

void f() {
    for (;;) {
        auto off = shm_offset.fetch_add(sizeof(long));
        if (off >= shm_size) break;
        *(long*)(shm + off) = off; // 赋值
    }
}
```

考察上面的程序：shm 是一块 16M 字节的内存，我测试的机器的 L3 Cache 是 32M，16M 字节能确保 shm 在 Cache 里放得下。f() 函数的循环里，视 shm 为 long 类型的数组，依次给每个元素赋值，shm_offset 用于记录偏移位置，shm_offset.fetch_add(sizeof(long)) 原子性的增加 shm_offset 的值（因为 x86_64 系统上 long 的长度为 8，所以 shm_offset 每次增加 8），并返回增加前的值，对 shm 上 long 数组的每个元素赋值后，结束循环从函数返回。

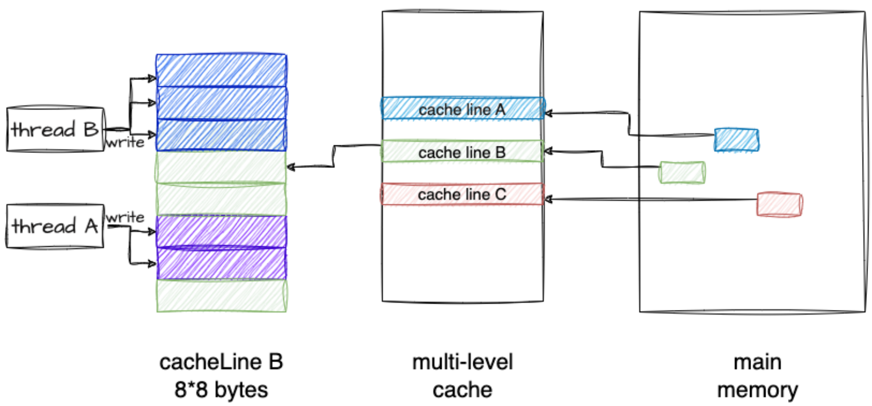
因为 shm_offset 是 atomic 类型变量，所以多线程调用 f() 依然能正常工作，虽然多个线程会竞争 shm_offset，但每个线程会排他性的对各 long 元素赋值，多线程并行会加快对 shm 的赋值操作。我们加上多线程调用代码：

```
std::atomic<size_t> step{0};

const int THREAD_NUM = 2;

void work_thread() {
    const int LOOP_N = 10;
    for (int n = 1; n <= LOOP_N; ++n) {
        f();
        ++step;
        while (step.load() < n * THREAD_NUM) {}
        shm_offset = 0;
    }
}

int main() {
    std::thread threads[THREAD_NUM];
    for (int i = 0; i < THREAD_NUM; ++i) {
        threads[i] = std::move(std::thread(work_thread));
    }
    for (int i = 0; i < THREAD_NUM; ++i) {
        threads[i].join();
    }
    return 0;
}
```

- main 函数里启动 2 个工作线程 work_thread。
- 工作线程对 shm 共计赋值 10 轮，后面的每一轮会访问 Cache 里的 shm 数据，step 用于 work_thread 之间每一轮的同步。

- 工作线程调用完 f() 后会增加 step，等 2 个工作线程都调用完之后，step 的值增加到 n * THREAD_NUM 后，while() 会结束循环，重置 shm_offset，重新开始新一轮对 shm 的赋值。

如图所示：



false-sharing-1

编译后执行上面的程序，产生如下的结果：

```
time ./a.out

real 0m3.406s
user 0m6.740s
sys 0m0.040s
```

time 命令用于时间测量，a.out 程序运行完成后会打印耗时，real 列显式耗时 3.4 秒。

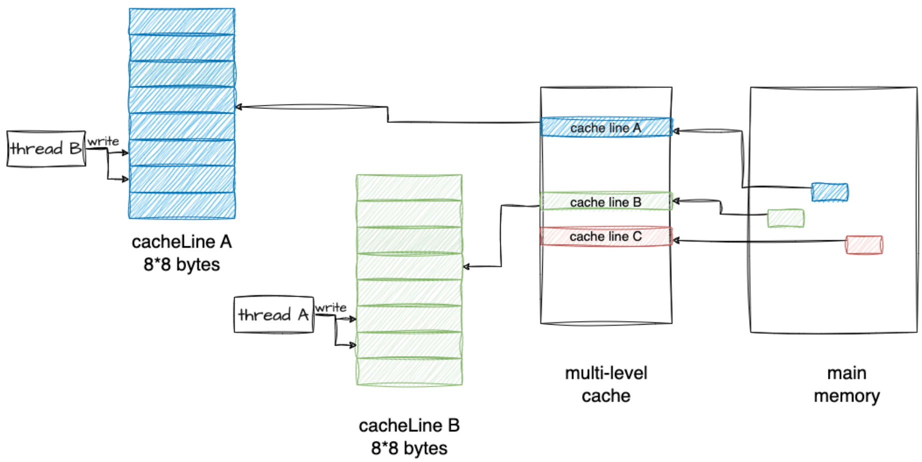
3.1 改进版 f_fast

我们稍微修改一下 f 函数，改进版 f 函数取名 f_fast：

```
void f_fast() {
    for (;;) {
        const long inner_loop = 16;
```

```
            auto off = shm_offset.fetch_add(sizeof(long) * inner_loop);
            for (long j = 0; j < inner_loop; ++j) {
                if (off >= shm_size) return;
                *(long*)(shm + off) = j;
                off += sizeof(long);
            }
        }
    }
```

for 循环里，shm_offset 不再是每次增加 8 字节（sizeof(long)），而是 8*16=128 字节，然后在内层的循环里，依次对 16 个 long 连续元素赋值，然后下一轮循环又再次增加 128 字节，直到完成对 shm 的赋值。如图所示：



no-false-sharing

编译后重新执行程序，结果显示耗时降低到 0.06 秒，对比前一种耗时 3.4 秒，f_fast 性能提升明显。

```
time ./a.out

real 0m0.062s
user 0m0.110s
sys 0m0.012s
```

f 和 f_fast 的行为差异

shm 数组总共有 2M 个 long 元素，因为 16M / sizeof(long) 得 2M：

1、f() 函数行为逻辑

- 线程 1 和线程 2 的 work_thread 里会交错地对 shm 元素赋值，shm 的 2M 个 long 元素，会顺序的一个接一个的派给 2 个线程去赋值。
- 可能的行为：元素 1 由线程 1 赋值，元素 2 由线程 2 赋值，然后元素 3 和元素 4 由线程 1 赋值，然后元素 5 又由线程 2 赋值…
- 每次分派元素的时候，shm_offset 都会 atomic 的增加 8 字节，所以不会出现 2 个线程给同 1 个元素赋值的情况。

2、f_fast() 函数行为逻辑

- 每次派元素的时候，shm_offset 原子性的增加 128 字节（16 个元素）。
- 这 16 个字节作为一个整体，派给线程 1 或者线程 2；虽然线程 1 和线程 2 还是会交错的操作 shm 元素，但是以 16 个元素（128 字节）为单元，这 16 个连续的元素不会被分开派发给不同线程。
- 一次派发的 16 个元素，会在一个线程里被一个接着一个的赋值（内部循环里）。

3.2 为什么 f_fast 更快

第一眼感觉是 f_fast() 里 shm_offset.fetch_add() 调用频次降低到了原来的 1/16，有理由怀疑是原子变量的竞争减少导致程序执行速度加快。为了验证，让我们在内层的循环里加一个原子变量 test 的 fetch_add，test 原子变量的竞争会像 f() 函数里 shm_offset.fetch_add() 一样激烈，修改后的 f_fast 代码变成下面这样：

```
void f_fast() {
    for (;;) {
        const long inner_loop = 16;
        auto off = shm_offset.fetch_add(sizeof(long) * inner_loop);
        for (long j = 0; j < inner_loop; ++j) {
            test.fetch_add(1);
            if (off >= shm_size) return;
```

```
                *(long*)(shm + off) = j;
                off += sizeof(long);
            }
        }
}
```

为了避免 test.fetch_add(1) 的调用被编译器优化掉，我们在 main 函数的最后把 test 的值打印出来。编译后测试一下，结果显示：执行时间只是稍微增加到 real 0m0.326s，很显然，并不是 atomic 的调用频次减少导致性能飙升。

重新审视 f() 循环里的逻辑：f() 循环里的操作很简单：原子增加、判断、赋值。我们把 f() 的里赋值注释掉，再测试一下，发现它的速度得到了很大提升，看来是 *(long)*(shm + off) = off 这一行代码执行慢，但这明明只是一行赋值。我们把它反汇编来看，它只是一个 mov 指令，源操作数是寄存器，目标操作数是内存地址，从寄存器拷贝数据到一个内存地址，为什么会这么慢呢?

3.3 原因

现在揭晓答案：导致 f() 性能底下的元凶是伪共享（false sharing）。那什么是伪共享？要说清这个问题，还得联系 CPU 的架构以及 CPU 怎么访问数据，回顾一下关于多核 Cache 结构。

背景知识

现代 CPU 可以有多个核，每个核有自己的 L1-L2 缓存，L1 又区分数据缓存（L1-DCache）和指令缓存（L1-ICache），L2 不区分数据和指令 Cache，而 L3 是跨核共享的，L3 通过内存总线连接到内存，内存被所有 CPU 所有 Core 共享。

CPU 访问 L1 Cache 的速度大约是访问内存的 100 倍，Cache 作为 CPU 与内存之间的缓存，减少对内存的访问频率。

从内存加载数据到 Cache 的时候，是以 Cache Line 为长度单位的，Cache Line 的长度通常是 64 字节，所以，那怕只读一个字节，但是包含该字节的整个 Cache Line 都会被加载到缓存，同样，如果修改一个字节，那么最终也会导致整个 Cache

Line 被冲刷到内存。

如果一块内存数据被多个线程访问，假设多个线程在多个 Core 上并行执行，那么它便会被加载到多个 Core 的的 Local Cache 中；这些线程在哪个 Core 上运行，就会被加载到哪个 Core 的 Local Cache 中，所以，内存中的一个数据，在不同 Core 的 Cache 里会同时存在多份拷贝。

那么，便会存在缓存一致性问题。当一个 Core 修改其缓存中的值时，其他 Core 不能再使用旧值。该内存位置将在所有缓存中失效。此外，由于缓存以缓存行而不是单个字节的粒度运行，因此整个缓存行将在所有缓存中失效。如果我们修改了 Core1 缓存里的某个数据，则该数据所在的 Cache Line 的状态需要同步给其他 Core 的缓存，Core 之间可以通过核间消息同步状态，比如通过发送 Invalidate 消息给其他核，接收到该消息的核会把对应 Cache Line 置为无效，然后重新从内存里加载最新数据。

当然，被加载到多个 Core 缓存中的同一 Cache Line，会被标记为共享（Shared）状态，对共享状态的缓存行进行修改，需要先获取缓存行的修改权（独占），MESI 协议用来保证多核缓存的一致性，更多的细节可以参考 MESI 的文章。

示例分析

假设线程 1 运行在 Core1，线程 2 运行在 Core2。

- 因为 shm 被线程 1 和线程 2 这两个线程并发访问，所以 shm 的内存数据会以 Cache Line 粒度，被同时加载到 2 个 Core 的 Cache，因为被多核共享，所以该 Cache Line 被标注为 Shared 状态。
- 假设线程 1 在 offset 为 64 的位置写入了一个 8 字节的数据（sizeof(long)），要修改一个状态为 Shared 的 Cache Line，Core1 会发送核间通信消息到 Core2，去拿到该 Cache Line 的独占权，在这之后，Core1 才能修改 Local Cache
- 线程 1 执行完 shm_offset.fetch_add(sizeof(long)) 后，shm_offset 会增加到 72。

- 这时候 Core2 上运行的线程 2 也会执行 shm_offset.fetch_add(sizeof(long))，它返回 72 并将 shm_offset 增加到 80。
- 线程 2 接下来要修改 shm[72] 的内存位置，因为 shm[64] 和 shm[72] 在一个 Cache Line，而这个 Cache Line 又被置为 Invalidate，所以，它需要从内存里重新加载这一个 Cache Line，而在这之前，Core1 上的线程 1 需要把 Cache Line 冲刷到内存，这样线程 2 才能加载最新的数据。

这种交替执行模式，相当于 Core1 和 Core2 之间需要频繁的发送核间消息，收到消息的 Core 的 Cache Line 被置为无效，并重新从内存里加载数据到 Cache，每次修改后都需要把 Cache 中的数据刷入内存，这相当于废弃掉了 Cache，因为每次读写都直接跟内存打交道，Cache 的作用不复存在，这就是性能低下的原因。

这种多核多线程程序，因为并发读写同一个 Cache Line 的数据（临近位置的内存数据），导致 Cache Line 的频繁失效，内存的频繁 Load/Store，从而导致性能急剧下降的现象叫伪共享，伪共享是性能杀手。

3.4 另一个伪共享的例子

假设线程 x 和 y，分别修改 Data 的 a 和 b 变量，如果被频繁调用，也会出现性能低下的情况，怎么规避呢?

```
struct Data {
    int a;
    int b;
} data; // global

void thread1() {
    data.a = 1;
}

void thread2() {
    data.b = 2;
}
```

空间换时间

避免 Cache 伪共享导致性能下降的思路是用空间换时间，通过增加填充，让 a 和 b 两个变量分布到不同的 Cache Line，这样对 a 和 b 的修改就会作用于不同 Cache Line，就能避免 Cache 失效的问题。

```
struct Data {
    int a;
    int padding[60];
    int b;
};
```

在 Linux kernel 中存在 __cacheline_aligned_in_smp 宏定义用于解决 false sharing 问题。

```
#ifdef CONFIG_SMP
#define __cacheline_aligned_in_smp __cacheline_aligned
#else
#define __cacheline_aligned_in_smp
#endif

struct Data {
    int a;
    int b __cacheline_aligned_in_smp;
};
```

从上面的宏定义，可以看到：

- 在多核系统里，该宏定义是 __cacheline_aligned，也就是 Cache Line 的大小
- 在单核系统里，该宏定义是空的

4 小结

pthread 接口提供的几种同步原语如下：

同步原语	出现背景	常见应用场景	优势	劣势	备注
互斥锁（mutex）	每次只有一个线程可以向前执行	大部分场景	使用简单、锁竞争不激烈的时候性能非常高	竞争激烈时候性能较低	第一选择，pthread有多种锁定特性，建议只使用标准互斥类型
读写锁（read_write_lock）	允许更高的并行度，一次只有一个线程可以占有写模式的锁，但是多个读线程可以占用读模式的读写锁	适用于读的次数远大于写的情况	读多写少的情况下性能较好	可能导致写饥饿、读饥饿。（取决于实现）	可以实现为：读优先、写优先、公平。
条件变量（condition_variable）	允许线程以无竞争的方式等到特定的条件发生、同时避免忙等待；拥有通知机制。	生产者-消费者模型	可以用于通知线程条件满足	唤醒线程、重新获取锁和重新检查条件可能导致额外的性能开销。	需要配合互斥量使用、需要check虚假唤醒和唤醒多个（posix标准允许唤醒一个以上线程）
自旋锁（spin）	线程并不希望在重新调度上花时间。不通过休眠使线程阻塞，而是通过忙等近似阻塞，用来实现一些其他类型的锁	锁被持有的时间短，	非抢占式内核避免中断、一些系统函数和系统库使用	自旋的时间可能会比预期长（时间片作用下）	用户层不推荐使用，因为互斥锁足够高效
屏障（barrier）	协调多个线程并行工作，每个线程等待直到全部线程都到达一点然后从该点进行执行	适用于固定数量的线程的 并行算法、数据初始化等	简化同步的代码	某些实现中，当线程在屏障上等待时会消耗 CPU 资源（忙等） 只适用于固定数量的线程	和Memory Barrier是两种，使用pthread_barrier_* 接口。 类似Java中的CyclicBarrier或者C++20的std::barrier

由于 linux 下线程和进程本质都是 LWP，那么进程间通信使用的 IPC（管道、FIFO、消息队列、信号量）线程间也可以使用，也可以达到相同的作用。 但是由于 IPC 资源在进程退出时不会清理（因为它是系统资源），因此不建议使用。

以下是一些非锁但是也能实现线程安全或者部分线程安全的常见做法：

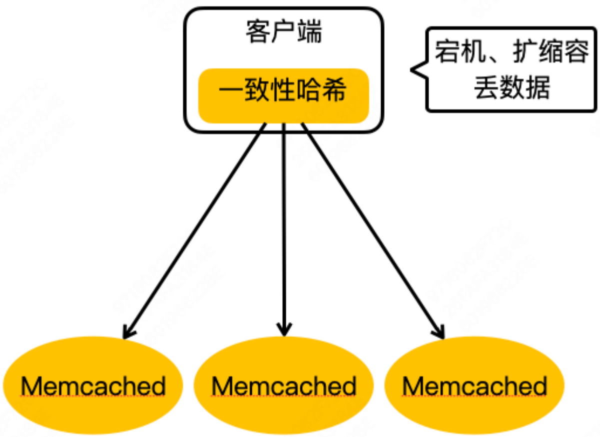
名称	简介	常见应用场景	优势	劣势	备注
原子赋值	简单类型的对齐读取和写入通常是原子的。 比如 int32、int64	双buffer切换时候的指针赋值； 共享内存中简单类型之间修改	无锁且实现简单		本质上单条处理器指令是原子的 并非所有处理器架构都保证是原子的
简单原子变量(atomic)	通过编译器、语言等实现原子的CPU指令	原生类型的自增自减和立即数赋值等、不强调先后只追求最终结果的正确	性能高、实现简单		gcc、C、C++、Java等都有实现 所有原子类型都不支持拷贝 没有浮点类型的原子变量
CAS(简单原子变量就是一种weak的CAS)	内存屏障	对执行的先后顺序有严格要求	性能高、实现简单		在竞争严重的时候，自旋可能非常浪费CPU
双buffer	在内存中保存两份	更新不频繁的数据	性能高、无需加锁	浪费空间	只适用于一写多读的场景
延迟删除双buffer（Double Buffering with Deferred Deletion）	在更新后短期内双buffer，而后删除旧版本，通过指针赋值的原子性切换到新数据	更新不频繁的数据 读多写少的场景	性能高、无需加锁	更新频率有限制	只适用于一写多读的场景
thread_local	每个线程持有一份数据，彻底摆脱线程同步	线程间无需实时交互	性能高、无需加锁	每个线程都有一个实例	
per-cpu变量	每个处理器都分配了该变量的副本	绑核后无锁读写	性能高、无需加锁		参考：DEFINE_PER_CPU，get_cpu_var等
RCU（Read-Copy-Update）	需要修改时候创建副本，然后切换副本，	读多写少的场景			参见：rcu_read_lock，https://liburcu.org/

可以看到，上面很多做法都是采用了副本，尽量避免在 thread 中间共享数据。最快的同步就是没同步（The fastest synchronization of all is the kind that never takes place），Share nothing is best。

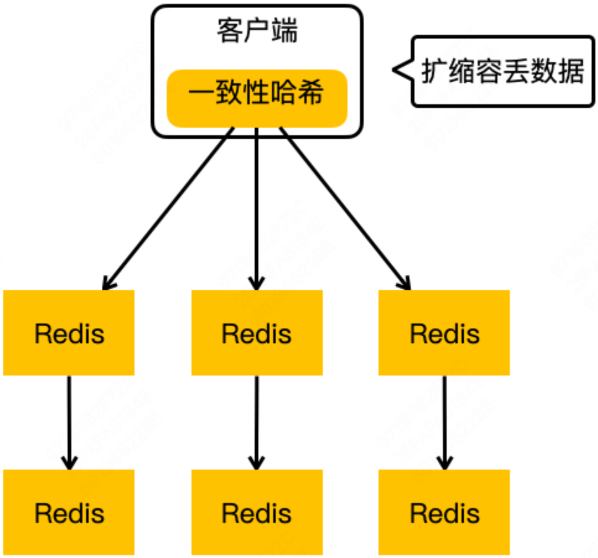
美团大规模 KV 存储挑战与架构实践

KV 存储作为美团一项重要的在线存储服务，承载了在线服务每天万亿级的请求量，并且保持着 99.995% 的服务可用性。在 DataFunSummit 2023 数据基础架构峰会上，我们分享了《美团大规模 KV 存储挑战与架构实践》，本文为演讲内容的整理。文章主要分为四个部分：第一部分介绍了美团 KV 存储发展历程；第二部分分享了内存 KV Squirrel 挑战和架构实践；第三部分阐述了持久化 KV Cellar 挑战和架构实践；最后一部分介绍了未来的发展规划。希望这些内容对大家有所帮助或启发。

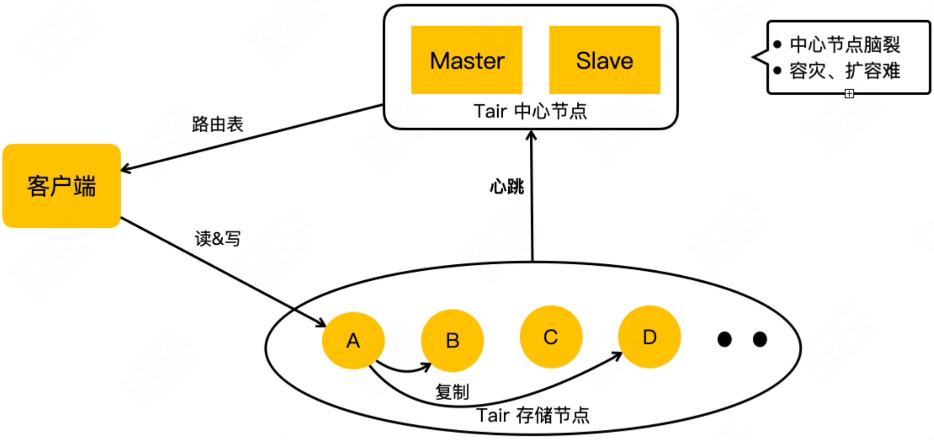
1 美团 KV 存储发展历程



上图就是美团第一代的分布式 KV 存储的架构，可能很多公司都经历过这个阶段。在客户端内做一致性哈希，然后在后端部署上很多 Memcached 实例，这样就实现了最基本的 KV 存储分布式设计。但这样的设计存在很明显的问题：比如在宕机摘除节点会时丢失数据；此外，在缓存空间不够需要扩容时，一致性哈希也会丢失一些数据，这样会给业务的开发带来很大的困扰。



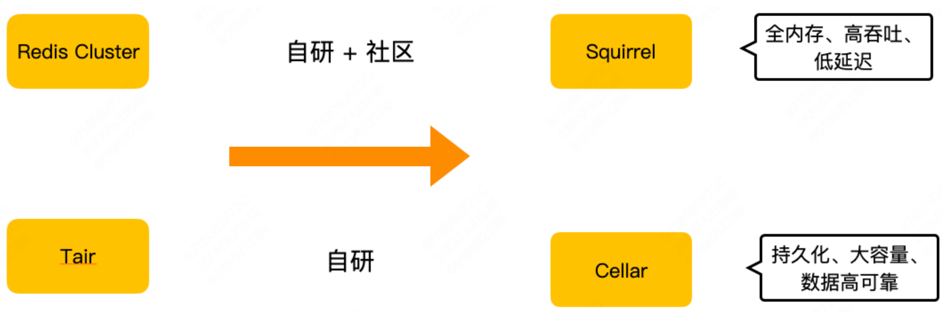
随着 Redis 项目的成熟，美团也引入了 Redis 来解决我们上面提到的问题，进而演进出来上图这样一个架构。可以看到，客户端还是一样，使用一致性哈希算法，在服务器端变成了 Redis 组成的主从结构。当任何一个节点宕机，我们可以通过 Redis 哨兵完成 failover，实现高可用。但有，还一个问题还是没有解决，如果扩缩容的话，一致性哈希仍然会丢失数据。



这时我们发现业界有一个比较成熟的开源 KV 存储：也就是阿里巴巴的 Tair 。2014 年，我们把 Tair 引入到技术内部，去满足业务 KV 存储方面的需求。Tair 开源版本的架构主要是三部分：最下边的是存储节点，存储节点会上报心跳到它的中心节点，中心节点内部设有两个配置管理节点，会监控所有的存储节点。如果有任何存储节点宕机或者扩容之类的行为，它会做集群拓扑的重新构建。客户端启动的时候，它会直接从中心节点引入一个路由表，这个路由表简单来说就是一个集群的数据分布图，客户端根据路由表直接去存储节点读写。之前我们 KV 遇到的扩容丢数据问题，它也有数据迁移机制来保证数据的完整性。

但是在使用的过程中，我们还遇到了一些其他问题，比如：它的中心节点虽然是主备高可用的，但它没有分布式仲裁之类的机制，所以在网络分割的情况下，它是有可能发生“脑裂”的，这种情况也给我们的业务造成过比较大的影响。在容灾扩容的时候，遇到过数据迁移影响业务可用性的问题。

另外，我们之前用过 Redis ，业务会发现 Redis 的数据结构特别丰富，而 Tair 还不支持这些数据结构。虽然我们用 Tair 解决了一些问题，但是 Tair 同样也无法完全满足我们的业务需求。于是，我们认识到在美团这样一个业务规模大、复杂度高的场景下，很难有开源系统能很好满足我们的需求。所以，我们决定在已应用的开源系统之上进行自研。



时值 2015 年，Redis 社区正式发布了它的集群版本 Redis Cluster。所以，我们紧跟社区步伐，并结合内部需求做了很多自研功能，进而演进出本文要介绍的全内存、

高吞吐、低延迟的 KV 存储 Squirrel。另外，我们基于 Tair，加入了很多美团自研的功能，演进出本文要介绍的持久化、大容量、数据高可靠的 KV 存储 Cellar 。

Redis 社区一直都很活跃，所以，Squirrel 的迭代是自研和社区并重，自研功能设计上也会尽量与社区架构兼容。Tair 开源版本已经多年没有更新，所以，Cellar 的迭代完全靠自研。后续内容上大家也能看到，因为这方面的不同，Cellar 和 Squirrel 在解决同样问题时可能会选取不同的方案。

这两个存储其实都是 KV 存储领域的解决方案。实际应用上，如果业务的数据量小，对延迟敏感，建议用 Squirrel ；如果数据量大，对延迟不是特别敏感，我们建议用成本更低的 Cellar 。

2 大规模 KV 存储的挑战

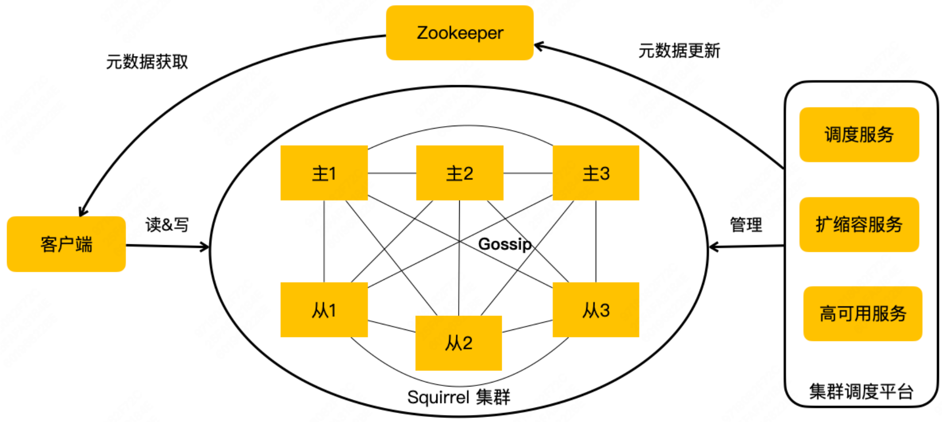
大规模 KV 存储的业务挑战主要有两点：

一个是扩展性。随着业务规模持续变大，业务会要求使用容量更大的集群。这个容量包括两方面，一方面是数据量，还有一方面是调用量。扩展容量，最常见的方法就是把集群水平扩展到更多的节点，但是当集群节点数达到一定规模后，再想扩展新节点也会遇到很多困难，这是扩展性上的第一个挑战。

还有一个问题是有些业务场景的调用容量是无法随着集群水平扩展而扩展的。比如，很多业务会使用 mget 进行批量读取。但随着集群节点数的增加，由于“木桶效应”，整个 mget 请求的长尾延迟会越来越高，进而导致服务的请求超时率持续上升。等集群达到一定规模之后，长尾延迟造成的可用性降低就超出业务的承受能力了。所以在水平扩展之外，我们还需要解决好节点垂直扩展上的挑战，来支持这种批量操作的业务场景。

另一个是可用性。随着集群规模变大，要保证可用性维持在与小规模集群同等的水平，其实是很困难的。但业务服务却不会因为集群规模变大而能接受可用性有所降低。所以，美团的挑战是如何保证集群可用性不会随着规模的变大而有所降低。

3 内存 KV Squirrel 挑战和架构实践



上图是美团的 Squirrel 架构。中间部分跟 Redis 社区集群是一致的。它有主从的结构，Redis 实例之间通过 Gossip 协议去通信。我们在右边添加了一个集群调度平台，包含调度服务、扩缩容服务和高可用服务等，它会去管理整个集群，把管理结果作为元数据更新到 ZooKeeper。我们的客户端会订阅 ZooKeeper 上的元数据变更，实时获取到集群的拓扑状态，直接对 Redis 集群节点进行读写操作。

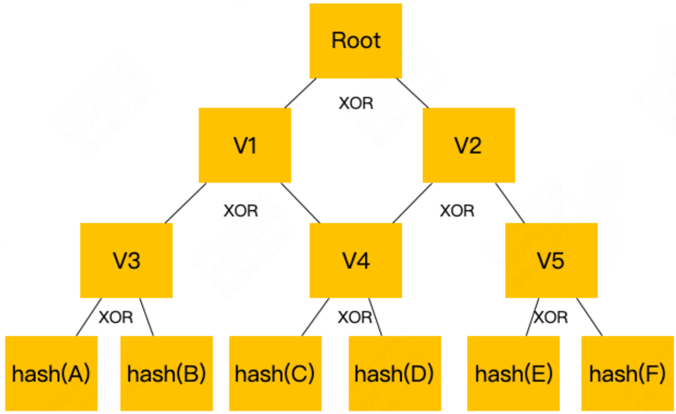
3.1 Squirrel 水平扩展的挑战

但是基于 Redis Cluster 架构的水平扩展，会有如下问题：

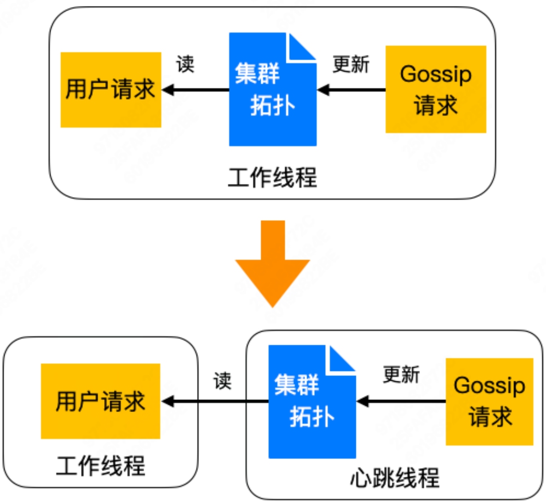
一个是 Gossip 的消息通信量是节点数的平方，随着集群节点数的增加，Gossip 通信的消息量会急剧膨胀。比如，我们实测对于一个 900 节点的集群，Gossip 消息的 CPU 消耗会高达 12%，远高于小集群的 Gossip 资源消耗，这样会造成极大的资源浪费。

除了资源的浪费以外，Gossip 消息过多，也会更多抢占用户请求处理线程的资源，进而会导致用户请求经常被 Gossip 消息的处理所阻塞，再导致用户请求产生更多的超时，影响服务可用性。

3.2 Gossip 优化



为了解决上述的扩展性问题，我们对社区的 Gossip 方案进行了优化。首先针对 Gossip 传输的消息，我们通过 Merkle Tree 对其做了一个摘要，把集群 Gossip 通信的数据量减少了 90% 以上。服务端节点仅需要对比 Hash 值即可判断元数据是否有更新，对于存在更新的情况也能快速判断出更新的部分，并仅对此部分元数据进行获取、更新，大幅降低了 Gossip 消息处理的资源消耗。同时，我们还增加了一个周期性的元数据全量同步功能，来解决可能因 Hash 冲突导致元数据无法更新的问题。



针对上述提到的 Gossip 消息处理影响业务请求的问题，我们把 Gossip 消息处理功能剥离到一个单独的心跳线程里，并且由心跳线程来更新集群拓扑的元数据。对于处理用户请求的工作线程，仅需要对元数据进行读操作，可以做到无锁读。这样的话，Gossip 请求处理就对业务请求完全没有影响了。

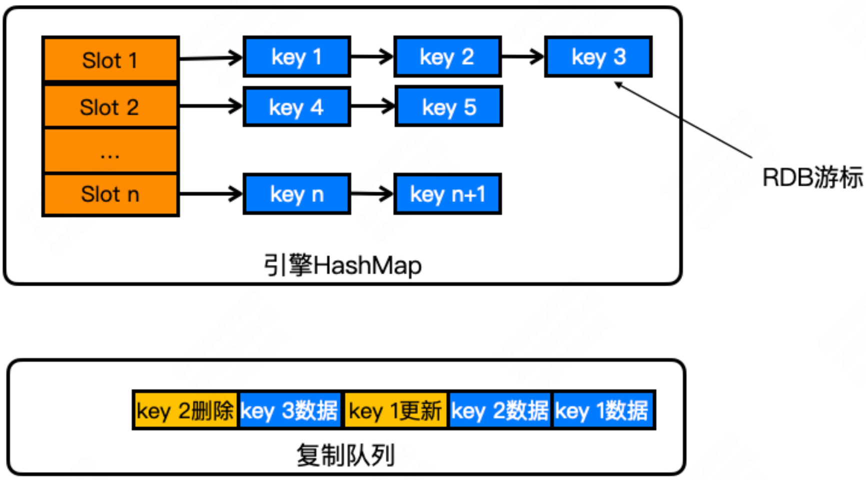
3.3 Squirrel 垂直扩展的挑战

对基于 Redis 研发的 Squirrel 来说，垂直扩展会存在如下问题：

首先是数据容量的问题。对一个内存存储来说，节点容量过大的话，很容易影响服务的可用性。例如，在主从节点要做数据同步时，Redis 节点需要通过 fork 产生子进程来生成全量数据的 RDB 快照。当一个 8GB 的节点做 fork 调用时，会由于页表项过多，造成进程出现 500 毫秒的阻塞。对于平均耗时只有几毫秒的 KV 请求来说，这 500 毫秒的阻塞会造成大量的超时。

还有就是处理量的扩展问题。虽然我们可以通过加从库去扩展集群的读能力上限，但主库的写处理能力却还是无力扩展的。而且，受限于主库的处理能力和机器带宽限制，加从库来扩展读能力也是有上限的。

3.4 forkless RDB



针对上述节点过大，fork 生成 RDB 会导致可用性降低的问题。我们实现了 forkless RDB 方案，这是一个不基于 fork，且不会中断服务的生成数据快照 RDB 的方案。

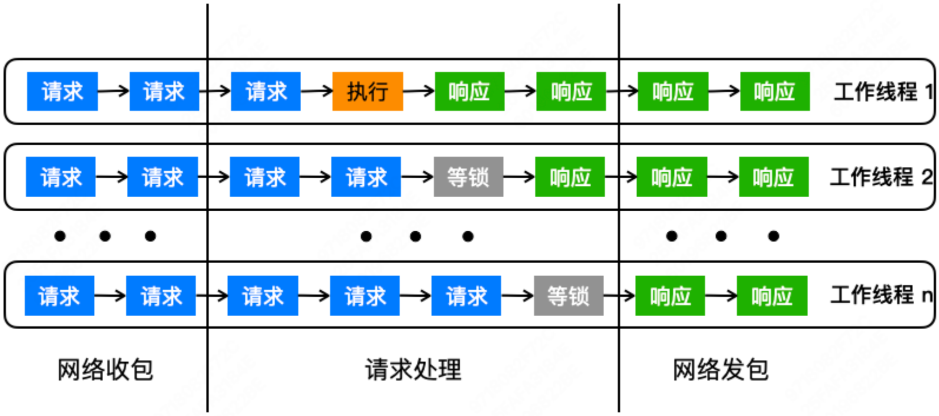
如上图所示，forkless RDB 的生成期间，它首先会停止哈希表的 rehash 过程，避免数据在哈希表之间的搬迁影响快照的一致性。然后，它会从头开始对整个哈希表的 key 做迭代，每迭代一个 key 就会把它 dump 一份出来放到复制队列里边。在迭代 key 的同时，它会对迭代的位置记录一个游标。

如果在迭代哈希表的过程中，里面的 KV 有变更的话，在这个游标之前的 KV 变更，也会把它放到复制队列里边，确保已经复制的 KV 能够持续获得后续的变更。如图所示，RDB 游标在 key 3，它会把之前已经迭代过的 key 1 更新、key 2 删除操作也插入到复制队列里边。在游标之后的 key，因为还没有做数据复制，所以等后续迭代到这个 key 时，把其最新值 dump 到复制队列就好。通过这样的方式，就实现了一个不需要 fork 就能获得一个一致性数据快照 RDB 的过程。

这个方案的优点很明显，生成 RDB 的过程不会阻塞服务请求处理，并且因为是实时的发送一个个 KV 数据，所以就不需要等 RDB 生成好就可以向从库复制数据了，大幅提升了数据同步的速度。但因为全量数据迭代、复制是在工作线程去做的，而不是在子进程内。所以，该方案会占用一部分工作线程的资源。另外，因为是以 KV 为粒度做复制的，所以，如果哈希表里面有大 KV 的话，可能会因为工作线程复制大 KV 耗时过长，造成用户请求等待耗时的上升。

3.5 工作多线程

对于处理量的扩展，社区有一个 IO 多线程的解决方案。但这个 IO 多线程只是把网络收发部分做了多线程处理，所以，其扩展能力是比较有限的。比如 4 个 IO 线程下，它只能把整体的吞吐提升一倍，就到极限了。而且因为此时工作线程已经到瓶颈了，再往上去加 IO 线程，不仅无法提升性能，反而会消耗更多的 CPU 资源。对此，我们的解决方案是工作多线程，也就是说把请求处理的过程也多线程化。



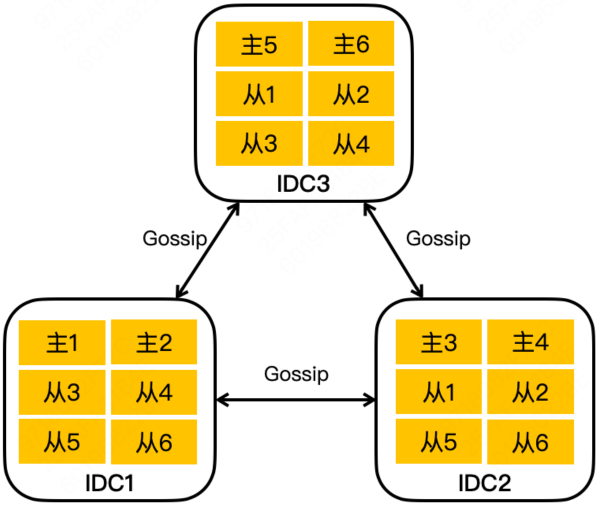
如上图所示，在工作多线程方案下，每个线程都会去处理请求，并且每个线程会完成从收包到请求处理，然后到发包的整个过程，是一个 Run-to-Completion 线程模型。相比 IO 多线程，它会减少很多线程切换，节省很多的 CPU 资源。同时对于请求处理的过程，我们也通过细致的梳理，尽量缩小了临界区的范围，以保证大部分的请求处理过程是在临界区之外的，来提升处理并发度。

如果一个工作线程需要加锁的话，它会先 try lock。如果加锁成功就继续执行了，但如果加锁失败的话，这个工作线程也不会阻塞等锁。它会先去注册一个管道的通知消息，然后就继续处理网络的收发包，还有非临界区的请求了。等到锁被释放的时候，这个工作线程会通过 epoll 获得管道里面的锁释放通知，然后去拿到这把锁。这个时候它就可以去处理临界区的请求操作了。

这样的话，在整个加锁、解锁的过程中，工作线程没有任何阻塞，仍然可以继续做网络收发、非临界区请求的处理，获得最大限度的处理能力。另外，对于新建 socket、数据复制等工作，跟工作线程的耦合很低，我们将其放到了单独的线程去执行，以尽量降低工作线程的负载。

通过实测，工作多线程方案的吞吐比社区 IO 多线程提升了 70%，相对于社区单线程提升 3 倍多。

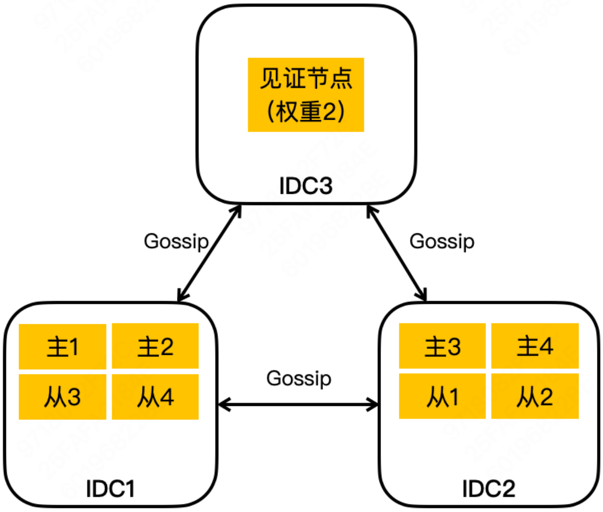
3.6 Squirrel 可用性的挑战



基于 Redis Cluster 的大规模集群可用性挑战主要是维持机房容灾部署很困难。如上图所示，由于 Redis Cluster 是去中心化的架构，所以部署上要求至少是三机房分布，以此来保证任何一个机房挂掉的时候，剩余的两个机房仍然能有过半的节点来选出新的主节点。比如一个上千节点的集群要扩容的话，可能需要几百个分布在三个机房的节点，一时之间其实很难凑齐这么多机房的资源。而当业务大促容量需求很急时，我们有时候只能牺牲机房容灾能力来满足业务的容量需求。

还有在成本方面，对于一些数据可靠性要求较低的业务，只需要两副本冗余就够了，极端情况下丢一点数据也是可以接受的。但受限于容灾要求，这些业务也只能使用三机房三副本部署，从成本角度考量很不划算。

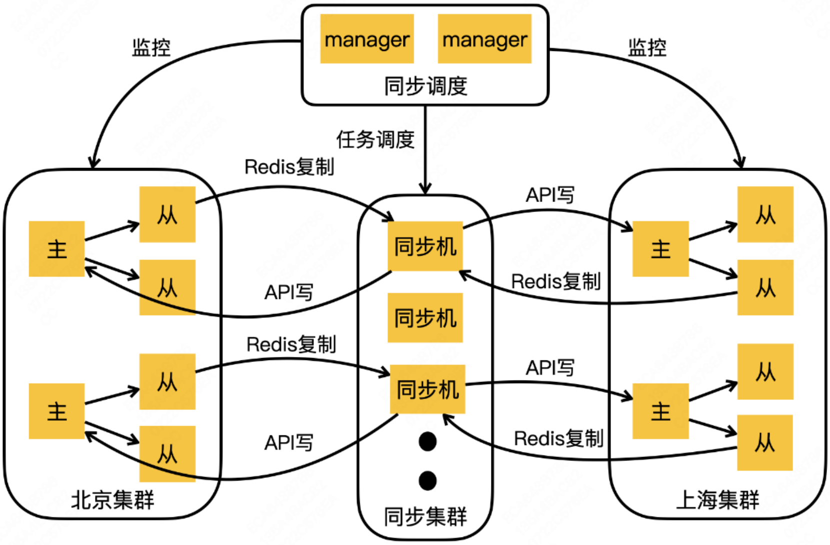
3.7 两机房容灾



受 Google Spanner 的见证者节点启发，我们在 Squirrel 集群也引入了见证者节点角色。同 Spanner 一样，Squirrel 见证者节点也不会存储数据，所以，它无法作为正常的主从库提供请求处理能力，也不能发起选主投票。但见证者节点可以在集群选主时参与投票，帮助存活的机房节点完成过半选主过程。

见证者节点还可以设置权重，这样只需要一个或几个高权重见证者节点，就能满足一个大规模集群的容灾部署需求了。由于见证者节点不存储数据，且节点数很少，虽然集群还是三机房部署，但实际几乎只需要两机房的资源就能满足机房容灾部署需求了，这样就大幅降低了集群维持容灾部署的难度，从而节省大量的机器成本。

3.8 跨地域容灾



Squirrel 跨地域容灾的架构如上图所示，它通过一个集群间同步服务在两个不同地域的集群之间做数据同步。这个同步服务首先伪装为上游集群节点的 slave 把它的 RDB 和增量 log 拉取过来，然后再把拉取到的数据转化成写请求发到下游的集群，从而实现了一个集群间的数据同步。通过这样的架构，我们解决了服务的跨地域容灾问题。并且，通过在集群间搭建正反两个方向的两个同步任务，就能实现集群间的双向同步。这样的话，用户服务就可以只在本地域写，但同时能读到两个地域分别写入的数据，解决了单向同步需要跨地域写的问题。

双向同步有两个经典问题需要解决：

一个是循环复制问题。我们为每个 Squirrel 集群标记了不同的 cluster id，并且记录了每个 KV 的初始写入 cluster id，同步服务会过滤掉与目标集群 cluster id 相同的数据，以避免发生循环复制。

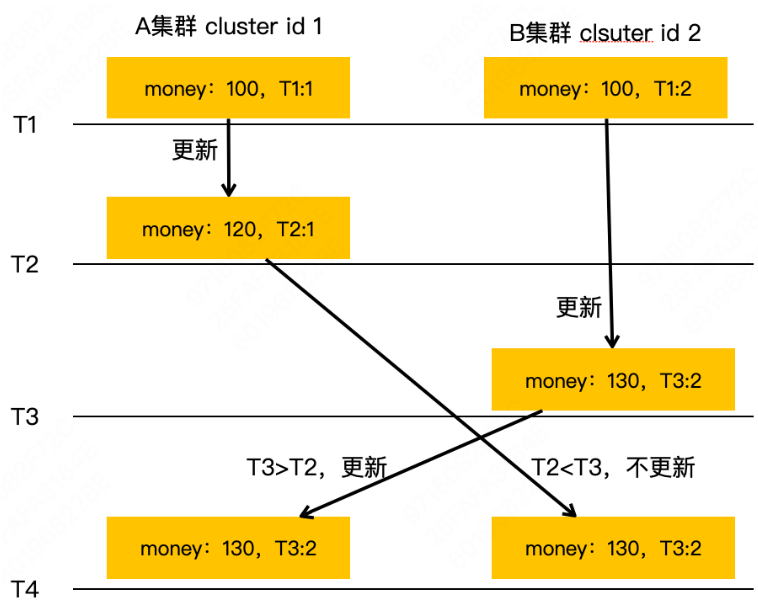
还有一个是数据冲突问题。我们一开始是通过业务层面保证在每个地域写不同的 Key 来解决的。但是在双向同步的运行过程中，还是会有一些极端场景可能会出现两个地

域并发写同一个 Key。比如像机房网络故障场景，业务会把故障机房的所有写入都切到正常机房。

但由于我们的集群间复制是异步的，可能故障机房有一些最新的 Key 变更还没有复制到正常机房的集群。而如果在业务将写切换到正常机房后，又写入了相同 Key 的不同变更，就会产生两个同步集群的数据冲突。在机房网络恢复之后，业务还是要把一部分流量切回到之前故障的集群上，恢复到跨地域容灾的架构。

但由于两个集群可能已经有数据冲突了，所以，在业务切回之前，就需要对数据做冲突校验和修复。但是对大数据量集群来说，数据校验和修复的耗时可能会长达数天。在这样长的时间内，只有一个单地域集群来支撑业务，无论是从容灾还是容量的角度来看，都是有较大风险的。

3.9 双向同步冲突自动解决



为了解决上述的双向同步数据冲突问题，我们实现了一个基于数据写入本地时间的 last write win 冲突自动解决功能。

如上图所示，在 T1 时刻 Key money 的值在 A、B 两个集群都是 100。T2 时刻，money 的值在 A 集群更新成了 120。但是在 A 集群的新值还没复制到 B 集群的时候，B 集群在 T3 时刻把 money 的值更新成了 130。这时候 A、B 集群会互相向对方复制各自写入的新值，A 集群收到 B 集群的值 130 后，会发现 B 集群 money 的更新时间大于自己（T3 > T2），它就会更新自己的 money 值为 130；B 集群也会收到 A 集群复制过来的 money 值 120，但它会发现这个值的更新时间小于自己本地值的更新时间（T2 < T3），就会忽略这个复制请求。通过这样一个基于更新时间的 last write win 策略，就可以达到最终一致性。

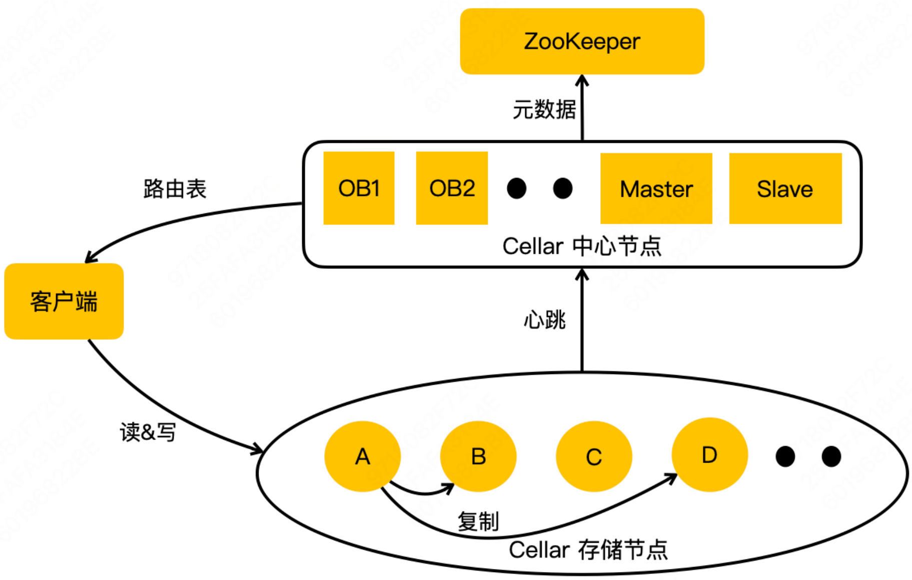
上述方案看起来简单，但是在复杂、大规模的业务场景下，还有很多问题要处理，所以，我们还做了以下的工作：

- **保存最近更新的时间戳**：当发生时钟回退时，我们会继续使用自己保存的时间戳，避免使用本地回退的时间导致数据也跟着发生了回退。（PS：对于时钟回退问题，我们调研过最新的 NTP 时钟同步不会像以前一样造成本地时钟的回退或跳变，现在它通过把时钟 tick 调快或调慢来完成类似的调整，所以，前述关于时钟回退的解决方案在最新的 NTP 同步机制下就不是必要的了。不过，为了保证我们的服务在任何系统下都能正常运行，我们最终还是实现了这个功能。）
- **记录写入数据的集群 id**：我们会为所有写入的 Key 保存写入的集群 id。当两个值的更新时间相同时，我们会比较集群 id，如果也相同，我们就知道是同一个集群先后写入但获取到相同本地时间的数据，会允许其写入；如果不同，我们仅会让集群 id 更大的值写入，来保证数据最终一致性。
- **由复制操作改为复制变更后的数据**：像 INCR 类接口，A 集群的 money T1 时刻通过 INCRBY money 20 变成了 120，然后 B 集群 T2 时刻通过 INCRBY money 30 变成了 130。A 集群收到 B 集群的复制时，因为时间戳比自己的本地值大，它会执行 INCRBY money 30 变成 150；然后 B 集群收到 A 集群的复制时，因为时间戳比自己的本地值小，它会把这个复制请求给忽略掉，就造成了数据冲突。针对这个问题，我们将所有操作的数据复制都改成

了复制操作后的数据，而不是这个操作本身，来解决类似 INCRBY 这种接口的数据冲突问题。

- **保存最近删除的 Key**：像删除类接口，A 集群 T2 时刻写入了 money：120，然后 B 集群在 T3 时刻删除了 money 这个 Key。A 集群收到 B 集群的复制时，由于其时间戳比本地值大，A 会把数据删了；但 B 集群收到 A 集群的复制时，由于本地已经不存在 money 这个 Key 了，它就会把 money 当做一个新 Key 进行写入，就造成了数据最终不一致。针对这个问题，我们通过保存最近一段时间删除掉的 Key 及删除时间戳，以便在删除集群收到对端复制过来的旧 Key 时进行甄别。

4 持久化 KV Cellar 挑战和架构实践



上图是我们最新的 Cellar 架构图，它跟阿里开源的 Tair 主要有两个层面的不同。

第一个是 OB，第二个是 ZooKeeper。我们的 OB 跟 ZooKeeper 的 Observer 是类似的作用，提供 Cellar 中心节点元数据的查询服务。它实时的与中心节点的 Master 同步最新的路由表，客户端的路由表都是从 OB 去拿。这样做的好处主要有

两点：第一，把大量的业务客户端跟集群的大脑 Master 做了隔离，防止路由表请求影响集群的管理；第二，因为 OB 只提供路由表查询服务，不参与集群的管理，所以它可以水平扩展，极大地提升了路由表的查询能力。

第二个是我们引入了 ZooKeeper 做分布式仲裁，解决了上述提到的 Master、Slave 在网络分割情况下的“脑裂”问题。并且通过把集群的元数据存储到 ZooKeeper，从而提升了元数据的可靠性。

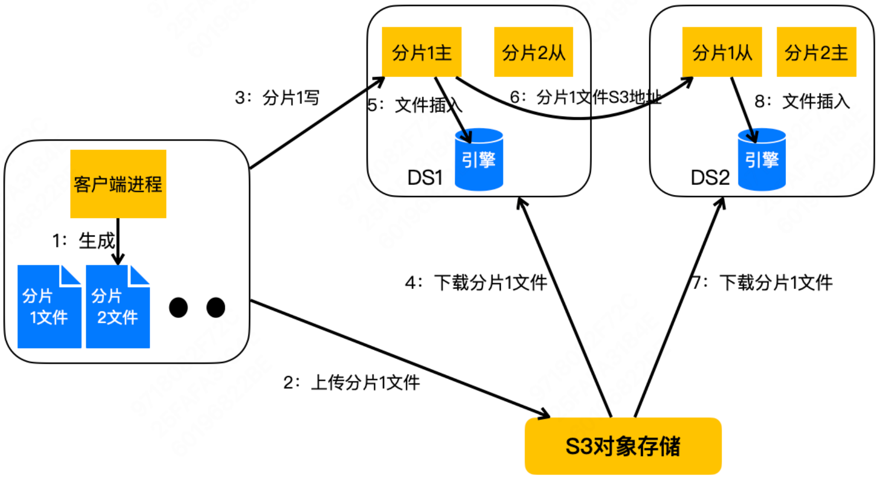
4.1 Cellar 垂直扩展的挑战

在 Cellar 架构下，不存在水平扩展的问题，但与 Squirrel 一样，它也有垂直扩展方面的挑战。而由于 Cellar 是持久存储，它也很少遇到单机数据容量的问题，而要解决的问题主要是处理容量的垂直扩展。而且，由于 Cellar 是持久化引擎、多线程模型，它要解决的处理容量扩展问题也是不一样的，具体如下：

- **引擎读写能力的不均衡性**：Cellar 是基于 LSM-Tree 引擎模型的持久化存储，这种引擎的多 Level compaction 会导致写放大问题，进而会造成其写处理能力比读低很多。所以，在一些写相对较多的场景，机器资源虽然还有空闲，但写处理能力却已经到瓶颈了。
- **线程间同步的开销**：想要提升处理容量，就需要增加线程数。而随着线程数的增加，线程间同步的开销在整个服务的 CPU 使用占比也会越来越高。所以，如果解决不好线程间同步的问题，想单纯地增加线程数来提升处理容量行不通。

4.2 Bulkload 数据导入

对于上述提到引擎写压力达到瓶颈的集群，我们调研后发现其在线的实时写入一般都是比较少的，高写入量主要是用户从离线批量写数据到线上 Cellar 集群带来的。基于此，我们开发了 Bulkload 数据导入能力来解决这个问题。



Bulkload 整体架构如上图所示，它在普通写入流涉及的客户端和存储节点之外，还引入了 S3 对象存储来做导入数据的中转。下面我们看下 Bulkload 具体的写入流程：Bulkload 首先会在客户端进程内生成分片内有序的数据文件并写到本地硬盘上。等客户端的数据文件写好之后，它会上传到对象存储，利用对象存储做数据文件的中转，解决了客户端与服务端之间直传大文件容易失败的问题。

分片 1 的数据文件写入到对象存储之后，客户端会将数据文件的存储地址告诉分片 1 的主所在的存储节点 DS1。然后 DS1 就会从对象存储下载分片 1 的数据文件，并把它直接插入到 LSM-Tree 引擎里面。因为这是一个完整的文件插入，所以，它可以消除引擎在普通写入时的内存排序和刷盘压力。同时，因为这个文件的数据是分片内有序的，所以，它在参与 Level 间 Compaction 时会与其他的引擎文件交叉很少，可以大幅减少多 Level compaction 的压力。

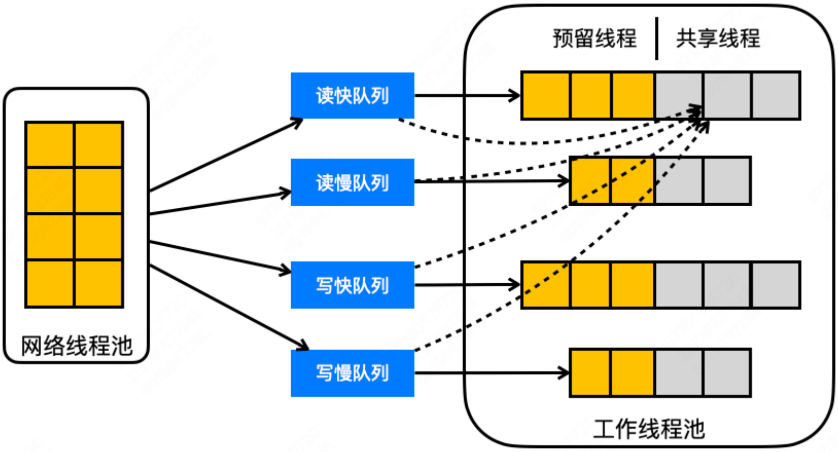
然后 DS1 会把分片 1 数据文件的对象存储地址复制发送到分片 1 的从所在的存储节点 DS2 。因为存储节点的复制只是传输数据文件的地址，所以复制速度是特别快的，也节省了很多传输的带宽。DS2 收到了分片 1 的地址后同样会从对象存储下载数据文件，并插入到引擎里面。

通过 Bulkload 解决方案，我们整体把数据离线导入的性能提升到旧版的 5 倍。比如

我们的一个存储广告特征的客户使用 KV 方式从离线导数据到在线需要 14 小时，受限于在线高峰期无法导数据，如果需要继续增加特征数据，就需要扩容集群了。而扩容集群一方面会因为“木桶效应”导致请求长尾延迟问题，另一方面 Cellar 成本的上升也会抵消一部分广告收益。而在 Bulkload 功能加持下，该客户导入相同规模数据仅需不到 3 小时，它可以在不增加 Cellar 资源的情况下，将广告特征规模增加数倍，大幅提升了广告的效果。

4.3 线程调度模型优化

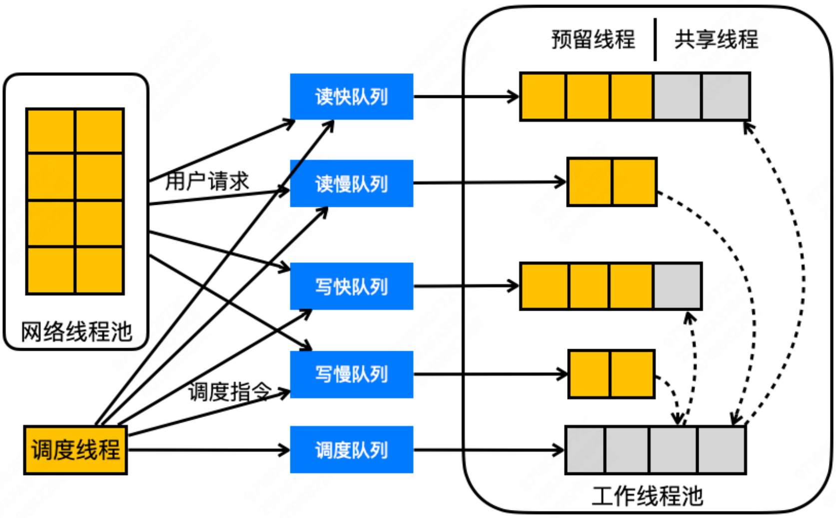
我们最初的线程模型与开源版 Tair 一样，网络线程池做收发包，收到的包经过一个队列转出到一个大的工作线程池做请求处理。这样的线程模型，很容易发生请求间的互相影响。比如用户有离线数据导入到 Cellar 的时候，就很容易导致在线读请求的超时。又比如当有大 Value 读写的时候，工作线程处理会比较慢、占用线程的时间会很长，导致正常 Value 读写的快请求只能在队列等待，进而导致大量超时。



所以，为了隔离在离线请求、快慢请求的处理，让服务资源优先保证核心流量的处理，我们后来把线程模型改造成如上图所示的 4 个队列 + 4 个线程池的结构，将请求分成 4 类（读快、读慢、写快、写慢）分别放到不同的队列和线程池去处理，进而来提升服务核心流量的可用性。

但是，工作线程池按照请求类型分离之后带来一个问题，就是不同业务场景、甚至同一业务的不同时段，不同类型请求量的占比是不一样的。所以，给每个线程池分配多少线程是一个很棘手的问题。

针对这个问题，我们增加了一个线程动态调度的逻辑：每个线程池都有一部分线程被设定为可共享线程，如果线程池比较空闲，共享线程就会去轮询其他的队列，处理一些繁忙线程池的请求，这样就达到了自适应调整各线程池资源的效果。但是在这样的架构下，虽然解决好了请求隔离性和不同请求类型线程资源的动态分配问题，但我们发现随着节点流量的上涨，共享线程对于其他队列的轮询会消耗越来越多的 CPU 资源，而且集群业务的负载分布与默认的线程数设置差异越大，这个消耗的占比也会越高。



为了解决上述线程池资源自适应调度带来的 CPU 消耗问题，我们对分离后的线程、队列模型做出了如上图的改造。改进后的线程模型最主要的特点是引入了一个调度线程和一个空闲线程池，这个调度线程会实时统计每个线程池的负载，来评估每个线程池是否需要增加或减少线程并做出调度动作，空闲线程池用来存放当前空闲的可用于调配的线程资源。

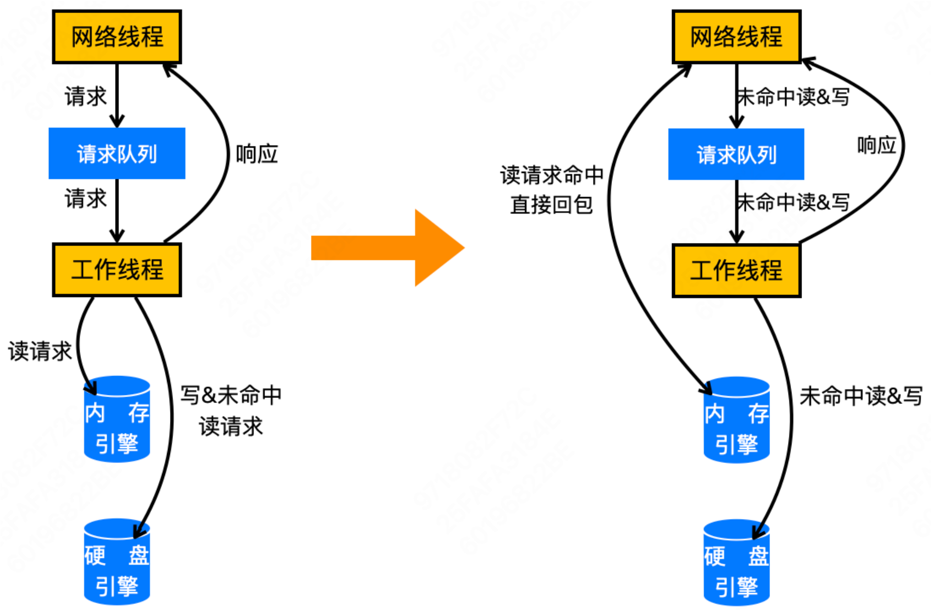
当调度线程评估后决定做线程资源调配时，它就会发送调度指令到相应队列中，当线

程池里的线程获取并执行了这个指令后，就实现了线程资源的调配。比如，它想给读快线程池增加线程，就会给空闲线程池的队列发送一个调度指令，空闲线程池的线程取到这个指令后，就会将自己加入到读快队列的线程池里面，去处理读快队列的请求。

当调度线程想对读慢线程池调减线程时，它会向读慢队列发送一个调度指令，读慢队列的线程获取到这个指令后，就会离开读慢线程池加入到空闲线程池。通过调度线程准实时的毫秒级负载统计、调度，我们实现了线程池资源的快速动态分配。对于每一个线程池的共享线程，也不再需要去轮询其他线程池的队列了，只需要专心处理自己队列的请求即可，大幅降低了线程池资源调度的 CPU 消耗。

通过上述的线程队列模型优化，服务在高负载场景下可以提高 30% 以上的吞吐量。

4.4 线程 RTC 模型改造



上图左侧画的是我们服务请求的 IO 处理路径：一个请求的处理流程会经过网络线程、请求队列、工作线程、内存和硬盘引擎。这个设计的问题是，请求在不同线程之间

流转会造成大量的 CPU 切换以及 CPU 高速缓存的 Cache Miss，进而造成大量的 CPU 资源消耗。在大流量场景下，这样的 CPU 消耗也是很可观的一笔资源。

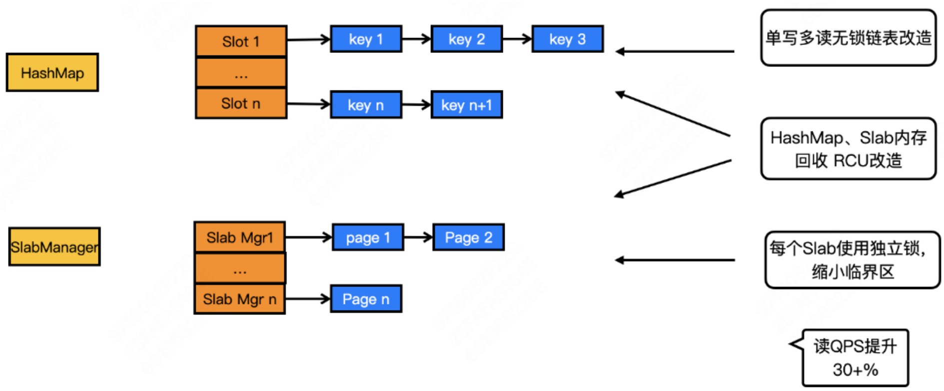
针对这个问题，我们对线程队列模型又做了如上图右侧所示的改造。新的模型下，我们让网络线程直接去做读请求的处理，对于能够命中内存引擎的读请求，其处理模型就是一个 RTC（Run-to-Completion）模型。

具体来讲，当网络线程收到一个请求之后，会先判断是否为一个读请求，如果是，就会直接去读内存引擎。我们服务的内存引擎会缓存硬盘引擎上的热点数据，如果内存引擎命中的话，网络线程就可以直接返回结果给客户端。这样在网络线程内就实现了请求的闭环处理，相比原来的模型可以去除所有因请求流转造成的 CPU 资源消耗。而对于写和读未命中内存引擎的请求，仍然需要经过原来的请求处理路径，去硬盘引擎读或者写数据。

新的线程模型，经实测在 80% 内存引擎命中率场景下，服务读吞吐可以提升 30%+。虽然新的线程队列模型只实现了读缓存命中请求的 RTC，但其实在线流量大多都是读多写少且热点数据明显、内存引擎命中率比较高的场景，所以，新模型上线后在大多数的业务集群都取得了明显的性能提升。

4.5 内存引擎无锁化

当单机请求量达到了一定规模之后，我们发现服务内的锁操作会占用很多的 CPU 资源。经分析发现，大多数的锁操作都发生在上节内容提到的内存缓存引擎上。如上节所述，所有请求都会经过内存引擎，且大部分请求都会在内存引擎命中并返回结果给客户端。所以，大部分请求都是纯内存处理，这个过程中的锁操作就很容易成为瓶颈。针对这个问题，我们对内存引擎做了无锁化改造，其改造后的结构如下图所示：

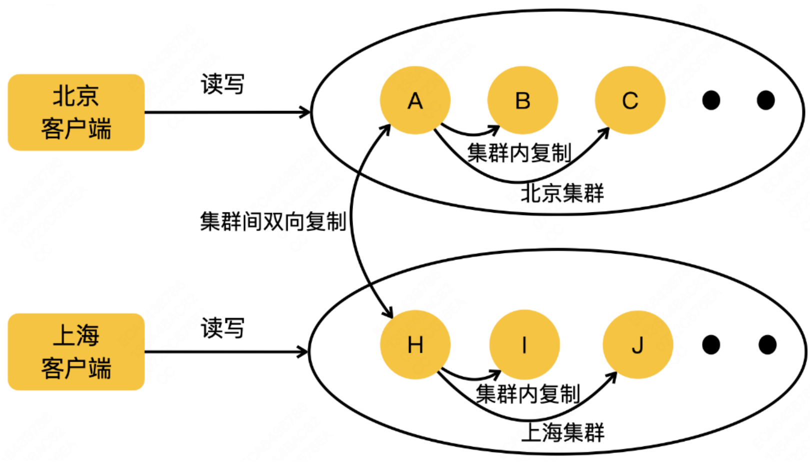


整体改造主要跟上图的 HashMap 和 SlabManager 两个数据结构有关（其他数据结构在图中已略掉）。HashMap 是存储 KV 数据的核心结构，它把 Key 通过 Hash 算法散列到不同的 Slot 槽位上，并利用链表处理 Hash 冲突；SlabManager 管理不同尺寸内存页的申请和释放，它利用链表把相同尺寸的内存页放到一起管理。

对于 HashMap，我们做了单写多读的无锁链表改造。同时，通过引入 RCU 机制实现了异步的内存回收，解决了读请求与写请求内存释放操作的冲突，实现了读请求处理全程的无锁化。写请求虽仍需要加锁，但我们对写做了锁粒度的优化，可以大幅提升并发度。比如我们把 SlabManager 的访问由一把大锁改成每个内存尺寸的管理链表单独一把锁，这样在分配和释放不同尺寸内存页的时候就可以实现并发。同时 RCU 机制下的内存异步回收，也解决了写线程回收内存时可能被阻塞的问题，进一步提升了写性能。

内存引擎通过无锁化加 RCU 技术的改造，读处理能力提升了 30% 以上。

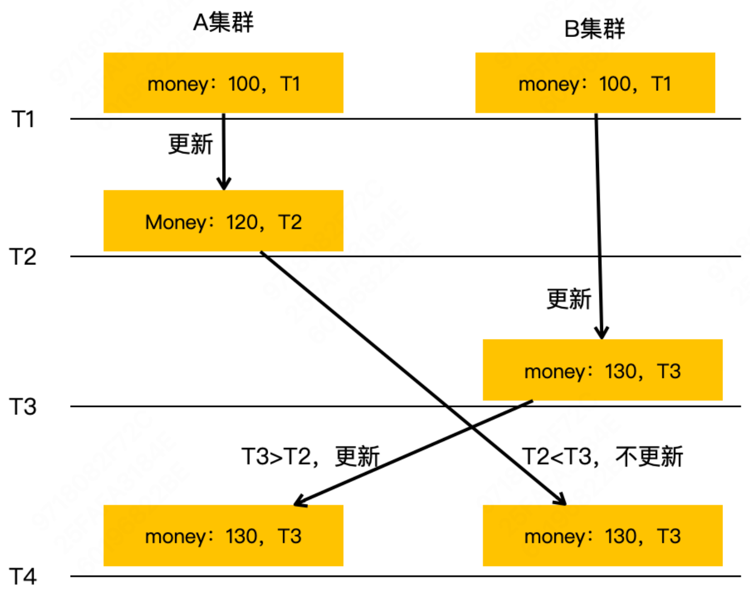
4.6 Cellar 可用性的挑战



同 Squirrel 一样，Cellar 也通过建设集群间数据同步能力，实现了跨地域的容灾架构。不同的是，Cellar 因为是自研，无需考虑与社区版本的兼容性，同时为了简化部署结构、降低运维成本，它把集群间数据同步功能做到了存储节点内部。如上图示例的北京集群 A 节点、上海集群 H 节点，在接收到写入之后，除了要做集群内的数据同步以外，还需要把写入数据同步到跨地域的另一个集群上。

Cellar 也可以通过配置两个方向的跨集群数据同步链路，实现完全的本地域读写。Cellar 由于采用了存储节点内建的方案，它的集群间复制通过使用定制的复制包来甄别客户写入和复制写入，并只为客户写入生成复制 log 来避免循环复制，相对 Squirrel 会简单一点。但同样的，这种架构也会遇到极端情况下，双向同步导致的数据冲突问题。

4.7 双向同步冲突自动解决



如上图所示，Cellar 也实现了类似 Squirrel 的基于数据写入本地时间的 last write win 冲突自动解决方案。但 Cellar 的方案有一点区别是，它没有通过在每条数据记录 cluster id 的方式解决时钟回退、两次变更写入的本地时间相同的问题，而是引入了 HLC（Hybrid Logic Clock）时钟来解决这个问题。

因为 HLC 可以保证每个集群写入数据的时钟是单调递增的。所以，接收端是不用担心对端复制过来的数据有时间戳相同的问题。而对于两个集群分别写入，时间戳相同且 HLC 的逻辑时钟刚好也相同的情况，可以通过比较集群配置的 cluster id（不会存储到每条 KV 数据内）来决定最终哪个数据可以写入。

5 发展规划和业界趋势

未来，根据技术栈自上而下来看，我们的规划主要覆盖服务、系统、硬件三个层次。

首先，在服务层主要包括三点：

- 第一，Squirrel && Cellar 去 ZK 依赖。如前所述，Squirrel 集群变更到客户端的通知是依赖 ZK 来实现的，Cellar 的中心节点选主和元数据存储也是依赖 ZK 实现的。但 ZK 在大规模变更、通知场景下，它的处理能力是无法满足我们的需求的，很容易引发故障。所以，Squirrel 会去掉对 ZK 的依赖，改为使用公司内的配置管理、通知组件来实现集群变更到客户端的通知。Cellar 会通过在中心节点间使用 Raft 协议组成 Raft 组，来实现选主和元数据多副本强一致存储。（注：本文整理自 DatafunSummit 2023 演讲，此工作当前已完成开发，处于灰度落地阶段。）
- 第二，向量引擎。大模型训练、推理场景有很多向量数据存储和检索需求，业界很多 NoSQL、SQL 数据库都支持了向量引擎能力。KV 存储作为高性能的存储服务，如果支持了向量引擎，可大幅提升大模型训练、推理的效率。
- 第三，云原生。当前美团的 KV 服务规模很大，相应的运维成本也比较高。所以，我们计划做一些服务云原生部署、调度方面的探索，向更高运维自动化水平迈进。

其次是系统层，计划对 Kernel Bypass 技术做一些探索和研发落地，比如新版内核支持的 io_uring、英特尔的 DPDK、SPDK 技术等。由于 KV 存储是典型的高吞吐服务，它的网络 IO、硬盘 IO 压力都很大，Kernel Bypass 技术可以大幅提升服务的 IO 能力，降低访问延迟和成本。

最后是硬件层，计划对计算型硬件的应用做一些探索，比如配备了压缩卡的 SSD，可以将服务引擎层使用 CPU 做的数据压缩工作卸载到压缩卡上，释放出 CPU 资源做更高价值的计算工作。KV 服务是典型的低延迟、高网络负载的服务。所以，我们也计划对 RDMA 网络做一些探索，以期进一步降低服务访问延迟、提升网络处理能力。

领域驱动设计 DDD 在 B 端营销系统的实践

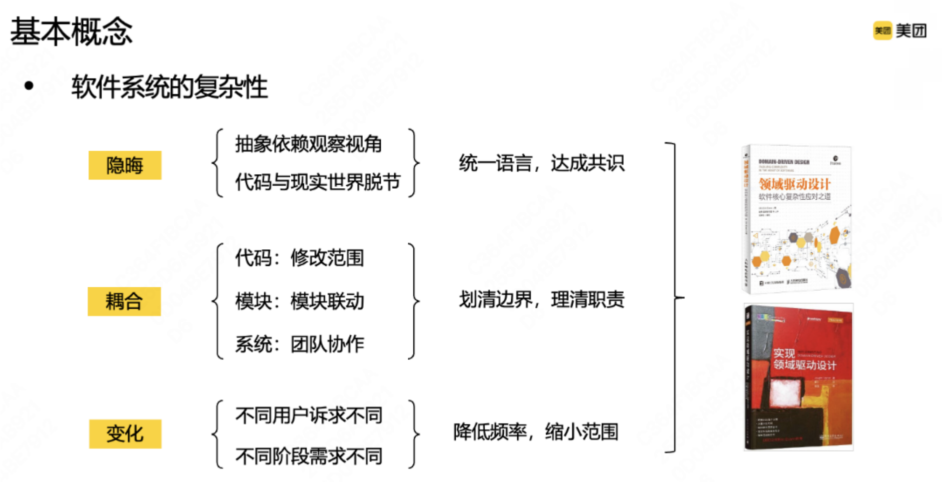
1 背景

通过营销活动实现客户 / 用户拉新、留存和促活是业界普遍采用的方法。为实现商户增长和留存，美团核心本地商业 / 商业增值技术部也构建了相应的营销系统来支撑商户的线上营销运营。在系统建设过程中，面临着业务体量大、行业跨度大、场景多样、客户结构复杂，需求多变等挑战。本文试图还原从 0 到 1 构建面向商户的营销系统过程中，并通过 DDD（领域驱动设计）来应对系统设计和建设中遇到的业务复杂度高、需求多变、维护成本大等问题。

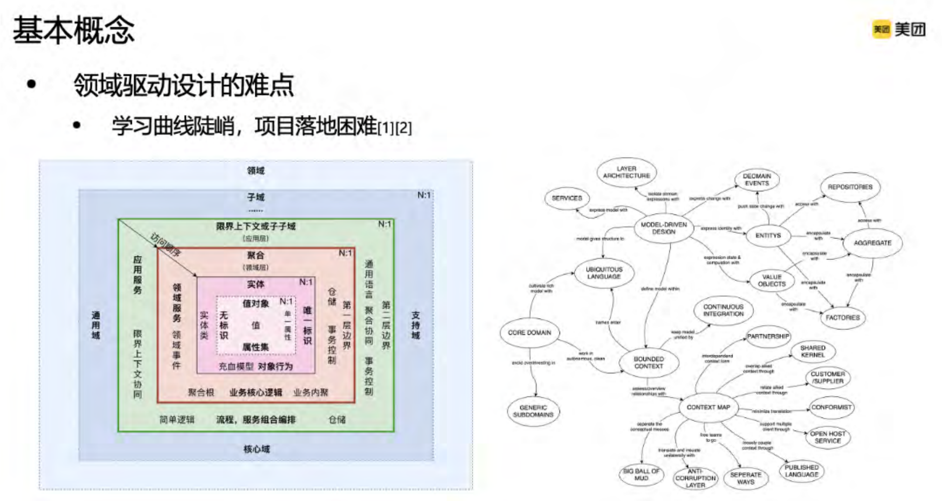
2 基本概念

软件系统的复杂性主要体现在三个方面。

- **隐晦**：一是抽象层面的隐晦，抽象系统时，每个人都有自己特定的视角，你需要站在对方的角度才能明白他为什么这么做；其次是实现层面的隐晦，代码是一种技术实现，通常与现实世界的业务概念脱节，无形中增加了理解成本。
- **耦合**：代码层面的耦合扩大了修改范围；模块层面的耦合需要跨模块 / 服务交互；系统层面的耦合则需要跨团队协作。从代码到模块再到系统，耦合的影响逐渐扩大，成本随之增加。
- **变化**：业务需求决定了系统功能，不同的用户需求不一样，不同的业务发展阶段需求在不断变化，系统功能要随着业务需求的变化不断调整，这时就涉及到系统改动的频次和范围。



DDD（Domain-Driven Design，领域驱动设计）是应对软件设计复杂性的方法之一，它能很好的解决上述三个问题，但其概念体系复杂（如下图所示），学习曲线陡峭，即便深入研读 DDD 的两本经典著作，项目落地时依然有点“捉襟见肘”。



在展开介绍 DDD 之前，这里先回顾一下历史：

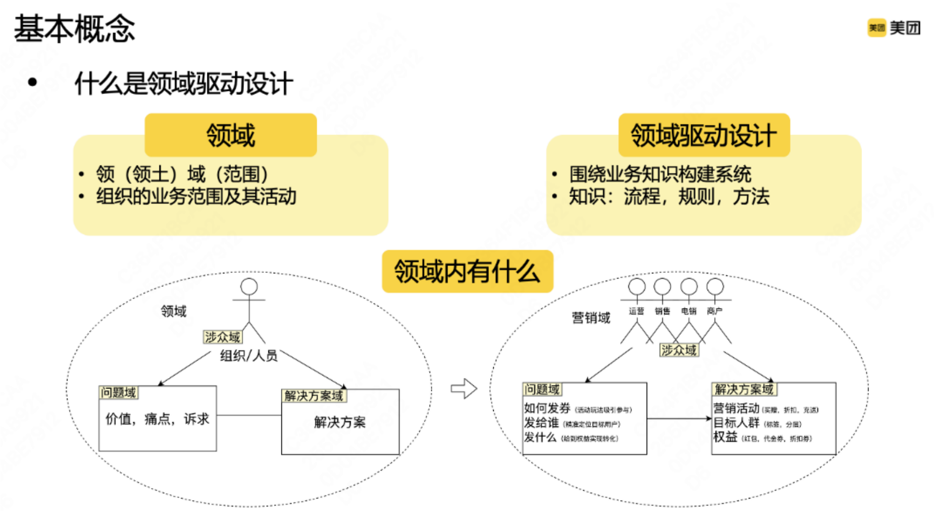
- 早期，计算机创新更多聚焦在语言方面，为软件工程师提供功能更强大的语言来操作计算机，充分使用计算机的算力。

- 60 年代，面向对象语言诞生，通过封装、继承、多态等特性进一步增强了语言的表达能力。
- 80 年代，出现面向对象的分析与设计，解决了如何构建类模型的问题，帮助我们更好地使用面向对象语言来实现系统，但没有解决如何把物理世界映射到计算机世界的问题。
- 2000 年，出现领域驱动设计方法，通过分析业务，抽取概念，建立对应的领域模型，再采用面向对象的分析与设计方法构建对应的类模型，达成了从物理世界到计算机世界的映射。

什么是领域？领域由三部分组成：领域里有用户，即涉众域；用户要实现某种业务价值，解决某些痛点或实现某种诉求，即问题域；面对业务价值，痛点和诉求，有对应的解决方案，这是解决方案域。什么是领域驱动设计？通俗地讲，针对特定业务，用户在面对业务问题时有对应的解决方案，这些问题与方案构成了领域知识，它包含流程、规则以及处理问题的方法，领域驱动设计就是围绕这些知识来设计系统。

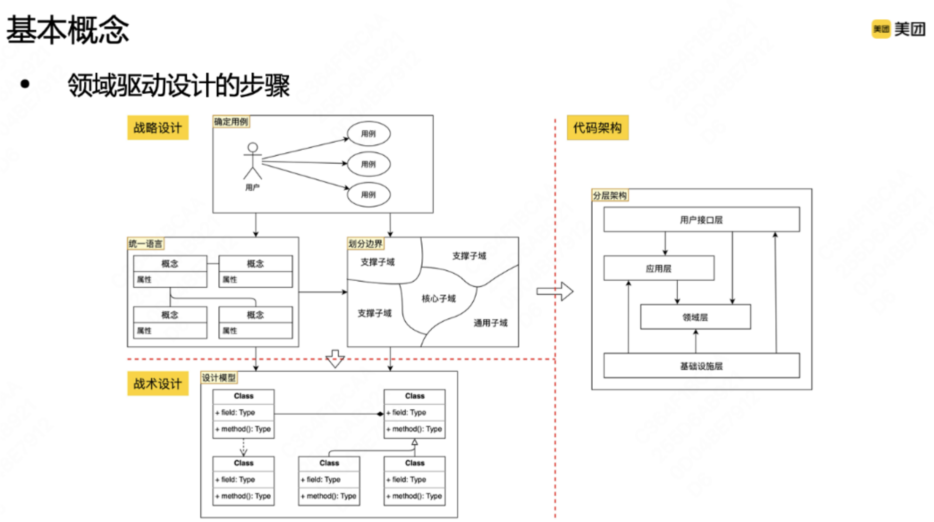
以营销为例，营销系统所服务的用户有 4 类：运营、销售、电销人员和商户。解决 3 个核心问题：如何发券、发给谁、发什么（红包还是折扣券）。解决方案：通过营销活动来承载发券，不同的活动类型对应不同的玩法（如买赠、折扣、充送等）；通过目标

人群来确定发给谁；通过权益来定义发什么（如：红包、代金券、折扣券等）。



本文将从战略设计、战术设计和代码架构分 3 个部分介绍领域驱动设计的落地：

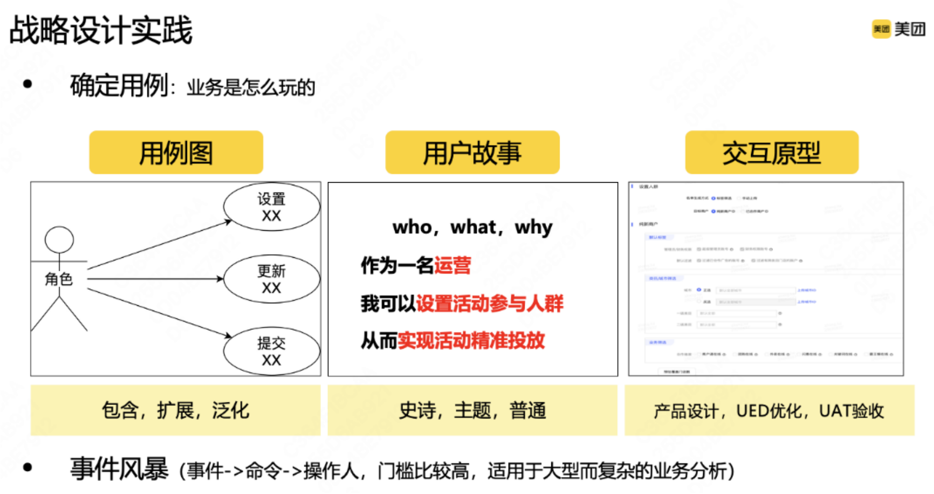
- **战略设计**：确定用例，统一语言和划分边界。
- **战术设计**：概念模型转化成类（代码）模型。
- **代码架构**：将系统设计映射为系统实现。



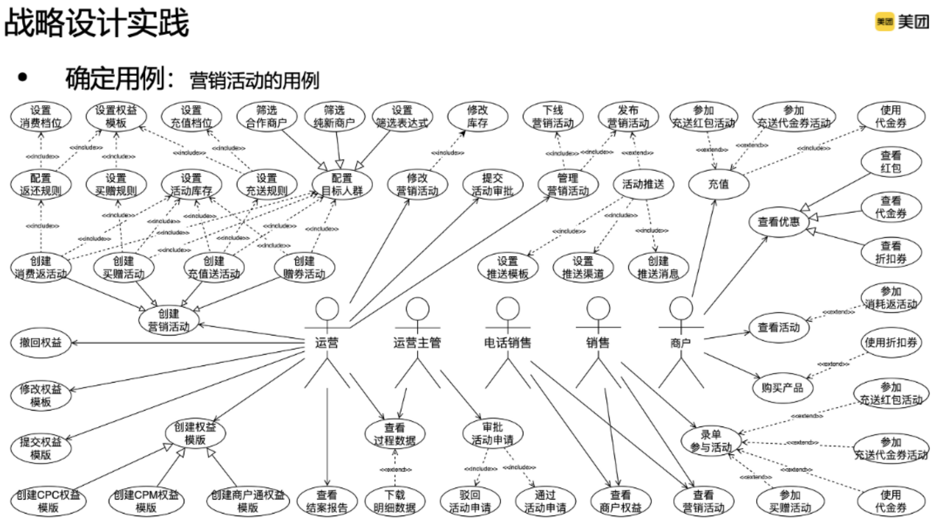
3 战略设计实践

战略设计之前，先要确定用例，也就是业务是怎么玩的，有几种常见的方法：

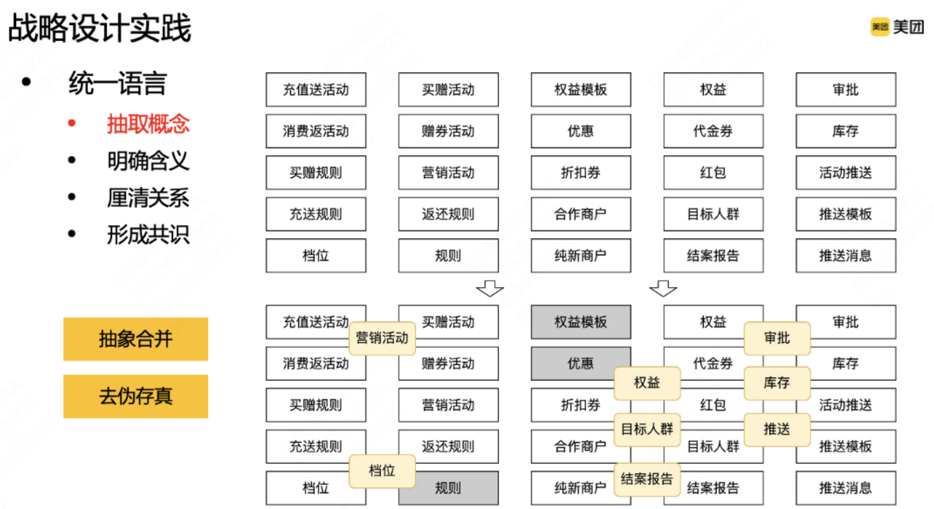
1. 用例图：最简单直观的表达了用户与系统的交互。
2. 用户故事：敏捷开发模式下用的较多，从 Who、What 和 Why 三个维度描述了业务需求。
3. 交互原型：用户操作的页面及其操作流程，其缺点是过于关注用户体验，而忽略了业务底层逻辑。
4. 事件风暴：关注业务的底层逻辑，但使用门槛较高，适用于大型而复杂的业务分析。



下图是营销系统的用例图（起初并没有这么完整，这是多次迭代后的结果）：



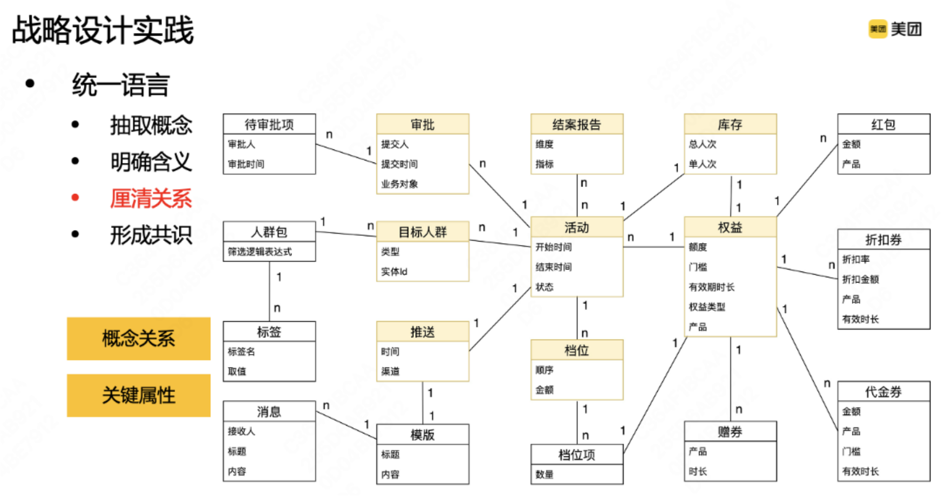
确定业务玩法后，接下来是统一语言。从用例里抽取概念，并对概念进行甄别（去伪存真，抽象合并）找到真正描述业务的概念。比如，有多种方式来描述活动规则：充值送规则、返还规则和档位等，技术可能会泛泛地称其为规则，业务人员则用档位来描述（比如充值送活动，充 1000 送 100 红包，充 2000 送 300 红包，充 3000 送 500 红包，那 1000、2000、3000 就是业务所认为的档位）。抽取概念时，尽量采纳业务侧的叫法，这样统一语言比较容易推行。



接着是明确概念的含义，概念由术语、Term（术语的英文版）和含义三部分构成。含义明确的术语就是统一语言，这些术语将用在日常需求沟通、产品文档，技术设计以及代码实现中。

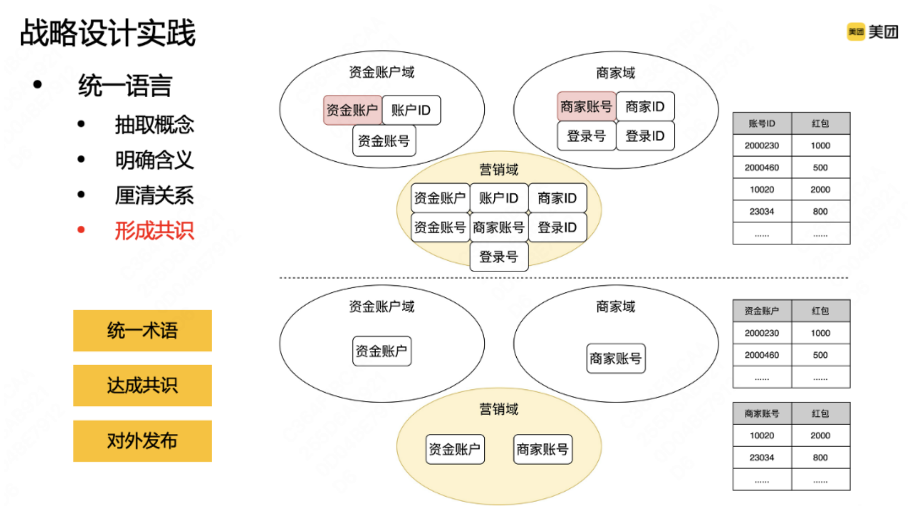
术语	Term	含义
营销活动	campaign	为达成拉新，留存，提LTV等业务目的进行的一系列运营动作......
目标人群	candidate	可以参与营销活动的商户......
权益	benifit	......
档位	gear	......
库存	stock	......
推送	push	......
审批	approval	......
结案报告	report	......
代金券	coupon	......
红包	redpack	......
......	......	......

明确概念后，接着理清概念之间的关系（1 对 1，多对 1，多对多），确定概念所代表的的业务实体的核心属性和行为，从而得到概念模型。后续在业务需求讨论、产品和技术方案设计时，基于这个概念模型，使用统一语言进行描述，大家能很容易对齐；同时精心抽出的概念和建立的概念模型更接近业务本质，为后续的战术设计打下了基础。



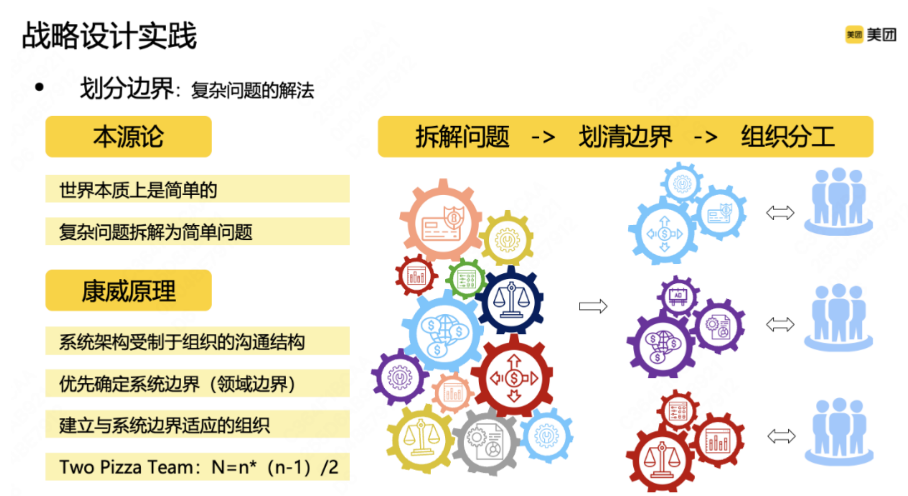
基于统一语言和概念模型，业务 – 产品 – 技术三个角色比较容易就需求达成共识，保障沟通的一致性。

缺少这些就很容易出问题，如：刚开始做营销系统时，在如何描述“商户”上，没有统一语言，资金域有三个概念来描述商户（资金账户、账号 ID、资金账号），商家域有四个概念描述商户（商家账号、商家 ID、登录号、登录 ID），到了营销域，不同的人采用不同的概念来描述商户，造成了沟通的混乱。给商户发红包时，“资金账户、账号 ID、资金账号、商家账号、商家 ID、登录号、登录 ID”这些概念都可以描述商户，但业务人员弄不清这些概念之间的区别，导致 ID 误用，红包发错。事后对这些概念进行了梳理和统一，营销域只关注资金账户和商家账号，系统功能上明确使用资金账户或商家账号来发送红包，这样就不易出错了。



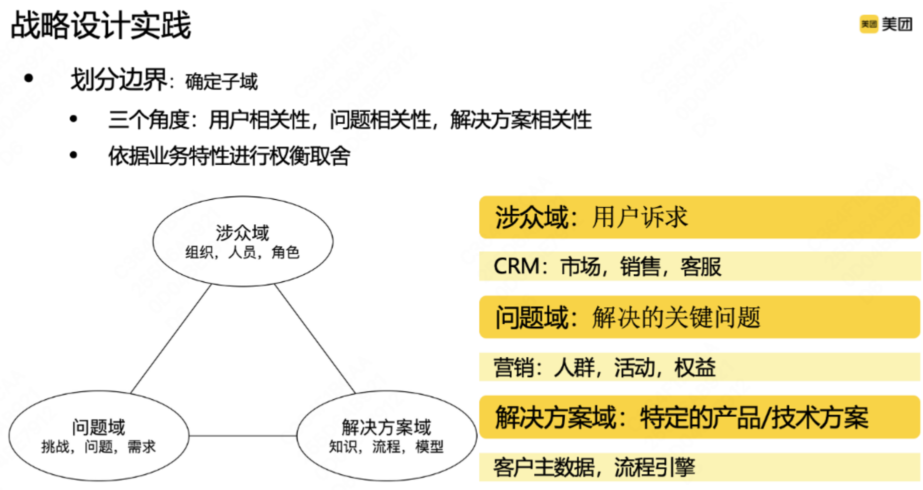
概念模型是一张大网，描述了概念间的关系以及关键属性，但还不能直接映射为代码模型，要映射为代码模型，还需拆解，化繁为简。

本源论认为世界的本质是简单的，复杂问题由多个简单问题构成；康威原理认为系统架构受制于组织沟通架构，系统落地时，首先要确定系统边界，再依据系统边界组织分工。这两个原理表明：我们可以将复杂问题拆解为多个简单问题，并针对团队资源组织分工协作。

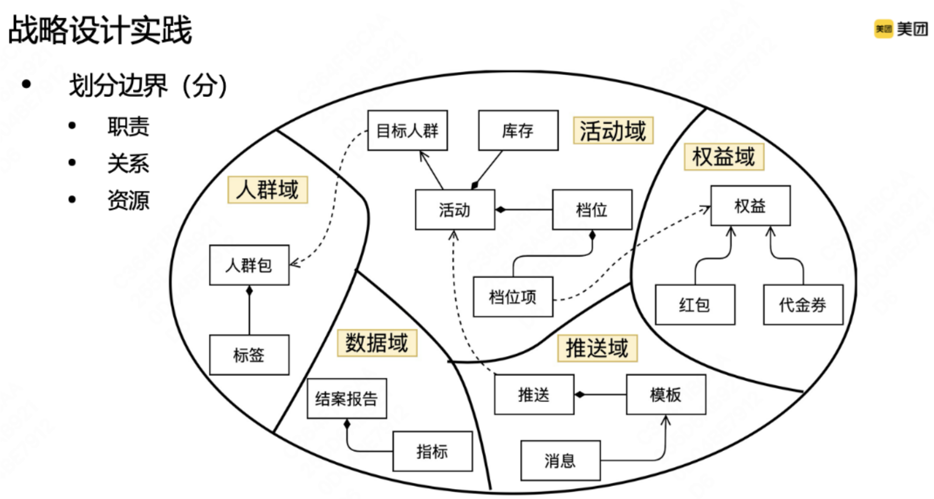


这里提供一种拆解方法（来自美团内部）给出了一种拆解方法：按纵和横两个维度来拆，纵是从业务价值和目标维度划分，横是从功能的通用性维度划分。这里尝试从业务角度来拆，没有系统支持时，业务要在线下运转，通常根据要达成的业务目标，将业务流程或业务组分拆解为多个节点，并定义每个节点的职责以及对应的规范和标准，安排对应的组织或人员执行。简单地说，就是从业务问题和解决方案出发，拆解到对应的人。因此基于业务的拆分通常能实现系统用户、业务问题和解决方案之间的一致性。业务系统是把业务的玩法从线下搬到线上，在进行系统拆分时，也可以使用这个思路。从三个层面来进行：

- **基于涉众域拆解**：也就是按用户相关性进行拆解，不同的用户使用不同的系统功能，如：CRM 由市场人员、销售人员、客服人员三类角色协同完成客户触达，签约合作，售后服务三大职能，针对这三个角色建设相应的系统能力。这种拆解方式比较简单，但也存在较大的局限性，可能导致功能的重复建设。
- **基于问题域拆解**：不同角色 / 用户要解决的问题是相同 / 相似的，可基于问题域进行拆解，如营销系统的用户包括销售、商户、销运等角色，但它核心是要解决如何发券（活动），发给谁（人群），发什么（权益）的问题。基于问题域的拆解相较于基于涉众域的拆解更加抽象，但也可能复用性不够。
- **基于解决方案域拆解**：不同的问题，可能有相同的解决方案，如 HR 域有请假审批、财务上有报销流程、CRM 领域存在客户资质审批，三个领域各自需要解决审批流程的问题，可以构建通用的审批流引擎来统一解决，这是基于解决方案域进行拆解。基于解决方案域的拆解最抽象，也最贴合业务本质，但也容易陷入过度设计的陷阱。

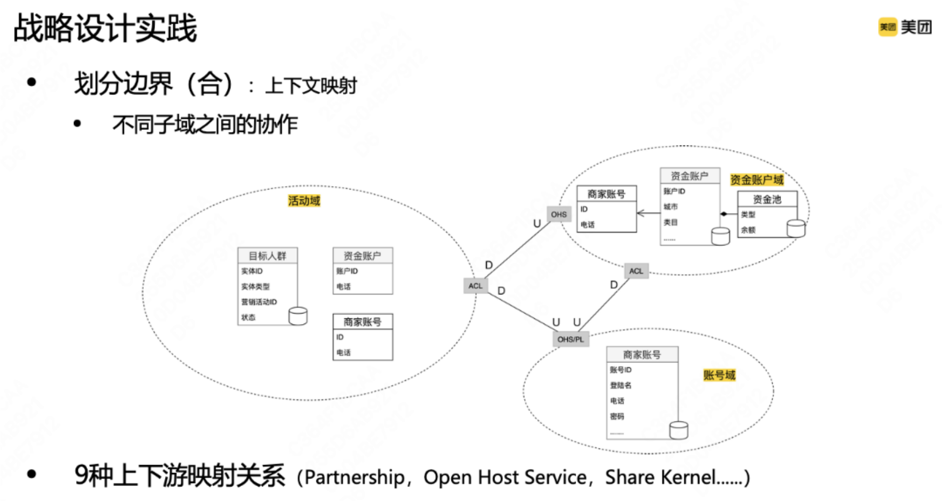


营销系统基于问题域拆解为五个子域（活动域，权益域，人群域，推送域，数据域），每个子域解决特定的问题，各子领域相对内聚和简单：

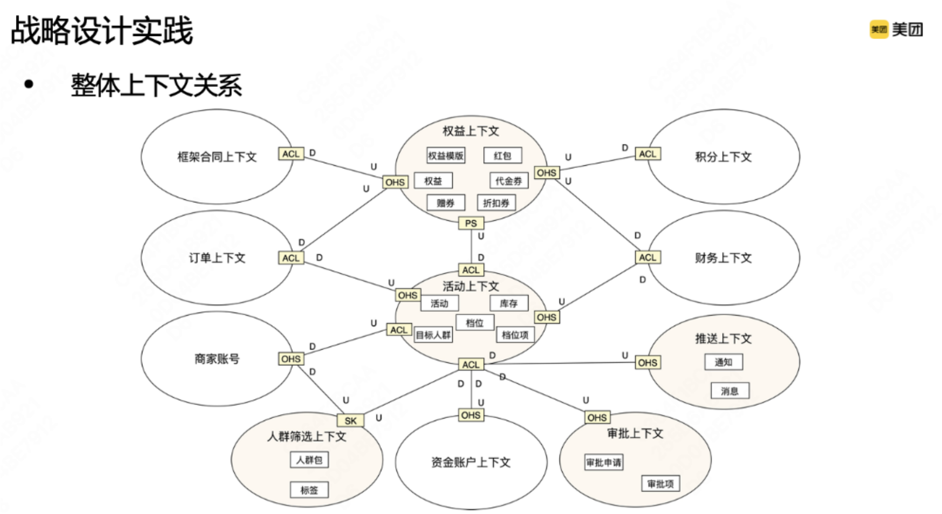


业务系统要运转起来，需要子域之间相互配合，这就要定义上下文映射，实现不同子域间的协作。如活动域关注的两个目标人群：一是资金账户（表示已签约的商户）；另一个是商家账号（表示未签约商户）。资金账户是财务域定义的，而商家账号是账号域定义的，两个概念都不是营销域原生概念。此时，营销域需通过某种方式依赖外部概

念，将外部概念映射到营销域，通过防腐层来对接外部服务来实现这种映射。领域驱动设计里定义九种上下游映射关系，这里不赘述：



下图是营销系统的整体上下文关系：

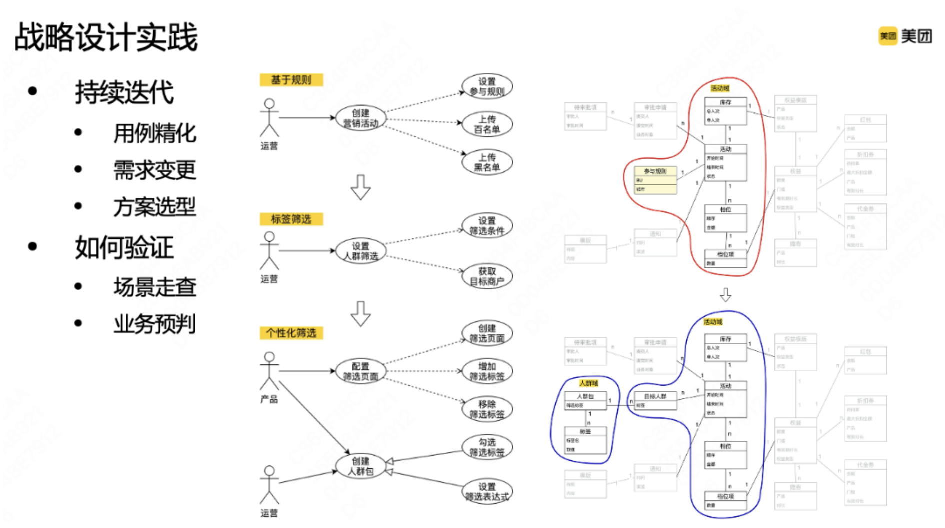


从用例分析，统一语言到子域拆分，初步完成战略设计，但这并非终局，战略设计是一个持续迭代的过程，迭代的来源主要有 3 个：

- **用例精化**：在探讨需求的过程中，用例不断丰富。
- **需求变更**：业务不断发展带来需求变化，进而影响用例及相关概念的内涵，概念模型亦随之调整和迭代。
- **方案选型**：当产品，业务或技术发生较大变化时，可能需要采用另一种方式实现它，这时所采用的概念会有所不同。比如早期构建营销活动域时，通过参与规则来定义谁可以参加活动，将商户与参与规则进行匹配，符合就能参与。这种方式带来的问题是无法提供一个完整的活动人群列表，除非将所有商户（5000 万 +）匹配一遍。随着业务方越来越重视活动参与商户的分层，触达和转化，引入目标人群的概念，通过目标人群来保存所有可参加活动的商户。从参与规则到目标人群，概念发生了变化，底层模型也完全不一样（参与规则是一套规则体系，而目标人群由筛选服务提供），实现了战略设计上的迭代。

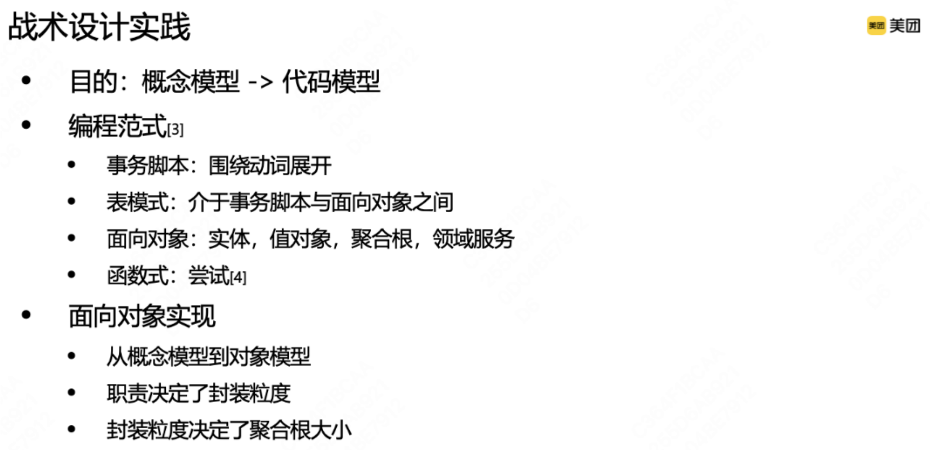
有了战略设计，构建了统一语言和概念模型后，如何验证概念模型呢？通常用两个方法：

- **场景走查**：把模型代入到所有的场景确认一遍，确定所抽象出来的概念模型和统一语言能正确描述它。
- **业务预判**：未来业务的变化会在哪里，当变化发生时，概念模型的内涵和外延是否方便扩展并支持到变化。



4 战术设计实践

战略设计得到了概念模型，战术设计则是将概念模型映射为代码模型，有很多编程范式，比如事务脚本、表模式、面向对象，函数式等，最好的方式是面向对象的实现。



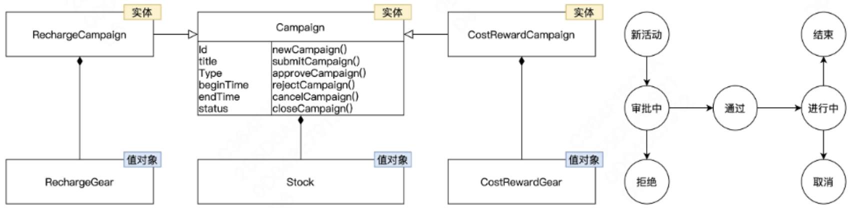
从概念模型到对象模型：

- 首先，概念是分层的，如营销活动是一个泛化概念，其下还有充值送活动、消费返活动，买赠活动等具体活动。构建对象模型时，通过派生 / 继承来实现概念分层。
- 其次，概念关系映射成对象关系，比如营销活动包含了档位和库存，那在构建营销活动对象时，可通过组合实现这种包含关系（档位对象和库存对象成为营销活动对象的属性）。
- 最后，概念的属性行为，可以直接变成对象的属性和行为；概念的状态机以及生命周期也会变成对象的状态机。

两类对象：实体和值对象，这两者的区别是是否有统一标识和自己的状态。

战术设计实践

- **概念模型 -> 对象模型**
  - 概念分层 -> 类分层：营销活动分为充值送活动，消费返活动等
  - 概念关系 -> 类关系：营销活动包含档位，库存
  - 概念属性，行为 -> 类属性，行为：活动包括标题，开始时间，结束时间，审批，发布
  - 概念状态 ->类的状态机
  - 两类对象：实体（状态，唯一标识），值对象（无生命周期，无唯一标识）

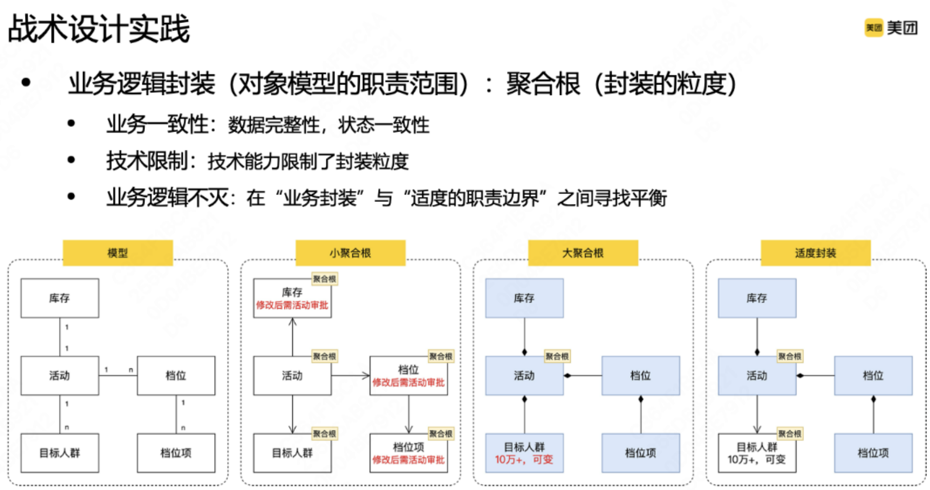


有了对象模型，还需通过聚合根完成封装，如何确定聚合根的粒度？营销活动包含活动、库存、档位、档位项、目标人群五个对象，如果采用小聚合根模式，一个对象对应一个聚合根，这样每个聚合根都很简单。但从业务角度看，库存或档位会影响活动的状态，如：修改了库存或档位，活动需要重新审批和上下线，这种业务上的耦合需要在技术上进行处理。此时，就得在小聚合根上构建领域服务来封装这些逻辑。

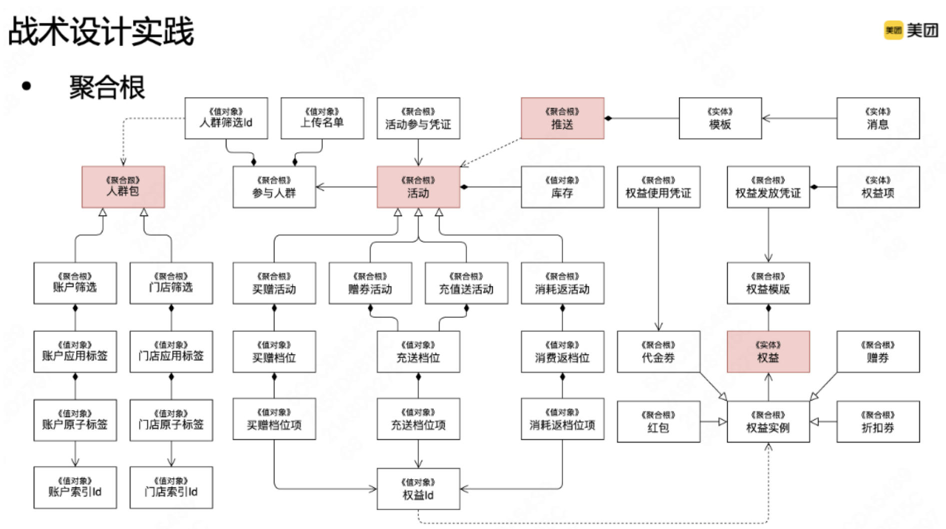
另外一种模式是大聚合根。围绕活动，把活动相关的概念（活动、库存、档位、档位项、目标人群）都封装起来，但聚合根比较复杂，影响活动加载（一些活动的目标人群上百万，懒加载可解决问题，但增加了复杂度）。

聚合根的设计要遵循一定的原则：

- 满足业务一致性、数据完整性、状态一致性。比如库存档位和活动状态要一致，在数据上也要完整，不存在没有档位的活动，也不存在没有库存的活动。
- 技术限制。有些实体会带来技术挑战，如数据量太大，可抽出来单独考虑。
- 业务逻辑不灭，在业务封装与适度的职责边界之间寻找平衡。不管是大聚合根还是小聚合根，业务逻辑永远都是存在的，就是看把它放在哪里。



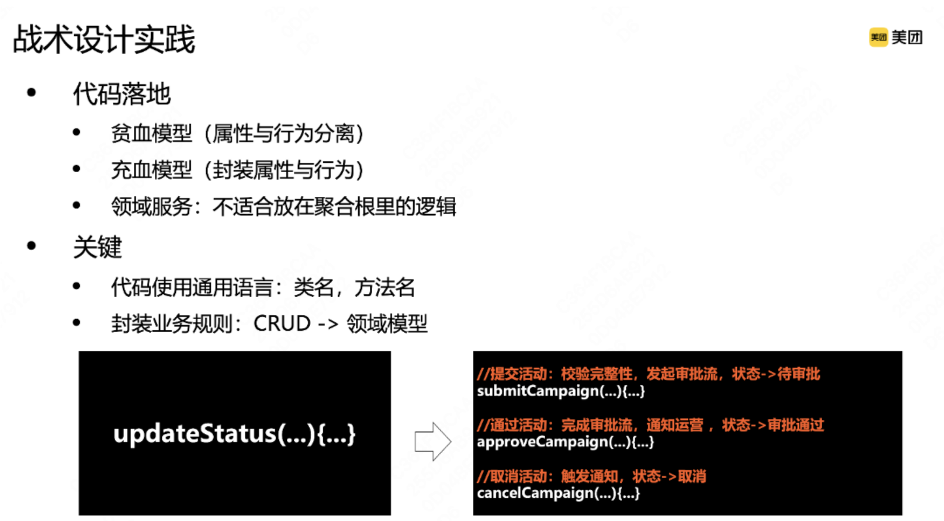
如下图是营销系统的聚合根：



聚合根已经非常接近代码实现，落地代码时，大家还会纠结用贫血模型还是充血模型。Spring MVC 通常运行在单例模式下，引入充血模型会增加理解成本和技术复杂度。另外，不适合放在聚合根里的领域逻辑，可以放在领域服务里，如：同时存在多个充值送活动时，用户只能参加优先级最高的一个，在充值送活动聚合根里会标识活

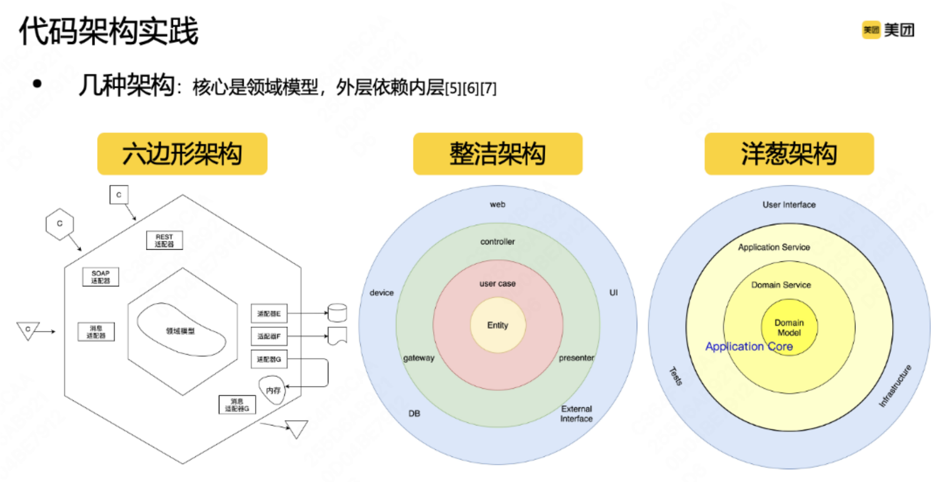
动的优先级，但挑选优先级最高的活动并非聚合根的职责，但确实是领域逻辑的一部分，此时可通过领域服务实现。

从概念模型，类模型到代码实现，整个过程都要使用统一语言。在落地代码时，代码要体现出业务含义，比如下图的例子，要避免左边 updateStatus() 这样的方法，它没有体现业务含义（必须阅读代码实现，才知道这个方法做了什么）；图中右边的 submitCampaign()，approveCampaign()，cancelCampaign() 则有明确的业务含义。

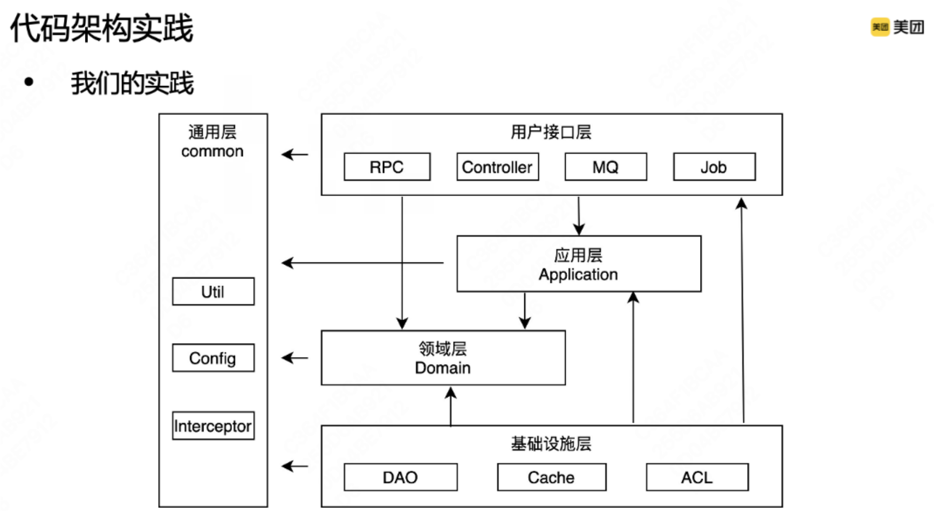


5 代码架构实践

完成战术设计后，如何组织代码架构？无论是六边形架构，整洁架构还是洋葱架构本质上都是围绕着领域模型展开，应用层、基础设施层和外部接口都依赖领域模型：



下图是我们团队的工程实践，与前面三个图本质上是一样的。领域层和应用层次放在中间（两者都属于领域逻辑），基础设施和用户接口依赖中间层：



6 总结

- 我们做的大部分系统都不是全新系统，如 CRM、HR 或 SCM 等，已经有很多业界实践，可充分借鉴这些实践，没必要自己创造新概念。

- 要重视统一语言。没有统一语言就不会有概念模型，没有概念模型就不可能有靠谱的代码模型，拿到需求后就开始设计代码模型是不靠谱的。
- 领域驱动设计是团队工作。现实中没有一个是严格意义上的领域专家，所有参与到这项工作的人都可以是领域专家，整个工作可以由技术团队主导，但一定要落地到产品和业务。
- 拥抱变化，持续迭代。模型是相对稳定的，但并非一成不变，业务理解的深度，抽象的角度与方式，业务的变化都会影响到领域模型，领域模型的建立是持续迭代的过程。

这里分享几个常见的误区：

- 深陷领域驱动设计的概念体系。在代码里生搬硬套领域驱动设计里的概念，比如聚合根、值对象、实体等，掰扯概念之间的细微差异，设计复杂的领域事件等。这反而增加理解成本，让系统变得复杂。领域驱动的精髓在于从业务出发，抽象出业务领域知识，构建概念模型，一步一步将这些概念模型映射成系统。至于如何采用聚合根、领域服务、实体、值对象、领域事件等，可以灵活取舍。
- 试图通过精心设计来获得领域模型。领域模型不是设计出来的，而是通过战略设计的几个步骤，从业务中抽象出来的，最重要是理解业务，对业务进行抽象。
- 使用了 DDD 就一定会产生好的领域模型的想法也不可取，我们知道飞机怎么造，但我们不一定能够造出好飞机，但如果我们知道这个方法，可以少走弯路。

在聊需求的那一刻，设计就开始了，统一语言就是设计的一部分。

解决方案域在模型维度分为四层：

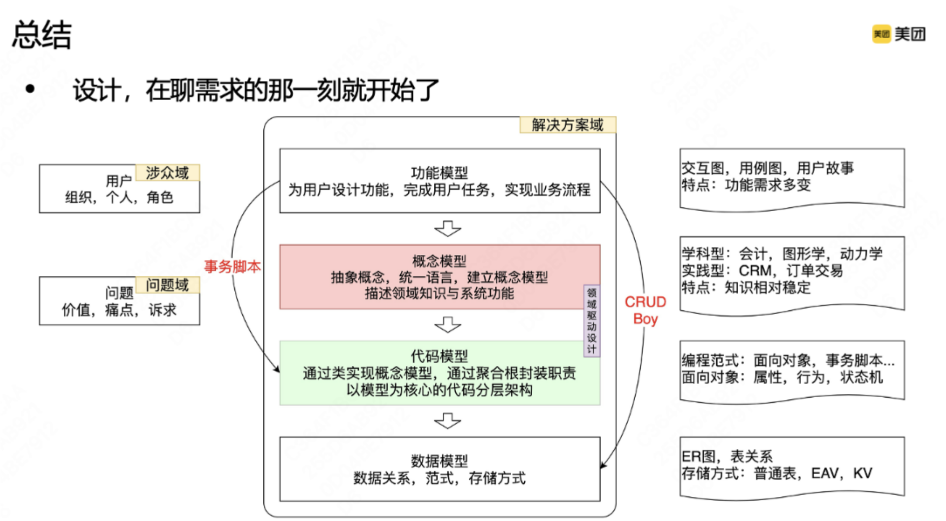
1. **功能模型**：产品表达给我们业务的玩法，我们把它变成了用例，从用例里抽取出功能模型。
2. **概念模型**：对功能模型进一步抽象，统一语言，形成概念模型。
3. **代码模型**：将概念模型映射为代码模型。

4. 数据模型：业务数据需要存储，需要设计对应的表结构。

这里有两个陷阱：

1. 看到功能模型后，就开始设计数据模型，考虑数据该怎么创建、怎么更新、什么时候该删除，沦落为 CRUD boy。
2. 看到功能模型后，就开始考虑操作数据的流程是什么，陷入到事务脚本陷阱。（对于一些简单的功能，不排斥使用事务脚本，但对于复杂功能，事务脚本的维护成本非常大）

另外，领域至少可以分为两大类：一是学科型，比如财务、会计、图形学、动力学，这类系统的设计须先深入理解学科知识；二是实践型，如 CRM、订单交易等，是业务经验的总结，这类系统的设计不妨参考前人的实践。当然，如果自己的业务具有独特性，那就只能靠自己摸索了。



7 参考资料

[1]《DDD 实战课》欧创新
[2]《领域驱动设计》Eric Evans
[3]《企业应用架构模式》Martin Fowler

[4]《实现领域驱动设计》Vaughn Vernon
[5]《The Clean Architecture》Robert C. Martin
[6]《The Onion Architecture》Jeffrey Palermo
[7] https://carlalexander.ca/what-is-software-complexity/
[8] https://martinfowler.com/bliki/BoundedContext.html

DDD 在大众点评交易系统演进中的应用

1 大众点评交易业务介绍

本文主要涉及境外出行、商场团购和内容商业化等三类交易业务场景。在大众点评 App 里，在境外城市站有美食、购物、商场、景点、门票、当地玩乐等频道入口，可以购买境外出行交易产品，在境内的逛街 / 商场频道可以找到商场团购优惠以及商场团购代金券。

此外，商家如果有推广需求可以在商家端 App（开店宝 App）“点星”入口购买达人的创作服务，最终达人交付的笔记，在点评 App 信息流里进行展示。具体来说，境外出行产品覆盖景点门票、餐厅订座和休闲娱乐；商场团购产品包含普通团单和秒杀团单，适用于商场的优惠活动；内容商业化产品则允许商家购买达人的图文或视频笔记，以此来推广自己的服务或产品。

2 领域驱动设计概述

2.1 什么是领域驱动设计

领域驱动设计是一种软件设计方法，它主要用于处理复杂业务需求。我们可以将其分解为“领域”、“驱动”和“设计”三个部分来理解。“领域”指的是特定的业务范围或问题域，如电商、医疗、保险等。确定领域后，我们就能明确核心的业务问题。例如，在电商中，核心问题可能涉及商品、库存、仓储和物流；在保险领域，则可能关注投保、承保和理赔等方面。

“设计”在 DDD 中通常指的是领域模型的设计，DDD 强调领域模型是系统的核心，它反映了业务概念和业务规则。“驱动”有两层含义：一是业务问题域驱动领域建模的过程；二是领域模型驱动技术实现或代码开发的过程。确保领域模型的准确性是关键，因为它可以保证代码实现能够真实反映并解决业务的核心问题。

领域驱动设计是一种处理高度复杂领域的设计思想，它通过分离技术实现的复杂性，围绕业务概念构建领域模型来控制业务的复杂性，以解决软件难以理解、难以演化等问题。领域驱动设计是一种设计思想，首先体现了分离的思想，它分离了业务复杂性和技术复杂性，其次体现了分治的思想，它通过领域模型、限界上下文或子域进行分治。

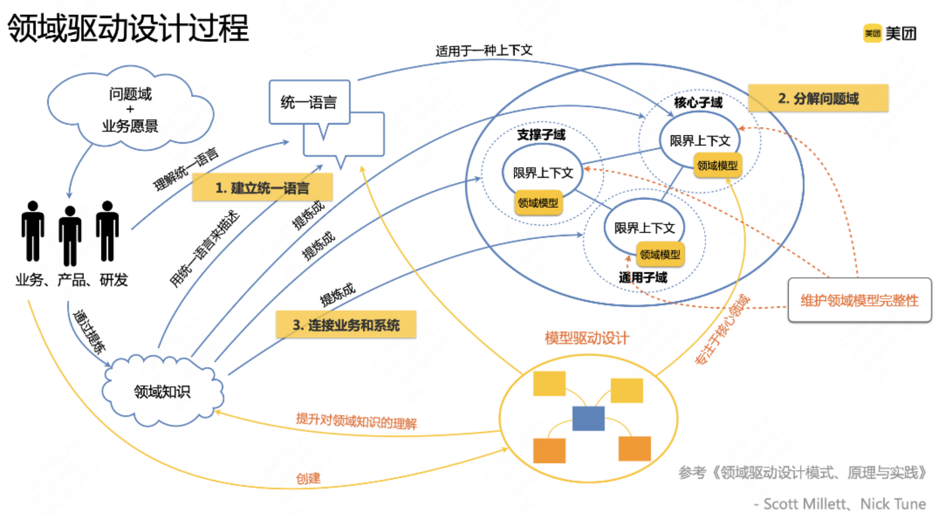
2.2 领域驱动设计核心概念

领域驱动设计涉及到的核心概念非常多，我们重点强调一下“统一语言”和“限界上下文”。“统一语言”贯穿领域驱动设计从战略设计到战术设计到最后的代码实现全过程，对于需求分析、知识提炼和最后代码的实现，都是非常重要的。

“限界上下文”是连接问题空间和解决方案空间的桥梁，一方面我们在问题空间分析问题时，它是语言的边界和模型的边界；另一方面，在解决方案空间我们通过限界上下文来确定应用的边界和技术的边界，从而帮助我们确定整个系统及各个限界上下文的解决方案。

2.3 领域驱动设计的过程

首先，领域驱动设计需要业务、产品、研发以及 QA 共同来参与，应基于对问题域以及业务愿景的理解，并进行充分讨论而达成统一认知，在这过程中提炼领域知识，并建立统一语言。同时在领域知识基础上进一步提炼，分解问题域为核心子域、支撑子域和通用子域，再通过模型驱动设计思想，设计领域模型，通过领域模型连接业务和系统，并且在模型驱动设计过程中，会有新的认知迭代。通过这些认知迭代进一步丰富统一语言，因此领域知识是一个不断迭代、螺旋式推进的过程。

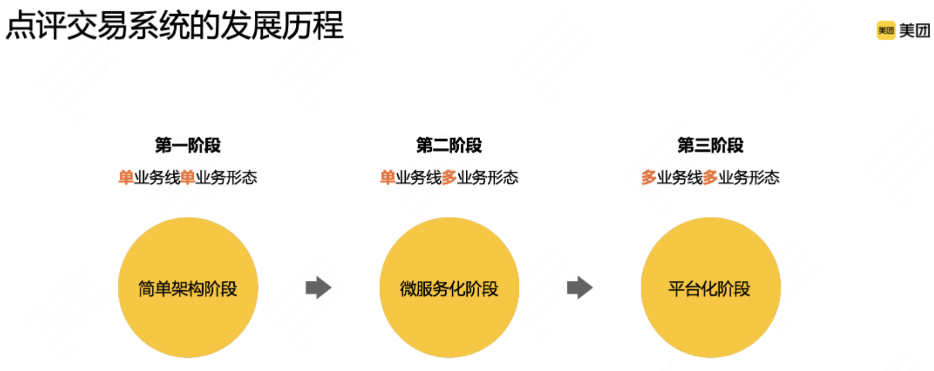


3 大众点评交易系统演进

点评交易系统的发展历程从业务视角和技术视角看，分别有三个阶段。从业务视角看：

- 第一阶段是单业务线单业务形态阶段，这个阶段我们只支持了境外出行交易业务场景，包含了预订的业务形态；
- 第二个阶段是单业务线多业务形态阶段，业务形态变得更加丰富；
- 第三个阶段即多业务线多业务形态阶段。

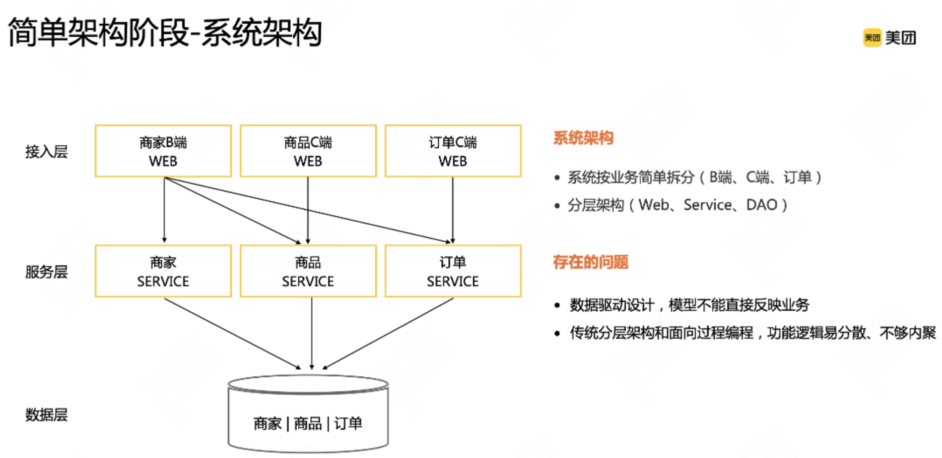
而从技术视角看，主要是经历了包括简单架构、微服务架构和平台化架构等三个阶段的演进。



3.1 简单架构阶段

这个阶段是我们业务和系统起步的阶段，当时我们只支持了预订形态的一两个品类的交易，整体上相对比较简单，同时我们当时团队的规模也很小，为了快速支持业务上从 0 到 1 这个过程中不断的探索和试错，我们在技术系统建设的主要思路是按照业务环节对业务功能模块做了一些简单的划分，从而做到能够快速的迭代和交付。

在这个阶段，我们的系统架构也相对简单，根据业务进行了基础的拆分。具体来说，接入层分为商家 B 端、商品 C 端和订单 C 端，而服务层则划分为商家、商品和订单三个部分。整体上采用了传统的 MVC 分层架构。这种架构在项目初期确实展现出了其优势，即“简单”和“快”。



然而，随着业务需求的不断增加和变得复杂，系统开始暴露出一些问题，主要可以归纳为两个方面：

首先，我们采用的是数据驱动设计，通常是先建立数据库表，这导致模型无法直观地反映业务实际情况。其次，由于采用了传统分层架构，我们将数据库表映射为持久化对象（PO），然后在服务层通过 CRUD 操作进行过程式编程。这在多个场景下出现了功能相似但又有所不同的需求时，经常导致重复编写相似的代码，最终造成了逻辑上的分散，系统整体的内聚性不足。

以订单退款逻辑为例，我们面临着包括订单确认前 / 后退款、履约前 / 后退款等多种场景，以及需要考虑由用户（买家）、商家（卖家）、客服和系统等不同角色发起的退款。虽然这些不同场景和角色发起的退款业务逻辑在很大程度上是相似的，但它们之间也存在一些差异。

在传统的 MVC 架构模式下，由于缺乏对业务领域的深入理解和沉淀，服务间的调用往往缺乏清晰的结构，导致逻辑交织在一起。此外，研发团队在系统迭代过程中可能没有足够重视高内聚和低耦合的设计原则。因此，系统内部往往会出现多处重复且相似的订单退款代码逻辑，这不仅降低了系统的可读性，也给系统的可维护性带来了挑战。

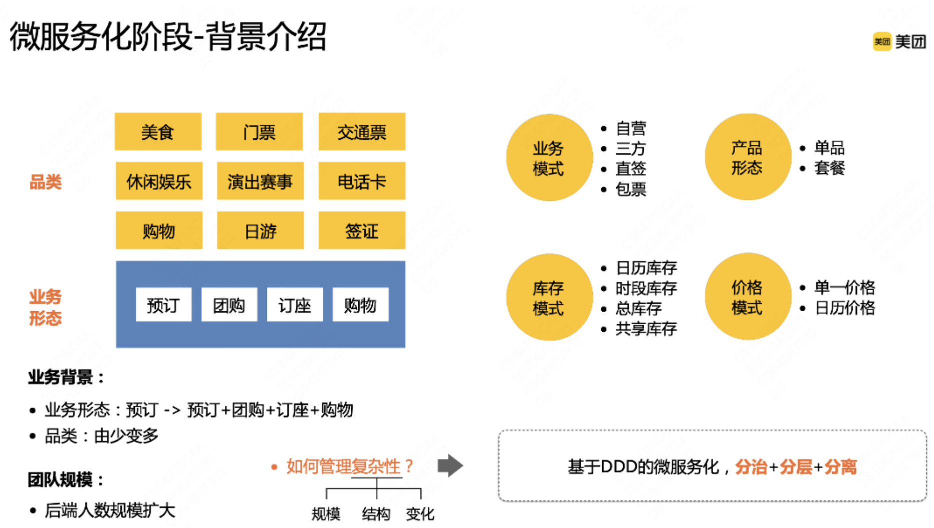
3.2 微服务化阶段

随着业务品类的增加和业务模式的多样化，我们的业务和系统复杂度迅速上升，团队规模也相应扩大。这种复杂性主要由三个因素造成：

首先，业务规模的扩张带来了系统规模和代码量的增加；其次，业务需求的累积导致了系统内部的重复代码、复杂的依赖关系，以及为了满足高可用性和高性能需求而引入的各种技术组件和并行、异步解决方案；最后，业务需求的频繁变动也增加了系统的复杂性。

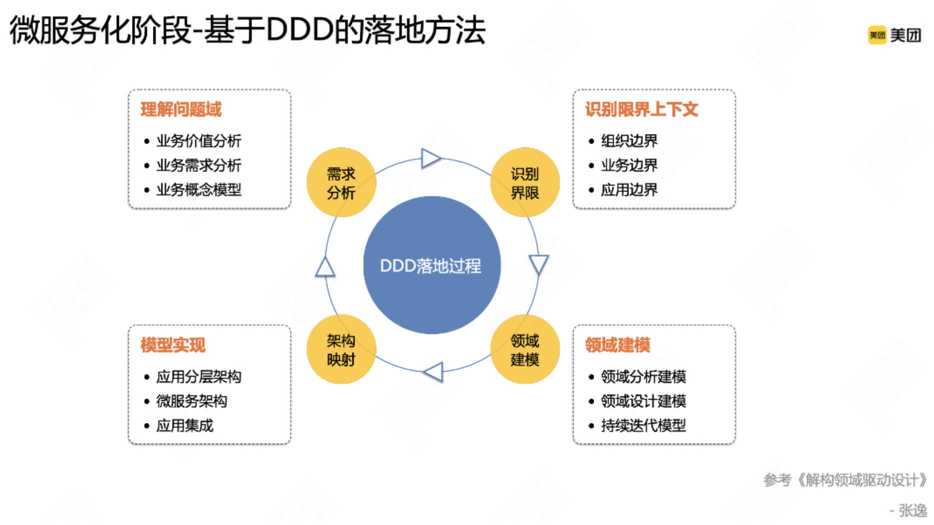
为了应对这些挑战，我们的主要思路是：通过分治的方法来管理软件规模，利用系统分层和关注点分离的原则来优化系统结构，以及通过隔离变化来应对频繁的需求迭代。这些策略都是领域驱动设计（DDD）的核心理念，基于此，我们实施了微服务架

构的拆分，以更好地管理和控制系统复杂性。



在领域驱动设计的落地方法上，我们参照行业实践内容并且结合自身的理解，我们将 DDD 的实施过程划分为以下四个阶段：

- **理解问题域**：这个阶段的核心是深入分析业务价值、需求以及构建业务概念模型。产出统一语言和子域划分，确保团队在业务理解上达成共识。
- **识别限界上下文**：在这一阶段，我们通过组织、业务和应用的边界来确定限界上下文，并且明确不同上下文间的关系和交互。
- **领域建模**：包括领域分析、设计建模，以及模型的持续迭代。这个阶段的目标是构建能够反映业务核心概念和规则的模型。
- **模型实现**：实现阶段主要依赖于应用分层架构、微服务架构和应用集成，确保领域模型能够在系统中得到有效实施。

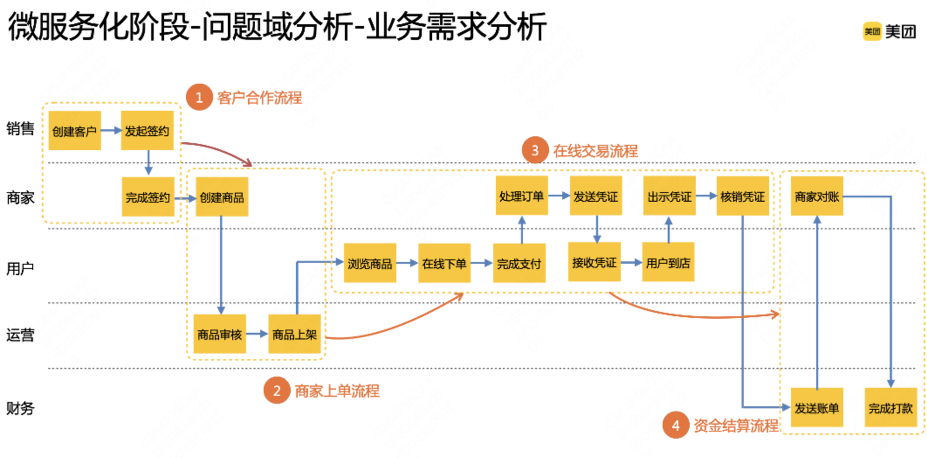


理解问题域

业务价值分析有助于评估系统的复杂性，并且可以指导我们识别最为关键的业务领域。业务需求分析是一个关键的知识提炼过程，其中涉及多种方法和工具，例如事件风暴、四色建模以及用例分析等，我们采用的是相对轻量的用例分析法。

在进行用例分析之前，我们首先需要对业务流程进行细致的分析。这一步骤通过拆解业务流程和环节，帮助我们发现和识别具体的业务用例。这里简化了交易业务流程的分析，因为大多数人对电商类业务流程较为熟悉。对于不熟悉的业务领域，我们将需要进行更深入的业务流程和场景分析。

我们将交易业务流程分为四个主要部分：客户合作流程、商家上单流程、在线交易流程和资金结算流程。有了这些流程的分解，我们就可以进入到具体的用例分析阶段。



在用例分析阶段，我们以商家上单流程和在线交易流程为例来说明。在商家上单流程中，涉及到的主要角色包括商家和运营。商家负责创建新商品、商品上架、商品下架以及更新商品库存等操作。而运营人员则参与商品审核，包括审核通过、审核驳回、查看审核列表等关键用例。至于在线交易流程，其参与方主要是买家、商家以及客服。买家的行为包括购买商品、支付、申请退款和查看订单等，商家则处理订单确认、发送凭证、核销凭证和订单检索等关键用例。客服则参与售后服务，涉及订单退款、订单赔付等核心用例。

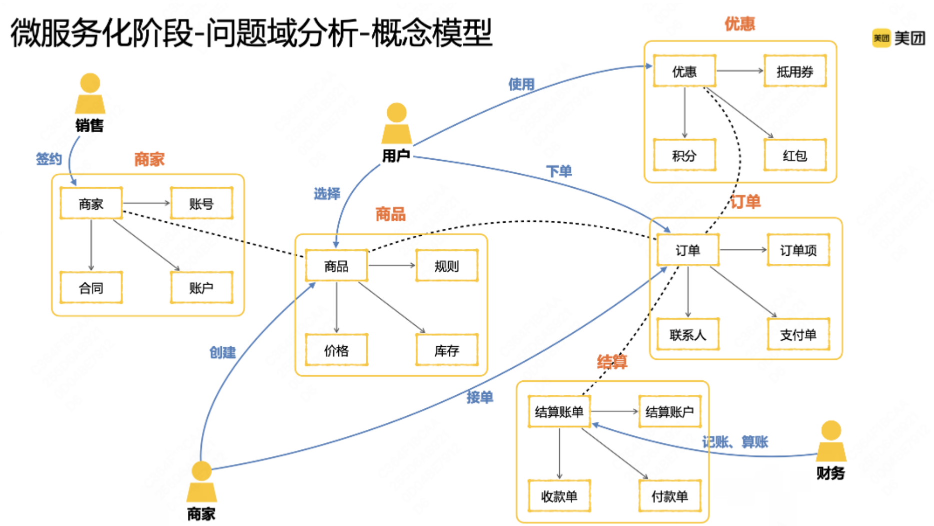
在完成业务流程和用例分析之后，我们可以根据相关性对问题进行初步分类，并划分为不同的子域，建立统一语言。为了更好地进行知识提炼，为识别限界上下文和建立领域模型提供必要的信息，我们需要深入分析每个用例，并制定用例规约来提取关键概念。

在实际操作中，我们没有严格制定用例规约，而是使用产品需求文档中的描述。在技术方案设计阶段，我们也会使用类似于时序图和接口描述的方法来详细阐述用例。无论采用哪种描述方式，关键在于坚持使用统一语言，这对于从描述中提炼出核心概念至关重要。这样做不仅有助于团队成员之间的沟通，也便于后续的设计和开发工作。

在业务流程分析、用例分析以及用例规约的制定和编写之后，我们对交易业务的领域

知识已经有了充分的了解，并构建了相应的概念模型。在这个模型中，销售签约商家，商家负责商品的创建，用户选择商品进行下单，下单购买过程中可能会使用优惠，在订单完成之后需要财务介入对商家进行结算。

对这些关键的概念进行归类之后，我们识别出了商家、商品、订单、优惠和结算等几个子域。这里或许会产生一个疑问：对于我们已经熟悉的领域，是否真的需要经过这样复杂的分析和提炼过程来划分子域？实际上，对于有经验的架构师而言，确实可以迅速地完成子域的识别和划分，这也展示了领域驱动设计过程中的一种艺术性。然而，这些系统性的分析步骤确保了即使是不熟悉领域的团队成员，也能够准确地理解业务并作出恰当的架构决策。



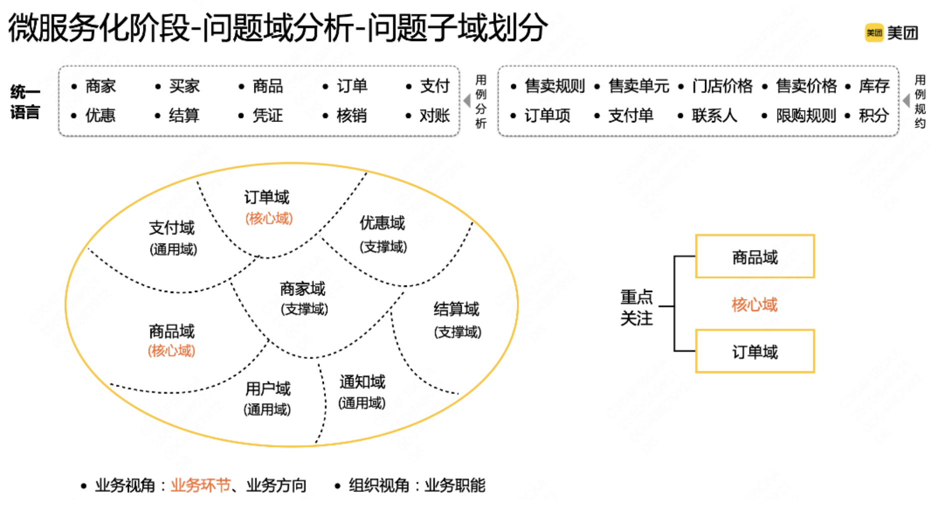
在问题域分析阶段主要的输出包括两大部分：一是统一语言，二是子域划分。

1. 在统一语言上，通过用例分析我们提炼了商家、买家、商品等统一语言，通过用例规约的整理对统一语言进行了丰富，包括售卖规则、售卖单元、订单项等等，我们可以使用这些统一语言进行交流并且用于后面的模型设计和代码实现。
2. 在子域划分上，我们最终识别出了如图所示的这样几个子域，结合我们在价

值分析阶段得到的为用户提供一站式服务体验，以及为商家提供一体化售卖平台的这样的核心价值，我们将商品域和订单域作为核心域进行重点建设。

此外，对于子域的划分方法，可以分别按照业务和组织两个视角来看，从业务视角上可以按照业务环节或业务方向进行划分，我们使用的其实就是按照业务环节来划分的，将商家合作到商品上单再到交易和结算的整个业务流程进行阶段划分，按照划分出来的每个环节确定子领域。

当目标系统为客户提供多个业务方向的产品时，可以根据业务方向进行子领域划分，比如银行系统可以从储蓄、理财、外汇等几个方向来进行拆分；当目标系统用于企业的管理时，可以从组织视角按照业务职能部门进行划分。



识别限界上下文

在对问题域进行了充分的分析之后，我们进入了限界上下文识别的阶段。前面提到了限界上下文的重要性，它是连接问题空间与解决方案空间的重要桥梁。一方面我们在问题空间分析问题时，它是语言的边界和模型的边界，也就是业务的边界，另外在解决方案空间我们通过限界上下文来确定应用的边界。

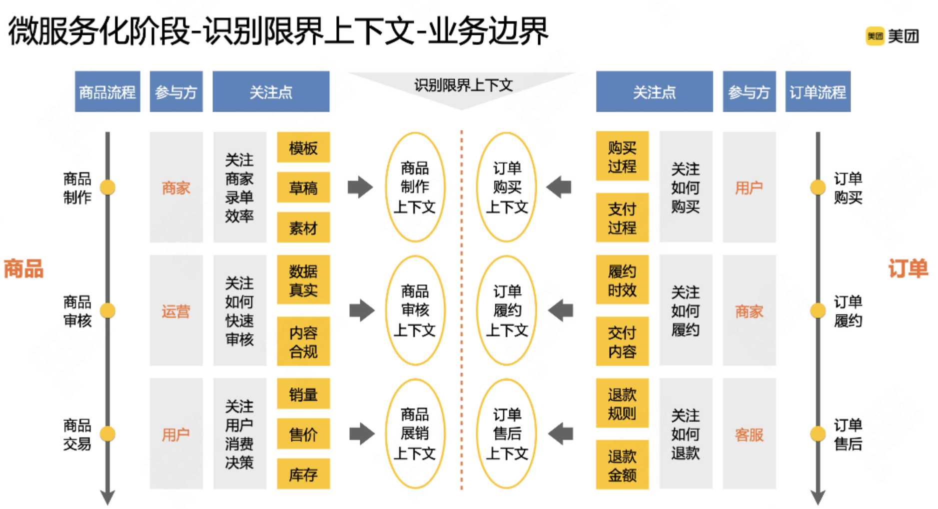
所以我们在限界上下文识别的时候，也主要是从业务边界和应用边界两方面来进行。

首先我们基于语义相关性和功能相关性对我们在问题域分析阶段所罗列的业务活动进行归类，优先考虑功能相关性，得到初步的限界上下文划分，在我们交易系统的分析过程中，这个结果与子域划分结果基本上是一致的。

那么限界上下文具体要到什么粒度呢，这里跟我们的业务复杂度、技术复杂度以及团队规模有一定的关系，结合我们的实际情况，我们对商品和订单这两个核心域的限界上下文做进一步的识别和划分。

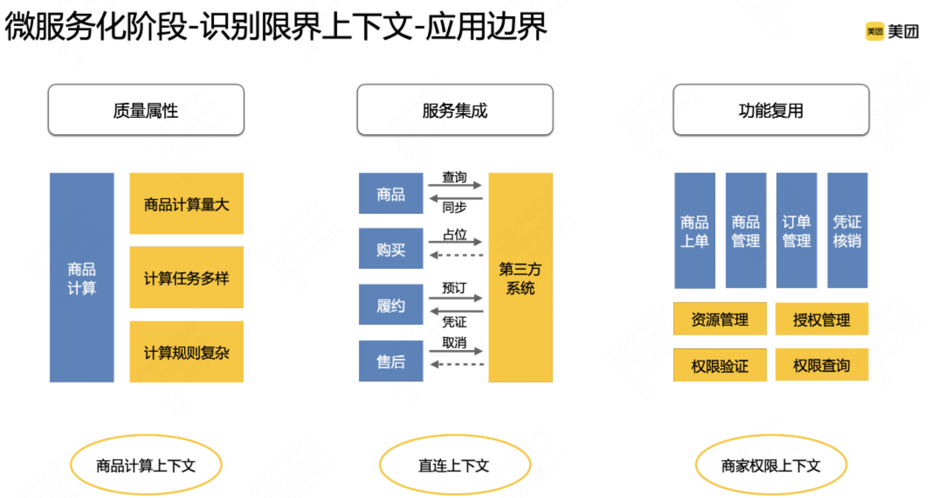
仔细思考后我们发现，尽管商品和订单是贯穿整个业务流程的核心概念，但在业务流程的不同阶段涉及不同的参与方和关注点，对应到系统能力上的诉求也不尽相同。

以商品为例，其涉及到商品的创建、审核发布以及用户端的展示销售等环节。在商品创建阶段，商家关注录单效率和商品制作过程的管理；在审核阶段，运营关注审核需求和审核效率；而在展示销售阶段，用户关注商品信息、价格库存以及如何做出购买决策。订单的情形也类似，在购买、履约和售后各个阶段，关注点也有所不同。因此，我们对商品和订单的限界上下文进行了细分，以确保系统设计能够更精准地满足各阶段的业务需求。

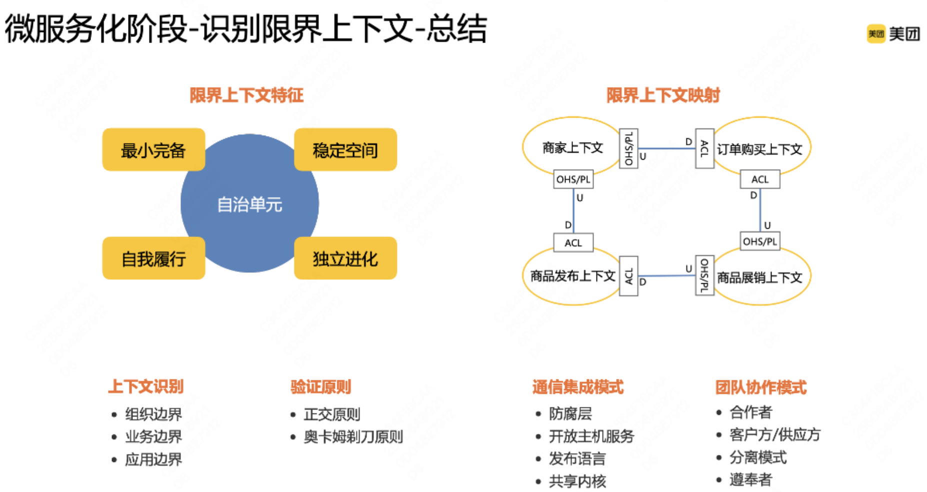


限界上下文的识别过程虽然本质上仍然是对问题域拆分和求解的过程，但同时限界上下文也是应用的边界和技术的边界，所以我们也需要考虑一些质量需求和技术因素，不过需要注意的是我们仍然要遵循先业务后技术的原则，并且在考虑技术因素时，仍然要保证领域模型的完整性和一致性。

我们从质量属性、服务集成和功能复用三个方面对限界上下文做进一步的划分，以商品计算为例，商品计算量大、任务多、规则复杂，为了避免影响正常的商品展示和售卖，所以从展销上下文进行了拆解。此外，我们的商品和订单都涉及到要与很多第三方的系统进行对接，这里面将第三方服务的集成划分为单独的直连上下文，从而隔离三方系统差异对内部商品和订单相关系统带来的变化。在功能复用上，我们考虑对多个限界上下文都涉及的功能进行提炼，作为单独的一个上下文，比如商家权限上下文。



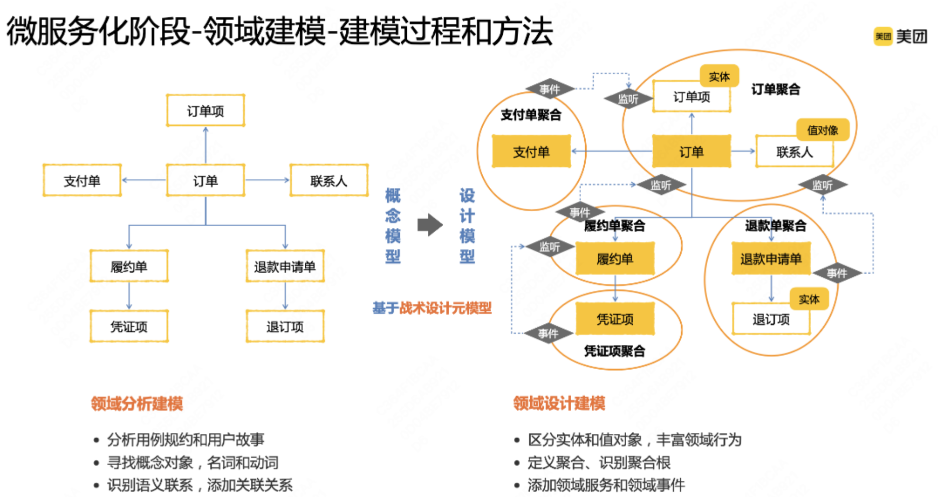
限界上下文封装了按照纵向切分的业务能力，那多个限界上下文如何协作来完成一个完整的业务场景呢，这就涉及到限界上下文的映射，按照通信集成模式和团队协作模式来划分，有多种映射关系，这里面我们用到最多的是通过防腐层、开放主机服务和发布语言三者联动来隔离上下游的变化、维护整个领域模型的稳定性。



领域建模

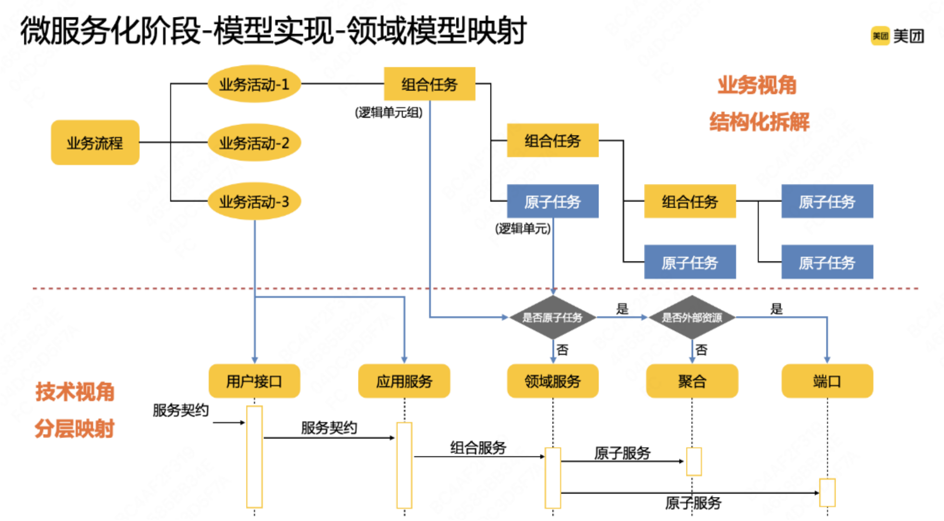
在领域建模阶段，我们整体上分为领域分析建模和领域设计建模。首先，主要是对用例以及用例规约和用户故事进行详细的分析，从中通过名词法和动词法寻找领域概念来构建我们的领域分析模型。在此基础上，我们基于 DDD 战术设计的元模型，识别出这些概念中的实体和值对象，并且根据业务规则的不变性设计聚合。

以订单为例，这里是我们简化之后的模型，包括订单、支付单、履约单、凭证以及退款单这样几个聚合，在存在状态变化时，聚合之间通过领域事件进行协作。

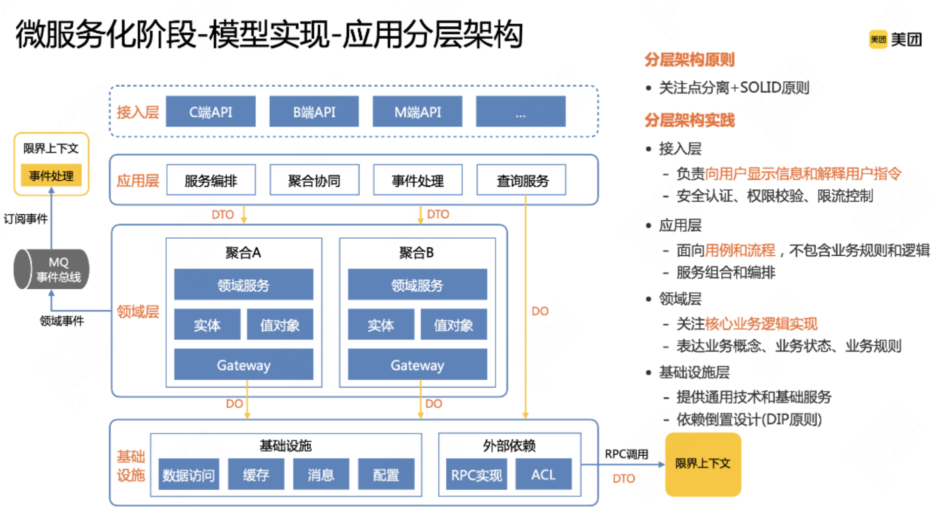


模型实现

在完成限界上下文的识别以及领域模型的设计之后，接下来进入到代码实现阶段，那我们如何将具体的业务流程或业务活动映射到我们的系统进行代码实现呢。这里我们首先是从业务视角对业务流程和业务活动进行分层结构化拆解，其实我们之前的用例分析和用例规约就是这个拆解过程，在拆解之后，我们按照一定的映射关系将其映射到用户接口、应用服务、领域服务、聚合和端口的实现上。



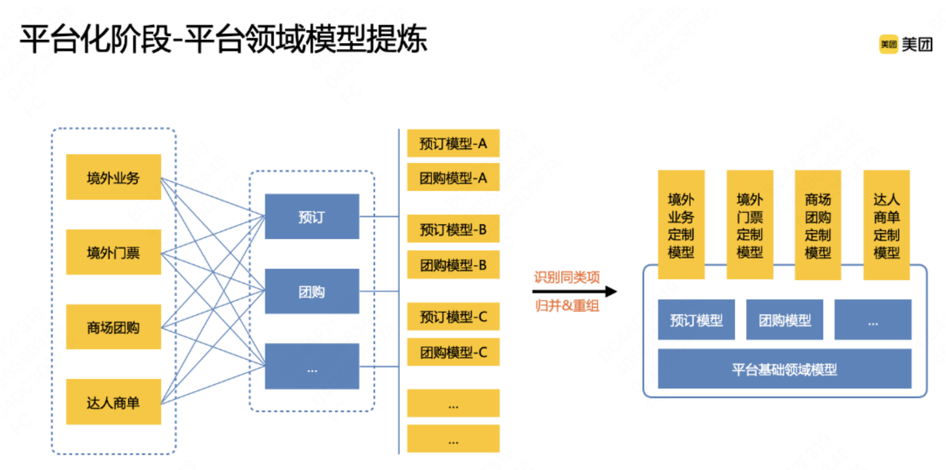
最后，我们按照限界上下文划分微服务，服务内部按照分层架构进行实现。整体上基于关注点分离和 SOLID 原则，分为接入层、应用层、领域层和基础设施层。最终需要维护领域层的稳定性，对上由接入层和应用层来隔离变化，对下由基础设施层通过依赖倒置的方式来隔离数据以及外部依赖的差异性和变化。



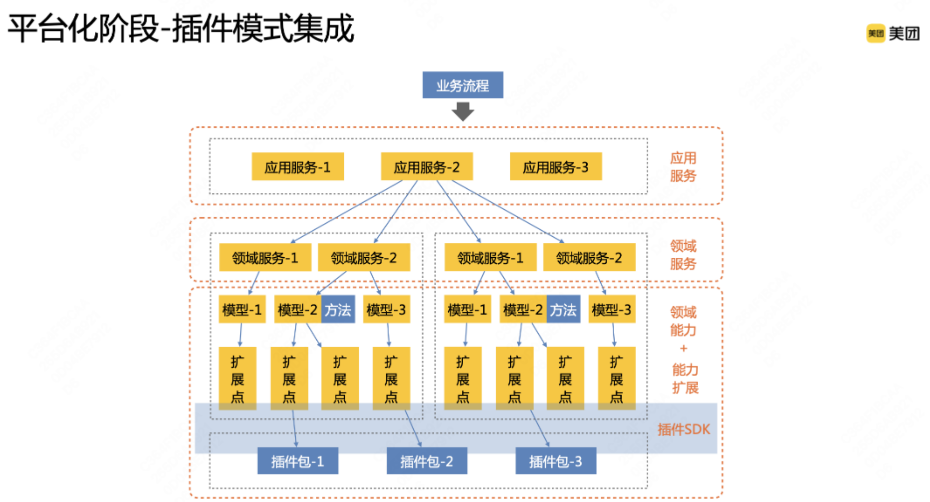
3.3 平台化阶段

随着业务的不断发展，出现了商场团购、内容商业化等更多的交易业务场景，在技术上可以通过平台化的思路将底层系统能力进行复用来提升各业务的支持效率。同时，DDD 的战略模式也在重点关注组织上如何更好的管理大型业务系统，因此我们可以结合 DDD 来构建平台领域模型和业务扩展模型，从而更加高效地完成平台化改造。

我们主要以业务最为复杂的境外交易业务作为基础的主领域模型，并按照 DDD 领域建模过程对商场团购和商业化业务进行拆解得到的领域模型与主领域模型进行映射匹配，经过同类项识别、归并和重组，得到平台领域模型和各业务的扩展模型。在我们的实际落地过程中，为了实现在多业务之间进行最大化复用的目标，我们在平台领域模型的构建上做了进一步的提炼，将平台领域模型拆解为基础领域模型，以及预订业务模型、团购业务模型等按照业务形态划分的领域模型。



此外，为了提升业务 BP 和平台团队的协作效率，在平台领域模型和业务领域模型划分的基础上，我们采用了基于插件化的集成开发模式。通过扩展点的定义，由各业务线在各自的插件包里基于业务扩展模型进行业务定制化实现，再集成平台领域模型和业务扩展模型，最后实现完整的业务流程和业务场景。



4 总结和思考

DDD 是一种开放的思想体系，其核心在于通过领域模型的建立来引导整个设计过程。

- 第一，本文认为战略设计的重要性可能要高于战术设计，因为它涵盖了对业务流程和核心概念的理解和组织。
- 第二，领域建模是一个动态的、迭代的过程，而非一成不变的瀑布式流程。这个过程类似于一个建模涡流，从战略设计到战术设计，不断迭代。在战术设计过程中，如果发现某些方面不合理，就需要对战略设计做出调整。同样，子域的划分和限界上下文的识别也是动态的，需要根据新的发现不断优化。
- 第三，DDD 不强迫采用特定的架构模式，它关注的是业务与技术复杂性是否得到了有效分离。无论是整洁架构、六边形架构还是传统的 DDD 分层架构，只要能够实现这一目标，它们都是可行的选择，即便是采用 MVC 分层架构，只要能够分离业务和技术复杂性，也同样适用。

最后，我们来简要强调一下工程师的思维模型，这些在领域驱动设计（DDD）的实施过程中也至关重要。一方面，工程师需要培养用户思维、业务思维和产品思维，这有助于深入理解业务和问题域。基于这样的理解，工程师可以运用结构化思维来分解问

题，并通过抽象思维来提炼模型。另一方面，结合分层、分治和工程思维，工程师可以有效地将设计转化为实际的代码实现。

5 参考资料

[1]《解构领域驱动设计》- 张逸
[2]《领域驱动设计 – 软件核心复杂性应对之道》- Eric Evans
[3]《领域驱动设计模式、原理与实践》- Scott Millett, Nick Tune
[4]《Business Model Generation》- Alexander Osterwalder

Spark 向量化计算在美团生产环境的实践

1 什么是向量化计算

1.1 并行数据处理：SIMD 指令

让我们从一个简单问题开始：假设要实现“数组 a+b 存入 c”，设三个整型数组的长度都是 100，那么只需将“c[i] = a[i] + b[i]”置于一个 100 次的循环内，代码如下：

```
void addArrays(const int* a, const int* b, int* c, int num) {
  for (int i = 0; i < num; ++i) {
    c[i] = a[i] + b[i];
  }
}
```

我们知道：**计算在 CPU 内完成，逻辑计算单元操作寄存器中的数据，算术运算的源操作数要先放置到 CPU 的寄存器中，哪怕简单的内存拷贝也需要过 CPU 寄存器。**所以，完成“c[i] = a[i] + b[i]”需经三步：

1. 加载（Load），从内存加载 2 个源操作数（a[i] 和 b[i]）到 2 个寄存器。
2. 计算（Compute），执行加法指令，作用于 2 个寄存器里的源操作数副本，结果产生到目标寄存器。
3. 存储（Store），将目标寄存器的数据存入（拷贝）到目标内存位置（c[i]）。

其中，加载和存储对应访存指令（Memory Instruction），计算是算术加指令，循环执行 100 次上述三步骤，就完成了“数组 a + 数组 b => 数组 c”。该流程即对应传统的计算架构：单指令单数据（SISD）顺序架构，任意时间点只有一条指令作用于一条数据流。如果有更宽的寄存器（超机器字长，比如 256 位 16 字节），一次性从源内存同时加载更多的数据到寄存器，一条指令作用于寄存器 x 和 y，在 x 和 y 的每个分量（比如 32 位 4 字节）上并行进行加，并将结果存入寄存器 z 的各对应分量，最

后一次性将寄存器 z 里的内容存入目标内存，那么就能实现单指令并行处理数据的效果，这就是单指令多数据（SIMD）。

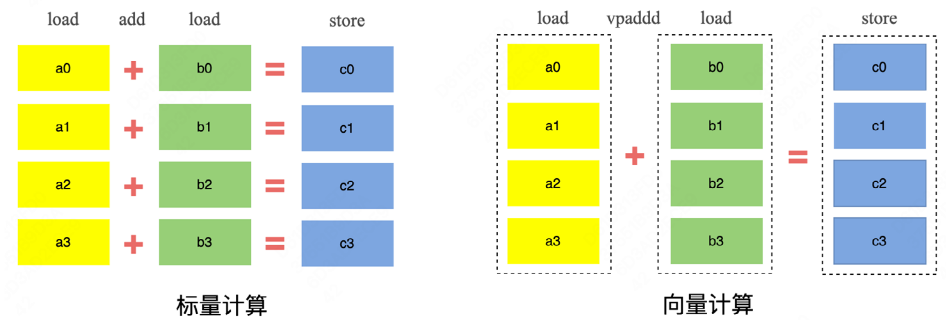


图 1：向量化计算“数组 a+b 存入 c”

单指令多数据对应一类并行架构（现代 CPU 一般都支持 SIMD 执行），单条指令同时作用于多条数据流，可成倍的提升单核计算能力。SIMD 非常适合计算密集型任务，它能加速的根本原因是“从一次一个跨越到一次一组，从而实现用更少的指令数完成同样的计算任务。”

1996 年，Intel 推出的 X86 MMX（MultiMedia eXtension）指令集扩展可视为 SIMD 的起点，随后演进出 SSE（1999 年）SSE2/3/4/5、AVX（2008）/AVX2（2013）、AVX512（2017）等扩展指令集。在 linux 系统中可以通过 lscpu 或 cpuid 命令查询 CPU 对向量化指令的支持情况。

1.2 向量化执行框架：数据局部性与运行时开销

执行引擎常规按行处理的方式，存在以下三个问题：

1. CPU Cache 命中率差。一行的多列（字段）数据的内存紧挨在一起，哪怕只对其中的一个字段做操作，其他字段所占的内存也需要加载进来，这会抢占稀缺的 Cache 资源。Cache 命失会导致被请求的数据从内存加载进 Cache，等待内存操作完成会导致 CPU 执行指令暂停（Memory Stall），这会增加延时，还可能浪费内存带宽。

2. 变长字段影响计算效率。假设一行包括 int、string、int 三列，其中 int 类型是固定长度，而 string 是变长的（一般表示为 int len + bytes content），变长列的存在会导致无法通过行号算 offset 做快速定位。
3. 虚函数调用带来额外开销。对一行的多列进行处理通常会封装在一个循环里，会抽象出一个类似 handle 的接口（C++ 虚函数）用于处理某类型数据，各字段类型会 override 该 handle 接口。虚函数的调用多一步查表，且无法被内联，循环内高频调用虚函数的性能影响不可忽视。

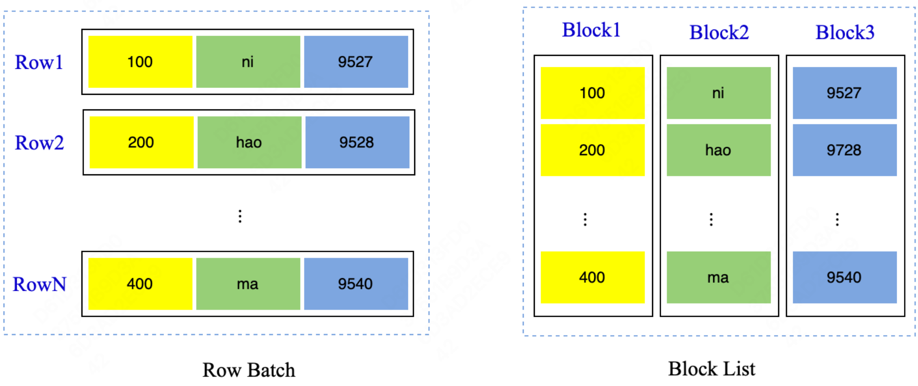


图 2：row by row VS blcok by block

因此，要让向量化计算发挥威力，只使用 SIMD 指令还不够，还需要对执行框架层面进行改造，变 Row By Row 为 Block By Block：

1. 数据按列组织替代按行组织（在 Clickhouse 和 Doris 里叫 Block，Velox 里叫 Vector），这将提高数据局部性。参与计算的列的多行数据会内存紧凑的保存在一起，CPU 可以通过预取指令将接下来要处理的数据加载进 Cache，从而减少 Memory Stall。不参与计算的列的数据不会与被处理的列竞争 Cache，这种内存交互的隔离能提高 Cache 亲和性。
2. 同一列数据在循环里被施加相同的计算，批量迭代将减少函数调用次数，通过模版能减少虚函数调用，降低运行时开销。针对固定长度类型的列很容易被并行处理（通过行号 offset 到数据），这样的执行框架也有利于让编译器做

自动向量化代码生成，显著减少分支，减轻预测失败的惩罚。结合模板，编译器会为每个实参生成特定实例化代码，避免运行时查找虚函数表，并且由于编译器知道了具体的类型信息，可以对模板函数进行内联展开。

```cpp
// 非向量化处理框架：先行后列
for (int i = 0; i < rows_per_batch; ++i) {
  Row row = row_batch->get_row(i);
  for (int j = 0; j < schema.column_num(); ++j) {
    void* data = row.get(schema.column(j).offset);
    switch (schema.column(j).type) {
      case TypeInt: // 一次一个元素
        handle_int((int*)data);
      case TypeString:
        handle_string((StringView*)data);
    }
  }
}
```

```cpp
// 向量化执行框架：一次一批（多行）
for (int i = 0; i < schema.column_num(); ++i) {
  switch (schema.column(i).type) {
    case TypeInt:
      handle_int((int*)blocks[i].data(), rows_per_batch);
    case TypeString:
      handle_string((StringView*)blocks[i].data(), rows_per_batch);
  }
}
```

图 3：向量化执行框架示例

1.3 如何使用向量化计算

1. 自动向量化（Auto-Vectorization）。当循环内没有复杂的条件分支，没有数据依赖，只调用简单内联函数时，通过编译选项（如 `gcc -ftree-vectorize`、-O3），编译器可以将顺序执行代码翻译成向量化执行代码。可以通过观察编译 hint 输出和反汇编确定是否生产了向量化代码。
    - 编译 hint 输出，编译：`g++ test.cpp -g -O3 -march=native -fopt-info-vec-optimized`，执行后有类似输出“test.cpp:35:21: note: loop vectorized”。
    - 反汇编，`gdb test + (gdb) disassemble /m function_name`，看到一些 v 打头的指令（例如 vmovups、vpaddd、vmovups 等）。
2. 使用封装好的函数库，如 Intel Intrinsic function、xsimd 等。这些软件包中的内置函数实现都使用了 SIMD 指令进行优化，相当于 high level 地使用了向量化指令的汇编，详见：https://www.intel.com/content/www/us/en/docs/intrinsics-guide/index.html。
3. 通过 asm 内嵌向量化汇编，但汇编指令跟 CPU 架构相关，可移植性差。
4. 编译器暗示：

◦ 使用编译指示符（Compiler Directive），如 Cilk（MIT 开发的用于并行编程的中间层编程语言和库，它扩展了 C 语言）里的 #pragma simd 和 OpenMP 里的 #pragma omp simd。

◦ Compiler Hint。通过 __restrict 去修饰指针参数，告诉编译器多个指针指向不相同不重叠的内存，让编译器放心大胆的去优化。

5. 如果循环内有复杂的逻辑或条件分支，那么将难以向量化处理。

以下是一个向量化版本数组相加的例子，使用 Intel Intrinsic function：

```
#include <immintrin.h> // 包含 Intrinsic avx 版本函数的头文件

void addArraysAVX(const int* a, const int* b, int* c, int num) {
  assert(num % 8 == 0); // 循环遍历数组，步长为 8，因为每个 __m256i 可以存储 8 个
32 位整数
  for (int i = 0; i < num; i += 8) {
    __m256i v_a = _mm256_load_si256((__m256i*)&a[i]); // 加载数组 a 的下一个
8 个整数到向量寄存器
    __m256i v_b = _mm256_load_si256((__m256i*)&b[i]); // 加载数组 b 的下一个
8 个整数到向量寄存器
    __m256i v_c = _mm256_add_epi32(v_a, v_b); // 将两个向量相加，结果存放在向
量寄存器
    _mm256_store_si256((__m256i*)&c[i], v_c); // 将结果向量存储到数组 c 的内存
  }
}

int main(int argc, char* argv[]) {
  const int ARRAY_SIZE = 64 * 1024;
  int a[ARRAY_SIZE] __attribute__((aligned(32))); // 按 32 字节对齐，满足某些
向量化指令的内存对齐要求
  int b[ARRAY_SIZE] __attribute__((aligned(32)));
  int c[ARRAY_SIZE] __attribute__((aligned(32)));
  srand(time(0));
  for (int i = 0; i < ARRAY_SIZE; ++i) {
    a[i] = rand(); b[i] = rand(); c[i] = 0; // 对数组 a 和 b 赋随机数初始值
  }

  auto start = std::chrono::high_resolution_clock::now();
  addArraysAVX(a, b, c, ARRAY_SIZE);
  auto end = std::chrono::high_resolution_clock::now();
  std::cout << "addArraysAVX took " << std::chrono::duration_cast<std::
chrono::microseconds>(end - start).count() << " microseconds." << std::
endl;
  return 0;
}
```

addArraysAVX 函 数 中 的 _mm256_load_si256、_mm256_add_epi32、_mm256_store_si256 都是 Intrinsic 库函数，内置函数命名方式是

- 操作浮点数：_mm(xxx)_name_PT
- 操作整型：_mm(xxx)_name_epUY

其中 (xxx) 代表数据的位数，xxx 为 SIMD 寄存器的位数，若为 128 位则省略，AVX 提供的 __m256 为 256 位；name 为函数的名字，表示功能；浮点内置函数的后缀是 PT，其中 P 代表的是对矢量（Packed Data Vector）还是对标量（scalar）进行操作，T 代表浮点数的类型（若为 s 则为单精度浮点型，若为 d 则为双精度浮点）；整型内置函数的后缀是 epUY，U 表示整数的类型（若为无符号类型则为 u，否在为 i），而 Y 为操作的数据类型的位数。

编译：g++ test.cpp -O0 -std=c++11 -mavx2 -o test。选项 -O0 用于禁用优化（因为开启优化后有可能自动向量化），-mavx2 用于启用 AVX2 指令集。

测试发现：非向量化版本 addArrays 耗时 170 微秒，而使用 Intrinsic 函数的向量化版本 addArraysAVX 耗时 58 微秒，耗时降为原来的 1/3。

2 为什么要做 Spark 向量化计算

从业界发展情况来看，近几年 OLAP 引擎发展迅速，该场景追求极致的查询速度，向量化技术在 Clickhouse、Doris 等 Native 引擎中得到广泛使用，降本增效的趋势也逐渐扩展到数仓生产。2022 年 6 月 DataBricks 发表论文《Photon- A Fast Query Engine for Lakehouse Systems》，Photon 是 DataBricks Runtime 中 C++ 实现的向量化执行引擎，相比 DBR 性能平均提升 4 倍，并已应用在 Databricks 商业版上，但没有开源。2021 年 Meta 开源 Velox，一个 C++ 实现的向量化执行库。2022 Databricks Data & AI Summit 上，Intel 与 Kyligence 介绍了合作开源项目 Gluten，旨在为 Spark SQL 提供 Native Vectorized Execution。**Gluten+Velox 的组合，使 Java 栈的 Spark 也可以像 Doris、Clickhouse 等**

Native 引擎一样发挥向量化执行的性能优势。

从美团内部来看，数仓生产有数万规模计算节点，很多业务决策依赖数据及时产出，若应用向量化执行技术，在不升级硬件的情况下，既可获得可观的资源节省，也能加速作业执行，让业务更快看到数据和做出决策。根据 Photon 和 Gluten 的公开数据，应用向量化 Spark 执行效率至少可以提升一倍，我们在物理机上基于 TPC-H 测试 Gluten+Velox 相 Spark 3.0 也有 1.7 倍性能提升。

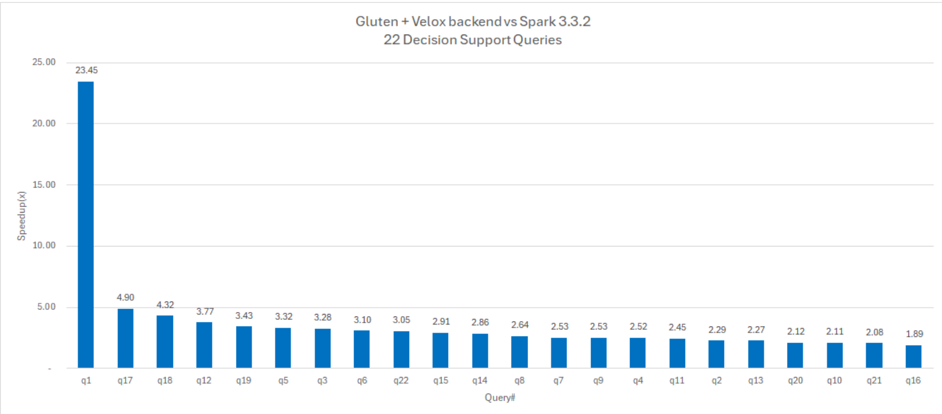


图 4：Gluten+Velox 在 TPC-H 上的加速比，来自 Gluten

3 Spark 向量化计算如何在美团实施落地

3.1 整体建设思路

1. 更关注资源节省而不单追求执行加速。Spark 在美团主力场景是离线数仓生产，与 OLAP 场景不同，时间相对不敏感，但资源（内存为主）基数大成本敏感。离线计算历史已久，为充分利用存量服务器，我们不能依赖硬件加速的手段如更多的内存、SSD、高性能网卡。我们评估收益的核心指标是总「memory*second」降低。
2. 基于 C++/Rust 等 Native 语言而非 Java 进行开发。Java 语言也在向量化执行方面做尝试，但 JVM 语言对底层控制力弱（如无法直接内嵌 SIMD 汇

编），再加上 GC 等固有缺陷，还远远谈不上成熟，而系统向的语言（C/C++、Rust）则成为挖掘 CPU 向量化执行潜能的首选。

3. 可插拔、面向多引擎而非绑定 Spark。虽然面向不同工作负载的各类大数据引擎层出不穷，但其架构分层则相似，一般包括编程接口、执行计划、调度、物理执行、容错等，尤其执行算子部分有较多可复用实现。如 Meta 内部主要大数据引擎有 Presto 和 Spark，建设一个跨引擎的执行库，优化同时支持 Presto 和 Spark 显然是更好的选择；OLAP 引擎向量化计算本身就是标配；流计算引擎出于性能考虑，也可以攒批而非一条条处理数据（mini batch），因此向量化执行也有性能提升空间。我们认为面向不同场景设计的大数据引擎，有可能共用同一个高性能向量化执行库。
4. 使用开源方案而非完全自研。Spark 有几百个 function 和 operator，向量化改造的工作量巨大，从性能、完成度、适配成本、是否支持多引擎、社区的活跃度等方面综合考虑，我们最终选择了 Gluten+Velox 的方案。
5. 迁移过程对用户透明，保证数据一致。Spark 的几百个 function 和 operator 都要通过 C++ 重新实现，同时还涉及 Spark、Gluten、Velox 版本变化，很容易实现出现偏差导致计算结果不一致的情况。我们开发了一个用于升级验证的黑盒测试工具（ETL Blackbox Test），可以将一个作业运行在不同版本的执行引擎上进行端到端验证，包括执行时间、内存及 CPU 资源使用情况、作业数据的对比结果（通过对比两次执行的行数，以及每一列所有数据 md5 的加和值来确定数据是否一致）。

3.2 Spark+Gluten+Velox 计算流程

通过 Spark 的 plugin 功能，Gluten 将 Spark 和向量化执行引擎（Native backend，如 Velox）连接起来，分为 Driver plugin 和 Executor Plugin。在 Driver 端，SparkContext 初始化时，Gluten 的一系列规则（如 ColumnarOverrideRules）通过 Spark Extensions 注入，这些规则会对 Spark 的执行计划进行校验，并把 Gluten 支持的算子转换成向量化算子（如 FileScan 会转换成 NativeFileScan），不

能转换的算子上报 Fallback 的原因，并在回退的部分插入 Column2Row、Row-2Column 算子，生成 Substrait 执行计划。在 Executor 端，接收到 Driver 侧的 LaunchTask RPC 消息传输的 Substrait 执行计划后，再转换成 Native backend 的执行计划，最终通过 JNI 调用 Native backend 执行。

Gluten 希望能尽可能多的复用原有的 Spark 逻辑，只是把计算部分转到性能更高的向量化算子上，如作业提交、SQL 解析、执行计划的生成及优化、资源申请、任务调度等行为都还由 Spark 控制。

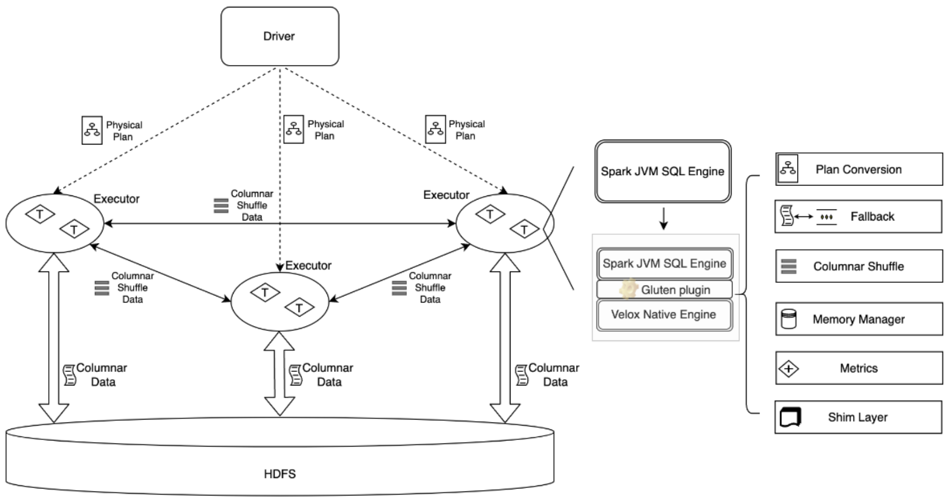


图 5：Spark+Gluten+Velox 架构图

3.3 阶段划分

在我们开始 Spark 向量化项目时，开源版本的 Gluten 和 Velox 还没有在业界 Spark 生产环境大规模实践过，为了降低风险最小代价验证可行性，我们把落地过程分为以下五个阶段逐步进行：

1. **软硬件适配情况确认**。Velox 要求 CPU 支持 bmi、bmi2、f16c、avx、avx2、sse 指令集，需要先确定服务器是否支持；在生产环境运行 TPC-DS 或者 TPC-H 测试，验证理论收益；公司内部版本适配，编译运行，跑通典

型任务。当时 Gluten 只支持 Spark3.2 和 Spark3.3，考虑到 Spark 版本升级成本更高，我们暂时将相关 patch 反打到 Spark3.0 上。这个阶段我们解决了大量编译失败问题，建议用社区推荐的 OS，在容器中编译 & 运行；如果要在物理机上运行，需要把相关依赖部署到各个节点，或者使用静态链接的方式（开启 vcpkg）。

```
cat /proc/cpuinfo | grep --color -wE "bmi|bmi2|f16c|avx|avx2|sse"
```

1. **稳定性验证**。确定测试集，完善稳定运行需要的 feature，以达到比较高的测试通过率，包括支持 ORC、Remote shuffle、HDFS 适配、堆内堆外的内存配置等。本阶段将测试通过率从不足 30% 提升到 90% 左右。
2. **性能收益验证**。由于向量化版本和原生 Spark 分别使用堆外内存和堆内内存，引入翻倍内存的配置，以及一些高性能 feature 支持不完善，一开始生产环境测试性能结果不及预期。我们逐个分析解决问题，包括参数对齐、去掉 arrow 中间数据转换、shuffle batch fetch、Native HDFS 客户端优化、使用 jemelloc、算子优化、内存配置优化、HBO 适配等。本阶段将平均资源节省从 -70% 提升到 40% 以上。
3. **一致性验证**。主要是问题修复，对所有非 SLA 作业进行大规模测试，筛选出稳定运行、数据完全一致、有正收益的作业。
4. **灰度上线**。将向量化执行环境发布到所有服务器，对符合条件的作业分批上线，增加监控报表，收集收益，对性能不及预期、发生数据不一致的作业及时回退原生 Spark 上。此过程用户无感知。

整个实施过程中，我们通过收益转化漏斗找到收益最大的优化点，指导项目迭代。下图为 2023 年某一时期的相邻转化情况。

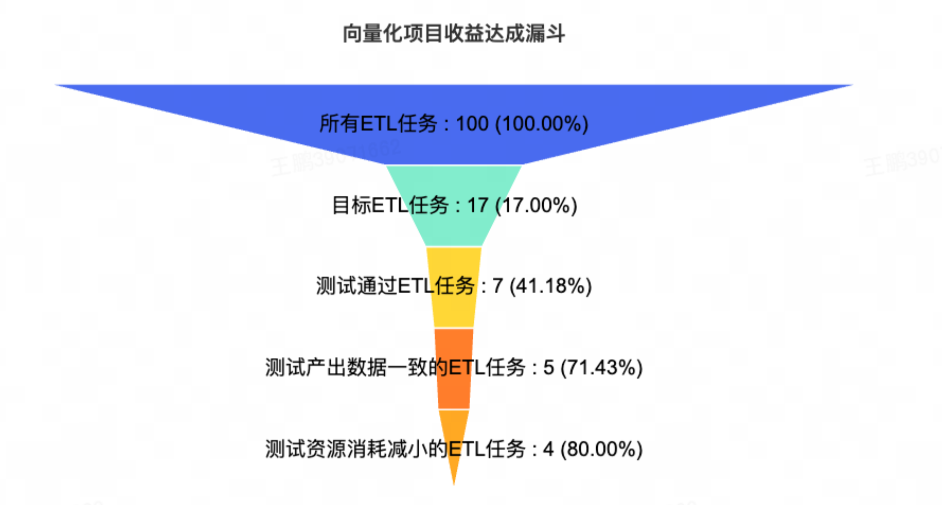


图 6：Spark 向量化项目收益转化漏斗图

4 美团 Spark 向量化计算遇到的挑战

4.1 稳定性问题

1. **聚合时 Shuffle 阶段 OOM**。在 Spark 中，Aggregation 一般包括 Partial Aggregation、Shuffle、Final Aggregation 三个阶段，Partial Aggregation 在 Mapper 端预聚合以降低 Shuffle 数据量，加速聚合过程、避免数据倾斜。Aggregation 需要维护中间状态，如果 Partial Aggregation 占用的内存超过一定阈值，就会提前触发 Flush 同时后续输入数据跳过此阶段，直接进入 ShuffleWrite 流程。Gluten 使用 Velox 默认配置的 Flush 内存阈值（Spark 堆外内存 *75%），由于 Velox 里 Spill 功能还不够完善（Partial Aggregation 不支持 Spill），这样大作业场景，后续 ShuffleWrite 流程可能没有足够的内存可以使用（可用内存 <25%*Spark 堆外内存），会引起作业 OOM。我们采用的策略是通过在 Gluten 侧调低 Velox Partial Aggregation 的 Flush 阈值，来降低 Partial Aggregation 阶段的内存占用，避免大作业 OOM。这个方案在可以让大作业运行通过，但是理论上提前触发 Partial

Aggergation Flush 会降低 Partial Aggretation 的效果。更合理的方案是 Partial Aggregation 支持 Spill 功能，Gluten 和 Velox 社区也一直在完善对向量化算子 Spill 功能的支持。

2. **SIMD 指令 crash**。Velox 对数据复制做了优化，如果该类型对象是 128bit（比如 LongDecimal 类型），会通过 SIMD 指令用于数据复制以提升性能。如下图所示，Velox 库的 FlatVector<T>::copyValuesAndNulls() 函数里的一行赋值会调用 T::operator=()，调用的 movaps 指令必须作用于 16B 对齐的地址，不满足对齐要求会 crash。我们在测试中复现了 crash，通过日志确定有未按 16B 对齐的地址出现。无论是 Velox 内存池还是 Gluten 内存池分配内存都强制做了 16B 对齐，最终确认是 Arrow 内存池分配出的地址没对齐（Gluten 用了三方库 Arrow）。这个问题可以通过为 LongDecimal 类型重载 operator= 操作符修复，但这样做可能影响性能，也不彻底，因为不能排除还有其他 128bit 类型对象存在。最终我们与 Gluten 社区修改了 Arrow 内存分配策略，强制 16B 对齐。

```
    rows.applyToSelected([&](vector_size_t row) {
      if (row >= source->size()) {
        return;
      }
      if (sourceValues) {
        rawValues_[row] = sourceValues[row];
      }
      if (rawNulls) {
        bits::setNull(
            rawNulls, row, sourceNulls && bits::isBitNull(sourceNulls, row));
      }
    });
  }
```

图 7：Crash 代码示例

4.2 支持 ORC 并优化读写性能

Velox 的 DWIO 模块原生只支持 DWRF 和 Parquet 两种数据格式，美团大部分表都是基于 ORC 格式进行存储的。DWRF 文件格式是 Meta 内部所采用的 ORC 分支版本，其文件结构与 ORC 相似，比如针对 ORC 文件的不同区域，可通过复用

DWRF 的 Reader 来完成相关数据内容的读取。

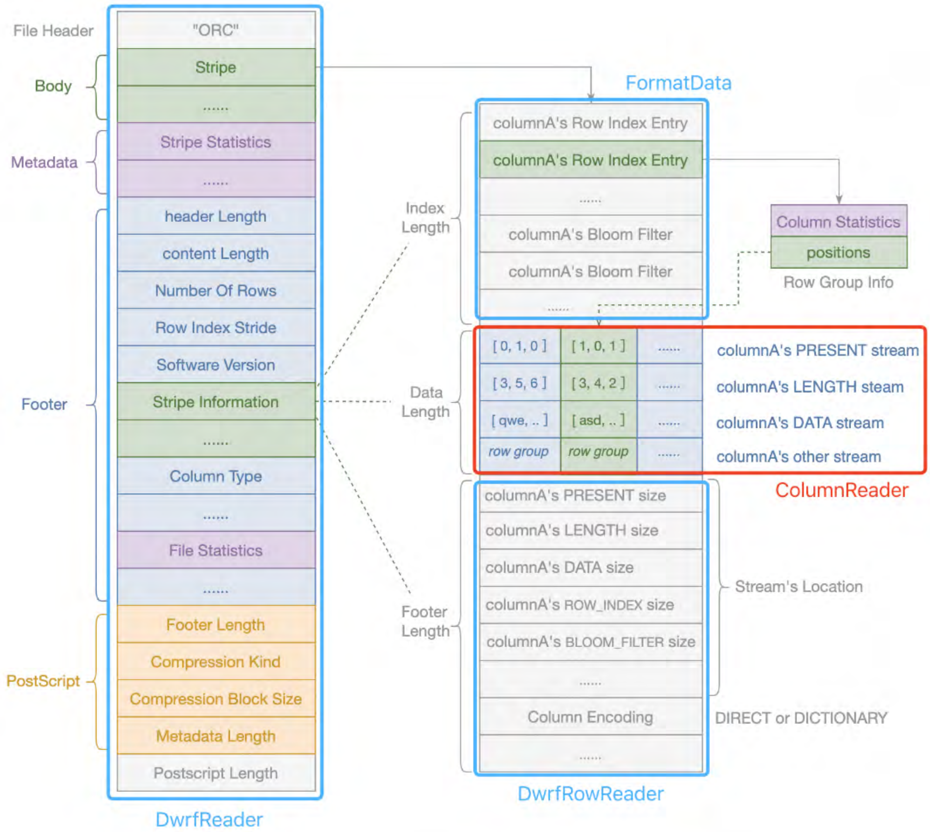


图 8：Dwrf 文件格式

- DwrfReader：用于读取文件层面的元数据信息，包括 PostScript、Footer 和 Header。
- DwrfRowReader： 用 来 读 取 StripeFooter， 以 便 确 定 每 个 column 的 Stream 位置。
- FormatData：用来读取 StripeIndex，从而确定每个 RowGroup 的位置区间。
- ColumnReader：用来读取 StripeData，完成不同 column 的数据抽取。

我们完善了 Velox ORC 格式的支持，并对读取链路做了优化，主要工作包括：

1. 支持 RLEv2 解码（Velox-5443）并在解码过程中完成 Filter 下推（Velox-6647）。我们将 Apache RLEv2 解码逻辑移植到了 Velox，通过 BMI2 指令集来加速 varint 解码过程中的位运算，并在解码过程中下推过滤不必要的数据。
2. 支持 Decimal 数据类型（Velox-5837）以及该数据类型的 Filter 下推处理（Velox-6240）。
3. ORC 文件句柄复用以降低 HDFS 的 NN 处理压力（Velox-6140）。出于线程安全层面的考虑，HdfsReadFile 每次 pread 都会开启一个新文件句柄来做 seek+read，客户端会向 NameNode 发送大量 open 请求，加重 HDFS 的压力。我们通过将文件的读取句柄在内部做复用处理（thread_local 模式），减少向 NN 发送的 open 请求。
4. 使用 ISA-L 加速 ORC 文件解压缩。我们对 ORC 文件读取耗时 trace 分析得出，zlib 解压缩占总耗时 60%，解码占 30%，IO 和其他仅占 10%，解压效率对 ORC 文件读取性能很关键。为此，我们对 ZlibDecompressor 做了重构，引入 Intel 的解压缩向量化库 ISA-L 来加速解压缩过程。

基于这些优化，改造后的 Velox 版 ORC Reader 读取时间减少一半，性能提升一倍。

图 9：Apache ORC 与改造后的 Velox ORC 读取性能对比，上为 Apache ORC

4.3 Native HDFS 客户端优化

首先介绍一下 HDFS C++ 客户端对 ORC 文件读取某一列数据的过程。第一步，读取文件的最后一个字节来确定 PostScript 长度，读取 PostScript 内容；第二步，通过 PostScript 确定 Footer 的存储位置，读取 Footer 内容；第三步，通过 Footer 确定每个 Stripe 的元数据信息，读取 StripeFooter；第四步，通过 StripeFooter 确定每个 Column 的 Stream 位置，读取需要的 Stream 数据。

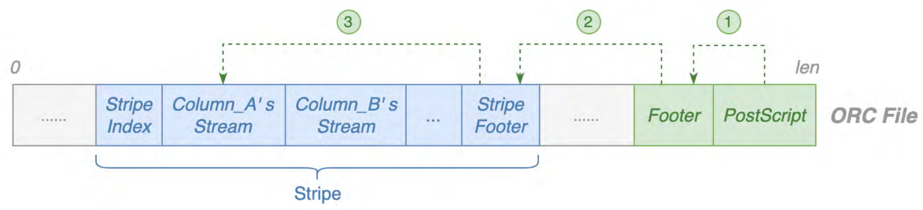


图 10：ORC 文件读取过程

在生产环境测试中，我们定位到两个数据读取相关的性能问题：

1. **小数据量随机读放大**。客户端想要读取 [offset ~ readEnd] 区间内的数据，发送给 DN 的实际读取区间却是 [offset ~ endOfCurBlock]，导致 [readEnd ~ endOfCurBlock] 这部分数据做了无效读取。这样设计主要是为了优化顺序读场景，通过预读来加快后续访问，然而针对随机读场景（小数据量下比较普遍），该方式却适得其反，因为预读出的数据很难被后续使用，增加了读放大行为。我们优化为客户端只向 DN 传递需要读取的数据区间，DN 侧不提前预取，只返回客户端需要的数据。

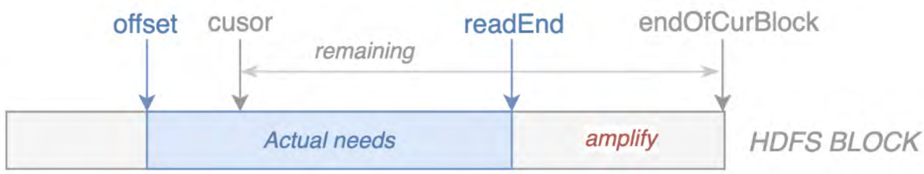


图 11：读放大过程示意图

2. **DN 慢节点导致作业运行时间变长**。我们发现很多大作业的 HDFS 长尾耗时非常高，HDFS 的平均 read 时延只有 10ms 左右，P99.99 时延却达到了 6 秒，耗时最长的请求甚至达到了 5 分钟，但在不启用 EC 场景下，HDFS 的每个 block 会有三副本，完全可以切换到空闲 DN 访问。为此我们对客户端的读请求链路做了重新的设计与调整，实时监测每个 DN 的负载情况，基于 P99.9 分位请求时延判定慢节点，并将读请求路由到负载较低的 DN 上面。

HDFS Native 客户端读优化之后，平均读写延迟降低了 2/3，吞吐提升 2 倍。

4.4 Shuffle 重构

Gluten 在 shuffle 策略的支持上，没有预留好接口，每新增一种 shuffle 模式需要较大改动。美团有自研的 Shuffle Service，其他公司也可能有自己的 Shuffle Service（如 Celeborn），为了更好适配多种 shuffle 模式，我们提议对 shuffle 接口重新梳理，并主导了此讨论和设计。

Gluten 中的 shuffle 抽象第一层是数据格式（Velox 是 RowVector，Gluten 引入的 Arrow 是 RecordBatch），第二层是分区方式（RoundRobin、SinglePart、Hash、Range），如果要支持新 shuffle 模式（local、remote），需要实现 2*4*2=16 个 writer，将会有大量冗余代码。分区具体实现应该与数据格式和 shuffle 模式无关，我们用组合模式替代继承模式。另外，我们在 shuffle 中直接支持了 RowVector，避免 Velox 和 Arrow 对应数据类型之间的额外转换开销。

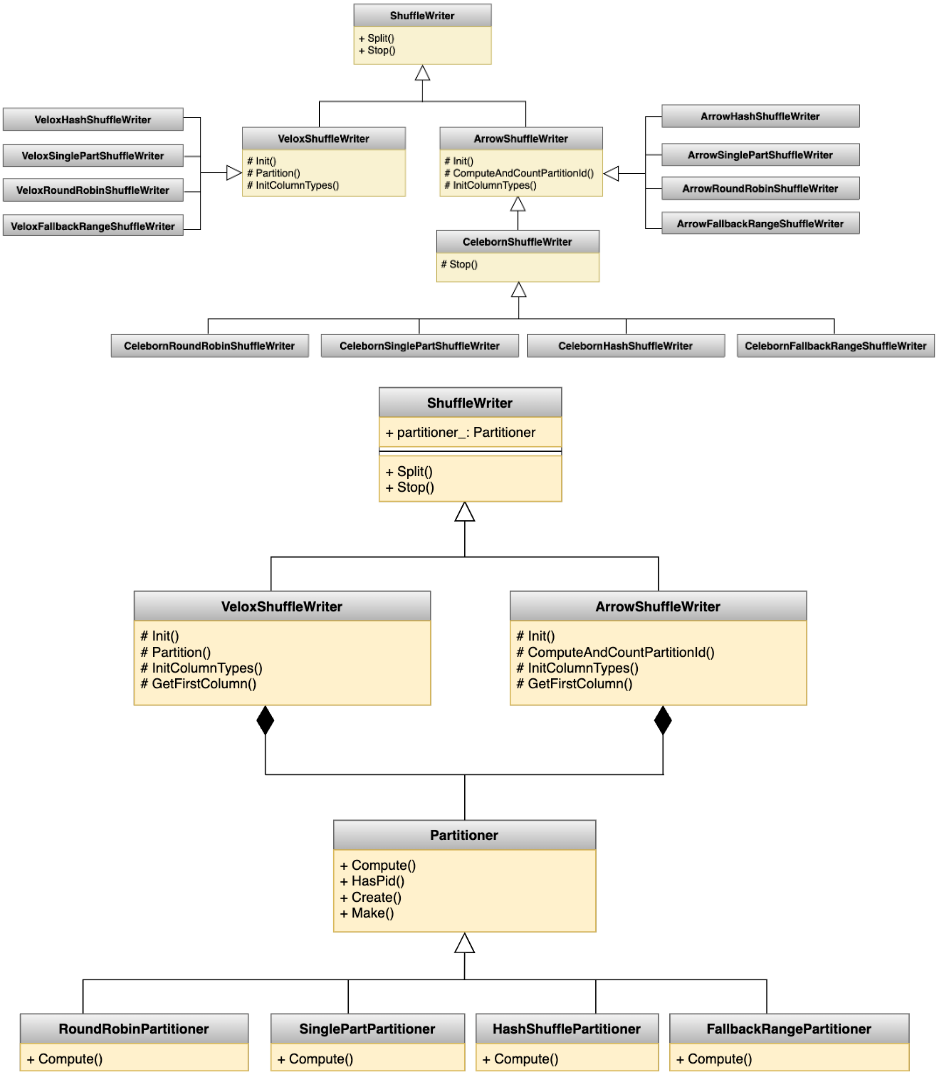


图 12：重构前后 shuffle 模块 UML 对比

4.5 适配 HBO

HBO（Historical Based Optimization）是通过作业历史运行过程中资源的实际使用量，来预测作业下一次运行需要的资源并设置资源相关参数的一种优化手段。美团过去在原生 Spark 上通过调配堆内内存取得了 8% 左右的内存资源节省。

Gluten 主要使用堆外内存（off-heap），这与原生 Spark 主要使用堆内内存（on-heap）不同。初期出于稳定性考虑 Gluten 和原生 Spark 的运行参数整体一致，总内存大小相同，Gluten off-heap 占比 75%，on-heap 占比 25%。这样配置既会导致内存利用率不高（原生 Spark 的内存使用率 58%，向量化版作业内存使用率 38%），也会使一部分作业 on-heap 内存配置偏低，在算子回退时导致任务 OOM。

我们把 HBO 策略推广到堆外内存，向量化计算的内存节省比例从 30% 提升到 40%，由于 heap 内存配置不合理的 OOM 问题全部消除。

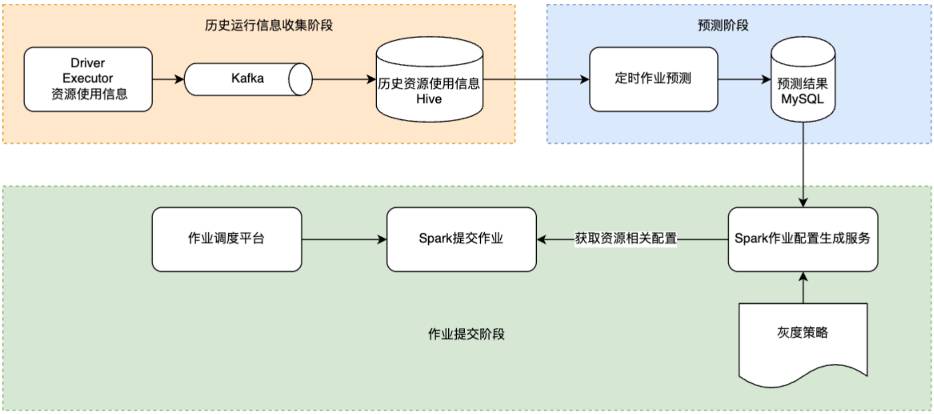


图 13：HBO 流程图

4.6 一致性问题

1. **低版本 ORC 数据丢失**。hive-0.13 之前使用的 ORC，Footer 信息不包含列名，只有 ID 用来表示第几列（如 Col1, Col2…）。Velox TableScan 算子在扫表的时候，如果下推的 Filter 里包含 IsNotNull(A)，会根据列名 A 查找该列数据，由于无法匹配到列名，会误判空文件，导致数据缺失。Spark 在生成读 ORC 表的执行计划时，通过访问 HiveMetaStore 得到表的 Schema 信息，并在物理算子 FileSourceScanExec 中保存了表的 Schema 信息。Gluten 对该算子进行 doTransform() 转换时，会把表的 Schema 信息序列化到 Substrait 的 ReadRel 里。接下来的 Substrait 计划转 Velox 计划

阶段，我们把表的 Schema 信息传给 Velox 的 HiveTableHandle，在构造 Velox 的 DwrfReader 时修正 ORC 文件 Footer 里的 Schema 信息（如果 Footer 的 Schema 不包含列名，就读取表 Schema 里的对应列的名称进行赋值），解决了这个问题。

2. **count distinct 结果错误**。比如这样一条 SQL：`select A, B, count(distinct userId), sum(amt) from t group by 1,2`，Gluten 会把 count(distinct userId) 变为 count(userId)，通过把 userId 加到 GroupingKey 里来实现 distinct 语义。具体处理过程如下：

步骤	处理过程	参数
1	Partial Aggregation	GroupingKey = <A, B, **userId**> AggregationExpression = partial_sum(amt)
2	Shuffle	Partitionkey = Hash(A, B, **userId**)
3	Partial Merge Aggregation (Spark) Intermediate Aggregation(Velox)	GroupingKey = <A, B, **userId**> AggregationExpression = merge_sum(amt)
4	Partial Aggregation for distinct	GroupingKey = <A, B> AggregationExpression = merge_sum(amt), **partial_count(userId)**
5	Shuffle	Partitionkey = Hash(A, B)
6	Final Aggregation	GroupingKey = <A, B> AggregationExpression = sum(amt), **count(userId)**

表 1：示例 SQL 在 Spark 中的处理步骤

在第 3 步的 Intermediate Aggregation 中，为了节省内存和加速执行，当 Velox 的 HashAggregate 算子满足触发 Flush 的条件时（HashTable 内存占用超过阈值或者聚合效果低于阈值），Velox 会标记 partialFull=true，触发 Flush 操作（计算 HashTable 里已经缓存数据的 Intermediate Result），并且后续输入的数据不再执行 Intermediate Aggregation 的计算，直接进入第 4 步的 Partial Aggregation。如果后续输入的数据里包含重复的 userId，count(userId) 会因为去重不彻底而结果错误。我们短期的修复方案是禁用 Intermediate Aggregation 的提前 Flush 功能，

直到所有数据都输入之后再获取该阶段的聚合结果。

这个方案的弊端有两个：1）HashTable 的内存占用会变大，需要同时开启 Hash-Aggregate 算子的 Spill 功能避免 OOM；2）直接修改了 Velox 的 HashAggregate 算子内部代码，从 Velox 自身的角度来看，没有单独针对 Distinct 相关的聚合做处理，随着后续迭代，可能影响所有用到 Intermediate Aggregation 的聚合过程。

鉴于此，Gluten 社区提供了一个更加均衡的解决方案，针对这类 Distinct Aggregation，生成执行计划时，Spark 的 Partial Merge Aggregation 不再生成 Intermediate Aggregation，改为生成 Final Aggregation（不会提前 flush、不使用 merge_sum），同时配合聚合函数的 Partial Companion 函数来返回 Intermediate 结果。这样就从执行计划转换策略层面规避这个问题，避免对 Velox 里 Final Aggregation 内部代码做过多的改动。

1. **浮点类型转换精度错误**。形如查询 `SELECT concat(col2, cast(max(col1) as string)) FROM (VALUES (CAST(5.08 AS FLOAT), 'abc_')) AS tab(col1, col2) group by col2;` 在 Spark 中返回 abc_5.08，在 Gluten 中返回 abc_5.079999923706055。浮点数 5.08 不能用二进制分数精确表达，近似表示成 5.0799999237060546875。Velox 通过函数 `folly::to<std::string>(T val)` 来实现 float 类型到 string 类型的转换，这个函数底层依赖开源库 google::double-conversion, folly 里默认设置了输出宽度参数 `DoubleToStringConverter::SHORTEST(可以准确表示double类型的最小宽度)`，转换时经过四舍五入之后返回 5.079999923706055。我们把宽度参数改为 `DoubleToStringConverter::SHORTEST_SINGLE(可以准确表示float类型的最小宽度)`，转换时经过四舍五入之后返回 5.08。

5 上线效果

我们已上线了 2 万多 ETL 作业，平均内存资源节省 40%+，平均执行时间减少

13%，证明 Gluten+Velox 的向量化计算方案生产可行。向量化计算除了能提高计算效率，也能提高读写数据的效率，如某个作业的 Input 数据有 30TB，过去需要执行 7 小时，绝大部份时间花在了读数据和解压缩上面。使用向量化引擎后，因为上文提到的 ISA-L 解压缩优化，列转行的开销节省，以及 HDFS Native 客户端优化，执行时间减少到 2 小时内。

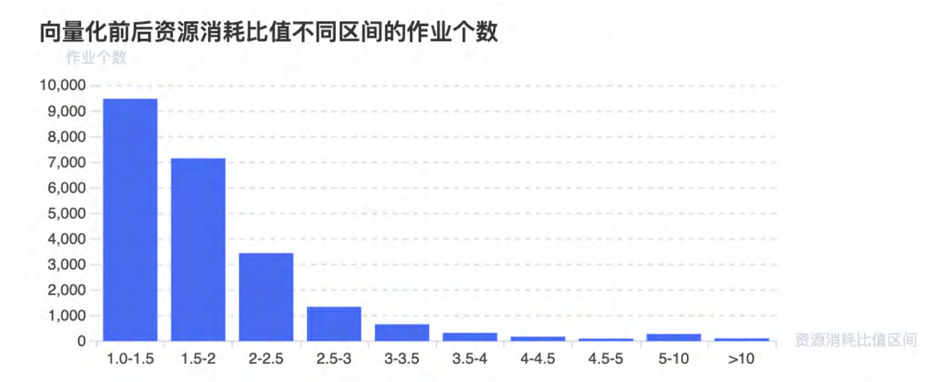


图 14：上线优化效果

6 未来规划

我们已上线向量化计算的 Spark 任务只是小部分，计划 2024 年能让绝大部分的 SQL 任务运行在向量化引擎上。

6.1 Spark 向量化之后对开源社区的跟进策略

Spark、Gluten、Velox 三个社区各有自己考虑和版本发布节奏，从一个社区到多个社区的引擎维护复杂度上升。我们的应对有二，一是计算引擎有不同层次，Spark 升级主要考虑功能语义实现、执行计划、资源和 task 调度，Gluten 和 Velox 的升级主要考虑物理算子性能优化，各取所长；二是尽量减少和社区的差异，公司内部适配只在 Spark 中实现，公司内的 UDF 以 git submodule 形式单独维护。

1. 升级到 Spark3.5。Gluten 最低支持的 Spark 版本为 3.2，23 年我们为了降低验证成本，选择在 Spark3.0 兼容 Gluten，但继续升级迭代成本比较高，

在推广之前，应该升级到更新的 Spark 版本。Spark3.5 将会是 Gluten 社区对 Spark3.x 上长期支持的稳定版本。高版本 Spark 也有一些额外收益，我们基于 TPC-H 实测，Spark3.5 相比 Spark3.0，「memory*second」减少 40%，执行时间减少 17%，根据之前升级经验，生产环境大约能达到一半效果。

2. 保持 Spark 版本长期稳定。高版本 Spark 对 Hadoop 版本的升级迭代带来比较高适配成本，内部迭代的 feature 也有比较高的迁移成本，因此我们平均 3 年才会升级一次 Spark 版本，更多是将需要的 feature 合并到内部分支。
3. 快速跟进 Gluten/Velox 新版本。升级到 Spark3.5 之后，我们内部 Spark 版本与 Gluten 社区的兼容性成本很低，并且向量化相关 feature 还会持续迭代，预计每半年可升级一次线上版本。

6.2 提升向量化覆盖率的策略

1. 扩大向量化算子和 UDF 范围。我们整理了影响权重最高的几十个算子回退问题与 Gluten 社区一起解决，对于大量内部 UDF，则会探索用大模型来将 UDF 批量改写为 C++ 版本的向量化实现。
2. 扩大 File format 支持向量化范围。美团内部有约 20% 的表为 textfile 格式，还有接近 10% 的表使用内部开发的 format，只能按行读取也不支持下推，加上行转列都会有额外性能开销，影响最终效果。我们将会把 textfile 全部转为 ORC，为自研 format 提供 C++ 客户端，进一步提升向量化计算性能。

7 致谢

感谢 Intel Gluten 合作伙伴高明、周渊、宾伟、韦廷、宏泽、莫芮、飞龙、马榕、镇辉、成成等的大力支持和辛勤付出，也感谢 Gluten 和 Velox 社区贡献者的开源精神和无私奉献。

大前端 | 如何突破动态化容器的天花板?

本文整理自美团技术沙龙第 83 期《前端新动向》(B 站视频)。长久以来,容器要实现动态化和双端复用,难免要牺牲掉一些性能。有没有办法让动态化容器的性能尽可能接近原生?美团金服大前端团队给出了一种解决方案,尝试突破动态化容器的天花板。

1 动态化容器的天花板

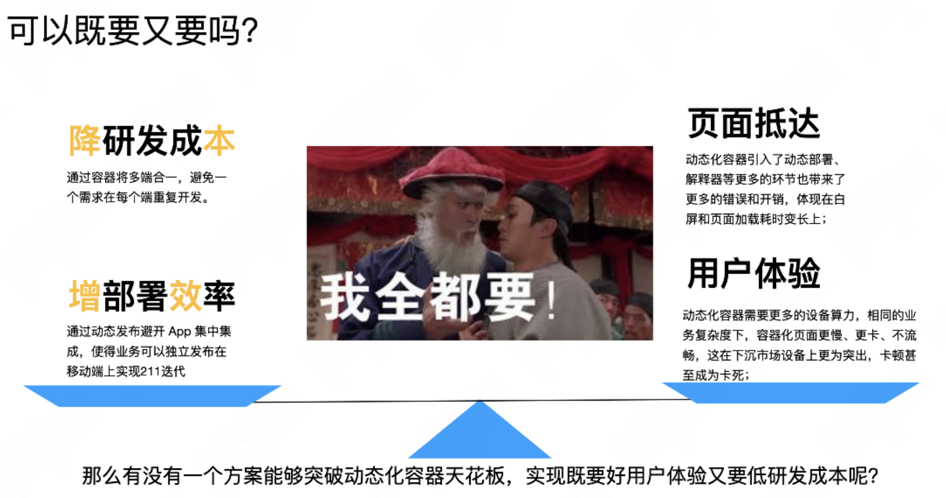
自 2015 年 React Native 推出至今 9 年时间,各类容器(动态化容器简称,下同)方案已经成为业界前端的普遍选择。业界有微信(小程序)、抖音(Lynx)、拼多多(Lego)、支付宝(Nebula/BirdNest)、京东(Taro-Native)等。美团也有 MRN、MMP/MSC 等容器。可以说容器是前端工程的关键基石,也是绕不开的话题。

过去我们做动态化改造主要为了解决以下问题:

1. **降研发成本**:通过容器将多端合一,避免一个需求在每个端重复开发,以改善研发成本结构。随着 HarmonyOS NEXT 的推广,这个优势将变得更大。
2. **增部署效率**:通过动态发布避开 App 集中集成,使得业务在移动端上可以独立部署和发布、实现 211 迭代,提升业务迭代面客效率。

然而凡事有利必有弊,有用必有费。动态化容器在解决上述问题的同时也带来以下问题:

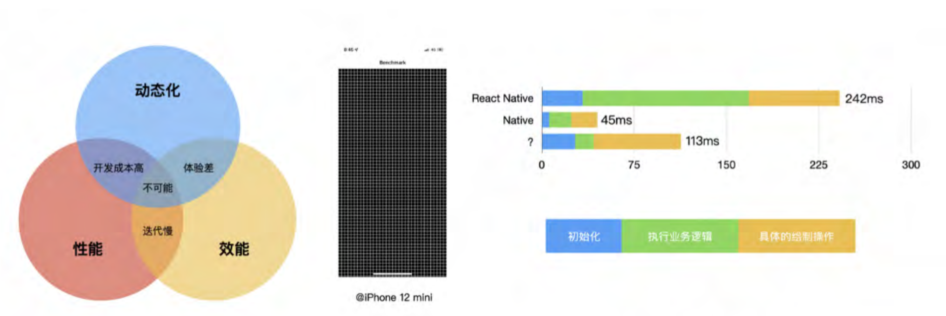
1. **降低页面成功**:动态化容器引入了动态部署、解释器等更多的环节。在增加整体复杂度的同时,更多的环节也带来了更多的错误和计算开销,具体体现在页面白屏和页面加载耗时增加上。
2. **牺牲用户体验**:动态化容器需要更多的硬件算力,相同的业务复杂度下,容器化页面相较原生页面更慢、更卡、不流畅,这在下沉市场设备上更为突出,卡顿甚至成为卡死。



动态化容器的绝对天花板是原生应用，目前事实天花板是 React Native/WebView。

定性地看前端容器天花板的问题，这里引述我们美团容器界的一位前辈的理论：性能、效能、动态化是动态化容器的不可能三角（下图左）。现有的通用的容器方案都是在这三个维度做“三选二”。

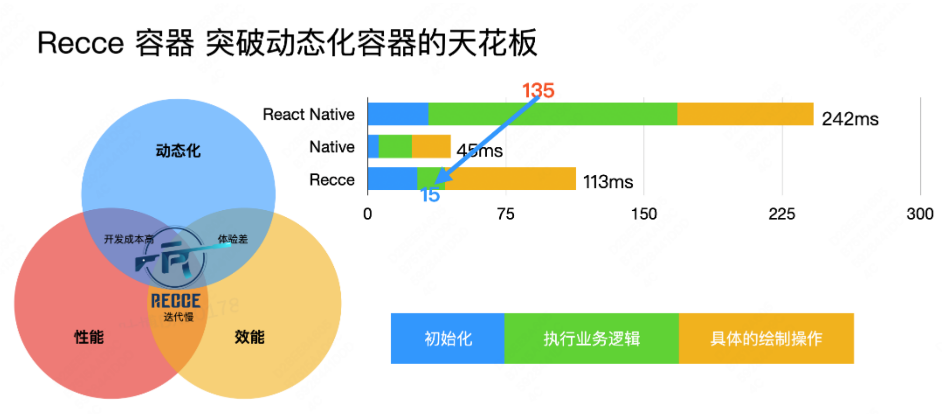
定量地看动态化容器，下图（右）展示了一个 3000 个相同视图节点的简单 Bench-mark 页面。没有额外逻辑，也没有网络请求，以 React Native 为例，在同一台设备上，在 React Native 上做一个这个页面，动态化页面加载耗时大约是 Native 原生页面的 5 倍。



定性看前端容器天花板

定量看动态化容器的天花板

这个测试结果不一定容易接受，但是很好理解。在相同的业务复杂度下，**动态化容器为了实现动态化，引入了一个逻辑解释器，增加了解释执行和与解释器通信的额外计算开销，这就是动态化页面性能表现差的主要原因**。既享受动态化的好处，又不牺牲用户性能体验，只在现有的方案上做选择是不可能达到的。不过金服大前端团队在这个问题上却是取得了一些实质性进展。



先说一下我们目前取得的成果：美团金服前端团队做了一个新的容器 Recce，然后在同样的测试页中，我们将执行业务逻辑部分的速度提升了 8 倍，整体的页面加载速度提升了一倍。而在实际的业务中，页面加载速度也实现了 3 倍的提升，在兼顾动态化和效能的前提下，实现了性能的大跨步提升，性能表现接近 Native 原生。下文将重点介绍 Recce 具体是怎么实现的，希望能够给大家提供一些帮助或借鉴。

2 容器分类及前期思考

首先，当我们计划做一个容器之前，需要先对现有容器建立基本的认识。

前端容器最重要部分之一在于绘制图形界面以完成人机交互，现有主要容器方案按照绘制方式可归纳为下列三类：

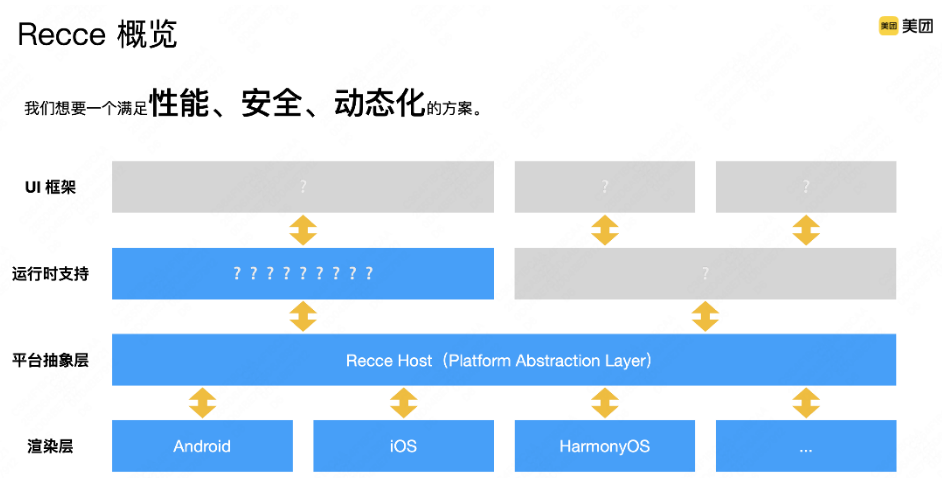
- 第一类可以称为基于 Web 的方案，这类的共同特点就是调用 WebView，通过 JavaScript 和 CSS 去绘制页面，然后通过 Web 提供的接口去和宿主通信。

- 第二类称之为“自己绘制”，它会调用更底层的 OpenGL 等图形的绘制框架，同时也会有自己的一套方案和语言去标记和编写自己的这些页面。
- 第三类则是去调用系统的 UI 框架，就是基于平台提供的 UI 框架去进行绘制，和第二种的区别是，会有一层抹平平台差异的平台抽象层。

对现有的容器再做进一步的结构分解，方便我们对不同容器方案之间做对比，这里拆分为四层：

- 第一层，也就是最上层的 UI 框架，跟我们直接平时写的代码相关，它会直接决定业务代码的样子。
- 第二层，其作用就像它的名字一样，也就是运行时支持，为运行 UI 框架提供支持，在这里会有解释器或者是标准库之类的东西。
- 第三层，会对不同平台去做一层不同的抽象，比如像 RN 会对视图操作之类的统一为相同概念和统一接口。
- 第四层，渲染层，对应着不同分类选择的各自的渲染方式。

那么下面，我们就会基于上图对现有容器这个分类和结构为认知基础，结合之前影响容器性能表现最大的因素在于逻辑解释器执行效率和逻辑解释器通信效率这个认知，再去考虑实现一个满足性能、安全、动态化的容器方案该怎么做。



3 Recce 的选型与搭建

所以如何选择 UI 框架、运行时支持、平台抽象层、渲染层 来实现一个高性能、安全的动态化容器呢。首先把渲染层放到最后，因为渲染层作为最底层并不影响上层选

型。然后我们首先讨论运行时支持怎么选，准确的说是解释器和编程语言怎么选，因为编程语言会影响上层 UI 框架，而解释器也影响到平台抽象层中的通信部分，所以接下来的讨论顺序是：**解释器 & 编程语言、UI 框架、渲染层以及整体架构**这四部分。

3.1 解释器 & 编程语言的选择

解释器以 Wasm 为主，JavaScript 为辅

前文也提到，我们期望能获得一个既能满足性能，又安全，同时还可以动态化这样的一个方案。既然必须动态化，就必须有逻辑解释器 [1]，问题就变成了怎样选一个性能好且安全的解释器，并且成本在可接受范围。现成的解释器还是不少的，前端范畴有 V8、JavaScriptCore、QuickJS 等 JavaScript 解释器，有符合 WebAssembly（后简称 Wasm）规范的 Wasmer、WasmEdge[2]；大家日常工作中会接触到的 Ruby、Python；还有游戏行业用的比较多的 Lua、C#[3]。

首先可以排除掉 Ruby 和 Python 这两个语言和解释器，无论是性能还是生态都不如 JavaScript。Lua 和 C# 也可以排除，主要是游戏生态和前端差距太远，晚饭想吃炸鸡，中午才开始孵蛋显然就来不及了。在 JavaScript 解释器和 Wasm 解释器两个大范围里，Wasm 解释器在性能和安全上较 JavaScript 解释器有决定性优势，生态上较 JavaScript 略差，但都在 W3C 标准范围内，一样可以运行在 H5 和小程序里，"晚饭想吃炸鸡，中午开始杀鸡还是来得及的"。

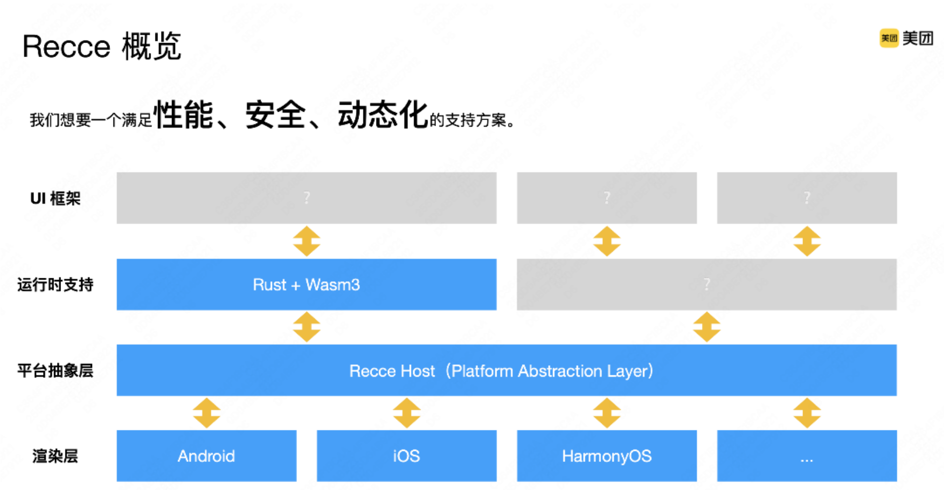
所以在解释器的选型上，就确定了 Wasm 解释器为主，JavaScript 为辅的基本策略。而 Wasm 解释器具体选择是 Wasm3，原因有两方面：第一，这是在不支持 JIT[4] 下最快的 Wasm 解释器；第二，对包大小占用非常少。

编程语言选择

在确定了 Wasm 解释器之后，编程语言的选择就变成了，在支持 Wasm 的语言里选择性能、安全和成本最优的。理论上可以编译成 WebAssembly 执行的语言非常之多，但真正成熟到可以上生产环境的只有 C、C++、Rust、Go 这四种 [5]。首先可以先排除掉 C，虽然性能好但是不支持高级抽象只适合用在嵌入式等极端场景，不适

合用来前端写业务。然后可以排除 C++，性能表现上 C++ 和 Rust 不相上下都好过 Go，但是错误管理和内存安全上输 Rust 一筹，并且 C++ 在前端业务层也是生态基本为零，不像 Rust 在前端生态发展迅猛。最后可以排除 Go，在性能表现上、类型系统设计、错误管理、还有前端生态上都输 Rust 一筹。

综上所述，在运行时我们就选择了 Rust 和 Wasm3，JavaScript 和 QuickJS 后面再进行介绍。



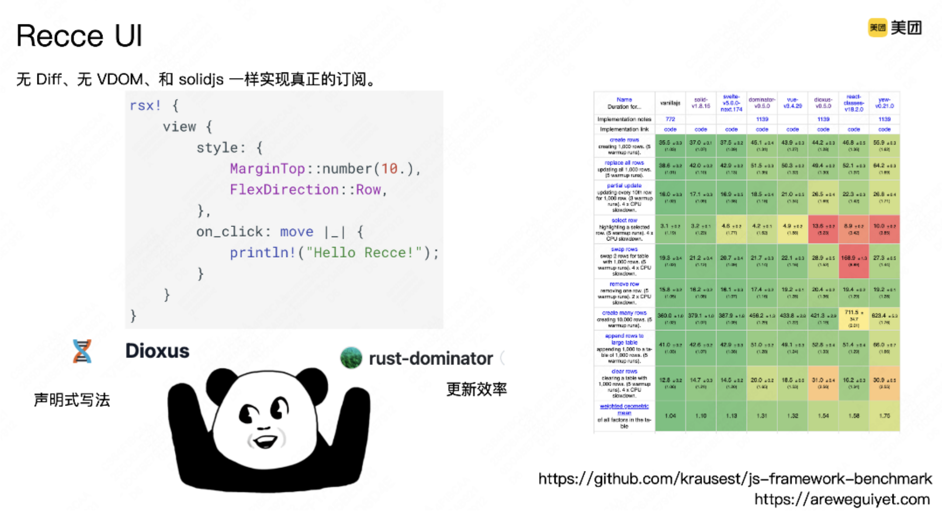
3.2 UI 框架

用 JavaScript/QuickJS 的部分可以复用 Vue 或者 React，这些先不提。用 Rust 则必须要为运行时的上层设计一套 UI 框架，确定应该怎么样编写页面。这项工作的挑战在于，它并不像 JavaScript 生态，有多年的积累可供参考或复用，比如经典的 React 和 Vue。当然好处在于也没有历史包袱，所以必须要结合 Rust 语言的特点才能更好地完成这个任务。

UI 框架大抵需要做到三点：（1）提供声明式 DSL 方便前端研发描述界面；（2）提供组件封装和状态管理能力完成业务逻辑和用户交互的衔接；（3）性能卓越。其中（1）和（2）在 JavaScript 生态中已经都有实现，（3）则未必，否则也不会有几十种 Web UI 框架并存这种局面。这个问题难在如何用 Rust 这门强类型纯静态语言去实现 JavaScript 弱类型动态语言实现的功能，并且要维持 Rust 零开销抽象的优势。

为了解决这个问题，我们参考了 GitHub 上开源的各种框架。一方面参考了 Dioxus 的 DSL 设计和 UI 封装，另一方面也保持了 Rust-Dominator 观察者模式订阅变更的更新效率，我们将这两个优点合并到一起，就得到了 Recce UI，就其特点，没有 Diff，也没有 VDOM，跟 SolidJS 一样实现真正的订阅，我们也尽可能地去保证可以高效地构建 UI。下图中的表格对应的就是一些 Web 项目下的性能对比，这个也并没

有直接对应到具体实践的容器上，因为我们也做了一个类似 Native 渲染的东西，所以这个表格对我们来说具备很好的参考价值，至少可以看到 Dominator 的更新效率还是很不错的。



3.3 渲染层

渲染层的选择则相对简单，如前述归纳实现一个渲染层大致三种方式：复用 Web-View、自建渲染绘制器、调用系统 UI 框架。

1. **复用 WebView**：如果追求高性能，这条路就不通，复用 WebView 意味着渲染指令 / 视图树 要用低效的方式以 WebView 的 JsCore 为跳板再去驱动 WebCore 做渲染。这和高性能就南辕北辙，还不如纯 H5 性能表现好。
2. **自建渲染绘制器**：这条路技术上是行得通的，但目前走不得：第一，从代码量上看 Chromium 内核有数百万行 C++，考虑到跨平台兼容则过千万行代码，这个规模和复杂度是超过美团 App 本身，即使照着写一遍，没有数十名 C++ 专家投入三五年之功是看不到成效的。第二，当世只有 Google 和 Apple 有这个能力做成 WebView。FireFox 的新浏览器计划半途而废，微软则直接放弃转投 Chromium。

3. **调用系统 UI 框架**：类似 RN 的方式，研发成本上是我们能接受的；渲染样式虽然没有 WebView 的 CSS 全，但是 Flex 足够支撑业务需求了，且保持是 W3C 的严格子集；RN 在性能上的问题主要出在通信层上，这个我们可以解决掉；最后也是最重要的是这个选择不是一个单向门，假如我们获得了驾驭 blink[6] 能力，那么就可以很低的成本平滑切换到 blink 上。

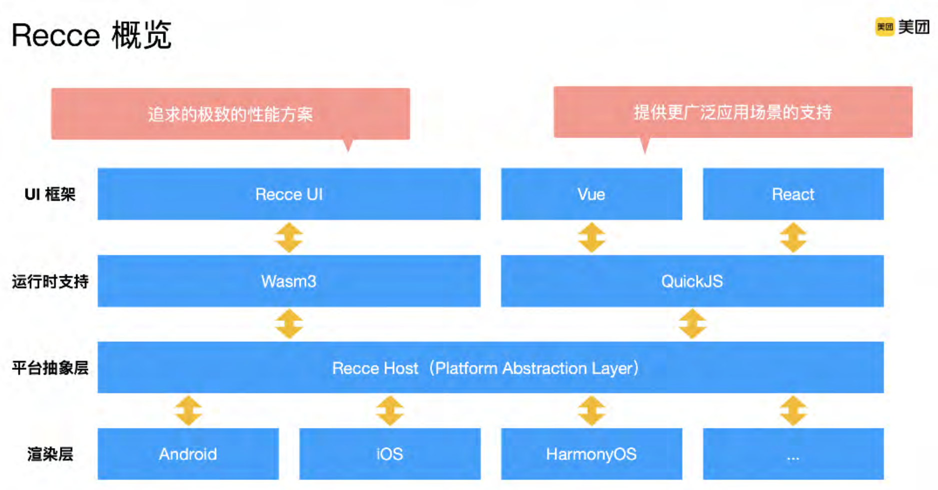
所以渲染层就是复用了 React Native 的 Native 部分，我们决定要站在这个“巨人”的肩膀上开始行动。毕竟 React Native 已经提供了非常优秀的组件封装，同时它也解决了 Android 和 iOS 在渲染层面的差异，因为这些接口基本上都是在系统 UI 下进行封装的，所以我们有理由相信这些接口本身的性能是良好的。

然后，剩下的问题就是设置属性、传递属性等成本，它们在实践的过程中，通常会成为页面渲染的一个瓶颈，事实上 React Native 也正在解决这个问题。基于此，我们决定保留下来 React Native 的 UIManager 的增、删、改等概念，以及 Yoga 布局，还有视图组件的封装，我们将这些保留了下来。后面，我们还会再讨论为什么没有使用 React Native 当前的属性转换的方式，这里不再展开讨论。

3.4 整体架构

最终，Recce 的概览如下图所示。这里重点讲下先前没有提及的 Recce Host 平台抽象层，这一层我们主要做了两件事：第一件，属性设置优化（或者叫渲染通信优化）；第二件，平台抽象。

属性设置优化后文会详细介绍，这里只说平台抽象。我们结合 WebIDL 的设计和 LLVM 的架构理解，在平台抽象层上下都实现了标准接口。类似 llvm 的 MIR 使得编译器前端和编译器后端可以独立迭代和接入，平台抽象层的标准接口设计使得只要遵循渲染指令标准的解释器或者渲染器都可以很容易接入。这就是我们很容易把 QuickJS + Vue/React 支持了的原因。Recce-js 可以使线上的大部分以 JavaScript 为主的前端页面获得更好的性能表现。同样 Recce 的鸿蒙适配的成本也非常低，不需要上千或者几千 pd 那么多。最重要的是未来替换性能更好的解释器或者语言或者 UI 框架都是简单可行的。这一层是 Rust 实现的高效、安全且使得容器整体架构易扩展和易维护。

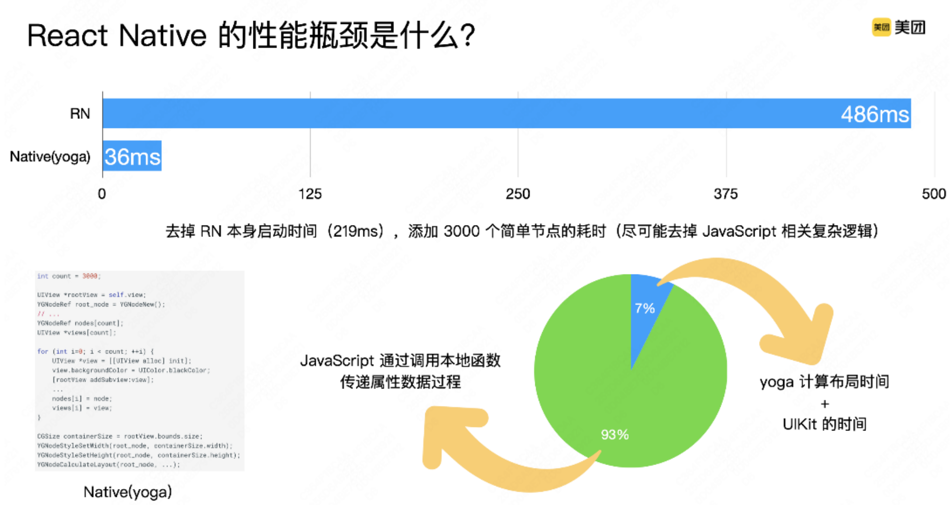


综上选型和搭建工作基本上已经完成了。接下来，我们再对 Recce 上的一些细节问题进行补充。

前面是讲道理，知已不易行更难。下文的“干货”可能才是决定成败的关键细节。

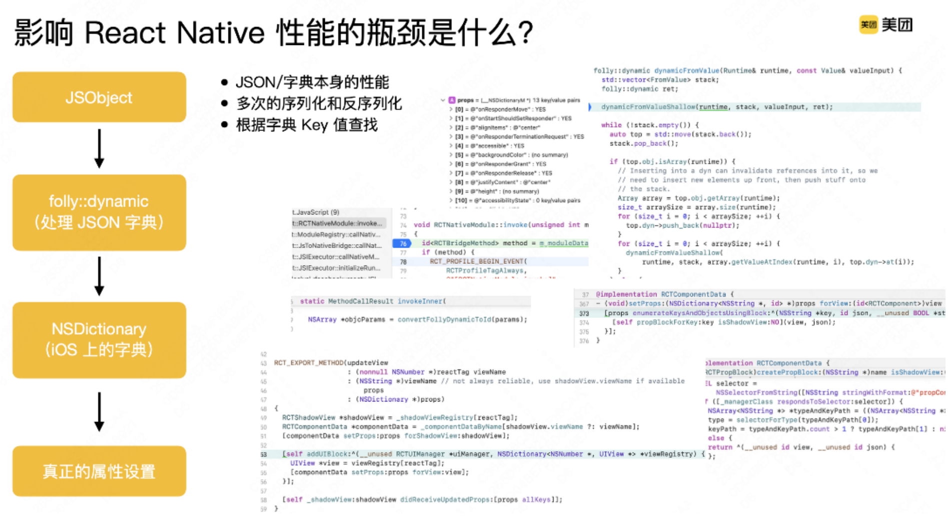
4 Recce 的一些细节问题

首先，就是上一节讨提到的，为什么没有使用 React Native 的属性转换？因为，我们发现属性转换是 React Native 一个性能瓶颈。其实为了评估这个问题，我们做了一个 3000 个节点的页面。当然，这个页面可能跟我们平时常见的页面长得不一样，但是它的渲染和布局成本和业务实际是比较接近的。

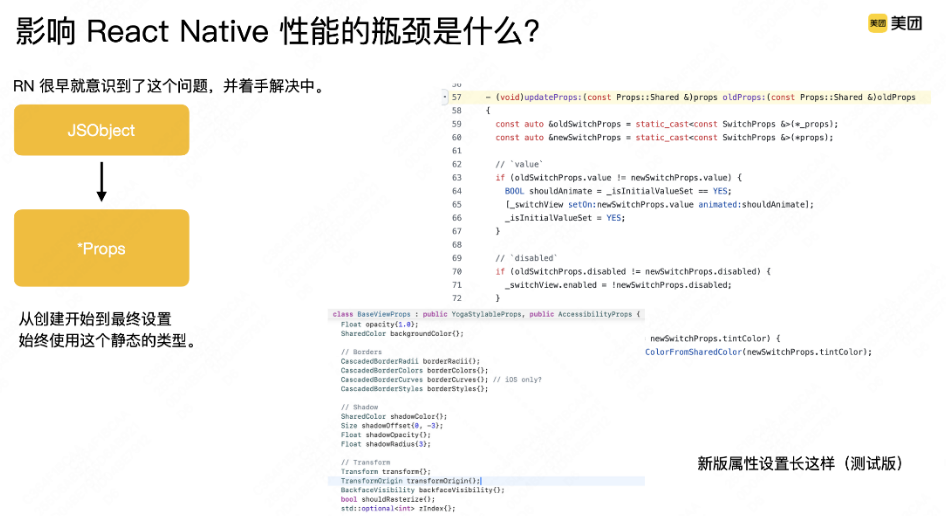


页面的逻辑我们写得尽可能简单，同时去掉了 React Native 的启动时间，然后我们启用了原生代码，但是调用了 Yoga 的布局计算，就这样写了一个页面去做对比。最终发现，二者的耗时差距非常大，因为我们的布局方式是一样的，调用的 UI 也完全是相同的，基于此，我们基本上就可以认定剩下的 93% 的时间都是为了将设置页面的各种各样的数据从 JavaScript 传递到具体的平台，也就是说属性转换会耗费大量的时间。

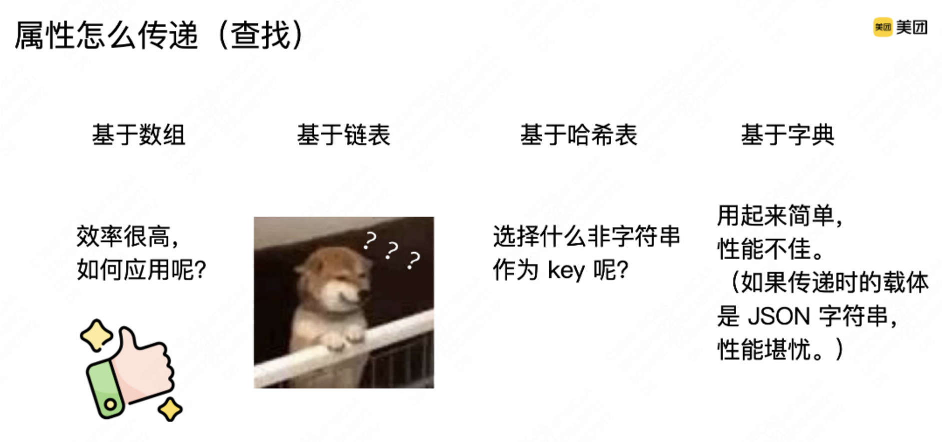
这里可以再深挖一下，为什么属性转换会耗费这么多的时间？我们稍微研究了一下这个问题后发现，主要还是因为 React Native 会有多次序列化和反序列化，这是一个类似于字典的东西，而且除了序列化的时间，还需要考虑构建字典本身需要的时间，还有执行字典的一些查找、设置等操作，最终我们还需要频繁地按照字典里面的 Key 值查找，查找到之后，再设置到具体的属性设置，而以上这些操作都会消耗掉不少的时间及内存。



事实上，React Native 官方也正在尝试解决这个问题，在官方公布的的一个实验性的新的框架中，React Native 直接将一个 JSObject 转换成了一个 C++ 的静态类型（*Props），然后在对应的各个平台中，直接使用了一个转换后的静态类型。如此以来，实际上就只存一次 JSObject 这样字典就可以，从创建开始到最终设置，始终使用这个静态类型，那么剩下的操作都会变得非常高效了。



同时，我们也会去思考应该怎么去解决属性传递以及查找的问题。这里可以简单看一下，我们常用的几种数据结构，都是基于数组、链表、哈希表、字典等等之类。但实际上，我们在这种场景下可以选择的可能就只有字典和数组，而 React Native 最常用的方式就是基于字典去构建各种各样的属性。但是基于字典这种方式并没有非常好的性能，如果传递的载体还是 JSON 字符串的时候，还需要承担 JSON 本身解析的任务。经过考量之后，我们最终决定采用基于索引的数组来构建一个个的属性值。



但是采取这种模式的话，维护每个数组属性上的索引将会变得非常的复杂，我们借助一个属性定义生成各端代码维护这个索引，我们在运行前约定好每个属性的索引值，当然，我们会放弃一定的兼容性。其中一个放弃的例子就是，当我们约定好一个属性的索引值之后，之后就不能再修改这个属性的索引值本身，否则就会遇到属性设置可能会发生错乱的问题。

基于数组

美团

我们借助一个属性定义生成各端代码维护这个索引：

```
// 定义属性
flex-wrap: enum('nowrap', 'wrap', 'wrap-reverse');
margin: ygvalue;
background-color: color;
```

```
Height::number(100.0);
```

```
static void Height_accept(rec_bin_reader_t *binReader, ...) {
  float height = bin_reader_get_f32(binReader);
  // ...
}

const PropAccept REC_PROPERTIES[206] = {
//  ...
  Height_accept,
//  ...
}
// 更新一个属性代码大概的样子
void setProp(rec_bin_reader_t * bin_reader) {
    int32_t index = bin_reader_get_i32(bin_reader);
    REC_PROPERTIES[index](bin_reader);
}
```

我们在运行前约定好每个属性的索引值。
* 当然会放弃一定兼容性

类似的，我们也可以把组件注册的标识从字符串修改为索引，由于这个属性和组件不太一样，就无法知道客户端会提供哪些原生的组件，所以至少要在运行时去使用字符串获取一次组件信息。在获取之后，就可以使用获取的这个索引，从而保持一个比较快的匹配效率。

基于（动态绑定）数组

美团

应用在（第三方）组件注册：运行时绑定一个索引。

```
// native
components = ["View" /*0*/, "Image" /*1*/];
// 注册组件
custom_view_id = components.register("CustomView" /*2*/);
// 这样我们就关联了 CustomView 为 2

// wasm 业务的代码：
custom_view_id = null;
get_custom_view_id() {
  if (!custom_view_id) {
    custom_view_id = get_id("CustomView");
  }
  return custom_view_id
}
```

* 真的不能在运行前获取吗？

* 这可能和 Java 的 String.intern() 有点像。

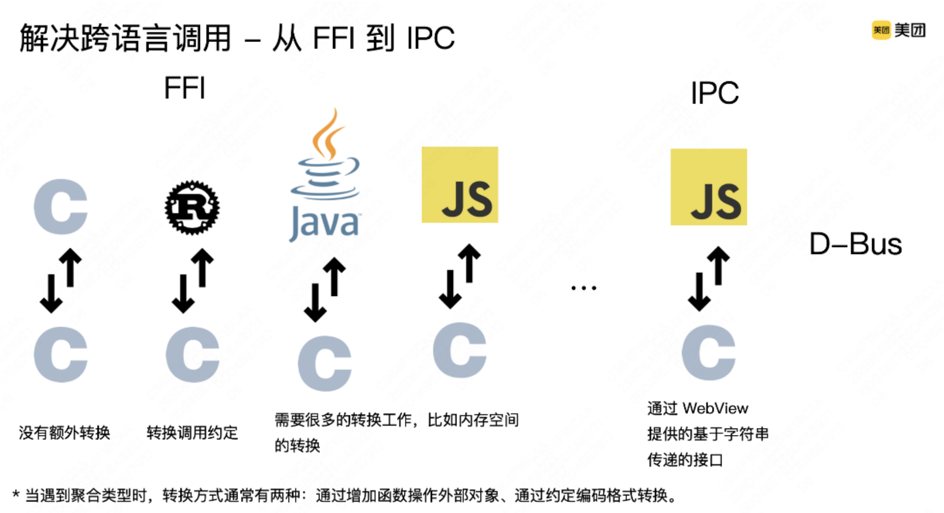
最近，我们还做了一个富文本的标签，这个标签跟前两者相比就变得更不一样了。这

里实际上输入的是一段 HTML 的字符串，所以输入的内容非常自由，我们没有办法像前两者一样使用一些静态的计算方式。但即便是富文本，仍然可以有一些已知的内容，比如像文本的样式、字符串等等，这些内容是可以提前知道的。而在这个层面，我们是可以做一些事情的，实际上可以基于所有输入的计算，得到一个完美的哈希函数，然后确保所有的输入不会发生碰撞。进而，在查找 Key 的时候就会变得快很多。

以上，就是我们在遇到属性传递时解决的各种各样的问题。而接下来，我们还需要解决一个问题——跨语言的调用问题。当然，在讨论这个问题之前，我们先简单地将这些调用划分成了四类：

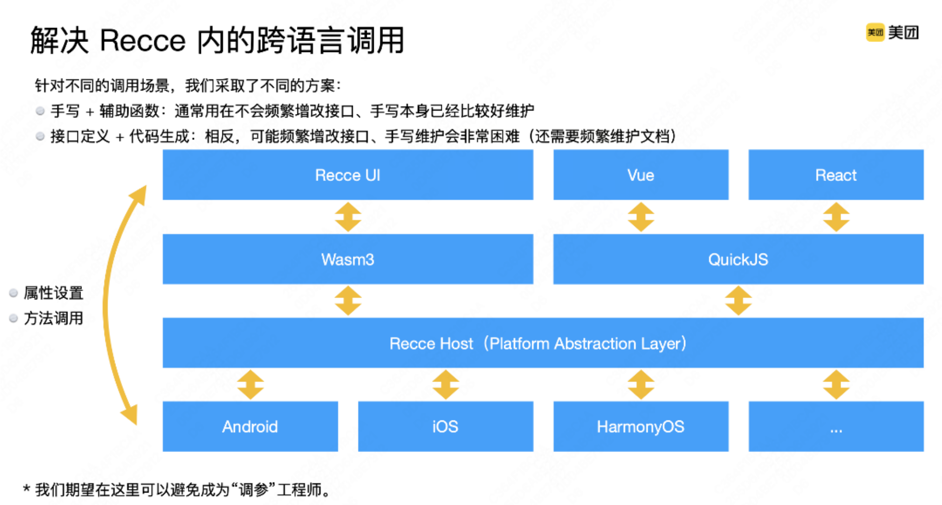
- 第一类是 C 与 C 之间的调用，其实 C 语言本身并没有什么转换，这里只是把它作为一个最特殊的场景，进行归类处理。
- 第二类是 Rust 去调用 C，这两种语言虽然也是直接在原生上去运行的，但是它们之间会增加了一些调用约定之类的转换。
- 第三类是 Java 和 JavaScript 去调用 C，这就需要借助解释器提供的接口去进行调用，其中也会涉及更多的转换的工作，比如说两个内存空间之间的拷贝。

- 第四类，我们认为它们其实更接近这个 IPC 的调用，比如像通过 WebView 提供的基于字符串传递的接口。



在这种场景下，我们会有更多的转换工作以及一些编码约定的设置，所以我们必须要找到一种基于字符串编码方式去传递各种复杂的数据。而具体应用到 Recce 内部的时候，其实并不存在类似于 IPC 场景的调用，所以只需要解决每一层之间语言的调用。但实际上，我们仍然面对着非常复杂的调用这个事实。

参考下图，可以看到每一层实际上都涉及到一个具体的跨语言调用。同时，还有刚刚提到的性属性设置，它则会跨过中间的所有层级，直接设置到具体的平台。最后要需要强调一下，涉及到具体的属性以及类似的一些方法调用，比如说打开相机，也会跨过中间层直接去做调用，而这个调用跟属性设置又存在很多不一样的地方。



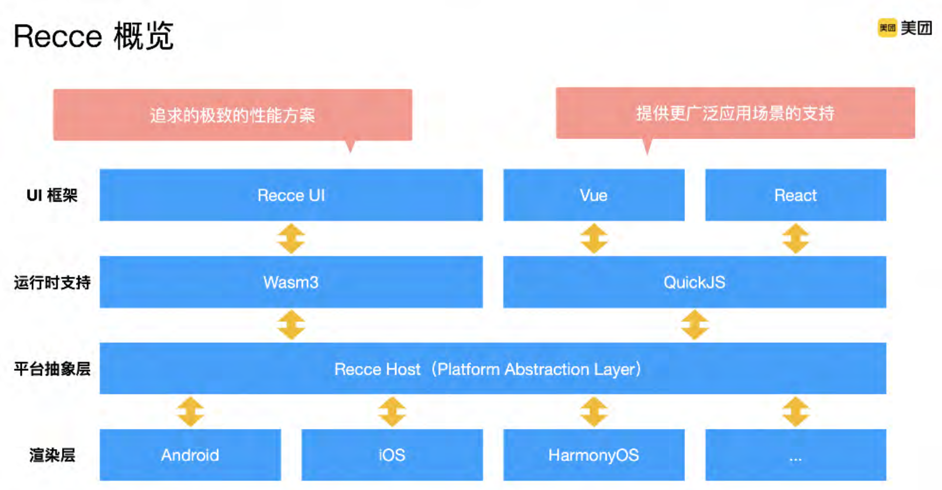
针对不同的调用场景，我们也采取了两种不同的解决方案：

- **手写 + 辅助函数**：通常用在不会频繁增改接口、手写本身已经比较好维护。
- **接口定义 + 代码生成**：可能频繁增改接口、手写维护会非常困难（还需要频繁维护文档）。

具体来讲，我们分成了以下 4 个场景：

1. **属性设置**：把定义的属性 props_gen 生成 Recce UI+ Vue + React + Android + iOS+HarmonyOs 等各种属性的操作代码。
2. **Rust（Wasm3）**：扩展 wit-bindgen 支持 Wasm3。
3. **QuickJS**：借助 Rust 宏封装一些本地函数（UI 操作接口）+ 自定义的二进制读写实现。
4. **业务方法调用 & 平台抽象层与 Platform 交互**：使用 mako 完成接口调用和文档生成。

5 总结和展望



最终我们获得了一个如上图的高性能、安全的动态化容器，可以以 Wasm 的方式支持原生级别的性能，也可以将 JavaScript 的前端工程的性能提升一截。

从某个角度看，像是我们把 RN 用 Rust 重写了，添加了 Wasm 解释器的支持。但用熟悉 WebView 架构的视角看，也可以看作是一个 WebEngine Lite，只是试图绘制暂时用的系统 UI。

文章最后做一下回望和展望。

回望：我们所做的所有架构和优化工作都可以概括为，区分本质复杂度和偶然复杂度，恰当的回应本质复杂度，降低偶然复杂度。

动态化容器的本质复杂度是什么？最主要的一条脉络是，渲染管线，以前端研发的编码逻辑和数据为输入，在管线中变为组件树 -> 虚拟文档树 -> 文档树 -> 视图树 / 样式树 -> 图层树 -> 呈现树，最终绘制到屏幕上为用户所眼见视图为输出。至于用什么 DSL、编程语言、解释器、编码方式等等其实是偶然复杂度。

Recce 的选型和搭建过程，实际上是围绕渲染管线进行优化，比如精简流程去掉了

VDOM 等环节，比如简化运行时、选用执行效率更高的解释器和编码方式等，再比如削峰填谷、消除瓶颈以提升渲染管线整体效率等。而这些工作都是为了降低容器本身的计算开销，因为动态化容器相对业务而言也是**偶然复杂度**。将终端有限的软硬件计算资源更多的留给**业务本质复杂度**是容器迭代的正确方向。

展望：目前 Recce 还在逐步完善和落地推广阶段，可以做的事情有很多：

- **改善开发体验**：比如引入 LLM 来降低 Rust 的学习门槛和开发成本，比如完善调试 & 脚手架工具链等等。
- **进一步优化性能**：比如以 Rust 原生方式运行以获得超过 Android 原生的性能表现；比如利用 Wasm 解释器线性内存特性，可以在 CI 上完成大部分的预计算，进一步提升加载性能表现；
- **自研渲染层**：这样一是能进一步提高性能，二也能降低多端维护成本，三是把样式能力集对齐到 WebView 可以实现和 H5、小程序的同构。当然这样其实就是 一个完整的 WebEngine Lite 了。

6 Q & A

Q：Recce 的性能如何？可以把这个问题更具象化一些吗？

A：目前从我们已经实践的范围中，在我们业务场景中能够找到最低端的 POS 设备上面，Recce UI 是可以和 Flutter 的性能表现媲美，而且我们是在动态解释运行，而 Flutter 是原生运行。也就是实现了原生级别的性能表现。

Q：以后 JS 没用了吗？

A：JS 有用的，但这个问题分两个层面回答。第一层，Recce 是支持这个 JS 运行时和相应生态的。比较极端的场景，比如说 POS 机，下沉市场的低端机，或者说对于性能要求很高的，比如说像 App 的首页冷启动，或者说像这个支付收银台等等这种场景下，优化性能的收益很大，那么就可以考虑用 Recce-rs 这个方案。如果不是这么极端的一般场景，我们用这个 Recce-js 的方案也能获得一个低成本，获得一个性

能优化。性能优化是没有止境的，根据业务场景的需要去选择，Recce 提供了更多且更好的选择。

第二层，移动端开发从一开始就有原生开发和 H5/JS 开发（PhoneGap 发布于 2009 年），但 JS 成为主流，始于 2015 年，Google 把 V8 适配到了 Android 上。这里就不是 JS 行不行，而是 V8 很行。 终端硬件的计算资源始终是有限的，V8 通过解释器层面 JIT/AOT 编译技术 大幅提升了 JS 的运行效率，使得 JS 铺开成为可能。但近几年 JS 引擎不再有大幅度迭代，手机硬件算力的升级速度也明显放缓，这就导致了终端上 软硬件计算容量的提升速度跟不上业务复杂度的提升速度。这才是当下 JS 技术栈面临的问题，也是为什么有如此多 JS 框架在卷性能的原因。

当然大家不用特别悲观，JS 和 Web 始终是互联网最重要的基础设施，事物是呈现螺旋发展趋势的（当然有人负责发展，有人负责螺旋），随着周期演变，新的软硬件技术升级又会推动 JavaScript 往前发展。比如如果一个使用 TypeScript/JavaScript 的高性能解释器出来，那么能够使现在的 JavaScript 工程性能大幅度提升，毕竟前端线上代码资产 90% 是以 JavaScript 形式存在的。

Q：为什么叫 Recce？

A：Recce/'rɛkiː/ 名字取自海豹部队的一把枪。选择这个名字的初衷是因为在 Recce-rs 选型的时候，目标是接近原生的性能表现、还要动态化，那么肯定要舍弃一部分开发体验的，切换 Rust 语言上会对研发同学增加学习成本、抬高门槛。因为预判到这一点，所以我们希望使用 Recce 的同学始终记得是经过了更多的训练的精英，要克服各种困难去完成高价值的任务。凡事有利必有弊，有用比有费，对于需要使用 Recce-rs 优化的场景 学习 Rust 就不是最难的技术问题。我们在为高价值场景提供更好的选择的同时，也将部分优化能力通过 Recce-js 反哺到一般业务场景上。不同的场景不同的成本结构，没有最好，只有合适的方案。

Q：Recce 为什么要追求原生级别的性能?

A：主要有三点原因。

一、前端提升性能体验可以提升用户体验和业务获客效率

前端部署和面客是重叠的过程，前端部署成功率影响业务面客。部署成功率低意味着，客户遇到白屏或不愿意等待最终放弃。提升性能可以提升前端部署成功率，进而可以提升用户体验和业务获客效率。

部署：代码离开开发环境，到用户终端设备运行前的这段环节为部署。

二、追求原生级别性能的动态化容器是公司业务的发展需要

美团业务的首要特点是低毛利，低毛利意味着业务发展需要做大规模，提升 UE，提升复购。

- 做大规模，意味着要争取广大下沉市场。那么在性能体验上就需要至少 Meet 友商，在下沉市场表现最好的应用（微信、拼多多）的主要功能都是原生实现的。争取下沉市场扩大规模在前端的命题意味着提供原生级别的性能体验，在下沉市场这可能不再是体验好坏而是能否使用的问题。
- 提升 UE，意味着必须要考虑研发人效，前端提研发人效就必须考虑跨平台同构，因此需要容器化，将多端开发降为一端开发。
- 提升复购，复购意味着多业态混合经营，那么在包大小约束下就必须要让大部分业务动态化。

综上，公司业务发展的需要决定了美团需要原生级别性能的动态化容器。

三、部署是前端工程领域的核心问题，容器是部署问题的主要答案

把时间拨回到 2010 年，彼时移动端开发方兴未艾，主要的技术栈只有 Native 和 H5（PhoneGap 为代表），其中 Native 性能表现好但不跨端、H5 迭代效率高但性能体验差。随着时间推移，Native 方向为了解决跨端研发成本和部署效率，发展出了 ReactNative、Flutter 等等方案，H5 方向为了改善性能，发展出了离线化、SSR/ESR、解释器优化等等。

可以说两个方向是相向而行，其实这是一个问题的两个面。究其根本在于前端代码运

行在不受我们控制且计算资源有限的用户设备上，不同于后端的主要问题是高并发，前端的主要问题是怎样把代码低成本跨过物权边界送到大规模且不同的用户的设备上并高效运行起来。

回顾过去，性能表现好原生要做动态化降低部署成本，部署成本低的 H5 要优化性能提升用户体验。换句话说，前端工程领域的核心问题是部署成本和用户体验的平衡。比如包大小就是原生开发部署成本高而非常困扰客户端开发的一个典型例子。

注释

[1] 逻辑解释器：在 iOS 中 Apple 不允许 JIT，只可以解释执行。
[2] WasmEdge：亚马逊用做云原生的 Wasm 解释器。
[3] C#：黑神话悟空就是用魔改 USharp 跑的游戏逻辑。
[4] 不支持 JIT ：Apple 不允许在 iOS 上 JIT。
[5] C、C++、Rust、Go 这四种
[6] blink：chromium 的布局渲染器。

KDD 2024 | 美团技术团队精选论文解读

ACM SIGKDD（Knowledge Discovery and Data Mining，简称 KDD）是数据挖掘领域的国际顶级会议。KDD Cup 比赛是由 SIGKDD 主办的数据挖掘研究领域的国际顶级赛事，从 1997 年开始，每年举办一次，是目前数据挖掘领域最有影响力的赛事。

本文精选了美团技术团队被 KDD 2024 收录的 5 篇长文进行解读，覆盖了用户意图感知、机器学习 & 运筹优化、在线控制实验、联合广告模型、实时调度决策等多个技术领域。这些论文都是美团与高校、科研机构合作的成果。希望能给从事相关研究工作的同学带来一些帮助或启发。

此外，大众点评技术部 / 内容智能算法团队组建的 BlackPearl 团队参加了今年 KDD Cup-OAG Bench（Open Academic Graph Benchmark）赛道的 WhoIsWho-IND、PST、AQA 三道赛题，基于大模型技术提出自反馈增强、嫁接学习等方案，三个赛题均以较大优势取得冠军！

01 Unified Dual-Intent Translation for Joint Modeling of Search and Recommendation

论 文 作 者：Yuting Zhang*（ICT, CAS），Yiqing Wu*（ICT, CAS），Ruidong Han（Meituan），Ying Sun（Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology），Yongchun Zhu（ICT, CAS），Xiang Li（Meituan），Wei Lin（Meituan）

备注：ICT, CAS 全称为 Institute of Computing Technology,Chinese Academy of Sciences

论文下载：PDF

论文类型：Long Paper

论文简介：推荐系统旨在帮助用户在众多候选商品中发现他们所喜爱的商品，并已服务于各种在线平台的数十亿用户。从直观上看，用户与商品的交互高度受到他们稳定的固有意图（例如，始终偏好高质量的商品）和变化的需求意图（例如，夏天想要一件 T 恤，冬天想要一件羽绒服）的驱动。然而，这两种意图在推荐场景中都是隐式表达的，这给准确感知用户意图带来了挑战。幸运的是，在通常与推荐系统共存于同一在线平台的搜索场景中，用户通过查询词显式表达了他们的需求意图。直观上，在这两种场景中，同一用户的交互可能受到相似的需求意图的影响，并且其固有意图是稳定的。

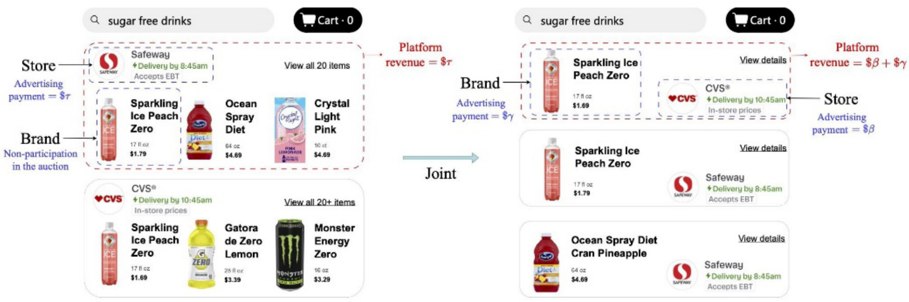
因此，利用这两个场景中的交互数据来相互增强或补充双重意图，并进行联合意图感知建模是可行的。然而，搜索和推荐的联合意图感知建模需要解决以下两个问题：（1）准确建模推荐中用户隐式的需求意图；（2）建模双重意图与交互商品之间的关系。为了解决上述问题，我们提出了基于双重意图转换的搜索推荐联合模型 (UDITSR)。为了准确模拟推荐中用户的需求意图，我们利用搜索数据中的真实查询作为监督信息来指导其生成。为了显式模拟 < 固有意图，需求意图，交互商品 > 三元组之间的关系，我们提出了一个双重意图转换传播机制，实现了在同一语义空间中学习三元组元素间的可解释关系。大量实验表明，UDITSR 在搜索和推荐任务中均优于现有的

SOTA 基线。此外，我们在美团外卖平台上进行了为期一个月的线上实验，平均提升了 1.46% 的 GMV 和 0.77% 的 CTR 指标。

02 Joint Auction in the Online Advertising Market

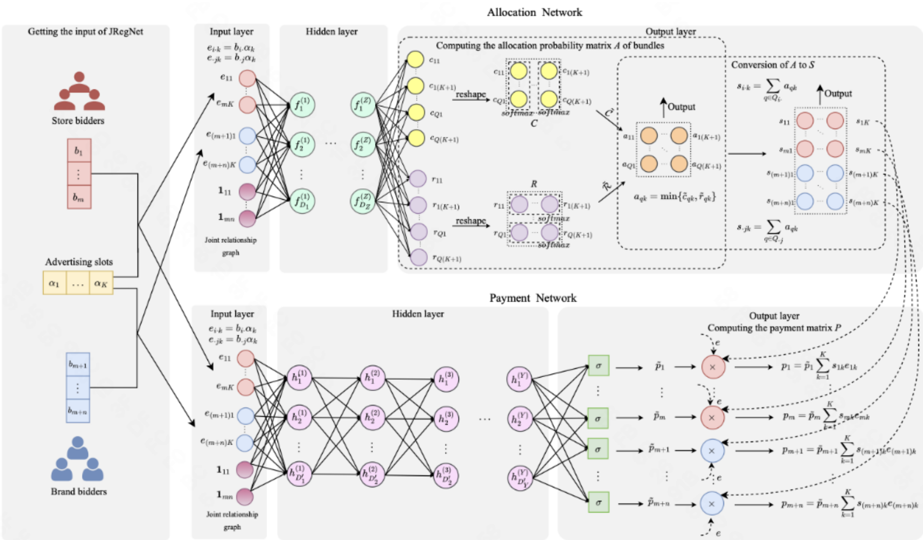
论文作者：Zhen Zhang（RUC, Gaoling School of Artificial Intelligence）, Weian Li（SDU, School of Software）, Yahui Lei, Bingzhe Wang（RUC, Gaoling School of Artificial Intelligence）, Zhicheng Zhang（RUC, Gaoling School of Artificial Intelligence）, Qi Qi（RUC, Gaoling School of Artificial Intelligence）, Qiang Liu（Meituan）, Xingxing Wang（Meituan）

论文类型：Long Paper



(a) Traditional advertising (b) Joint advertising

论文简介：在线广告是电子商务平台的主要收入来源。在当前的广告模式中，定向目标是愿意支付额外费用以提升其店铺位置的在线店铺主。另一方面，品牌供应商也希望在店铺中宣传其产品以提升品牌销量。然而，目前使用的广告模式无法同时满足店铺和品牌供应商的需求。



为了解决这个问题，我们创新性地提出了一个名为「联合拍卖」的联合广告模型，允许品牌供应商和店铺共同竞标广告位，满足双方的需求。然而，传统的广告拍卖机制不适合这一新颖场景。

在本论文中，我们提出了一种名为 JRegNet 的神经网络架构，用于最优联合拍卖设计，以生成满足近似 DSIC 和 IR 的近似最优机制。最后，我们在模拟数据和真实数据上进行了多项实验，证明与已知基线相比，我们提出的联合拍卖模型取得了较好的成果。

03 STATE: A Robust ATE Estimator of Heavy-Tailed Metrics for Variance Reduction in Online Controlled Experiments

论文作者：Hao Zhou*（Meituan），Kun Sun*（Meituan），Shaoming Li（Meituan），Yangfeng Fan（Meituan），Guibin Jiang（Meituan），Jiaqi Zheng（Nanjing University），Tao Li（Meituan）

论文类型：Long Paper

论文下载：PDF

论文简介：在线控制实验是评估营销活动效果的重要工具。其中，方差缩减方法可以有效地提高实验灵敏度，从而使用更少的样本或更短的实验周期得到置信的实验结论。一些典型的方法如 CUPED/CUPAC/MLRATE 等已逐步部署到各大公司的实验平台来提高实验的统计功效。然而，这些方法通常是建立在业务指标服从高斯分布的假设之上，无法正确地表征重尾分布的业务指标，从而效果提升有限。

在本论文中，我们将 t 分布与机器学习工具相结合，来表征重尾指标，通过变分 EM 优化方法，推断得到一个鲁棒的 ATE 估计器，我们称之为 STATE。它有效地缓解了离群点的干扰并显著降低了 ATE 估计的方差。此外，我们通过利用无偏的线性变换，将 STATE 方法从计数度量（Count Metric）扩展到比率度量（Ratio Metric）。我们在合成数据集和美团外卖的业务数据上都证明了 STATE 方法的有效性。与最先进的估计器（CUPAC/MLRATE）相比，STATE 降低了 ATE 估计量 50% 左右的方差，这表明它只需一半的观测值或一半的实验时间即可达到相同的统计功效。

04 Decision Focused Causal Learning for Direct Counterfactual Marketing Optimization

论 文 作 者：Hao Zhou*（Meituan），Rongxiao Huang*（Nanjing University），Shaoming Li（Meituan），Guibin Jiang（Meituan），Jiaqi Zheng（Nanjing University），Bing Cheng（Meituan），Wei Lin（Meituan）

论文类型：Long Paper

论文下载：PDF

论文简介：营销优化对于在线互联网平台的用户增长起着重要作用。现有的研究通常将这个问题表述为预算分配问题，并利用两个完全解耦的阶段，即机器学习（ML）与运筹优化（OR）来解决。然而，ML 阶段的学习目标没有考虑下游 OR 阶段的优化任务，这导致 ML 阶段模型的预测精度可能与决策质量不呈正相关。从而，降低模型预

估误差，不一定提升优化任务的决策效果。

本论文提出了一种基于决策的因果学习方法（DFCL），将 ML 与 OR 两个阶段集成到一个端到端的因果学习框架中，使得机器学习模型能以下游 OR 阶段的优化目标作为损失函数，从而保证 ML 阶段与 OR 阶段优化方向的一致性。其次，DFCL 克服了营销场景中的预算不确定性，反事实推断问题以及计算效率问题等多个技术挑战，使得 DFCL 可以实现针对大规模在线用户营销场景的直接反事实优化。离线实验和在线 A/B 测试都证明了 DFCL 相对于传统因果推断方法的有效性。目前，DFCL 已在美团的多个营销场景部署并应用。

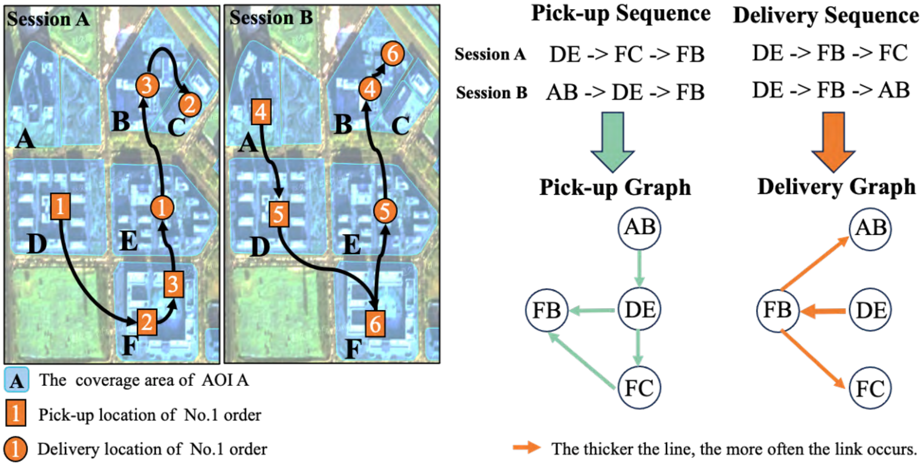
05 Harvesting Efficient On-Demand Order Pooling from Skilled Couriers: Enhancing Graph Representation Learning for Refining Real-time Many-to-One Assignments

论 文 作 者：Yile Liang（Meituan），Jiuxia Zhao（Meituan），Donghui Li（Meitu-an），Jie Feng（Tsinghua University），Chen Zhang（Tsinghua University），Xuetao Ding（Meituan），Jinghua Hao（Meituan），Renqing He（Meituan）

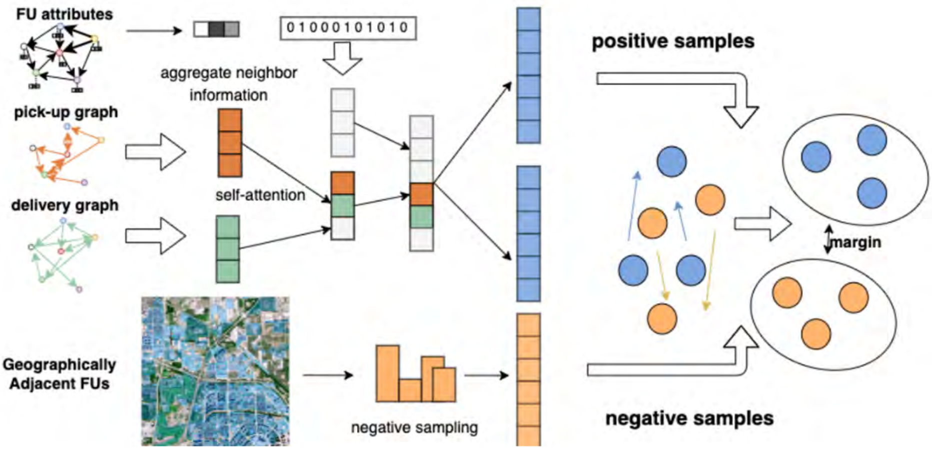
论文类型：Long Paper

论文下载：PDF

论文简介：即时配送平台通过周期性批量调度的方式，将大量订单分配给配送员进行履约。在实时调度决策过程中，将多个订单合并指派，构造高聚合、高顺路的配送路径是提升配送员收入、优化配送员工作体验的关键。由于订单分配问题的复杂性和实时性，传统方法难以在有限时间内快速生成高质量的订单分配方案。此外，线下环境中频繁出现的不确定性因素（如门禁通行管控等），也增加了平台在感知能力和分配决策上的难度。然而，经验丰富的配送员由于对环境更加熟悉，具有更强规划能力，可有效帮助系统提高感知能力，并为实时调度决策提供“灵感”。



基于此，本论文利用有经验的配送员的实际轨迹数据，构建了效率感知网络，以挖掘嵌入在配送员轨迹中的订单合并潜力。具体而言，通过丰富的时空行为数据，将订单表示为低维向量，并通过轻量地相似性计算，对庞大的解空间进行实时剪枝，有效降低订单分配问题求解复杂度，从而快速识别高质量的订单分配方案。



同时，为了提升异构图神经网络在即时配送场景的学习效果，对传统算法的 Loss、负采样等模块进行了针对性优化。该网络已部署在美团的配送系统中，线上结果显示能在实时性能要求下提升订单分配质量，优化了配送员的工作体验。

美团科研合作

美团科研合作致力于搭建美团技术团队与高校、科研机构、智库的合作桥梁和平台，依托美团丰富的业务场景、数据资源和真实的产业问题，开放创新，汇聚向上的力量，围绕机器人、人工智能、大数据、物联网、无人驾驶、运筹优化等领域，共同探索前沿科技和产业焦点宏观问题，促进产学研合作交流和成果转化，推动优秀人才培养。面向未来，我们期待能与更多高校和科研院所的老师和同学们进行合作。欢迎老师和同学们发送邮件至：meituan.oi@meituan.com。

KDD 2024 | OAG-Challenge Cup 赛道三项冠军技术方案解读

大众点评技术部 / 搜索与内容智能团队组成的 BlackPearl 队伍，参加了 2024 年 KDD 2024 OAG-Challenge Cup 赛道的 WhoIsWho-IND、PST、AQA 三道赛题，以较大优势包揽了该赛道全部赛题的冠军，本文对这三个赛道的夺冠方案分别进行了解读，希望对大家有所帮助或启发。

KDD 2024 OAG-Challenge Cup 与学术数据挖掘研究相关，学术数据挖掘的最终目标是加深我们对科学的发展、本质和趋势的理解。它提供了发掘巨大的科学、技术和教育价值的潜力。例如，从学术数据中进行深度挖掘可以协助政府制定科学政策，支持公司人才发现，并帮助研究人员更有效地获取新知识。学术数据挖掘包含很多以学术实体为中心的应用，比如论文检索、专家发现和期刊推荐等。然而，学术知识图谱挖掘相关的数据基准的缺乏严重限制了该领域的发展。KDD 2024 OAG-Challenge Cup 主要包括三道经典学术知识图谱挖掘问题，包括：

- **论文同名消歧（WhoIsWho-IND）**：目前，在线出版物数量的迅速增加使得同名消歧问题变得更加复杂。 此外，现有消歧系统的不准确导致了错误的作者排名和奖项作弊的情况。 本任务在给定每位作者的个人资料，包括作者姓名和发表的论文情况下，需要开发一个模型来检测论文中错误分配给该作者的论文。此外，数据集还提供了所有涉及论文的详细属性，包括标题、摘要、作者、关

键词、地点和发表年份。

- **论文源头追溯（PST）**：随着科技的飞速发展，论文数量呈爆炸式增长。全球每年发表数百万篇论文，且数量持续攀升。根据 Scopus 数据库，截至 2021 年，全球已发表 2.2 亿篇学术期刊论文，涵盖自然科学、社会科学和人文科学等各领域。对于研究者来说，从众多文献中把握技术发展的脉络变得愈加困难。PST 任务要求在给定一篇论文 p 的全文的情况下，从这篇论文中找出 ref-source。ref-source 即最重要的参考文献（叫做“源头论文”），一般是指对本篇论文启发性最大的文献。每篇论文可以有一篇或多篇 ref-source，也有可能没有 ref-source。对于论文的每一篇参考文献，论文源头溯源都要给出一个范围在 [0, 1] 的重要性分数。
- **学术论文问答（AQA）**：在这个技术蓬勃发展，信息迅速更新的时代，为研究人员和大众提供多领域的高质量前沿学术知识已成为当务之急。AQA 给定专业问题和一组候选论文，目标是检索到最相关的论文来回答这些问题。

BlackPearl 团队在本次竞赛中，创新性的采用大模型来解决学术挖掘领域中的论文同名消歧、论文源头追溯、学术论文检索三个经典难题，显著优于特征工程、GNN、BERT 等传统方案。本文将对 BlackPearl 团队在此次三赛道的夺冠方案进行分别解读，更多完整细节和代码请移步 GitHub。

论文同名消歧（WhoIsWho-IND）

在论文同名消歧任务中，由于每条样本下的论文数量过多以及每篇论文包含的信息量较大，导致输入文本超长，现有的文本聚类方法不能充分的利用这些信息。故赛题的难点在于如何构造大模型输入形式、如何在资源有限的情况下利用更多信息以及充分发挥大模型的潜力。

针对以上难点，BlackPearl 将原始的聚类任务转化为比较任务，构建了一套基于自反馈增强的迭代式大模型文本聚类链路。此外，BlackPearl 还使用了 Train-Time Difficulty Increase（TTDI）和 Test-Time Augmentation（TTA）[1] 等技术，进一

步提升效果。算法框架如下图 1 所示：

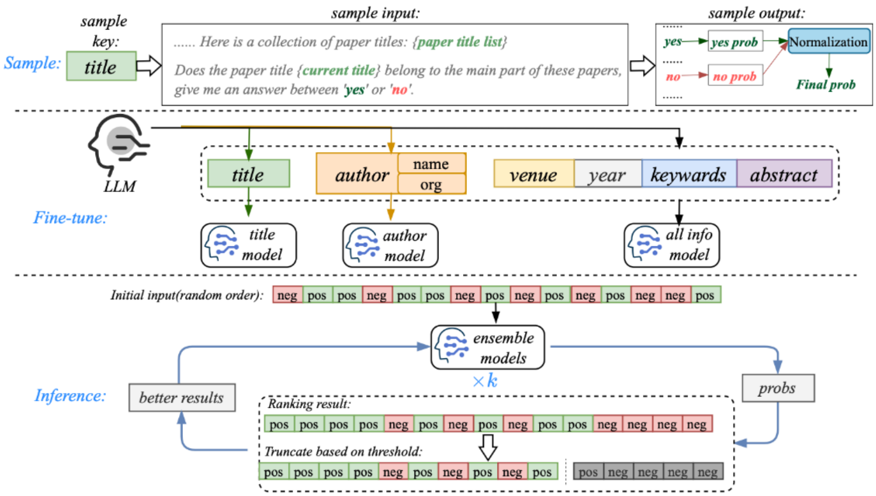


图 1：WhoIsWho-IND 解决方案。图的最上面部分是使用标题作为信息源的一个示例。在微调阶段，我们使用多源数据对多个模型独立微调。集成模型是指从多个微调模型中推断出的结果的加权平均。K 表示迭代自精炼轮数

在将任务转化为比较任务后，从经验上看，输入的参考文献越多，模型接收的输入中的参照信息也更丰富，对模型判断当前 paper 是否正确也越有帮助，我们通过实验也验证了这一点。然而，在固定最大输入长度的限制下，在输入中拼接的参考文献越多，每篇论文所包含的信息就越少。

为了尽可能利用更多信息以及尽量减少训练所用资源，我们使用了一种拆分策略、在微调阶段我们使用多源数据对多个模型独立微调，以确保每个模型能够专注于特定的信息源，我们使用 deepspeed[2] 的 zero1 来在训练时长和显存占用方面取得平衡，微调方法用 LoRA[3] 和 QLoRA[4]。通过实验，我们确定了标题和作者是最关键的两个信息，其他信息则不在单独训练模型以避免资源浪费。为了利用到其他信息，我们对所有可用信息源进行微调得到一个综合模型，用于模型结果集成，进行信息互补。

由于我们使用比较任务来确定当前论文是否属于主要（正确）类别，我们自然会认为，参考文献中正确论文的比例越高，模型对当前论文的正确性判断就越有信心。基于

此，我们提出了迭代自精炼 (IRF) 方法，该方法不需要额外的模型训练，通过不断精炼参考论文中正确论文的比例来获得更好的结果。通过将大模型预测的若干 paper 正确概率进行排序 + 阈值截断，使得大模型比较任务下一轮输入的参考 paper 中正样本浓度提升，从而使得模型输出结果时更自信，最终显著提高了识别正确论文的概率。由于第一轮迭代时我们还没有拿到预测概率，因此初始输入中的参考文献是随机采样的。

为了使模型在推理阶段应对更具挑战的样本，我们在训练阶段增加任务难度，以防止任务变得简单。例如，减少最大训练长度、适当增加训练输入中错误论文的比例，从而提升模型的鲁棒性。

在比较任务中，模型输出的概率不应受参考文献输入顺序的影响。针对这一问题，我们充分利用 TTA，在将每个样本中的参考文献输入模型之前，对其顺序进行 shuffle，并对多个结果进行平均，以获得更稳健的结果。

我们做了大量实验验证了我们方法的有效性，对比实验和消融实验如下所示：

Methods	Dev	Test
GCN[7]	0.586	–
GCCAD[2]	0.634	–
LGB[6] Ensemble	–	0.799
LGB + ChatGLM	–	0.813
Title model	0.757	0.767
Author model	0.715	–
All info model	0.758	–
Ours	**0.794**	**0.834**

Methods	Dev	Test
Title model	0.757	0.767
+TTA	0.761	0.772
Author model	0.715	–
+TTDI	0.727	0.788
Ensemble model	0.772	0.808
+ISR, k = 1	0.786	0.827
+ISR, k = 2	0.791	0.831
+ISR, k = 3	**0.794**	**0.834**

表 1、表 2　整体实验效果以及消融研究

总的来说，我们的核心上分点如下：

- Task Format Conversion：将聚类任务转化为比较任务，在输入中给出一些参考论文，并确定当前论文是否属于主要类。
- Train-Time Difficulty Increase (TTDI)：训练阶段增加任务难度，让模型跳

出“舒适区”，使其能够在推理过程中更好地处理具有挑战性的示例。

- Test-Time Augmentation (TTA)：在测试时对输入数据施加多种变换（此题为打乱参考文献的输入顺序），并对这些变换后的数据进行模型预测。最终，汇总这些预测结果（例如取平均值或进行投票），以获得更稳健和准确的最终预测。
- 自反馈增强的迭代式大模型文本聚类：针对比较任务，通过不断精炼参考论文中正确论文的比例来获得更好的结果。

论文源头追溯（PST）

在论文源头追溯任务中，我们面临三大挑战：数据集标签分布差异、冗长的 HTML 格式标识符、超大规模无标注数据集辅助信息召回。具体而言，该任务存在规则标注的和人工标注的两类数据集，且这两类数据集的标签分布存在显著差异。规则标注的数据集数据量大，但标签置信度低、噪声较多且有效信息分散；人工标注的数据集标签置信度高，且与测试集分布一致，但数据量较少。此外，数据集中还存在大量 HTML 格式标识符，这些标识符文本长度可达数万 Token，却包含极少的有效信息。同时，该任务存在的超大规模无标注数据集（DBLP 数据集）拥有各论文完善的辅助信息，但需要自行召回有效信息。

故赛题的难点在于如何充分利用不同置信度的训练数据集及超长的上下文信息、提取高噪声数据中的有效增益信息。针对这两个难点，团队利用嫁接学习 [5] 的思想分别提出 Grafting-Learning For DataSet 技术和 Grafting-Learning For LongText 技术，将 BERT-Like 模型的复杂文本语义匹配能力嫁接到 LLM 中，提高样本置信度。同时，团队提出的 Automatic RAG & Feature Engineering 技术能够自动召回辅助信息，进一步去除超大规模无标注数据集中的高噪声。算法框架如下图 2 所示：

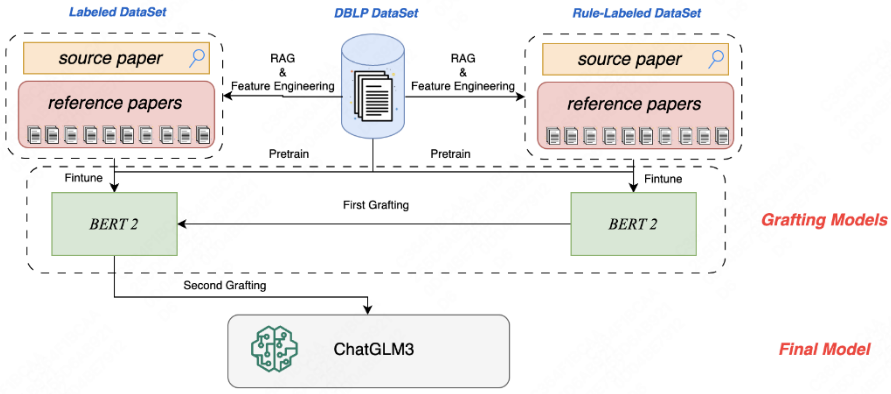


图 2：PST Solution By BlackPearl。RAG & Feature Engineering 指我们提出的自动召回及处理辅助信息技术，Pretrain 指利用 DBLP 数据集进行 BERT 模型的 MLM 预训练。方案经过两次嫁接学习，最终由 ChatGLM3 产出预测概率

- **Grafting-Learning For DataSet**：多个不同数据集之间的标注规则存在差异，由于本次存在规则标注的数据集，其标注质量较低但数据集规模极大，巧妙利用该数据集能够带来较大的指标提升。我们提出了 Grafting-Learning For DataSet 技术，在规则数据集上对 BERT 进行微调，并将其最后一层隐状态作为人工标注数据集的额外特征，用于训练第二个 BERT 模型。这种方法巧妙地嫁接了规则数据集的有效正标签信息。同时，我们在消融实验下发现嫁接学习的方式比普通迁移学习（即在规则标注数据集微调后的 BERT 模型再用人工标注数据集微调）更具鲁棒性。由于两类数据集的标签分布差异较大，普通迁移学习甚至会带来负收益，而嫁接学习能够有效保留有益信息、摒除无益信息。
- **Grafting-Learning For LongText**：长文本训练和推理会显著增加时间和显存消耗。为缓解该现象，团队提出了 Grafting-Learning For LongText 技术，将多来源的文本分别经过不同的 BERT 进行有监督微调，最终每条数据都将得到多个不同来源的模型预测概率，此时噪声文本中的有效信息增益都被 BERT 模型提取完成，再将预测概率输入到 ChatGLM 中，即可用较短的文本和 BERT 预测概率进行最终的模型判断。此方法能够有效避免在 Attention

计算时由于文本过长导致时间开销平方级爆炸增长，将其切割后利用小模型 BERT 提纯噪声，只保留过滤后的 BERT 预测概率用以做最终判断。同时，由于采用了多个 BERT 模型提纯去噪，高噪声对最终结果的影响进一步降低，模型输出结果更加置信。

- **Automatic RAG & Feature Engineering**：在超大规模无标注数据集（DBLP 数据集）中，存在着每篇论文对应的少量辅助信息，同时也存在大量相似论文的噪声信息。团队提出的 Automatic RAG & Feature Engineering 技术能够自动分析每篇论文的重点，于 DBLP 数据集中找到其对应的辅助信息，同时在 RAG 链路过程中就完成了一系列特征工程的构建。在辅助信息和特征工程的帮助下，模型能够以更短的输入产出更可信的结果。

学术论文问答（AQA）

在这个技术蓬勃发展，信息迅速更新的时代，为研究人员和大众提供多领域的高质量前沿学术知识已成为当务之急，本次学术论文问答任务要求参与者开发一个模型，能够通过检索相关论文来回答专业问题，本质上是一个检索任务，即给定用户问题，检索出最相关的论文，评估指标是 MAP@20。

本赛题数据集来源于从 StackExchange 和知乎网站检索问题帖，提取答案中提到的论文 URL 作为标签，从业务和数据角度上理解，带有复杂噪声的数据是该任务的主要难点，主要原因在于数据来源于互联网网站用户问题的引用数据集，不可避免的存在大量噪音。比如用户为什么引用这篇论文的认知标准是不一样的，理论上一个问题可以对应多个论文，所以噪音之大可想而知。采用开源向量模型对文本进行召回时，存在开源模型召回的论文从语意上跟原文非常接近，但正确答案可能排名靠后的问题。甚至在采用 bert 类 SimCSE 模型进行对比学习微调后模型性能进一步降低。

针对噪声大，学习难的问题，我们提出了三个核心方法去解决噪声大、学习难的问题。算法框架如图 3 所示：

- **LLM for Vector**：在 LLM 时代前期，一般文本相似度表征会采用自编码模

型，参数量一般不超过 1B，在大模型时代，LLM 的表征能力明显优于自编码器模型的表征能力，本次赛题我们采用了目前向量表征能力较好的 7B 模型 SFR-Embedding-Mistral[6]。

- **Hard example 挖掘**：在训练向量表征的模型时，比较通用的提升方法是负样本的挖掘，在训练向量表征模型时，负样本的挖掘通常是提升模型表现的重要手段。在赛题中，由于数据呈现出的是“相似并不一定正确”的规律，负样本的挖掘尤为重要。
- **Boosting**：Boosting 本质上是一种集成模型的思路，我们在 Boosting 思想的基础上，我们提出了一种迭代的负样本挖掘过程。在每次迭代中，模型能够召回难度更高的负样本，逐步累积具有更高挑战性的负样本集，进而指导模型学习。

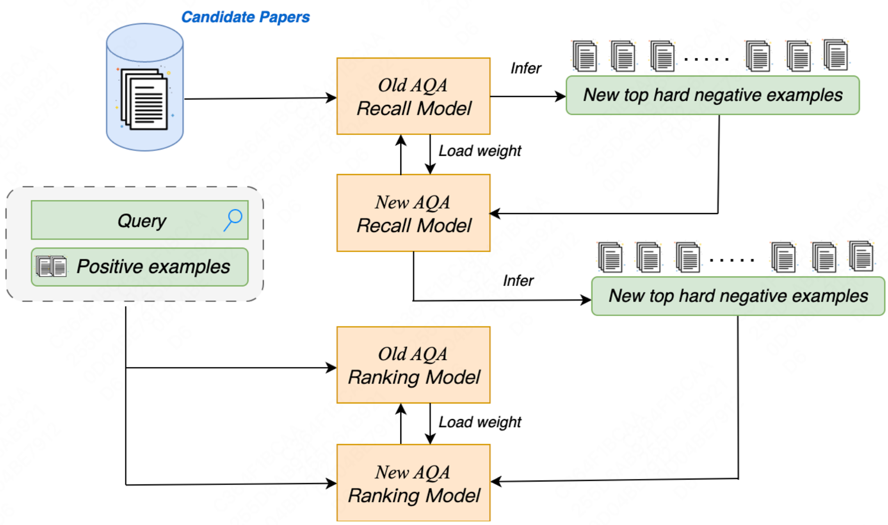


图 3：AQA Solution by BlackPearl 。Old AQA Recall Model 和 Old AQA Ranking Model 代表上一轮的基模型，New AQA Recall Model 和 New AQA Ranking Model 代表在基模型的基础上微调得到的模型

接下来介绍我们实现的模型框架细节：

- **召回（Recall）**：在此次竞赛中，用户查询的候选论文集包含成千上万的论文。

为了确保效率和实用性，将任务分成召回和排序两个过程是一个切实有效的策略，我们基于 SFR-Embedding-Mistral 进行指令微调，微调采用对比学习损失函数，每一个 batch 迭代过程中，对每一个正样本，我们随机从 100 个难负样本中抽取 3 个负样本，并联合 batch 内其他负样本计算一个样本的损失。每次迭代我们固定轮次 10 轮，学习率 1e-4，若采用 QLORA 进行微调可进行单卡微调。

- **排序（Ranking）**：在排序优化中，我们同样采取指令微调的方式，基于 SOLAR-10.7B-Instruct-v1.0 模型，微调采用交叉熵损失函数，每一个 batch 迭代过程中，对每一个正样本，我们随机从 100 个难负样本中抽取 3 个负样本计算一个样本的损失函数，每次迭代我们固定轮次 10 轮，学习率 1e-4，若采用 QLORA 进行微调可进行单卡微调。
- **Boosting 迭代**：在召回模型和排序模型的训练中，负样本是至关重要的。我们制定了一种困难负样本挖掘方法来进一步提高模型的性能即采用迭代的方法进行困难负样本挖掘，如图 3 所示。在召回模型的训练中，在初始迭代时，我们使用开源的 SFR-Embedding-Mistral 模型来检索前 100 个困难负样本。这些样本然后用于微调以获得增强的模型。在随后的迭代中，利用上一轮改进的模型来检索前 100 个困难负样本，并进一步进行微调。对于排序模型，我们继续使用由召回模型检索到的困难负样本进行训练。这个迭代过程确保每一轮训练都能提升模型在上一轮中的性能。

如果未经微调，即使是先进的基于 LLM 的模型也表现不佳。然而，通过实施我们的微调策略，我们在第一次迭代中观察到初步提升了 0.07。如下表所示，MAP@20 分数在后续的每次迭代中逐步提高。到第六次迭代时，改进的速度开始减缓。最终，我们共进行了八次迭代，并通过 Rank avg 融合，最终分数达到了 0.301，相对于原生的基座模型获得较大提升。

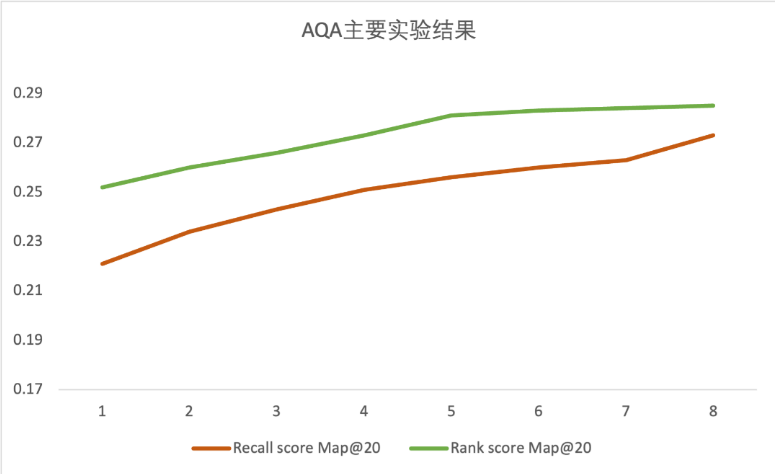


图 4：AQA 主要实验结果

在本次大赛中，BlackPearl 团队同学通过大模型技术解决了多个问题，再次让我们看到了前沿科技的力量。面向未来，大众点评技术部将不断深入探索大模型技术，充分挖掘其内在潜力，通过先进的 AI 技术，使点评 App 能够更精准地服务于用户，让 AI 帮大家更懂美食，更会生活。

写在后面

就在今天，美团技术团队公众号粉丝刚好突破了 40 万，感谢大家十年来的支持与厚爱，陪我们一路同行。我们常说，种一棵树最好的时间是十年前，其次是现在。从十年前开始，我们就信仰耐心和坚持的力量，愿意持续去做一些正确、有积累的事情。

十年来，美团技术团队公众号一直在努力践行，已经将 600 多篇美团内部优秀的技术文章分享给了大家，很开心能够跟大家一起学习交流、共同进步。恰逢中秋佳节即将到来，美团技术团队提前祝大家阖家欢乐，健康平安 ~~

参考文献

[1] Wang, G., Li, W., Ourselin, S., & Vercauteren, T. (2019). Automatic brain tumor segmentation using convolutional neural networks with test-time augmentation.

In Brainlesion: Glioma, Multiple Sclerosis, Stroke and Traumatic Brain Injuries: 4th International Workshop, BrainLes 2018, Held in Conjunction with MICCAI 2018, Granada, Spain, September 16, 2018, Revised Selected Papers, Part II 4 (pp. 61 - 72). Springer.

[2] Rasley, J., Rajbhandari, S., Ruwase, O., et al. (2020). Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (pp. 3505–3506).

[3] Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., & Chen, W. (2021). Lora: Low-rank adaptation of large language models. arXiv preprint arXiv:2106.09685.

[4] Dettmers, T., Pagnoni, A., Holtzman, A., & Zettlemoyer, L. (2024). Qlora: Efficient finetuning of quantized llms. Advances in Neural Information Processing Systems, 36.

[5] Jiangli Club. (n.d.). 嫁接学习的提出与具体用例 . Retrieved from http://jiangliclub.com/article?article_id=72.

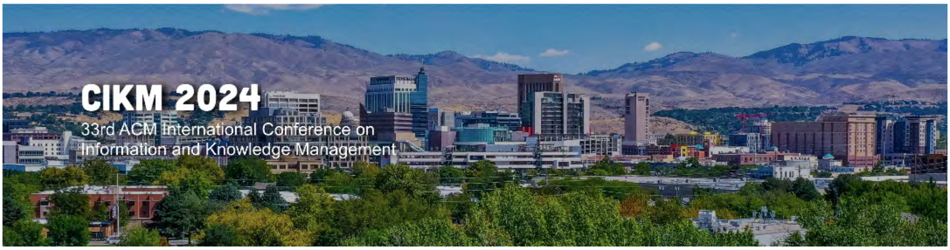
[6] Meng, R., Liu, Y., Joty, S. R., Xiong, C., Zhou, Y., & Yavuz, S. (2024). SFR-Embedding-Mistral: Enhance Text Retrieval with Transfer Learning. Salesforce AI Research Blog. Retrieved from https://blog.salesforceairesearch.com/sfr-embedded-mistral/.

[7] Zhang, F., Shi, S., Zhu, Y., Chen, B., Cen, Y., Yu, J.,… & Tang, J. (2024). OAG-Bench: A Human-Curated Benchmark for Academic Graph Mining. arXiv preprint arXiv:2402.15810.

CIKM 2024 | 美团技术团队精选论文解读

本文精选了美团技术团队被 CIKM 2024 收录的 8 篇论文进行解读，覆盖了自监督学习、解释生成、CTR 预测、跨域推荐、向量召回、图像生成、时效预测等多个技术领域。这些论文有美团独立研究，还有跟高校、科研机构合作的成果。希望能给从事相关研究工作的同学带来一些帮助或启发。

CIKM 是信息检索、知识管理和数据库领域中顶级的国际学术会议，自 1992 年以来，CIKM 成功汇聚上述三个领域的一流研究人员和开发人员，为交流有关信息与知识管理研究、数据和知识库的最新发展提供了一个国际论坛。大会的目的在于明确未来知识与信息系统发展将面临的挑战和问题，并通过征集和评估应用性和理论性强的顶尖研究成果以确定未来的研究方向。2024 年，CIKM 共收到全球 1496 篇论文投稿，最终 347 篇被接收，接收率约为 23%。

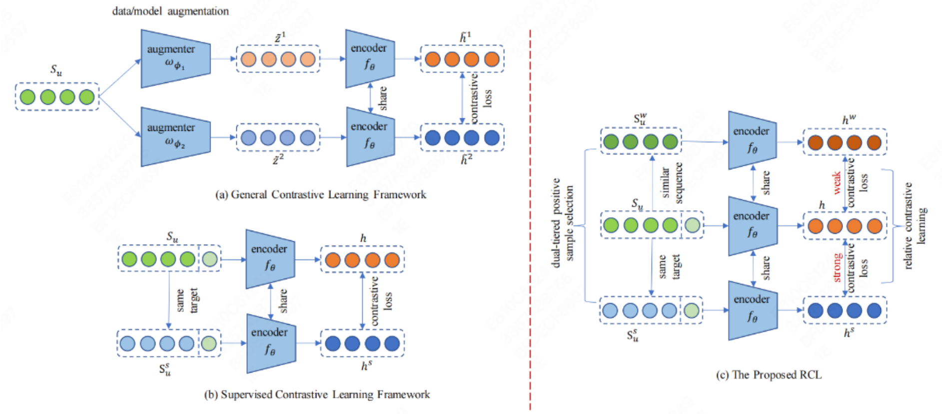


01 论文标题：Relative Contrastive Learning for Sequential Recommendation with Similarity-based Positive Sample Selection

论文作者：Zhikai Wang（Shanghai Jiao Tong University），Yanyan Shen（Shanghai Jiao Tong University），Zexi Zhang（Shanghai Jiao Tong University），Li He（Meituan），Yichun Li（Meituan），Hao Gu（Meituan）

论文类型：Poster（Full Research Paper track ）

论文地址：PDF



论文简介：推荐领域通常面临严重的数据稀疏性问题，对比学习作为一种自监督学习方法，通过提供额外的自监督信号来增强序列推荐模型的训练。现有方法通常依赖数据增强策略来创建正样本并促进表示不变性，但往往一些策略涉及到内容排序和内容替换可能会无意中改变序列中的用户意图信息。基于自监督对比学习的方法通过选择相同目标序列（与同一目标物品的交互序列）作为正样本，为基于增强的对比学习方法提供了一种替代方案。然而，基于 SCL 的方法存在相同目标序列稀缺的问题，因而缺乏足够的对比学习信号。

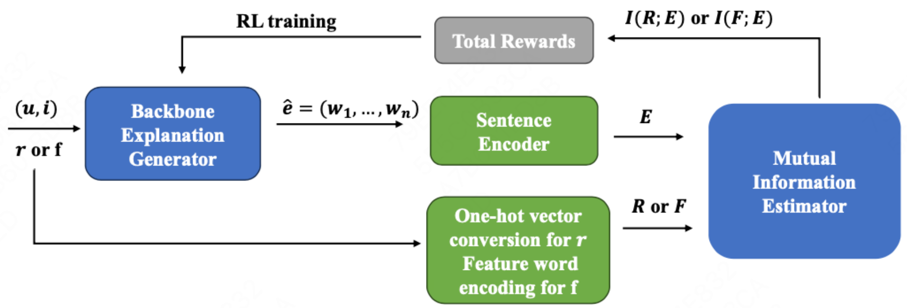
本论文提出使用相似序列（具有不同的目标内容）作为额外的正样本，并引入了一种名为相对对比学习 (RCL) 的新方法用于序列推荐。所提出的 RCL 包含两级正样本选择模块和相对对比学习模块，前者模块选择相同目标序列作为强正样本，并使用相似序列作为弱正样本，后者模块采用加权的相对对比损失，确保每个序列与其强正样本的表示更接近，而不是弱正样本。在公开数据集和点评业务数据集上 RCL 都优于现有方法，该算法在论文接收前已在大众点评首页信息流推荐场景落地并取得显著的效果。

02 论文标题：Aligning Explanations for Recommendation with Rating and Feature via Maximizing Mutual Information

论 文 作 者：Yurou Zhao（Renmin University of China），Yiding Sun（Renmin University of China），Ruidong Han（Meituan），Fei Jiang（Meituan），Lu Guan（Meituan），Xiang Li（Meituan），Wei Lin（Meituan），Weizhi Ma（Tsinghua University），Jiaxin Mao*（Renmin University of China）

论文类型：Research Track Full Paper

论文下载：PDF



论文简介：为用户提供基于自然语言的解释以证明推荐有助于提高用户满意度并赢得用户信任。然而，由于当前的解释生成方法通常被训练以模仿现有用户评论为目标，生成的解释往往与推荐商品的预测评分或是一些重要特征不一致，导致这些解释不能真正地帮助用户在推荐平台上做出明智的决策。为了解决这个问题，本文提出了名为 MMI（最大化互信息）的优化框架，以增强生成的自然语言解释与推荐商品的预测评分 / 重要特征之间的一致性。

具体来说，本文使用互信息（Mutual Information, 简称 MI）作为解释与预测评分 / 商品特征一致性的衡量标准，并训练一个基于 MINE 方法的互信息估计神经网络，将此

神经网络作为后续的 MI 估计器。然后，我们将一个训练好的解释生成模型视为主干模型，基于来自 MI 估计器的奖励对其进行基于强化学习的微调。微调过程会指导原先的主干生成模型学习会生成与商品的预测评分及重要特征更一致的解释。在三个公开数据集上的实验表明，MMI 框架可以提升不同的主模型，使它们在与推荐商品的预测评分和重要特征的一致性方面优于现有的模型。此外，本文通过用户实验验证了 MI 增强的解释确实有助于用户的决策，并且由于它们更好的一致性特点，与其他方法生成解释相比更能让用户满意。

03 论文标题：Enhancing CTR prediction through Sequential Recommendation Pre-training: Introducing the SRP4CTR framework

论文作者 Ruidong Han（Meituan），Qianzhong Li（Meituan），He Jiang（Meituan），Rui Li（Meituan），Yurou Zhao（Meituan），Xiang Li（Meituan），Wei Lin（Meituan）

论文类型：Short Paper

论文下载：PDF

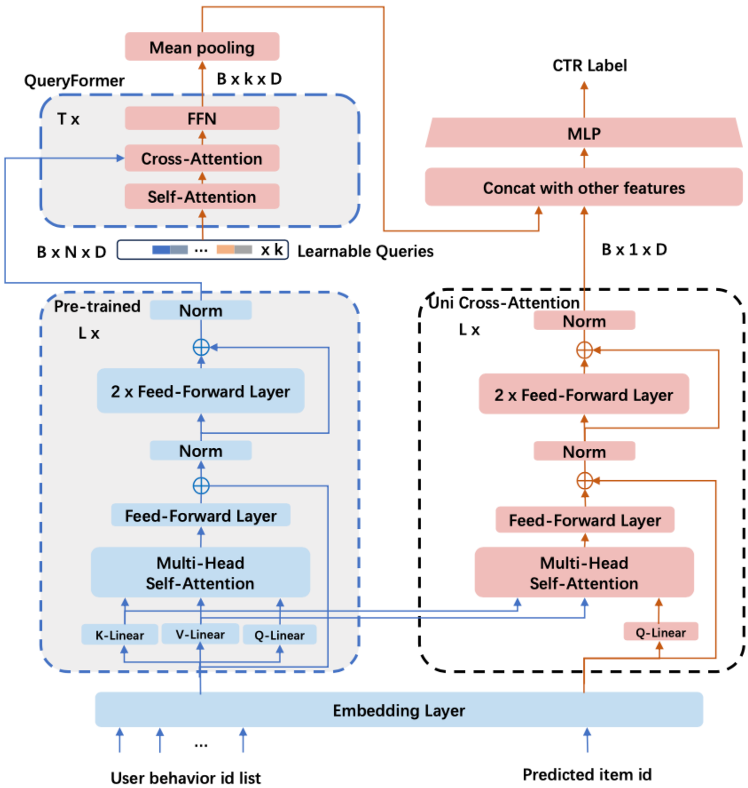


Figure 2: Overall architecture of our SRP4CTR method. The gray area represents the part that can be accelerated through folded inference.

论文简介：理解用户兴趣对于点击率（CTR）预测任务至关重要。在序列推荐中，通过自监督学习从用户历史行为中进行预训练可以更好地理解用户动态偏好，展现出与 CTR 任务直接集成的潜力。以往的方法将预训练模型集成到下游任务中，仅用于提取语义信息或单独的用户兴趣编码，然后将这些信息作为特征加入下游模型。然而，这些方法忽略讨论了下游任务中的额外推理成本，且没有考虑如何将预训练模型中的信息高效的转移到 CTR 任务预测的特定估计项中。

为了解决这一问题，本文提出了增强 CTR 预测的序列推荐预训练框架（SRP4C-TR）。首先，我们系统的讨论了引入预训练模型对推理成本的影响。随后，我们引入了一种新的预训练方法来尽可能保证编码时信息的完整性。在微调过程中，本文还引

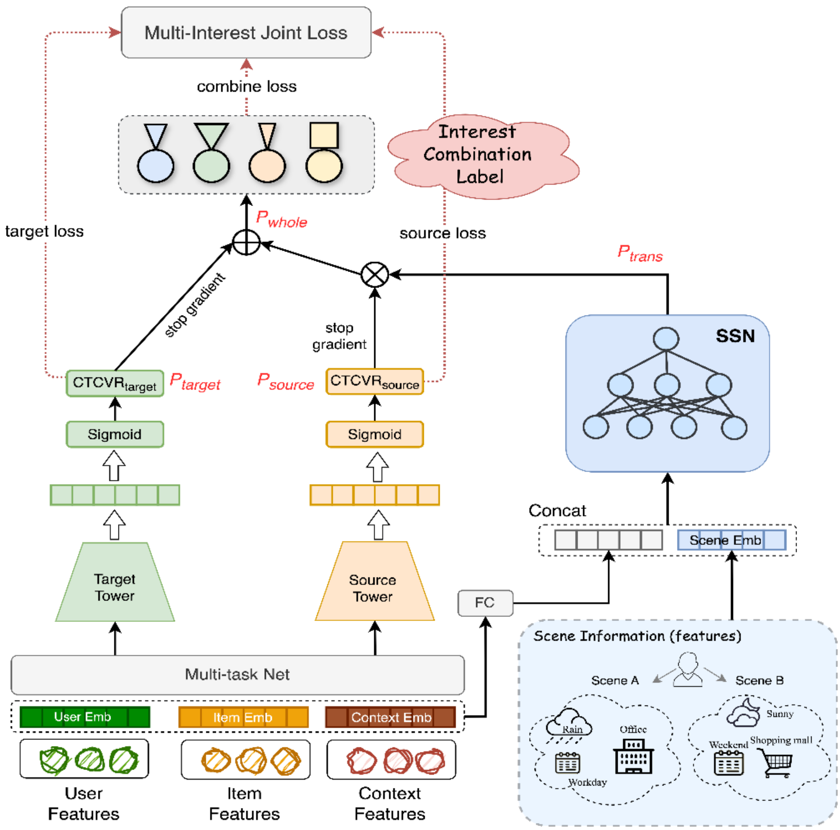
入了一个交叉注意力模块，以较低的成本建立了估计项与预训练模型之间的桥梁。此外，本文采用了一种新的自查询技术，以促进从预训练模型到工业 CTR 模型间的知识转移。离线和在线实验表明，本文的方法优于以前的基线模型。

04 论文标题：EXIT: An EXplicit Interest Transfer Framework for Cross-Domain Recommendation

论文作者：Lei Huang（Meituan），Weitao Li（Meituan），Chenrui Zhang，Jinpeng Wang（Meituan），Xianchun Yi（Meituan），Sheng Chen（Meituan），

论文类型：Applied Research Paper

论文下载：PDF



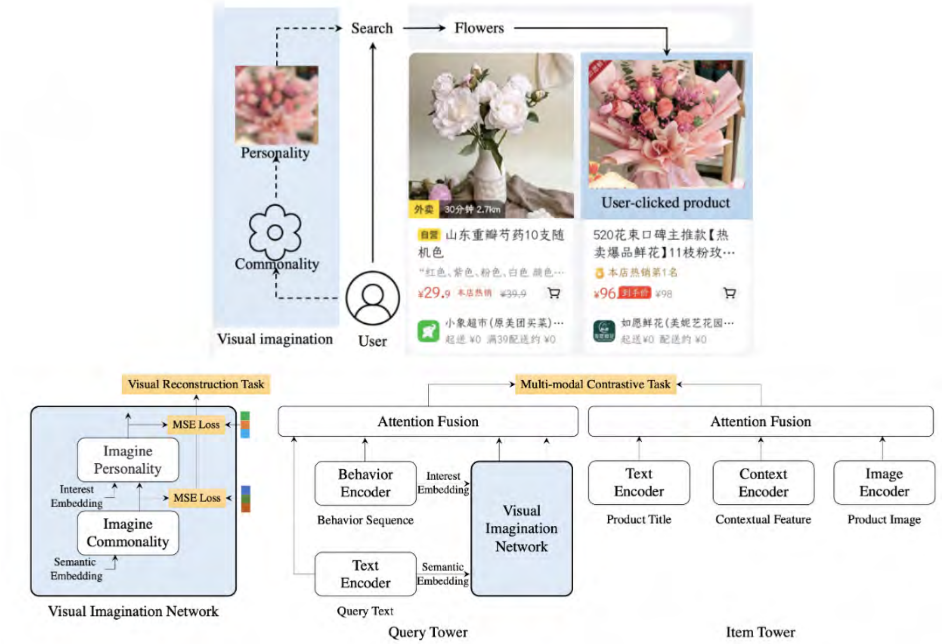
论文简介：跨域推荐是指利用其他领域的知识增强推荐系统对用户兴趣的预测精度，在工业应用中受到了广泛关注。现有的隐式建模跨域推荐方法并未考虑不同域之间服务功能和商品展现形式的差异，导致在落地过程中产生严重的负迁移问题。例如，用户在金刚和搜索中大量表达对医药、闪购等紧急需求的兴趣，直接将这些信号用于推荐系统并大量推送医药和闪购产品显然不合适。为解决这一挑战，本文提出了一种显式兴趣迁移框架，通过显式建模不同的用户上下文场景下源域信号向目标域迁移的概率，实现对目标域兴趣信号的筛选。无需复杂的网络结构及繁琐的模型训练过程，本文提出的显式框架能快速在工业推荐系统落地，为跨域推荐提供了一种简单而有效的解决方案。该算法已在美团首页推荐系统部署上线。

05 论文标题：VIER: Visual Imagination Enhanced Retrieval in Sponsored Search

论文作者：Yadong Zhang（Meituan），Yuqing Song（Meituan），Siyu Lu（Meituan），Qiang Liu（Meituan），Xingxing Wang（Meituan）

论文类型：Short Paper

论文下载：PDF



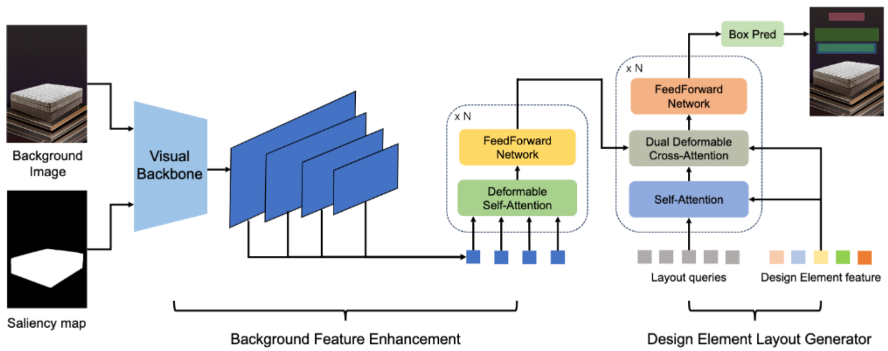
论文简介：向量召回是搜索系统的重要能力之一，通过将搜索词、用户和商品的信息编码为稠密向量，为用户检索出高质量的候选。然而，在即时零售场景下，搜索词（Query）存在两类极端问题：1）短 Query 通常为模糊意图，例如鲜花，2）长 Query 包含大量噪声实体，例如巧克力鲜花费列罗。这两类情况导致很难识别用户确切的搜索意图。实际上，消费者对于搜索的商品有心理图像预期，反映了他们特定的购买意图，基于此，本文提出了视觉预期增强的多模态检索模型来建模用户的潜在视觉偏好。具体来说，通过重建 Query 共性和用户个性的图像表征，并且与语义、用户行为序列等多模态信息相融合，增强了对用户搜索意图的理解，从而提升检索效果。通过搜索广告系统中的在线 A/B 实验，该方法在相较于基线在收入、点击和点击率等关键指标上取得了显著提升。此外，审稿人评价本文是多模态增强搜索理解的首批论文之一。

06 论文标题：Design Element Aware Poster Layout Generation

论 文 作 者：Yinan Li（Meituan），Jia Chen（Meituan），Yin Bai（Meituan），Jia Cheng（Meituan），Jun Lei（Meituan）

论文类型：Full Research Paper

论文下载：PDF



论文简介：图像创意在广告系统中起着重要的作用，利用商户素材制作优质的图片创意能够帮助广告主吸引更多的用户、获得更多的点击。海报布局生成领域虽然在近期取得了显著进展，但现有的方法多关注于对海报背景的理解，而忽略了设计元素（例如文字、Logo 和底纹）对布局的影响，这导致生成的布局经常存在明显的视觉瑕疵，包括（1）尺寸不当，例如将较短的文字放入较大的文本框中或将长文本放入较小的文本框中，以及（2）图像失真，例如拉伸变形的 logo 图标。 为此，本文定义了一个新的布局生成任务：感知设计元素的海报布局生成，该任务要求生成的布局不仅适配于背景图片，还要与给定的设计元素相匹配。

本文提出了一种名为 Design Element aware Transformer (DET) 的编码器 - 解码器网络，以生成既适合背景图像，又适配设计元素的合理布局。其中，编码器通过可变形自注意力从背景图像及其显著性图中提取细粒度的多尺度表示，解码器接收背景特征，通过可变形交叉注意力的方式将给定的设计元素内容特征、期望的宽高比特征

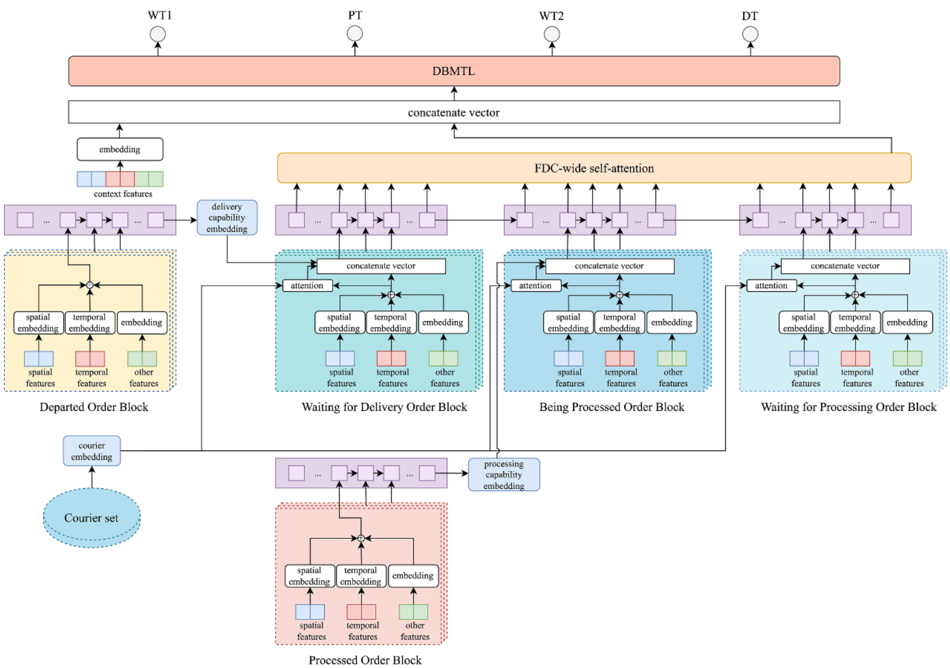
与背景特征进行关联，生成最适合各个给定的设计元素布局位置。同时，本文提出了一种新的评估指标 AspDiff，用于衡量生成的布局与给定设计元素的匹配程度。 在三个公开海报数据集上的定量和定性评估表明，与其他布局生成方法相比，DET 生成的布局框能更好的适配给定的设计元素，取得更好的视觉效果。该算法在论文接收前已应用上线，在展示、联盟等站内外主要广告场景中落地。

07 论文标题：Process-Informed Deep Learning for Enhanced Order Fulfillment Cycle Time Prediction in On-Demand Grocery Retailing

论文作者：Jiawen Wei（Meituan），Ziwen Ye（Meituan），Chuan Yang（Nankai University），Chen Chen（Meituan），Guangrui Ma（Meituan）

论文类型：Applied Research Paper

论文下载：PDF



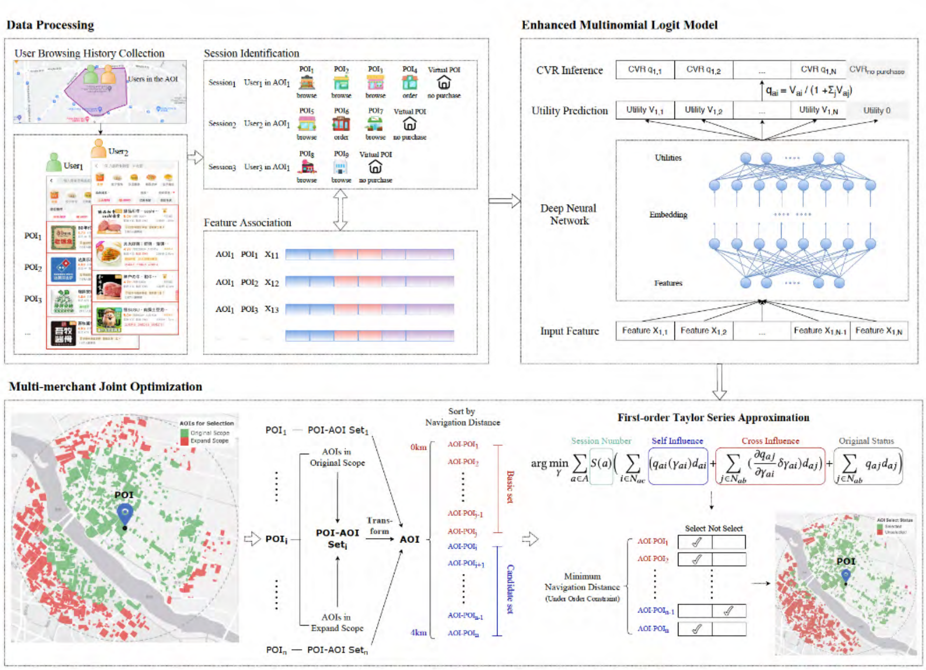
论文简介：在即时零售（On-demand Grocery Retail, OGR）领域，准确预测订单履约周期时间（Order Fulfillment Cycle Time, OFCT）对于提高客户满意度和运营效率至关重要。由于自营前置仓 + 配送的模式，小象这类 OGR 平台有着与即时外卖配送（On-demand Food Delivery, OFD）平台不一样的运营调度策略和可用数据，面临着截然不同的时效预测挑战。本文阐述了两种领域下时效预测问题的区别，提出了一种基于物理世界 OGR 订单履行过程的深度学习模型，用显式建模的方式刻画了订单量、生产能力、配送能力及调度策略等多种因素对 OGR 履约时效的影响。具体来说，我们使用多个循环神经网络（RNN）模块来动态评估生产和配送阶段的产能负载，并结合一系列专门的注意力模块来捕捉订单 - 订单之间以及潜在骑手 - 订单之间的相互作用对调度系统中订单履约优先级的影响。此外，我们的方法还利用深度贝叶斯多目标学习（DBMTL）来识别订单履约前序阶段对后续阶段的影响。该方法在小象数据集上已验证其优越性。我们的研究在 OFCT 预测方面取得了显著进展，为寻求优化履约流程和提升客户体验的 OGR 平台提供了深刻洞见。

08 论文标题：《Collaborative Scope: Encountering the Substitution Effect within the Delivery Scope in Online Food Delivery Platform》

论文作者：Yida Zhu（Meituan），Liying Chen（Meituan），Chen Zheng（Meituan），Jia Shi（Meituan），Daping Xiong（Meituan），Zewen Huang（Meituan），Shihao Ren（Meituan），Shuiping Chen（Meituan），Jinghua Hao（Meituan），Renqing He（Meituan）

论文类型：Applied Research Paper

论文下载：PDF



论文简介：商家配送范围定义了为顾客提供服务的地理区域，决定了用户所能看到的商家集合。在绘制这些范围时，构建准确的订单转化预估模型和合理刻画配送难度，是平衡平台规模与配送体验及效率的关键。目前的方法忽略了用户在选择商家时商家之间存在的替代关系这一前提，从单个商家的视角预估转化率。此外，由于大规模商家组合优化求解的复杂性，也增加了多商家联合决策的难度。

基于此，本论文将问题建模为多商家选品问题，提出了一种基于机器学习 + 组合优化的决策算法框架，从用户视角预估不同商家集合下的订单变化，确保其符合商家间存在替代关系的先验假设。为了解决求解效率问题，通过在组合优化中引入一阶泰勒级数近似的方法进行了优化。该算法框架在美团拼好饭业务上已全量，线上结果显示，在规模不受影响的情况下，通过降低订单交付距离，显著提高了交付效率。

ACL 2024 | 美团技术团队精选论文解读

本文精选了美团技术团队被 ACL 2024 收录的 4 篇论文进行解读，论文内容覆盖了训练成本优化、投机解码、代码生成优化、指令微调（IFT）等技术领域。这些论文是美团技术团队跟高校、科研机构合作的成果。希望能给从事相关研究工作的同学带来一些帮助或启发。

ACL 是计算语言学和自然语言处理领域最重要的顶级国际会议，由国际计算语言学协会组织，每年举办一次。据谷歌学术计算语言学刊物指标显示，ACL 影响力位列第一，是 CCF-A 类推荐会议。ACL 成立于 1962 年，世界上影响力最大、最具活力的国际学术组织之一，它每年夏天都会召开大会，供学者发布论文，分享最新成果，它的会员来自全球 60 多个国家和地区，是 NLP 领域最高级别的国际学术组织，代表了国际计算语言学的最高水平。

以下内容是 4 篇论文的解读：

01 Speculative Decoding via Early-exiting for Faster LLM Inference with Thompson Sampling Control Mechanism

论文类型：Long Paper

论文下载：PDF

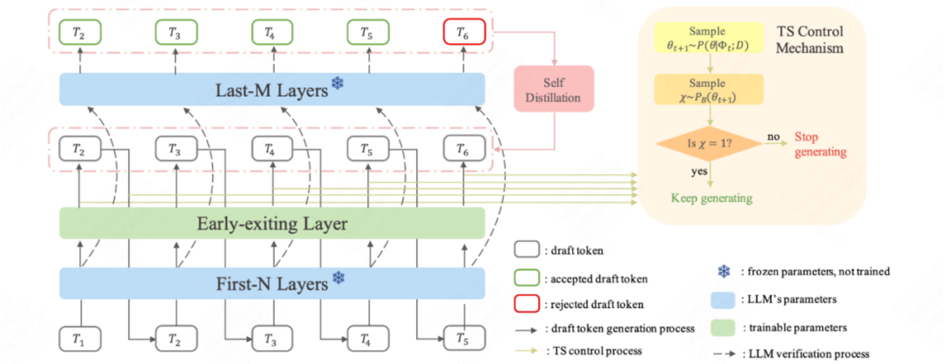


Figure 2: The framework of EESD which consists of three components: (1) Early-exiting layer which generate draft tokens efficiently and effectively; (2) Self-distillation which distills knowledge from the LLM (the target model); (3) TS control mechanism which can predict the optimal timing of terminating the draft token generation in each round. We divide the LLM (the target model) into two parts: the first-N layers and the last-M layers.

论文简介：近期，大型语言模型（LLMs）的发展突飞猛进，随之而来的就是推理成本上涨，这已经成为实际应用中较大的一个挑战。为了应对这些挑战，我们提出了一种名为「早期退出投机解码（EESD）」的全新方法，该方法实现了无损加速。

具体而言，EESD 在前 N 层之后加入早期退出的结构，并使用这一部分来生成草稿令牌（Draft Token）。为了提升这些初步令牌的质量，我们还结合了一种自我蒸馏方法。这种早期退出的设计不仅降低了部署和训练的成本，还大大提高了令牌（Token）生成的速度。

除此之外，我们还引入了一种新的采样机制，该机制利用了汤普森采样来调生成过程，并自动确定每一轮中草稿令牌的数量。然后，我们使用原始的 LLM 来验证这些草稿令牌，通过一次前向传递来确保最终输出的文本与原始的自回归解码保持一致。

在 13B 和 70B 的模型上的实验结果表明，我们的方法在文本生成速度上比以往的方法有显著的提升，这充分证明了该的方法的有效性。

02 Graph-Structured Speculative Decoding

论文类型：Long Paper

论文下载：PDF

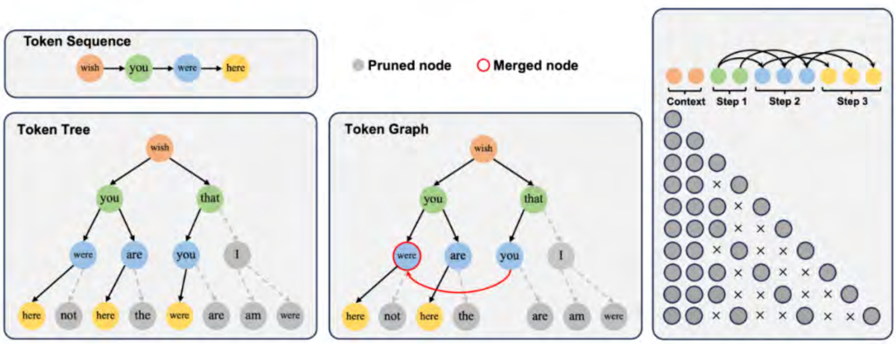


Figure 2: Overview of our method. (Left) GSD advances beyond TSD and SSD by implementing pruning strategies along with a re-occurring node merging technique. (Right) An illustration demonstrates the process by which the token tree (or graph) is flattened to a sequence. The sequence is then paired with a customized attention mask designed to uphold the proper dependencies between tokens to perform efficient drafting and verifying.

论文简介：投机解码已经崭露头角，它使用小型语言模型创建一种假设序列，然后由大型语言模型（LLM）进行验证，从而加快了 LLM 的推理速度。这种方法的效果主要取决于草稿模型的性能和效率如何平衡。在我们的研究中，我们试图通过生成多个假设，而不仅仅是一个，来增加被接受到最终结果的草稿令牌数量。这样，LLM 就有了更多的选择，并可以选择最长的、符合其标准的序列。我们的分析发现，由草稿模型产生的假设中有许多公共的令牌序列，这暗示了我们可以优化计算。

因此，我们引入了一种新的方法，使用有向无环图（DAG）来管理草拟的假设。这种结构使我们能够有效地预测和合并重复的令牌序列，大大降低了草稿模型的计算需求。我们将这种方法命名为图结构投机解码（GSD）。我们在多种 LLM 中应用了

GSD，包括一个参数达到 700 亿的 LLaMA-2 模型，结果发现文本生成速度提高了 1.73 倍到 1.96 倍，显著超过了标准的投机解码。

03 DolphCoder: Echo-Locating Code Large Language Models with Diverse and Multi-Objective Instruction Tuning

论文类型：Long Paper

论文下载：PDF

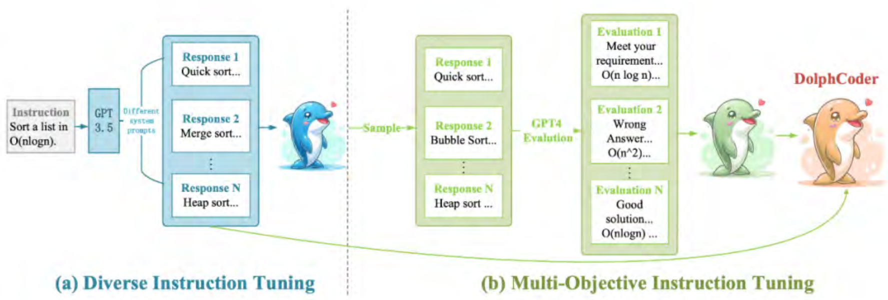


Figure 1: The overall architecture of our proposed diverse instruction tuning with self-evaluating for code generation, DolphCoder. Stage (a) denotes Diverse Instruction Tuning (DIT) and Stage (b) denotes Multi-Objective Instruction Tuning (MOT) for self-evaluating.

论文简介：在代码相关任务中，大型语言模型已经展现出出色的性能。为了提高预训练的 Code LLMs 的代码生成性能，一些工作已经提出了几种指令调优方法。

在本论文中，我们介绍了一种带有自我评估的多样化指令模型（DolphCoder），用于代码生成。它学习多样化的指令目标，并将代码评估目标结合起来，以增强其代码生成能力。我们的模型在 HumanEval 和 MBPP 基准上取得了优越的性能，为未来的代码指示调优工作提供了新的见解。我们的主要发现是：（1）增加具有不同推理路径的多样化响应可以提高 LLMs 的代码能力。（2）提高评估代码解决方案的正确性的能力也同时提高了创建代码的能力。

04 Learning or Self-aligning? Rethinking Instruction Fine-tuning

论文类型： Long Paper

论文下载： PDF

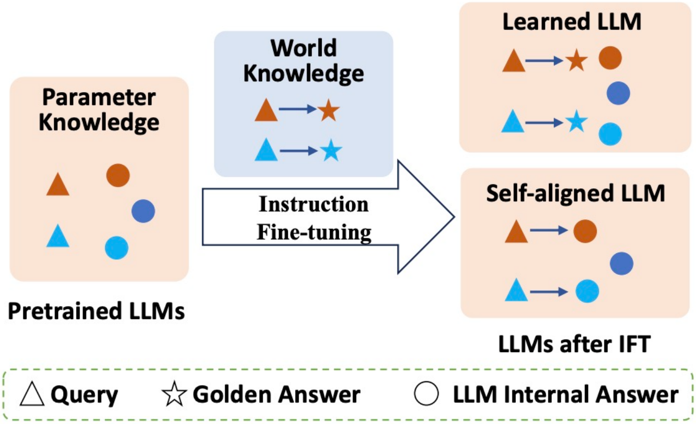


Figure 1: Two potential mechanisms for instruction fine-tuning. 1) Learning, which injects world knowledge in IFT data into the LLMs; 2) Self-aligning, which aligns queries with knowledge already in LLMs with similar behavioral norms. Elements with the same color are related.

论文简介： 指令微调（IFT）已经成为大型语言模型构建的核心步骤之一，当前主要是应用于模型行为模式的转换和注入特定领域知识。但指令微调对大模型输出的影响机制仍缺乏深入分析，对于指令微调给模型带来的增益，是由于指令微调过程带来的额外领域知识增益，还是其成功对齐了期望的输出空间从而实现了更好的知识表达机制尚不清楚。为此，本文设计了一个知识扰动的分析框架，来解耦合模型行为模式转换与额外知识注入的作用，以探索大模型指令微调的底层机制。

实验表明，试图通过指令微调学习额外知识往往难以产生积极影响，甚至可能导致明显的负面影响。而在指令微调前后保持内部知识一致性是实现成功指令微调的关键因素。研究结果揭示出指令微调的潜在机制，即指令微调的核心作用机制并不是让模型去「学习」额外知识，而是将模型内部现有的知识进行一种自我对齐，从而给模型带来增益。

SIGIR 2024 | 美团技术团队精选论文解读

本文精选了美团技术团队被 SIGIR 2024 收录的 3 篇论文进行解读，第一篇论文围绕如何利用深度学习，来整合广告拍卖和混排；第二篇论文扩展定义了全用户纵向联邦推荐范式，并首次提出基于检索增强的纵向联邦推荐框架 ReFer，解决了跨域特征缺失问题；第三篇论文提出了一种新颖的框架——解耦对比超图学习，并应用于下一个兴趣点推荐任务中。这些论文有美团技术团队的独立产出，也有跟高校、科研机构合作的成果。希望能给从事相关研究工作的同学带来一些帮助或启发。

SIGIR 的全称为 ACM Special Interest Group on Information Retrieval（ACM 国际信息检索大会），是中国计算机学会 CCF 推荐的 A 类国际学术会议，也是人工智能领域智能信息检索方向最权威的国际会议。根据会议官方统计，这次会议共收到 1148 篇长文投稿，其中有 791 篇有效长文投稿，仅有 159 篇长文被录用，录用率为 20.1%。

01 Deep Automated Mechanism Design for Integrating Ad Auction and Allocation in Feed

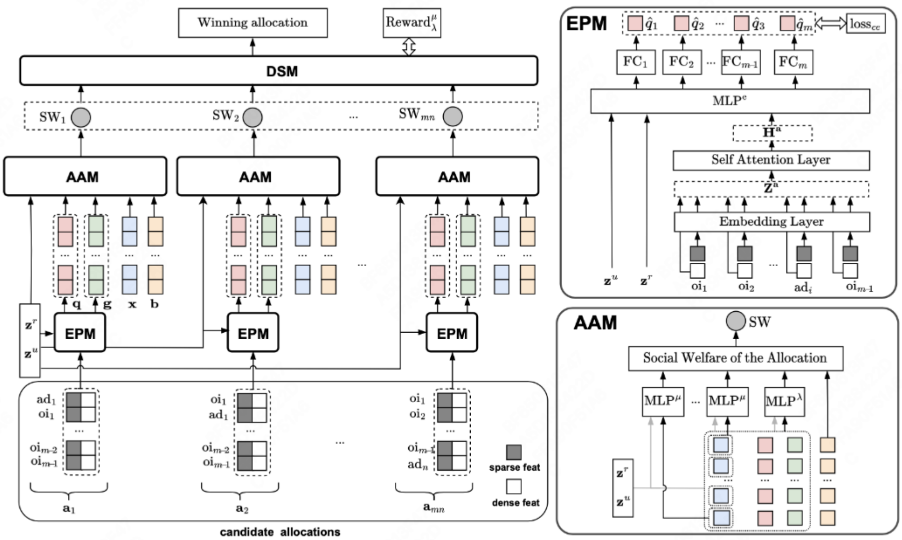
融合信息流广告拍卖与混排的深度自动机制设计

论 文 作 者：Xuejian Li*（Meituan），Ze Wang*（Meituan），Bingqi Zhu（Meituan），Fei He（Meituan），Yongkang Wang（Meituan），Xingxing Wang（Meituan）

备注：* 为共同一作。

论文类型：Long Paper

论文下载：PDF



论文简介：电子商务平台通常展示一个包含自然结果和广告的有序列表来响应用户的页面请求。这个列表是广告拍卖和混排的结果，直接影响平台的广告收入和总交易额（GMV），其中广告拍卖决定展示哪个广告及其计费，混排决定广告和自然结果的展示顺序。主流做法将广告拍卖和混排分为两个独立阶段，但这存在两个问题导致次优的结果：1）广告拍卖没有考虑外部性，例如实际展示位置和上下文对广告点击率（CTR）的影响；2）混排利用拍卖获胜广告的计费动态决定展示位置，未能维持广告

机制的激励兼容性(IC)。

因此,本文提出了一个深度自动机制,整合了广告拍卖和混排,确保在考虑外部性的情况下实现 IC 和个体理性(IR),同时最大化广告收入和 GMV。该机制将候选广告和自然结果的有序列表作为输入,对于每个候选广告,在自然结果有序列表的不同位置插入广告,生成所有候选分配。对于每个候选分配,页面级别模型将整个分配作为输入,输出每个广告和自然结果的预测结果,以建模全局外部性。最后,基于深度神经网络建模的自动拍卖机制选择最优分配并确定计费。该机制同时决定了广告的排名、计费和展示位置,在离线实验和在线 A/B 测试中,产生的广告收入和 GMV 高于最先进的基线。

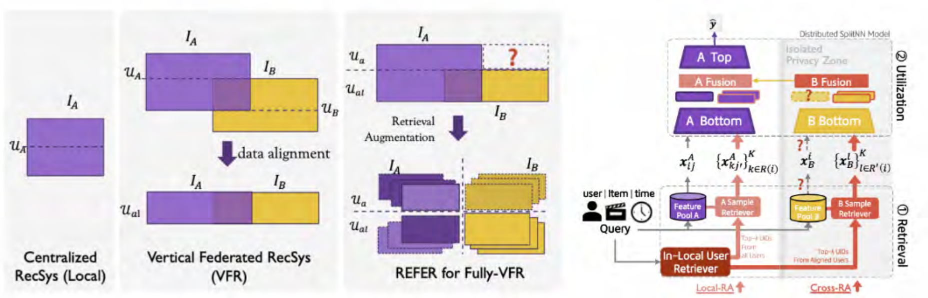
02 ReFer: Retrieval-Enhanced Vertical Federated Recommendation for Full Set User Benefit

ReFer: 一种面向全用户增益的检索增强式纵向联邦推荐框架

论文作者: Wenjie Li(Tsinghua), Zhongren Wang(Tsinghua), Jinpeng Wang(Tsinghua), Shu-Tao Xia(Tsinghua), Jile Zhu(Meituan), Mingjian Chen(Meituan), Jiangke Fan(Meituan), Jia Cheng(Meituan), Jun Lei(Meituan)

论文类型: Research Track Full Paper

论文下载: PDF



论文简介：随着跨企业数据流通的需求增长、和数据隐私保护监管日益严格，纵向联邦学习（Vertical Federated Learning，VFL）这一隐私机器学习技术被更多地应用于推荐系统中。然而传统联邦方案忽略了大量非交叉用户数据，不仅降低了训练过程中用户兴趣信息的丰富度，还导致模型只能对数量有限的交叉用户进行预测，极大降低了商业落地的性价比。

为解决这一问题，本论文扩展定义了全用户纵向联邦推荐范式（Fully Vertical Federated Recommendation，FullyVFR），并首次提出基于检索增强的纵向联邦推荐框架 ReFer。该框架提出了一种通用的二阶段分布式检索方案及其配套的分布式注意力融合机制，解决了跨域特征缺失问题，缓解了跨用户群的兴趣偏差，显著提高了全体用户在联邦模型上的性能增益。在公共数据集和美团业务数据集上的实验结果均显示，ReFer 能在多个任务场景下提升全体用户群的推荐性能。

03 Disentangled Contrastive Hypergraph Learning for Next POI Recommendation

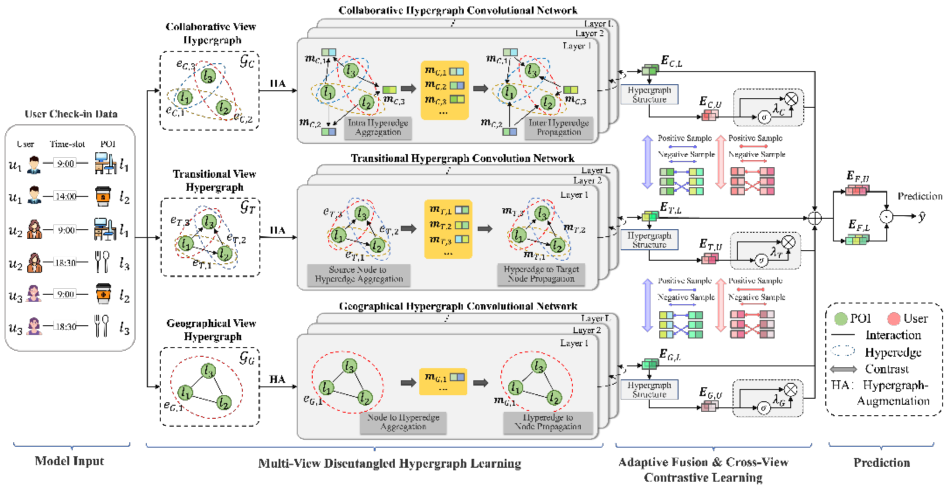
解耦对比超图学习用于下一兴趣点推荐

论文作者：Yantong Lai（IIE CAS; UCAS），Yijun Su（IIE CAS），Lingwei Wei（IIE CAS），Tianqi He（Meituan），Haitao Wang（Meituan），Gaode Chen（IIE CAS; UCAS），Daren Zha（IIE CAS），Qiang Liu（Meituan），Xingxing Wang（Meituan）

备注：IIE CAS 全称为 Institute of Information Engineering, Chinese Academy of Sciences；UCAS 全称为 University of Chinese Academy of Sciences

论文类型：Research Track Full Paper

论文下载：PDF



论文简介：下一个兴趣点（POI）推荐是一项重要且流行的任务，旨在为用户提供下一个感兴趣的位置建议。现有的大多数基于序列和图神经网络的方法已探索了多种途径来建模用户的访问行为，并取得了较好的性能。然而，目前仍有两个关键问题尚未得到充分关注：1) 大多数先前的研究忽视了用户偏好会受不同且不断变化的多方面决策因子影响，导致学到的用户表征耦合且次优；2) 许多现有方法未能充分建模不同用户决策因子之间的重要协同关联，阻碍了捕捉因子间互补推荐增强的能力。

为了解决这些挑战，本文提出了一种新颖的框架——解耦对比超图学习（DCHL），并应用于下一个兴趣点推荐任务中。具体而言，本文设计了一个多视图解耦超图学习组件，分别从协同、转移和地理视图解耦建模用户 -POI 交互行为，并针对性设计各视图感知的超图卷积网络学习解耦的 POI 表征。另外，本文提出了一个自适应融合方法来自动融合多视图用户表征，并采用了跨视图对比学习方法捕捉视图间的协同关联，实现用户表征和 POI 表征的表示增强。最后，本文在三个真实世界数据集上进行了充分实验，验证了所提方法相较多类别先进基线方法的优越性。

CVPR 2024 | 美团技术团队精选论文解读

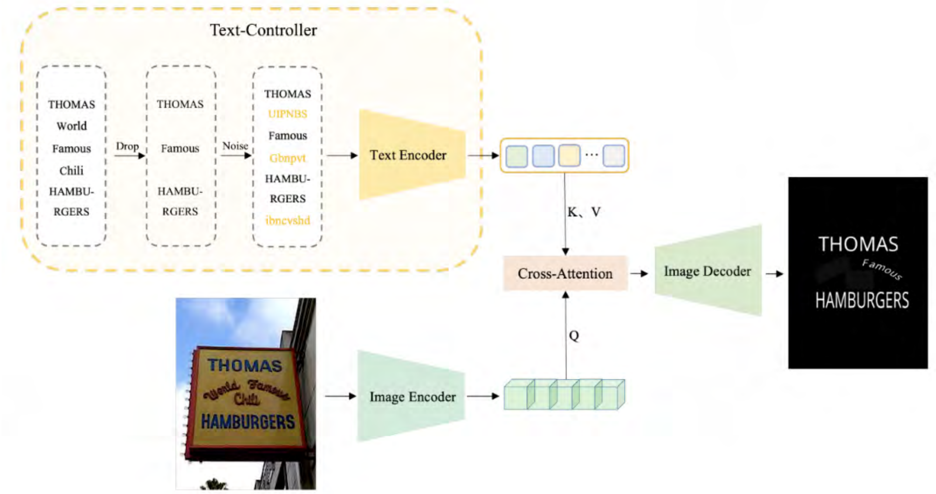
CVPR 全称为 IEEE Conference on Computer Vision and Pattern Recognition，国际计算机视觉与模式识别会议。该会议始于 1983 年，与 ICCV 和 ECCV 并称计算机视觉方向的三大顶级会议。根据谷歌学术公布的 2022 年最新学术期刊和会议影响力排名，CVPR 在所有学术刊物中位居第 4，仅次于 Nature、NEJM 和 Science。

本文精选了美团技术团队被 CVPR 2024 收录的 7 篇论文进行解读，这些论文既包括 OCR 预训练、长尾半监督学习等基础学习范式升级，也包括图生视频、数字人驱动、视听分割（AVS）等视觉 AIGC 技术创新。这些论文有美团视觉智能部的独立产出，也有跟高校、科研机构合作的成果。希望能给从事相关研究工作的同学带来一些帮助或启发。

01 | ODM: A Text-Image Further Alignment Pre-training Approach for Scene Text Detection and Spotting

论文作者：Chen Duan（Meituan），Pei Fu（Meituan），Shan Guo（Meituan），Qianyi Jiang（Meituan），Xiaoming Wei（Meituan）

论文地址：PDF



论文简介：近年来，文本 – 图像联合预训练技术在多个领域展现出了卓越的性能。然而，在光学字符识别（OCR）任务中，将文本提示与图像中相应的文本区域对齐是一个挑战，现有的基于 MIM（Masked Image Modeling）或者基于 MLM（Masked Language Modeling）的方法存在一定的局限性。

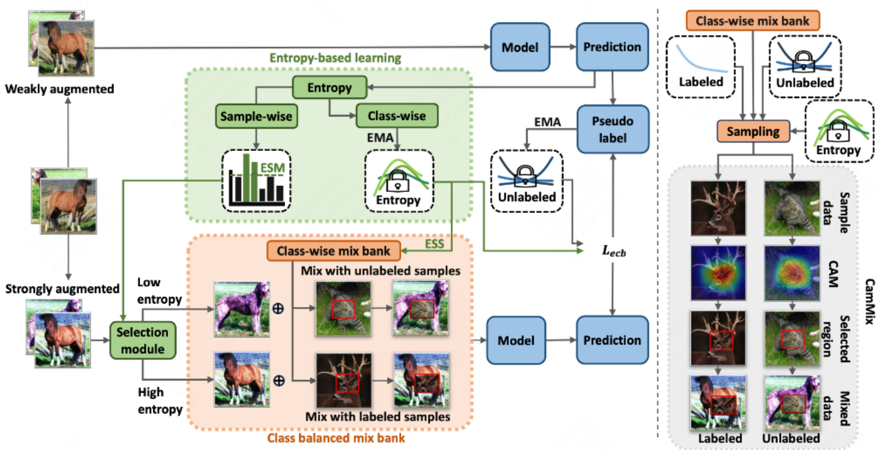
本文提出了一种创新的预训练方法，称为 OCR-Text Destylization Modeling（ODM），它可以将图像中不同风格的文本转换为基于文本提示的统一风格文本。通过 ODM，我们可以更好地对齐文本提示和图像中 OCR 文本，并使预训练模型适应场景文本检测和端到端任务中复杂多样的字体风格。此外，我们还设计了一种新颖的标签生成方法，并将其与我们提出的文本控制器模块相结合，有效降低了 OCR 任务中的标注成

本，使得更多未经人工标注的数据能够被用于预训练。在多个公共数据集上的实验表明，我们的方法在场景文本检测和端到端识别任务中显著提高了性能，并超过了现有的预训练方法。

02 | BEM: Balanced and Entropy-based Mix for Long-Tailed Semi-Supervised Learning

论文作者：Hongwei Zheng（Meituan），Linyuan Zhou（Meituan），Han Li（SJTU），Jinming Su（Meituan），Xiaoming Wei（Meituan），Xiaoming Xu（Meituan） 备注：SJTU（Shanghai Jiao Tong University）

论文地址：PDF



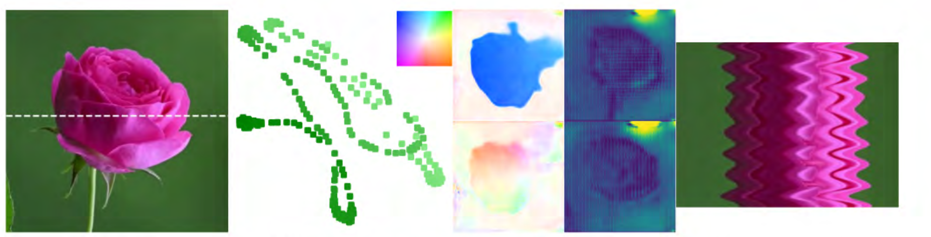
论文简介：长尾半监督学习（LTSSL）最近受到了广泛关注。本文探讨了数据混合在 LTSSL 中的应用。传统的数据混合方法通常采用批量混合，无法解决类不平衡的问题。此外，类的平衡不仅与数据量有关，还与类的不确定性有关，而类的不确定性与数据量的分布存在差距。例如，一些有足够样本的类可能拥有无法区分的特征，从而导致高不确定性。

为此，本论文介绍了基于平衡和熵的混合（BEM），这是一种开创性的混合方法，可

重新平衡数据量和不确定性的类别分布。具体来说，利用类平衡混合库来存储类数据，并根据对数据分布的估计对其进行采样混合，从而重新平衡类数据量。此外，我们还引入了一种基于熵的学习方法来重新平衡类的不确定性，包括基于熵的采样策略、基于熵的选择模块和基于熵的类平衡损失。实证结果表明，在多个基准测试中，BEM 与重新平衡方法相辅相成，显著提高了重新平衡方法的性能。作为首个利用数据混合来改进 LTSSL 的策略，BEM 证明了其在补充再平衡方法方面的多功能性。在不同的数据分布、数据集和 SSL 学习者之间，证明了 BEM 在补充再平衡方法方面的通用性。

03 | Animating General Image with Large Visual Motion Model

论文作者：Dengsheng Chen（Meituan），Xiaoming Wei（Meituan），Xiaolin Wei



(a) I_0　(b) $\hat{z}_{1,\cdots,K}$　(c) $\hat{p}_{1,\cdots,K}$　(d) X-t slices

论文简介：传统基于光流构建的图像驱动算法往往受限于一些特定的使用场景，例如人脸表情驱动、手势驱动等，而无法广泛用于预测任意场景的动态特征。我们认为这主要是由于相关领域缺乏大规模高质量的训练数据和学习能力足够强的模型结构导致。

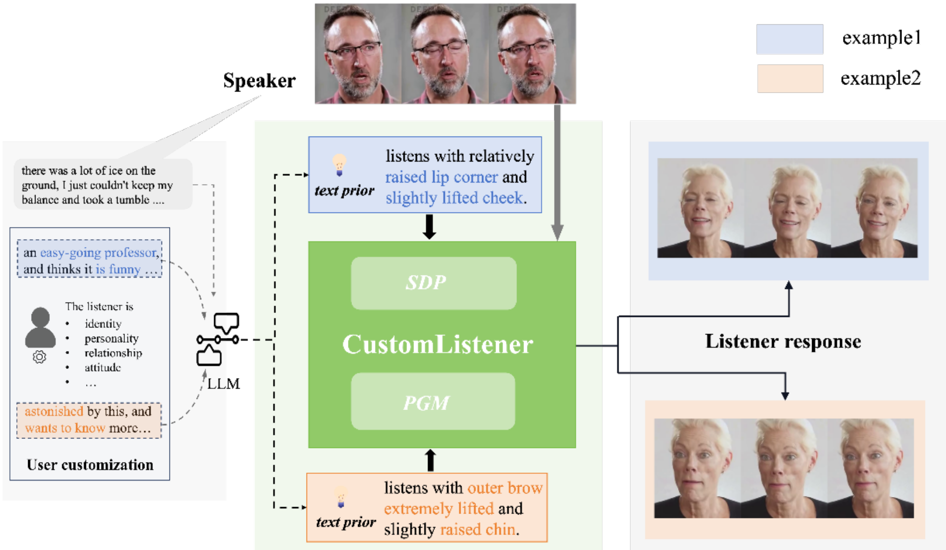
鉴于近期扩散模型在文生图上表现出令人惊艳的效果，我们首次尝试构建一个大规模的网络结构用于预测复杂场景的光流，并称之为大型视觉运动模型（LVMM）。LVMM 主要由神经渲染网络（R），光流预测网络（P），压缩和重建网络（E 和 D）以及一个潜在空间的扩散模型 e 构成。整个模型需要经过三个阶段的独立训练。

首先，LVMM 通过光流预测网络 P 生成给定两张图像之间存在的光流信息，然后通过神经渲染网络 R 用于将光流信息渲染成逼真的图像运动效果。在这个阶段，我们发现可以使用两个不同的网络分支分别预测光流中的高频和低频信息（如上图 c 所示）。其中高频信息则善于捕捉物体边缘的一些细微的运动特征，而低频信息则用于描述物体整体的运动趋势。两种特征相互补充，从而有效的估计出各种复杂场景下细微的运动差异。在完成 R 和 P 的联合训练以后，我们固定光流预测网络 P 的参数，将它作为一个数据预处理的操作，用于训练压缩和重建网络（E 和 D）。在实验过程中发现，直接使用扩散模型 e 去预测光流训练得到的模型无法取得很好的泛化性能，这是因为光流特征的推断往往严重依赖物体的视觉特征。如果想要得到足够好的泛化性，我们需要设计算法更好的将物体的视觉和运动特征解耦。我们发现通过压缩网络 E 可以将光流信息中的视觉和运动特征分别映射到两个不同的空间，从而保证扩散模型 e 的有效训练。最后，扩散模型 e 通过在高维特征空间解构物体的视觉特征和运动特征来准确预测图像中蕴含的动态特征，从而驱动静态图像表现出符合自然规律的动态效果，大大增加了图像的视觉吸引力。

04 | CustomListener: Text-guided Responsive Interaction for User-friendly Listening Head Generation

论文作者：Xi Liu*（Meituan），Ying Guo*（Meituan），Cheng Zhen（Meituan），Tong Li（Meituan），Yingying Ao（Meituan），Pengfei Yan（Meituan）

论文地址：PDF



论文简介：近年来，数字人生成技术逐渐发展并应用于虚拟对话交互场景中，通过模拟真实 Speaker 和 Listener 的表情和肢体语言，来创造生动和更具沉浸感的交流场景。然而，现有 Listener 生成中，用户只能通过简单情绪标签去控制 Listener 属性，可控力有限。本文中，我们提出 CustomListener，用户可以使用任意自由文本自定义想要的 Listener 属性（身份、性格、行为习惯、社会关系等），模型结合自定义的文本属性以及交流场景中 Speaker 的讲话内容 / 语音 / 动作，实时生成合理且逼真的 Listener 反应。

具体而言，我们首先基于 ChatGPT，依据用户定义文本和 Speaker 讲话内容，得到指导 Listener 动作的静态文本先验，从语义层面分析刻画来得到 Listener 的行为基调。该静态先验只提供了窗口时间内 Listener 的静态基调动作，然而对话中，Listener 的行为需要配合 Speaker 的实时状态，来确定静态基调动作的完成节奏和幅度信息。为实现这种 Speaker-Listener 行为的协调性，SDP 模块根据 Speaker 语音 - 静态文本先验的响应式交互来获得基调动作的完成节奏引导，根据 Speaker 动作对交互结果进行精炼来获得基调动作的幅度引导，由此将静态文本先验转换为包含 Listener 动作完成节奏和幅度信息的动态肖像 Token。为实现长视频生成的片

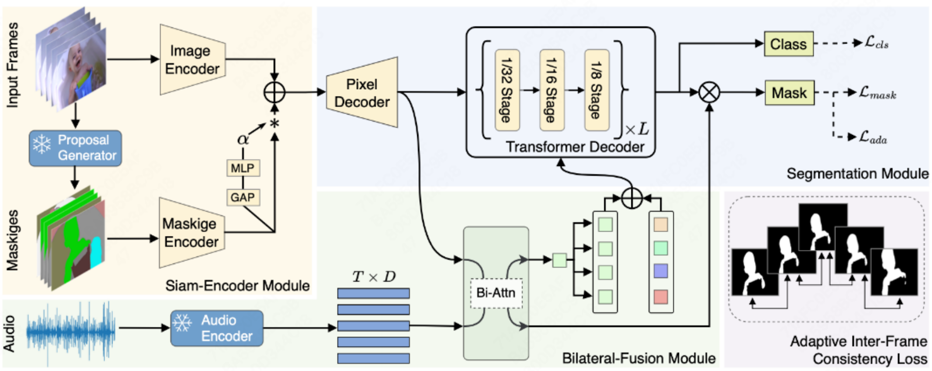
段间连贯性，PGG 模块基于片段间动态肖像 tocken 的相似性生成运动先验，以此保持片段间 Listener 行为的连贯性和属性的一致性，并基于以运动先验和动态肖像 Token 为条件的 diffusion 结构，最终实现听者的可控生成。

05 | Cooperation Does Matter: Exploring Multi-Order Bilateral Relations for Audio-Visual Segmentation

论文作者：Qi Yang（UCAS,CASIA），Xing Nie（UCAS,CASIA），Tong Li（Meituan），Pengfei Gao（Meituan），Ying Guo（Meituan），Cheng Zhen（Meituan），Pengfei Yan（Meituan），Shiming Xiang（UCAS,CASIA）

备注：UCAS（School of Artificial Intelligence, University of Chinese Academy of Sciences）；CASIA（Institute of Automation, Chinese Academy of Sciences）

论文地址：PDF



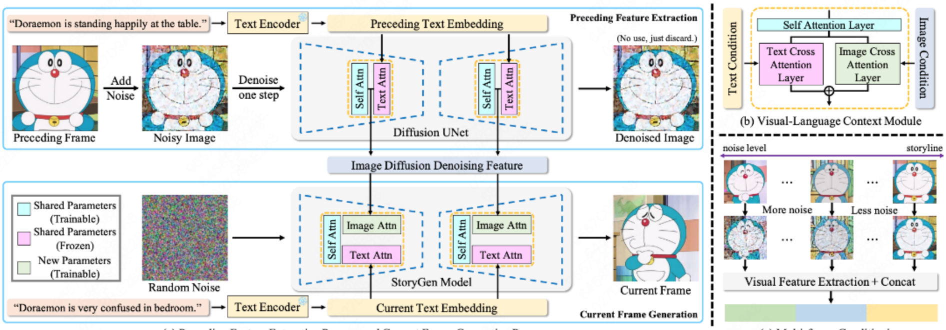
论文简介：人类的视觉注意力常受听觉引导，即我们倾向于专注发声目标。基于此，我们引入了视听分割（AVS）任务，旨在像素级分割视频中的发声目标。该任务需对场景进行音频驱动的像素级理解，极具挑战性。

本论文提出了一种创新的视听 Transformer 框架，名为 COMBO，即 COoperation of Multi-order Bilateral relatiOns。该框架首次探讨了视听分割中三种双边纠缠关系：像素纠缠、模态纠缠和时间纠缠。针对像素纠缠，图像和发声目标掩码之间存在像素级关系，图像中的无关背景往往会影响掩码预测的精度，目前大部分方法所依赖的基础分割模型如 SAM（Segment Anything Model）系列，在通用分割任务中展示出了很好的鲁棒性和泛化性，但迁移到 AVS 任务中后，无法达到很好的性能，因为 AVS 目的是得到所有发声目标的像素级分割，而 SAM 是在无语音引导条件下的类别级分割，无法直接进行适配。因此我们采用了孪生编码模块，利用先验知识生成更精确的视觉特征。针对模态纠缠，两种模态之间存在内在联系，如图像可以用文字描述，声音可以对应图像中的目标物，已有的方法往往聚焦在音频模态对视觉模态的影响，而忽略了视觉对音频的影响，相较于以上单边融合方法，我们认为两种模态的相互融合能带来更优的效果，因此设计了双边融合模块，来实现视觉特征和听觉信号的双向对齐，该模块使视觉特征更聚焦在发声目标，同时使语音信号更关注视觉目标。针对时间纠缠，在视频序列中，能够根据过去的帧序列结果来估计当前帧，同时也可以根据当前帧结果预测未来帧，基于以上时序间内在关系，我们引入了一种自适应帧间一致性损失算法。综合实验和消融研究表明，COMBO 在 AVSBench-Object 和 AVSBench-Semantic 数据集上均优于现有的最先进方法。

06 | Intelligent Grimm – Open-ended Visual Storytelling via Latent Diffusion Models

论文作者：Chang Liu*（SJTU,Shanghai AI Laboratory），Haoning Wu*（SJTU），Yujie Zhong（Meituan），Xiaoyun Zhang（SJTU），Yanfeng Wang（SJTU,Shanghai AI Laboratory），Weidi Xie（SJTU,Shanghai AI Laboratory）

备注：SJTU（Shanghai Jiao Tong University）



(a) Preceding Feature Extraction Process and Current Frame Generation Process (c) Multi-frame Conditioning

论文地址：PDF

论文简介：生成模型最近在文本到图像生成方面展示了出色的能力，但在生成连贯的图像序列方面仍然存在困难。在本研究中，我们专注于根据给定的故事情节生成连贯图像序列的新颖而具有挑战性的任务，称为开放式视觉叙事。我们的工作有以下三个贡献：

为了完成视觉叙事的任务，我们提出了一种基于学习的自回归图像生成模型，称为 StoryGen，它具有一个新颖的视觉 – 语言上下文模块，可以在依据相应的文本提示和之前的图像 – 字幕对的条件下生成当前帧；

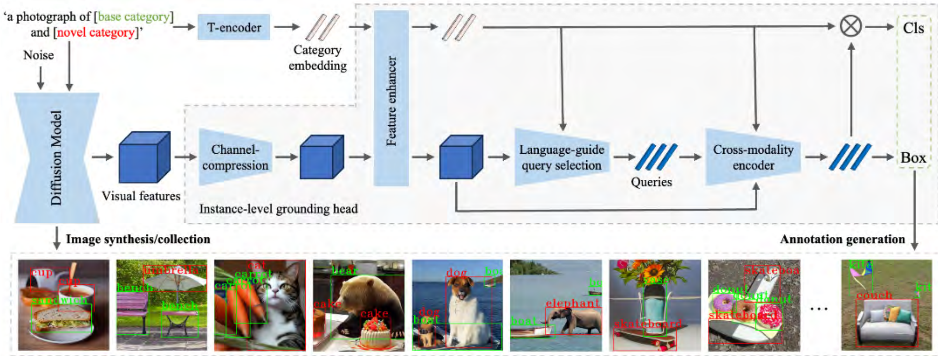
为了解决视觉叙事数据的不足，我们通过从在线视频和开源电子书中收集配对的图像 – 文本序列，建立了处理流水线，构建了一个具有多样化人物、情节和艺术风格的大规模数据集，命名为 StorySalon；

定量实验证明了我们的 StoryGen 的优越性，我们展示了 StoryGen 可以推广到未见过的角色而无需任何优化，并生成具有连贯内容和一致人物的图像序列。

07 | InstaGen: Enhancing Object Detection by Training on Synthetic Dataset

论 文 作 者：Chengjian Feng（Meituan），Yujie Zhong（Meituan），Zequn Jie（Meituan），Weidi Xie（SJTU），Lin Ma（Meituan）

备注：SJTU（Shanghai Jiao Tong University）



Generated training images for object detection

论文地址：PDF

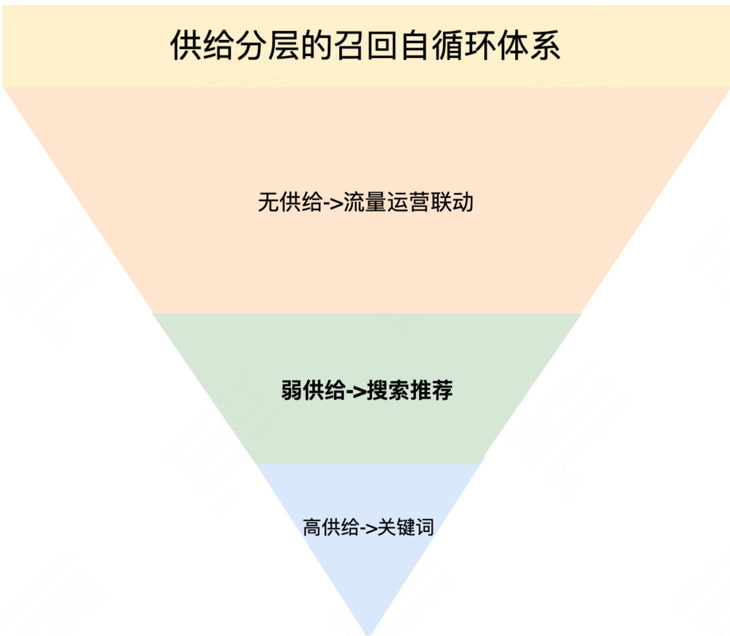
论文简介：近年来，文本到图像的生成模型在生成高质量图像方面取得了显著的成功，这为使用合成图像训练视觉系统提供了可能。现有的文本到图像生成模型通常可以根据某些自由形式的文本提示来生成图像。尽管这些生成的图像看起来很逼真，但无法满足训练复杂系统的需求，因为这些系统通常需要有实例级的注释，例如目标检测需要物体边界框。

在本文中，我们探索了一种创新的数据集合成范式，用于训练目标检测器以提高其性能，例如扩展类别或改进检测能力。具体而言，我们成功地将一个实例级的检测头（Grounding head）集成到一个预训练的生成模型中，以增强其在生成图像中定位物体实例的能力。检测头通过使用来自现成目标检测器的监督，以及一种针对目标检测器未覆盖的类别的新颖自训练方案，将类别名称的文本嵌入与扩散模型的区域视觉特征进行对齐。我们进行了详细的实验，结果表明这个增强版的生成模型，即 InstaGen，可以作为一个数据合成器，通过使用其生成的样本来增强目标检测器的性能，无论是在开放词汇（+4.5 AP）还是数据稀缺的情况下（+1.2 – 5.2 AP），都比现有最先进的方法表现出更好的性能。

百亿大规模图在广告场景的应用

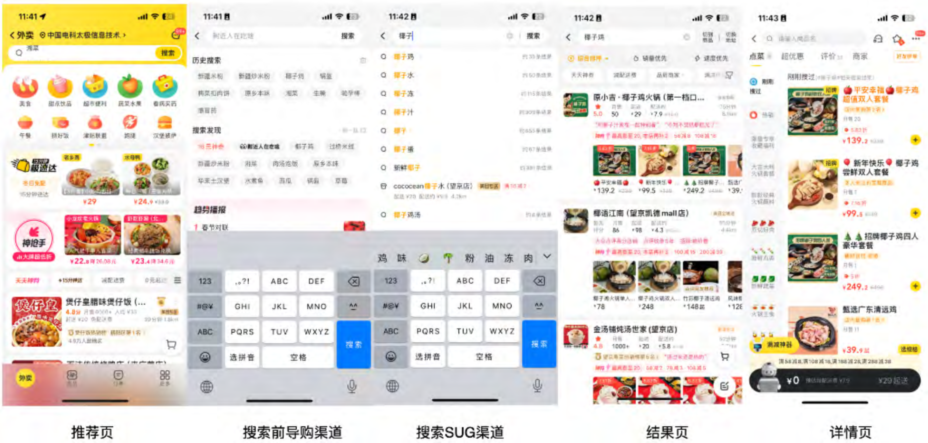
1 引言

美团外卖在线服务正成为日常生活中必不可少的服务，其中召回作为外卖广告系统的第一个环节，主要承担着从海量商品中寻找优质候选的角色。相比于业界召回系统，外卖场景召回阶段存在 LBS 限制，因此外卖搜索广告[1]提出供给分层的自循环召回体系：无供给区域，实现流量运营联动提升流量召回上限；高供给区域，通过关键词、向量召回提升召回效率；弱供给区域，通过搜索推荐进行弱供给填充，提高候选效率。搜索推荐目标是解决用户搜索意图不明确、供给受限制的流量下，从满足用户需求的角度出发进行的用户 –> 供给匹配，提高弱供给流量变现效率、用户搜索效率。



1.1 外卖广告搜索推荐业务及挑战介绍

用户进入外卖场景，整体浏览路径为推荐页、搜索页，进入搜索页之后整体浏览路径为搜索前导购渠道、搜索 SUG 渠道、主动搜索渠道、结果页、详情页，搜索推荐主要目标是解决搜索意图不明确、供给受限制的候选匹配问题，主要覆盖搜索前导购渠道（搜索发现）、搜索 SUG 渠道、结果页【 POI+SPU 】组合推荐、结果页相关填充等场景。



搜索推荐覆盖如上多个场景，具有场景多且场景输入交互和展现形态异构的特点，第一个挑战是如何统一建模异构多场景业务，提高弱供给匹配效率（多渠道）。外卖用户需求变化多样，从用户行为中可以发现，用户有在不同场景之间比较，需求发生演化至逐渐收敛的特点，例如用户从推荐转搜索、搜索换 Query、结果页反复对比、最终成单或者离开，第二个挑战是如何实时、准确捕捉用户需求的演变，完成用户与供给的高效匹配（即时化）。

针对搜索推荐业务多渠道、即时化特点，业界语义向量召回、个性化向量召回一般解决方案和问题是：

- 针对输入交互和展现形态差异较大的多种异构业务，不同业务样本组织方式差异较大，由于向量召回以线性方式组织样本，导致异构业务样本难以统一，因

此一般每个向量模型基于当前场景数据或者多场景数据进行单场景精细化建模，存在迭代效率低、小场景迁移能力弱的问题；

- 通过长短期序列建模，精细化刻画在不同时段内用户需求变化关系。时间段划分的序列内，存在数据稀疏性高、兴趣圈封闭、兴趣演变刻画粒度粗的问题。

搜索推荐业务的多个场景输入交互和展现形态差异较大，难以应用传统的具有相同目标、相似特征的多场景个性化向量召回建模方法，图结构作为多维非规则立体结构，由多种异构类型节点和节点间关系组成，适合通过异构图统一搜索推荐多异构场景。

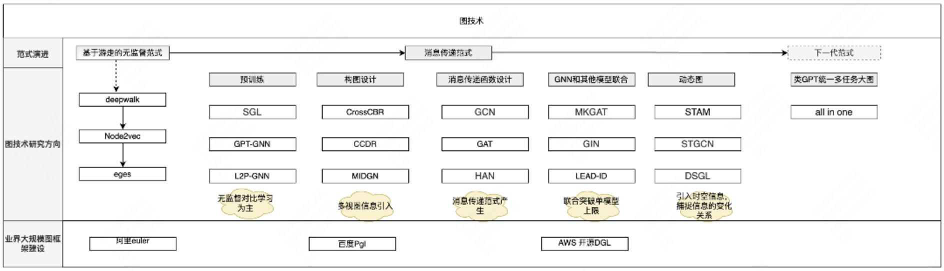
图技术具有异构节点关系关联能力、高阶关系聚合能力、稀疏节点高阶表征的特点，通过关系聚合、关联能力缓解小场景难以学好、稀疏节点难以表征好的问题，因此我们提出多场景异构大图统一建模解决搜索推荐渠道多带来的迭代效率低、异构场景难以统一、小场景难以学好的问题。用户需求具有不同场景间相互比较，需求演变至逐渐收敛的特点，这种即时性的变化特点，我们以多场景异构大图为基座提出异构动态图在线建模刻化需求演变关系，解决兴趣演变刻画粗、数据稀疏性高的问题。

1.2 图技术和引擎介绍

最近几年工业界和学术界在图领域研究取得了不错的进展，我们在这里对图深度学习的范式演进、主流研究方向、图引擎发展进行梳理 [2][3][4][5][6][7]。

图神经网络范式演进主要由基于图游走的无监督范式 -> 基于聚合的消息传递范式 -> 下一代范式，从浅层无监督深度学习到统一全场景图深度学习发展。在主流的基于聚合的消息传递范式下，主要研究方向分为消息传递函数设计、构图设计、图预训练、联合训练、动态图等主流方向。

图神经网络范式演进决定了未来走向图多任务统一方向，我们期望在范式演进路线上找到搜索推荐业务如何统一建模多场景异构业务；消息聚合范式下动态图、联合训练方向主要解决图新增节点、新增变化关系如何刻画，我们期望在动态图方向找到建模用户需求变化关系的方案。



相比传统深度学习引擎，图学习引擎需要具备图构建、图采样和图运算的能力。随着图技术发展越来越火热，图技术由学术界逐渐推广到工业界，引擎发展由支持图技术基本功能向更高效的支持大规模图方向发展。当前已有很多针对不同场景的开源图训练引擎 [8][9]。图学习业务场景的图模型规模越来越大，训练时间也越来越长，因此训练引擎 [8][9] 需要同时支持较大的图规模端到端训练和较快的训练速度。

在当前开源的框架中，单机的训练引擎可以发挥 GPU 的计算优势，但是存储有限，无法支撑业务 TB 级别内存和模型参数的大规模图学习训练任务。分布式的训练引擎可以通过横向扩展来支持大规模的图学习任务，但是优化多机图采样之间需要进行密集的通信造成瓶颈，使得各台机器都无法发挥 GPU 的计算能力，导致训练速度难以满足工业界需求。因此我们联合美团机器学习平台建设了一套图学习训练引擎，能够同时满足速度和规模两方面的需求。

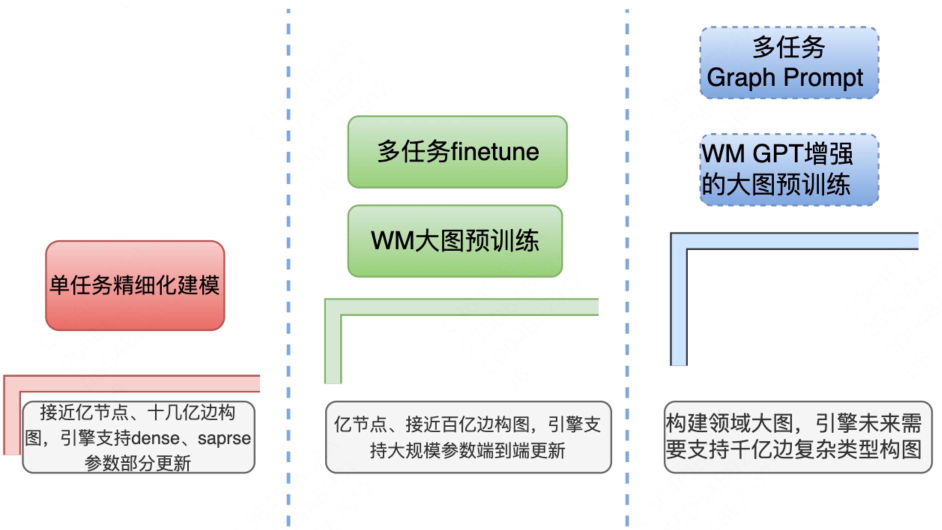
2 异构大图在搜索推荐业务的演进

我们提出多场景异构大图统一建模解决搜索推荐渠道多带来的迭代效率低、异构场景难以统一、小场景难以学好的问题。用户需求具有不同场景间相互比较，需求演变至逐渐收敛的特点，这种即时性的变化特点，我们以多场景异构大图为基座提出异构动态图在线建模刻化需求演变关系，如下阐述多场景异构大图和异构动态图在线建模的迭代演进。

2.1 外卖多场景异构大图

从业务逐步扩增、基建逐渐完善、技术逐渐发展的现状，我们多场景异构大图由单场

景精细化图建模 -> 多场景统一的大图预训练 + 下游任务微调 -> 联合 GPT 增强式检索的大图预训练 + 下游任务 Prompt 微调进行迭代，最终构建外卖领域 Graph 模型。



随着迭代的发展及数据规模的变化，图引擎的技术能力需要由支持小规模图快速迭代，到支持百亿边图规模、全参数端到端训练，最终实现支持千亿边规模领域大图训练能力的跨越。落地于搜索前导购渠道（搜索发现）、结果页【POI+SPU】组合推荐、结果页相关填充等多个场景，取得了较为明显的业务效果；在学术层面，相关论文已被 CIKM 2023 收录。

2.1.1 单场景图建模

基于 EM（Expectation Maximization）框架的单意图语言增强降噪图

背景：将之前的图神经网络直接应用于该异构图宽泛检索任务会遇到噪声交互。噪声交互主要来源于用户的随机误点（例如，在一个查询中共同点击“汉堡”和“沙拉”）以及全场景行为序列之间 Session（用户在搜索引擎中从开始到结束的连续行为）点击（例如，“肯德基”和“海底捞”），以及由于消息传递方案更容易受到噪声的影响。

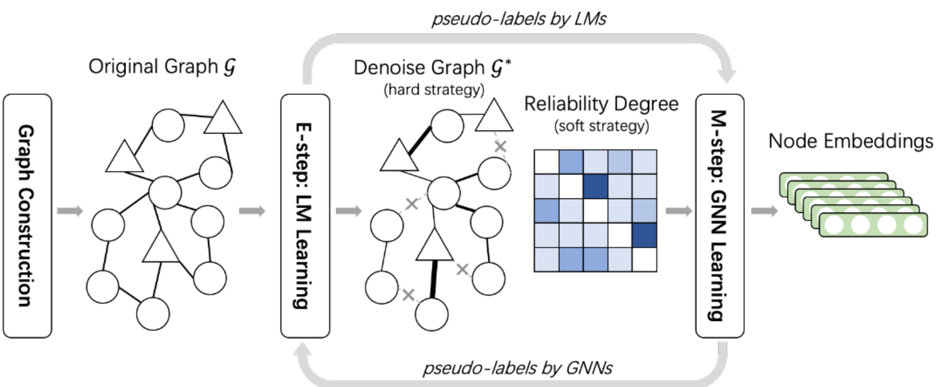
动作：之前工作主要聚焦于结构相似性或者基于规则的语义相似性降噪，不同层面存在稀疏表示和节点覆盖问题，因此我们提出基于变分 EM 框架进行 LM 和 GNN 联合训练，通过联合训练融合结构和语义信息进行图结构降噪。具体而言，在单意图去噪中，我们基于 LMs（(Language Models）估计每次图交互的可靠性程度，并基于可靠度为 GNNs(Graph Neural Networks) 设计了硬去噪和软去噪策略，如下公式所述，此外用变分 EM 框架将语言模型和图神经网络结合起来，以避免联合训练需要不可承受的计算成本，最终通过联合训练融合结构和语义信息进行降噪。

结果：EM 联合训练和软硬降噪（对比只有硬降噪图）带来离线 Recall +3.7%。

$$\mathbf{h}_v^{(k)} = \mathbf{W}^{(\mathrm{V})}\mathbf{h}_v^{(k-1)} + \sum_{u \in \mathrm{NB}(v)} \frac{\widetilde{r}_{u,v}}{\sum_{j \in \mathrm{NB}(v)} \widetilde{r}_{j,v}} \mathbf{W}^{(\mathrm{V})}\mathbf{h}_u^{(k-1)}$$

$$\widetilde{r}_{u,v} = \underbrace{\mathbb{I}\left(r_{u,v} \geq \gamma\right)}_{\text{hard}} \cdot \underbrace{\left(r_{u,v} + \frac{1}{\sqrt{d}}\left(\mathbf{W}^{(Q)}\mathbf{h}_v^{(k-1)}\right)^{\top}\left(\mathbf{W}^{(K)}\mathbf{h}_u^{(k-1)}\right)\right)}_{\text{soft}}$$

简化图神经网络的注意力头的数量为一（实际为多头），节点v的第k层表示$h_v(k)$通过邻居$\mathrm{NB}(v) \subseteq \mathrm{V}$表征聚合而来。其中$\mathrm{W}(Q)$、$\mathrm{W}(K)$、$\mathrm{W}(V) \in \mathrm{R}d \times d$是网络参数；$r_{uv}$作为硬降噪指标，通过$LMs$相似度得分计算得到边的可靠度；$\gamma$是硬降噪阈值，当可靠度得分小于阈值，聚合函数直接删除该邻居节点，当可靠度得分大于阈值，通过可靠度得分作为先验知识，和Masked Target-Attention共同决定邻居聚合的权重关系。



基于对比学习的多意图差异化建模

背景：将之前的图神经网络直接应用于该异构图宽泛检索任务会遇到意图不可区分性的问题。用户搜索词表达了多种多样的意图，对于同一个曝光卡片，具有不同意图的用户可能会关注不同部分（菜品、商家等），但是现有的图神经网络通常忽略意图之间的差异统一建模。

动作：我们提出多意图差异化建模，通过多意图对比学习方式解决之前忽视意图之间差异性问题。具体的我们在语言模型（LMs）中引入了意图感知节点，能够为同一个节点获得不同意图表示。GNNs 中通过设计聚合函数让每个意图节点更多地关注来自具有相同意图的边的邻居节点（公式如下）。最后提出了一个多意图对比学习目标（公式如下），以明确而有效地指导图模型显示建模不同意图的差异性。详细信息可以去阅读我们的论文 LEAD-ID[10]。

结果：多意图对比学习带来离线 Recall + 1.8%，多意图表征带来离线多业务平均 Recall + 3.8%。

$$\widetilde{\mathbf{h}}_{v^{(s)}}^{(k)} = \mathbf{W}^{(\mathrm{V})}\widetilde{\mathbf{h}}_{v^{(s)}}^{(k-1)} + \sum_{u \in \mathrm{NB}(v)} \frac{\widetilde{r}_{u,v}^{(s)}}{\sum_{j \in \mathrm{NB}(v)} \widetilde{r}_{j,v}^{(s)}} \mathbf{W}^{(\mathrm{V})}\widetilde{\mathbf{h}}_{u^{(s)}}^{(k-1)}$$

$$\widetilde{r}_{u,v}^{(s)} = \frac{\mathbf{w}_s^{\top}\mathbf{w}_{S(u,v)}}{\|\mathbf{w}_s\| \cdot \|\mathbf{w}_{S(u,v)}\|} \cdot \widetilde{r}_{u,v}$$

其中 $w\cdot$ 是参数，$S(u,v)$ 表示边 (u,v) 的意图；请注意，图神经网络（GNNs）只聚合相同意图的邻居的表示，即 $h_{u(s)}^{(k-1)}$。

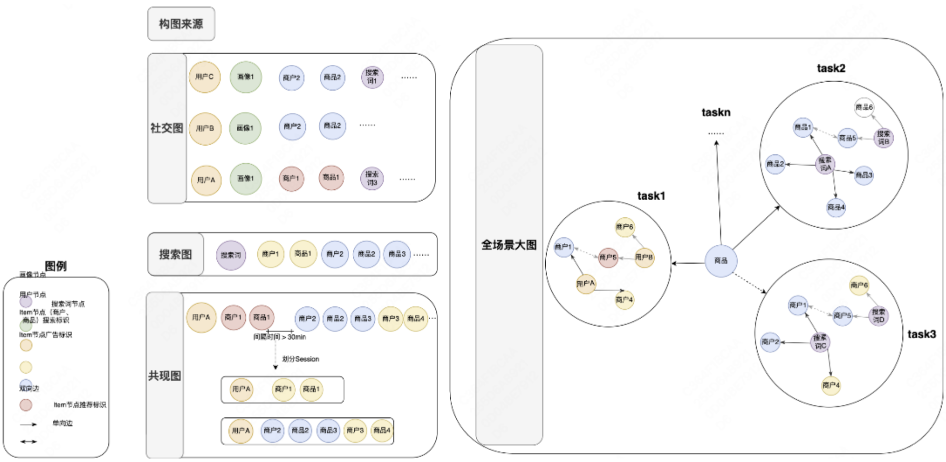
$$\mathcal{L}_{\mathrm{CL}}^{(\mathrm{item})} = \sum_{v \in \mathcal{V}_{\mathrm{item}}} -\log \frac{\exp(\mathbf{h}_{v(s)}^{\top}\mathbf{h}_{v(s)}^{+})}{\sum\limits_{\substack{s' \in S \\ s' \neq s}} \exp(\widetilde{\mathbf{h}}_{v(s)}^{\top}\widetilde{\mathbf{h}}_{v(s')}) + \sum\limits_{\substack{u \in \mathcal{V}_{\mathrm{item}} \\ v \neq u}} \exp(\widetilde{\mathbf{h}}_{v(s)}^{\top}\widetilde{\mathbf{h}}_{u(s)})}$$

对于节点表示$\mathbf{h}_{v(s)}$，我们通过dropout边和节点表征得到其正例$\mathbf{h}^{+}_{v(s)}$；为了鼓励模型区分各种意图，我们将不同意图的同一节点的表示视为强负例$\mathbf{h}_{v(s')}$，其中$s' \neq s$；同样，同一意图的不同节点被视为简单的负例。采用InfoNCE来最大化正对的一致性并最小化负对的一致性。

2.1.2 WM 多场景大图预训练

WM 大图构建

我们以外卖全场景作为数据源进行异构类型构图，实现一个大图支持多场景多业务。如下图所示，我们以用户画像、用户全行为序列、搜索点击序列 Session 内序列等为数据源进行大图构图；商品作为多场景共性连接节点，自定义业 Meta-path 作为单场景子图构建方法，构建具有实际任务意义的搜索商品子图、搜索商户子图、用户商品子图等。



其中图节点包 User、Item、POI、搜索词；边包括 User 点击、成 Item，搜索词点击、成单、加购 item、POI，用户序列 Item、POI 的 Session 内点击、成单等；大图整体规模亿节点、百亿边。

多场景统一大图预训练

背景：为了实现一个大图支持多场景多业务，提高迭代效率，我们在语义联合增强图降噪网络基础上进行统一多场景大图预训练。相比于上述单场景语言增强降噪图，大

图预训练主要挑战为如何进行多场景的语言模型和图模型预训练。

动作：语言模型采用 BERT 为 Base，采用底层多场 Share-bottom 共享，顶层异构节点差异化建模统一搜索推荐多个场景，获得多种类型节点表征。统一大图预训练阶段无差异性高阶聚合所有邻居节点必然带来噪声干扰，因此我们通过自定义场景 Meta-path 显示定义场景子图，多场景子图内进行高阶聚合、多场景子图间底层共享节点表征。模型以无监督链接预测任务作为目标，通过 LMs 和 GNNs 联合训练进行统一大图预训练任务。

结果：优化多任务样本混合比例离线多任务平均 Recall + 4%。

2.1.3 生成式模型增强的大图预训练、Prompt 微调

背景：上述统一多场景大图预训练 +Finetune 范式主要有几个问题，首先预训练任务和下游任务之间固有的训练目标有差距，导致预训练无法最大化发挥能力，其次此范式下每个任务都需要大量样本有监督训练，微调成本高且新任务泛化能力弱，在 Prompt 范式之前，多场景训练方法集中在模型框架结构优化，设计复杂且可迁移性弱，因此借鉴 GPT 新范式设计图领域统一多场景模型。

动作：生成式模型实现语义理解模型具有统一多场景任务设计简单、可迁移性强等优点，因此通过生成增强检索（GAR）方式进行搜索推荐多场景语义模型设计，然后通过 GAR 生成式检索模型和 GNN 联合训练进行统一大图预训练任务。具体而言，GAR 通过底层共享基于开源模型领域微调后的模型为基座、以对比学习为目标设计双塔结构、多场景多样 Prompt 设计样本结构，以 SFT 方式进行多场景任务训练实现搜索推荐多场景语义模型；如上所述，大图预训练阶段通过自定义场 Meta-path 显示定义场景子图，多场景子图内进行高阶聚合、多场景子图间底层共享节点表征，模型以无监督链接预测任务作为目标，最后 GAR 和 GNN 联合训练实现统一大图预训练任务。下游设计多场景 Soft-prompt 进行 SFT，具体 Soft-prompt 初始化向量进行表示，通过融合预训练节点表征 Soft-prompt 表征作为最终节点表征，多场景以训练少量参数、小样本进行下游任务微调。

结果：相比于多任务 BERT，GAR 带来所有任务离线指标上涨，多任务平均 Recall +1%；zero-shot 评估下游任务，soft prompt 微调（对比不进行下游任务微调），下游多任务平均 Recall +10%。

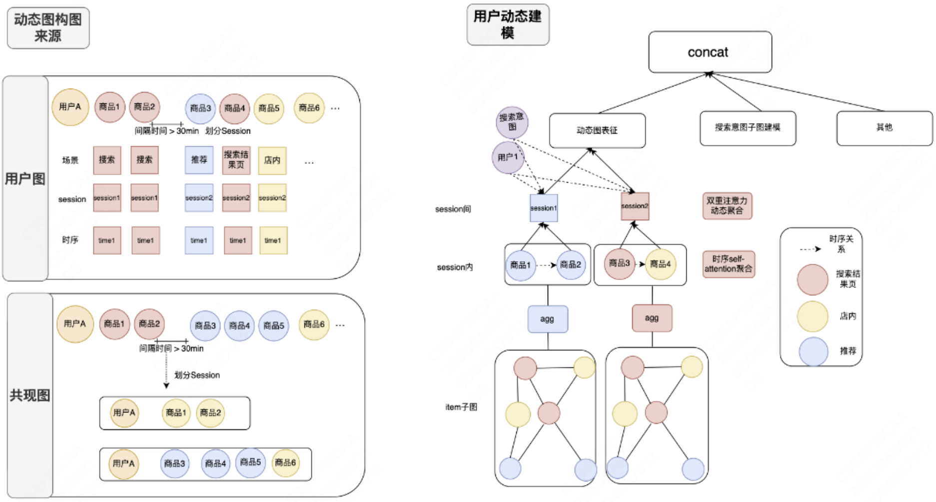
2.2 异构大图在线建模

由于用户需求变化关系有即时性、场景间相互比较逐渐收敛的特点，因此我们基于多场景异构大图建设图在线引擎，通过图在线建模完成用户与供给的高效匹配，提高流量使用和用户搜索效率，业务收益取得了较为明显的效果。

用户需求变化的动态图建模

背景：考虑用户需 Session 之间兴趣独立、Session 内部用户在不同场景间相互比较，需求演化至逐渐收敛的特点，提出基于动态图的用户 Sessionlevel 建模刻化用户需求的变化关系。

动作：Sessionlevel 建模加剧了序列的稀疏性、加大了表征难的问题，我们利用图的高阶聚合能力，沿用之前“软硬降噪”聚合函数，通过高阶聚合操作丰富序列中所有节点的表征能力。Sessionlevel 分为 Session 内部建模和 Session 间建模，Session 内部场景拆分为推荐、搜索中、搜索后，通过基于场景的时序 Self-attention 建模需求演化关系，Session 间基于当前实时搜索意图、用户信息双重注意力动态聚合，整体建模用户需求。用户搜索场景下搜索词表达用户即时意图，因此我们在上述语言增强降噪预训练图的基础上，基于搜索词和候选商品关系、商品共现关系构建搜索商品子图，为用户召回精确候选；最终搜索子图表征和动态图表征进行融合，整体结构如下图所示：



结果：用户 Sessionlevel 建模离线 Recall + 1%。

3 大规模图引擎 GraphET 工程建设

3.1 大规模图引擎训练框架建设

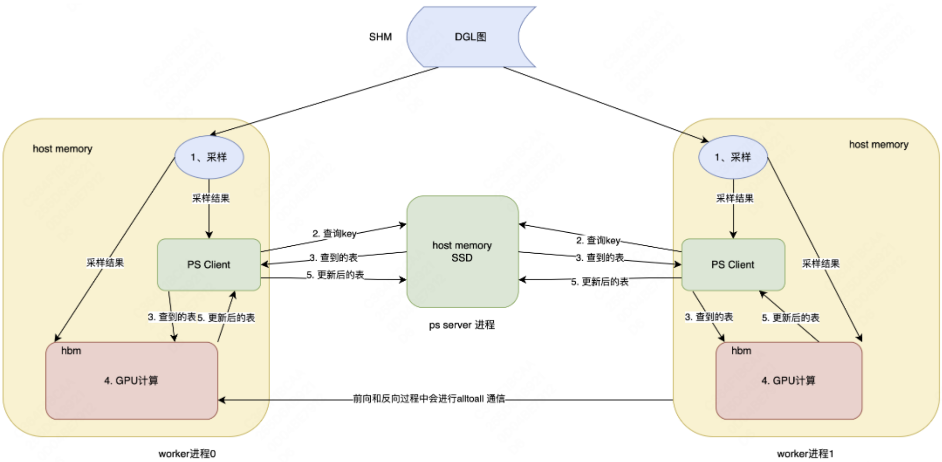
图学习业务场景的图模型规模越来越大，业务已经迭代到了几亿节点百亿边的规模，以 10 亿节点、100 亿条边的图模型为例，图结构本身采用 COO 格式保存在内存中，要占约 100GB 的内存（10GB*4*2 + 1GB*8）。在采样过程中随机游走会用 CSR、CSC 两种格式保存中间结果，以及训练过程中的内存占用，内存占用已经有了 300GB。

每个节点中还有用户定义的特征，以一个 256 维的节点特征为例，10 亿个节点总共需要 256* 4*1GB = 1TB。节点通常不会只有一类特征，边上也会有各种维度的边特征，这样的图规模常见集群中的 1TB 内存的无法保存。为了保证业务效果，节点和边的 Sparse、Dense 特征需要和模型参数进行端到端全量更新，TB 级别参数 GPU 训练更新开源图学习框架不支持。

因此我们在开源的图学习训练框架 DGL（Deep Graph Library）v0.7 基础上，研发

了一套大规模图神经网络的训练框架 GraphET，服务于公司多个业务线。该框架支持亿级别节点、百亿级别边离线图训练流程高效 pipline（图构建 / 采样 / 聚合 / 端到端建模）Pytorch Dgl Serving 在线向量计算，方便实现学术界任意复杂图模型工程在线化。

GraphET 训练系统的架构如下图所示：



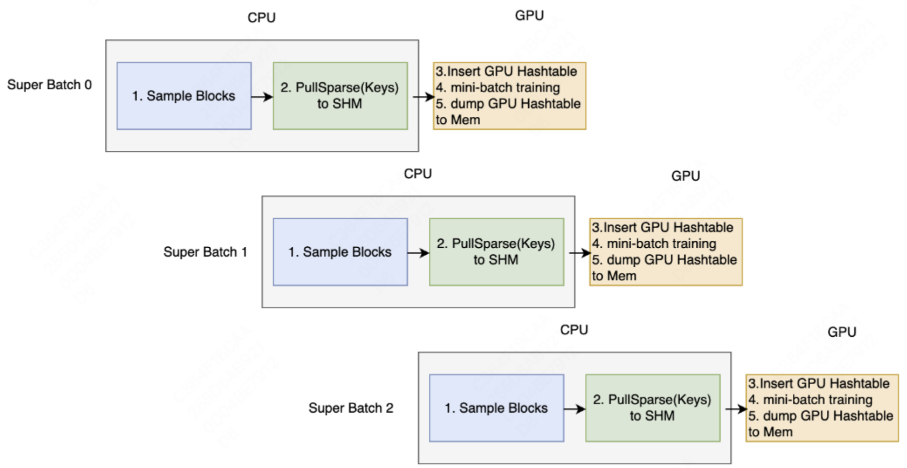
系统由负责模型训练的 Worker 进程和负责 Hashtable 保存的 Parameter Server 进程两部分构成。为了降低内存开销，将 DGL 图结构存到共享内存中，在多个 Worker 进程间共享同一份图结构。图中的节点和边上的特征保存在 Parameter Server 中，每次采样后会向 Parameter Server 发送需要查询的节点，将查询到的 Embedding 放入 SHM。Mini-batch 训练前将 Embedding 加载到 GPU 上，训练过程中用 alltoall 通信来获取节点 / 边特征，训练结束后将 Embedding 写回 PS 完成更新。系统支持显存 / 内存 /SSD 多级存储，根据特征的访问频次来将特征放置在合适的位置，在不影响系统吞吐的情况下，提高了 DGL 可以支撑的图的特征规模。

worker 进程

在我们设计的架构下，模型训练过程中涉及 Super-batch 粒度的训练样本采样、样本特征查询、Mini-batch 粒度的 GPU 训练和特征更新，不同阶段对硬件特点的需

求是不同的，具体来说对为了充分发挥不同硬件的功能，最大化利用 GPU 的计算优势，提升模型整体训练速度，我们通过三级流水线来加速模型训练。

- 训练样本采样是 CPU 密集型任务；
- 样本特征查询是 SSD IO 密集型任务；
- GPU 训练是计算密集型任务。



在流水线中，每个 Super-Batch 都包括采样、获取特征、训练三个阶段。样本采样阶段是独立的，采样结果放入 Queue 中；获取特征阶段由 PS Client 向 PS 发送异步请求拉取特征参数放入 SHM；训练阶段阶段将特征放到 GPU 上，训练后将新的 Embedding 写回 SHM。多级流水线之间通过消息队列和共享内存通信。

worker 进程对重复查询 Embedding 做了两方面优化：

- 采样后，在查询特征前会对多 GPU 采样出的 Key 进行去重。由于 Worker 进程一个 Super-batch 采样多个 Mini-batch，邻居较多的节点可能会被重复采样，去重后每个 key 在 PS 端仅查询一次；
- 每个 Mini-batch 训练时，所有 Key 按照 Key%Worker_num=i 的方式存储在 Worker i 对应的显存中，GPU 进程间 alltoall 通信前会对 key 去重以减少卡间通信。

PS 进程

PS 主要负责 PS 负责存储、查找和更新 Embedding 参数，支持两种存储方式：Full_memory 和 Ssd_kv_store。在 Full_memory 模式中所有的参数都是存在内存中，这相当于将参数存储在 SHM 中。在 Ssd_kv_store 模式中，所有的参数都存在 SSD 中，内存作为 SSD 的 Cache 仅存储部分参数，这种方式可以存储更多的参数，但需要考虑 Cache 命中率，避免内存中存储的参数太少，导致 SSD 读写速度成为性能瓶颈。

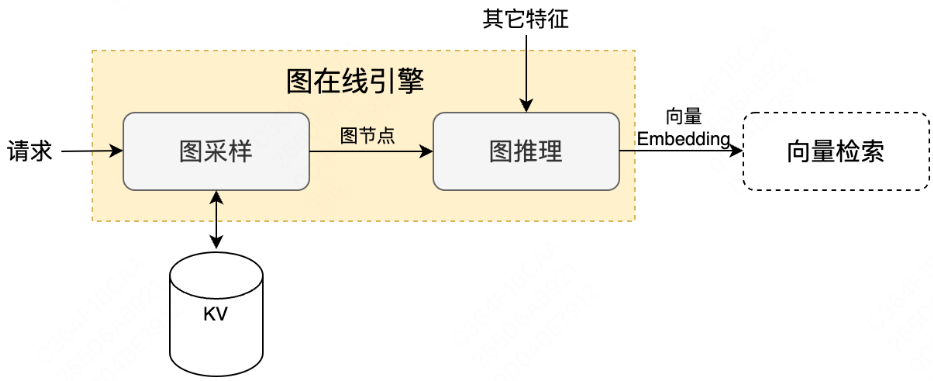
PS 以 KV 形式存储 Embedding 参数，使得 Embedding 参数在 PS 和 Worker 进程中的 PS Client 之间共享。为了优化内存使用效率，将所有 Hashtable 的 KV 对统一存储在一块大的共享内存中，内存中的 Hashtable 中存储指向共享内存中对应 Value 的指针（Offset）。

我们在 SSD 引擎方面做了多方面的优化：

- **SSD 聚合读优化**。SSD 上的 Key 查询是以 Group 为单位进行数据读取，而查询 Key 的分布很随机，导致读到 PageCache 的 Group 数据被频繁换入换出，影响查询性能。因此，我们将待查询的 Key 集合按照 Group 进行提前聚合，聚合后再进行 SSD 查询，一方面降低 I/O 读取次数，另一方面也能更好利用 PageCache 来提升查询性能。
- **对象池优化**。在 Key 查询过程中，需要频繁创建小对象（Cache 结点、Block 结点等），虽然底层已使用 TCMalloc 优化，但内存分配释放的开销仍不容小视。因此，我们引入定长对象池，在连续大内存上维护小对象的分配和释放操作，减少系统调用，提升服务性能。
- **文件 GC 优化**。由于 Compaction 操作，SSD 文件可能包含很多无效 Group 数据，但只有文件中 Group 全部为无效状态时才会触发文件删除，导致有效 Group 占比很低的文件迟迟得不到删除，占用磁盘空间，对 SSD 读写性能也产生影响。因此，我们引入异步 GC 线程，定期合并有效 Group 占比低的文件，删除无效文件，降低磁盘占用。

3.2 图引擎在线框架建设

随着图训练引擎支持大规模图落地，图节点和边变化关系更新、实时新增图节点、实时预测图表征能力成为制约业务效果的瓶颈。因此基于图模型离线训练流程，建设图在线引擎。图在线引擎建设包括两部分内容：图采样和图推理，如下图所示：



- **图采样**：将图模型训练过程中用到的多跳图节点，进行整合拼装后写入 KV Serving，提供高效图采样（后续会迁移至图数据库，实现实时采样）；
- **图推理**：将图采样节点以及其它特征输入到图模型中，进行在线前向推理，输出向量 Embedding 用于后续的向量检索召回。下面也将重点介绍我们在图推理方面的相关建设工作。

图推理遇到的挑战

Python 在线推理：图模型基于开源 DGL 框架进行训练和导出。虽然 DGL 框架支持 Pytorch 和 Tensorflow 两种 backend，但 Pytorch 相比 Tensorflow，无论是新功能特性的迭代效率方面，还是公司训练平台的支持方面都更加突出，因此在线推理部署的图模型是基于 DGL+Pytorch 的模式进行训练和导出。

Pytorch 本身是支持将模型序列化成 TorchScript 格式，进行 C++ 部署和推理加速，但 DGL 框架是基于 Pytorch 进行二次开发，无法序列化成 TorchScript 格式进行 C++ 部署，只能通过 Python 部署的方式进行推理，这就需要在现有 C++ 推理

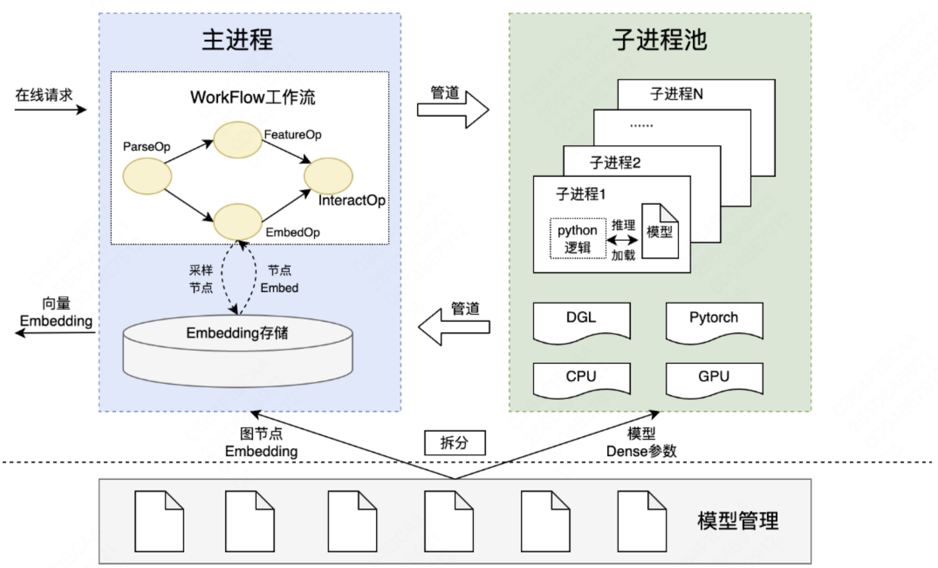
框架的基础上进行底层能力升级，支持 Python 部署模式的 backend，这对框架的 WorkFlow 推理流程、模型管理模式、进程部署方式等方面都是不小的挑战。

单机显存瓶颈：Python 由于全局解释器锁 GIL 的限制，导致单进程模式无法并行处理请求，一方面导致多核 CPU/GPU 无法被充分利用，资源被浪费，另一方面请求被串行积压，导致耗时上涨，这对于在线推理服务是不能接受的。

因此，为了避免 GIL 锁的影响，需要通过部署多进程的方式进行模型推理，支持在线请求的并发处理。但多进程部署方式，需要每个进程都加载一份模型数据，这无疑会受到单机显存的约束，模型越大，单机可部署的进程数就越少，进而限制处理请求的并发度，影响在线推理性能。因此，如何降低单进程可加载的模型数据量，提高并行部署的进程数量，是我们需要思考的问题和挑战。

图推理框架建设

针对上面梳理的问题和挑战，并结合业务现状和系统现状，我们进行了在线图推理框架的建设，系统架构如下图所示：



从上图可以看出，在线图推理框架由 1 个主进程 + N 个子进程组成，主进程负责 WorkFlow 工作流的调度，包括在线请求接收、解析、特征 / 图节点 Embedding 数据准备以及与子进程间的数据交互，最终返回向量 Embedding 结果；子进程负责以 Python 的方式进行模型的加载和推理，并将推理结果返回给主进程。主进程每次会从子进程池中选取空闲子进程，并通过管道进行通信。

多进程架构：解决 Python GIL 锁造成的单进程 CPU/GPU 利用率低的问题

将 Python 执行逻辑部署在多个进程中，通过单进程内串行执行请求，可有效避免 Python GIL 锁带来的限制，通过进程间并行处理请求，可充分利用 CPU/GPU 多核资源，提升服务性能和吞吐。主进程和子进程池之间，交互流程类似于“生产者 – 消费者”模式，通过引入管道、epoll 等机制，保证进程间通信高效执行。

模型拆分：解决模型过大造成的单机显存对子进程数量限制的问题

图模型包括亿级节点和几十亿条边，模型大小在几十 G 左右，默认全部加载到 GPU 中。考虑到模型加载后会出现膨胀现象，实际占用的 GPU 显存会更大，而 GPU 显存资源有限，加载单个模型都会存在显存溢出风险，很难支撑多进程加载多模型的模式。

经过分析，我们发现模型结构中存储了大量图节点 Embedding 数据，而图模型网络 Dense 参数只占百兆左右，同时发现单机内存大小要远大于 GPU 显存，且处于空闲状态。因此，我们在离线侧将图模型进行了拆分，将图节点 Embedding 部分加载到主进程内存中，且只需加载一次，而将模型 Dense 参数加载到 GPU 显存中，虽然每个子进程都需加载一份，但 Dense 参数体量较小，单个进程占用显存可控，可大幅提升子进程部署数量。

统一通信协议：解决不同策略模型的低成本快速迭代问题

不同策略模型对特征 / 采样 Embedding 的处理方式都有所不同，如果放在框架层进行适配，时间成本和人力成本都很高，影响模型的快速迭代。因此，我们制定了主进程 –> 子进程 –>Python 逻辑全流程的统一通信协议，通过标准化、规范化的通信数据格式，将特征 / 采样 Embedding 数据逐层传输到子进程 Python 逻辑中，而子进

程 Python 逻辑中才会真正执行模型定制化逻辑，算法同学可以按需修改，并作为模型的一部分被子进程加载，从而保证在服务框架层面稳定不变的情况下，动态支持不同策略模型的快速迭代。

4 总结和展望

图神经网络作为图结构数据建模方法，在搜推广领域展现出巨大潜力，业界头部公司均结合各自业务特点自建图引擎和图技术落地应用。

本文主要介绍大规模图框架在外卖广告场景的落地。基于对外卖搜索广告场景分析，提出搜索推荐业务解决 LBS 场景下弱供给问题。搜索推荐业务面临着多渠道、即时化的挑战。我们提出多场景异构大图，通过单场景精细化建模 -> 大图预训练 + 下游任务微调 -> 大图预训练 + 下游任务 Graph Soft Prompt 解决多渠道问题，异构图在线建模通过基于 Sessionlevel 的动态图建模用户需求变化关系。

为了满足亿节点百亿边大规模图端到端训练、在线实时推理，基于开源 DGL 框架研发了一套大规模图神经网络的训练、推理框架 GraphET，支持离线图训练流程 Pipline（图构建 / 采样 / 聚合 / 端到端建模），DGL Serving 在线推理，方便实现学术界任意复杂图模型工程在线化。

未来我们还将在以下方向继续进行探索：

- 借鉴 GPT 思想，搜推广领域通用 Graph 模型建设及落地；
- 构建领域大图，引擎需要支撑千亿边、复杂类型构图能力；
- 图在线引擎加速及支撑更大规模图在线推理框架建设。

5 参考资料

[1] Daniel C Fain and Jan O Pedersen. 2006. Sponsored search: A brief history. Bulletin-American Society For Information Science And Technology 32, 2 (2006).

[2] Grover A, Leskovec J. node2vec: Scalable feature learning for networks[C]// Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining. 2016: 855-864.

[3] Procopio L, Tripodi R, Navigli R. SGL: Speaking the graph languages of semantic parsing via multilingual translation[C]//Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. 2021: 325-337.
[4] Lu Y, Jiang X, Fang Y, et al. Learning to pre-train graph neural networks[C]//Proceedings of the AAAI conference on artificial intelligence. 2021, 35(5): 4276-4284.
[5] Velickovic P, Cucurull G, Casanova A, et al. Graph attention networks[J]. stat, 2017, 1050(20): 10-48550.
[6] Han H, Zhang M, Hou M, et al. STGCN: a spatial-temporal aware graph learning method for POI recommendation[C]//2020 IEEE International Conference on Data Mining (ICDM). IEEE, 2020: 1052-1057.
[7] Sun X, Cheng H, Li J, et al. All in One: Multi-Task Prompting for Graph Neural Networks[J]. 2023.
[8] Wang M Y. Deep graph library: Towards efficient and scalable deep learning on graphs[C]//ICLR workshop on representation learning on graphs and manifolds. 2019.
[9] Lin Z, Li C, Miao Y, et al. Pagraph: Scaling gnn training on large graphs via computation-aware caching[C]//Proceedings of the 11th ACM Symposium on Cloud Computing. 2020: 401-415.
[10] Zhou X, Wang R, Li H, et al. LEAD-ID: Language-Enhanced Denoising and Intent Distinguishing Graph Neural Network for Sponsored Search Broad Retrievals[C]//Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 4460-4464.
[11] Sun X, Cheng H, Li J, et al. All in One: Multi-Task Prompting for Graph Neural Networks[J]. 2023.

大众点评内容搜索算法优化的探索与实践

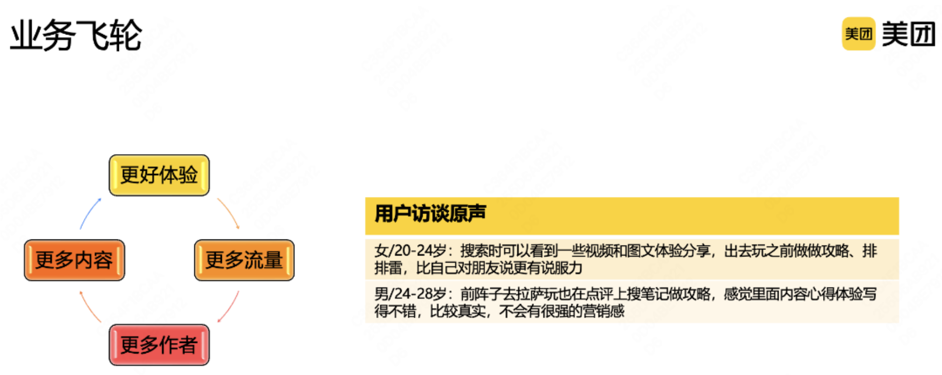
1 现状与挑战

美团在本地生活服务领域深耕多年，在帮助用户完成交易的同时，积累了丰富的图文视频内容供给。依托于这些内容供给，我们可以满足用户更丰富的需求类型，从交易环节扩展到交易前的种草、交易后的体验分享环节，将大众点评建设成为本地吃喝玩乐的社区。

在大众点评的用户中，有相当高比例会通过搜索来查找本地信息，而内容搜索是辅助用户决策、促进社区氛围的重要工具。例如当用户搜索“火锅”时，除了能看到火锅相关的商户和团单，还可以看到图文、视频、评价、笔记等多种形态和类型供给呈现；搜索“圣诞节活动”时，直接以双列内容形式呈现搜索结果，可以更加生动形象。

通过持续优化内容搜索体验，可以带来更多内容消费流量，进而吸引更多的用户转化为作者，激励创作出更多的内容，而有了更多的内容之后，又可以进一步带动体验提升，最终形成一个良性循环。从实际效果来看，内容搜索的价值也得到了用户的认

可，如下图是用户访谈原声，可以看到通过内容搜索结果逐步拓展了用户对搜索功能的认知。



内容搜索与典型类型的搜索如网页搜索、电商搜索、商户搜索等相比，有如下差异点：

- 在优化目标上，网页搜索更强调搜索满意度，电商搜索更看重商品交易总额，商户搜索更关注用户到店消费意向率，而内容搜索既要考虑搜索满意度，又要考虑点击和点击内容后的停留时长、点赞 / 收藏 / 转发 / 评论等交互行为。
- 在地域约束上，网页搜索和电商搜索没有特别强的地域限制，而商户搜索和内容搜索却有非常强的 LBS 区域限制，因为用户在美团点评的搜索场景下更希望查找附近的商户和内容。
- 在供给类型上，网页搜索、电商搜索、商户搜索结果类型较为单一，而内容搜索有非常多的类型，比如笔记、评价、旅游攻略、菜谱等。
- 在结构化程度上，电商搜索和商户搜索相对较高，因为有商家和销售维护相应信息；网页搜索一般结构化程度比较低，可被检索的网页大部分信息是非结构化的；内容搜索的供给中既包括图片、视频、文本等非结构化信息，也有内容关联的作者、商户、关联话题等结构化信息，整体呈现半结构化的特点。
- 在供给规模上，电商搜索和商户搜索供给量级相对可控，因为商品、商户的生产维护成本较高；而网页搜索和内容搜索的供给生产成本低，规模会相对更大一些。

- 在更新频率上，一个商品从上线到下架、一家店从开业到关停，需要相当长的时间周期，而内容和网页生产和更新频率都更快一些。

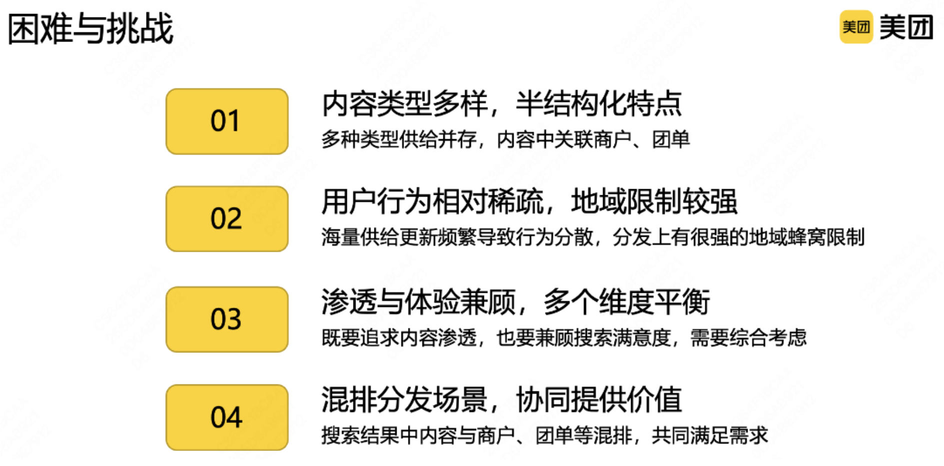
从以上对比来看，内容搜索在各个维度上与典型的搜索类型存在很大区别，这就需要结合自身特点，进行相应的技术选型和方案设计。

行业对比

搜索类型	网页搜索	电商搜索	商户搜索	内容搜索
优化目标	搜索满意度	商品交易总额GMV	意向率	满意度/渗透/交互/时长
位置约束	弱	弱	强	区域限制强
供给类型	单一	单一	单一	笔记/评价/攻略/菜谱...
结构化程度	低	高	高	图片/视频/文本 作者/关联商户/话题/活动...
供给规模	大	中	小	大
更新频率	快	中	慢	快

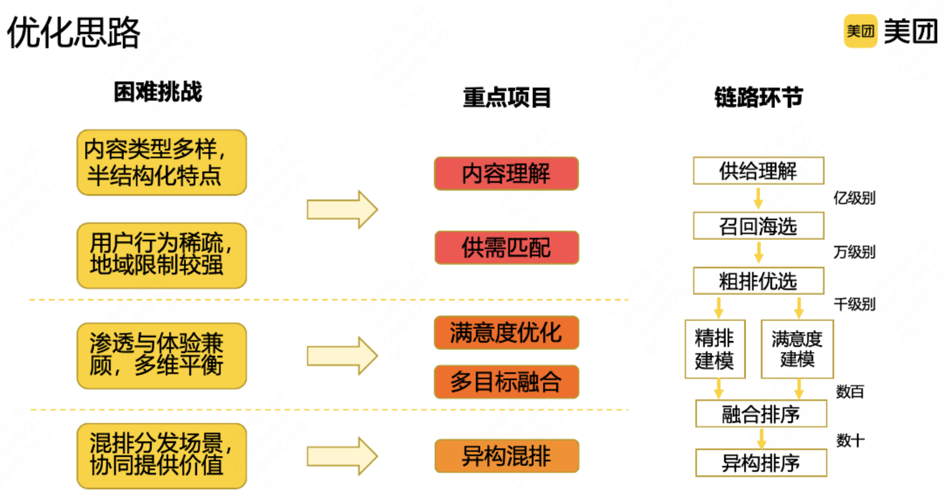
我们对面临的困难挑战进行总结，主要包括以下四个方面：

1. 多种类型供给并存，且供给中既有结构化的信息，又有非结构化的信息。
2. 内容供给量级大且更新频繁，导致用户行为分散，单篇内容较难获取到足够的用户行为数据；在分发过程中又有较强地域限制，形成类似蜂窝状的消费规律，进一步加剧了用户行为稀疏的问题。
3. 在优化过程中既要拉动内容消费指标，也要兼顾搜索满意度，在推进中需要综合平衡多个维度。
4. 在最终搜索结果中，内容与商户、团单等以混排形式呈现，需要与其他类型搜索结果协同发挥价值，共同满足用户需求。



2 内容搜索优化实践

下面我们会从面临的问题和挑战出发，分享如何通过链路各环节，持续优化内容搜索的体验。

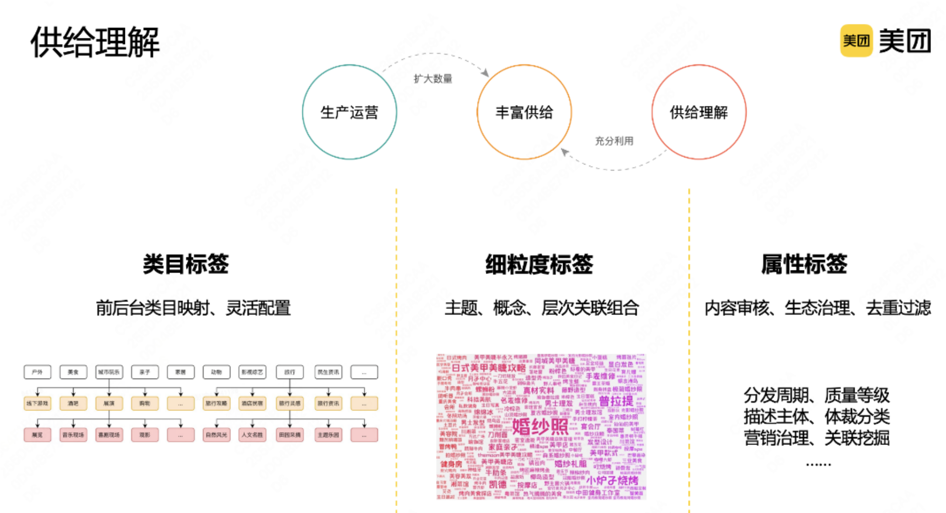


2.1 供给理解

面对用户持续创作生产的海量内容，我们需要对其进行充分理解，包括显式标签和隐

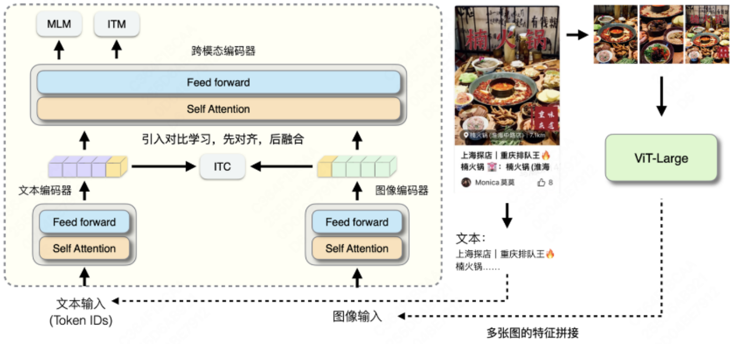
式表征两部分工作。显式标签体系主要包括：

- **类目标签**：通过构建分发前台后台两套标签，可以实现前后台类目灵活映射。当需要进行前台类目体系调整时，可以通过调整映射层快速支持，减少对后台打标任务的影响。
- **细粒度标签**：类目标签个数有限，在推荐搜索等分发场景还需要更细粒度的刻画，为此构建主题标签、概念标签等，相互之间有一定的关联和组合关系。
- **属性标签**：前两类标签更多关注内容在讲什么，而属性标签更侧重于刻画内容本身是什么，比如是否涉政涉黄、是否重复、是否命中生态治理等。



除了显式标签，分发链路很多环节还需要更加泛化的隐式表征。结合实际场景特点，我们自研了多模态预训练模型，通过引入对比损失把图文表征对齐到统一特征空间，并结合自监督对比学习训练范式、掩码学习、图文匹配等优化，提升了跨模态交互效果。

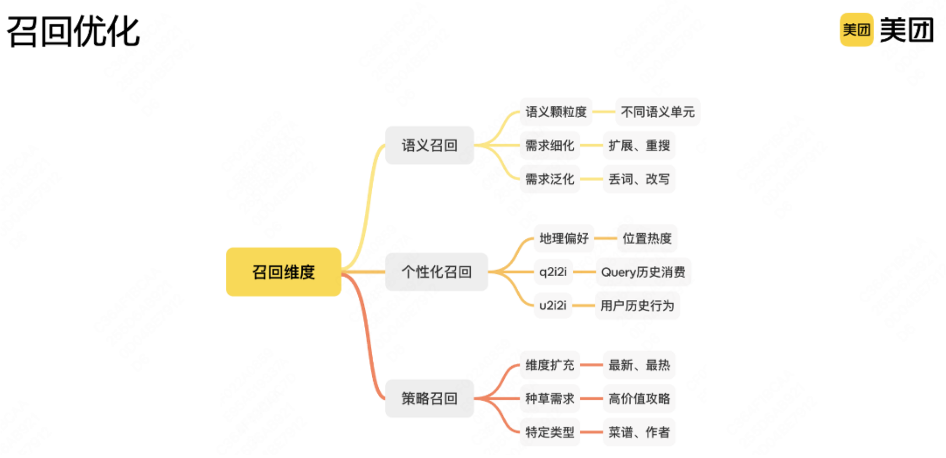
供给理解



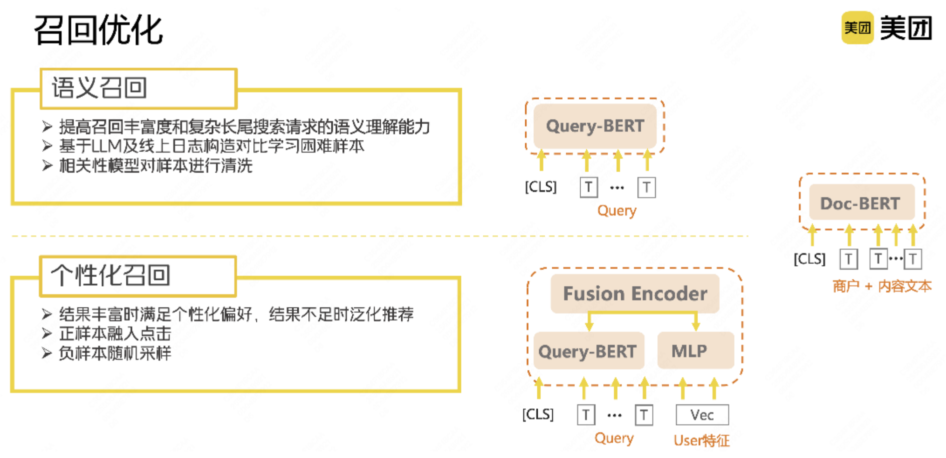
2.2 召回环节

作为最前置环节，召回决定了一次搜索查询所能拿到的候选总集合，直接影响到后续环节的效果天花板。搜索场景的召回主要包括：

- **语义召回**：搜索召回需要首要保证结果相关，为此对语义召回进行了多维度的设计，包括不同颗粒度的语义单元召回、对用户需求进行细化和泛化处理。
- **个性化召回**：结合用户地理偏好、特定区域偏好与用户历史消费内容相似度等，设计召回通路满足个性化需求。
- **策略召回**：基于用户不同场景的实际需求设计对应策略，包括最新最热内容的召回、更符合种草需求的高价值攻略召回、定向搜索作者内容或特定类型如菜谱的召回等。



其中语义和个性化召回有很大部分通过隐式实现，语义召回更侧重搜索词自身信息的刻画，而个性化召回还融入了用户偏好、上下文等很多信息。



2.3 排序环节

排序包括粗排、精排、多目标融合排序、异构混排等多个环节，随着逐层筛选，打分量级依次减小，可以使用结构更复杂、规模更大的模型。

介于召回和精排之间的粗排环节，需要兼顾准确性和全面性、权衡打分能力和时延性能，发挥承上启下的作用。为此引入用户在全域的行为样本，达到系统层面的纠偏

作用；我们通过表征蒸馏、分数蒸馏和顺序蒸馏等方法，提升模型表达能力；在常见 Query-Doc 双塔结构基础上，引入交叉塔（如交叉点击率、时长等），提高特征交互能力。

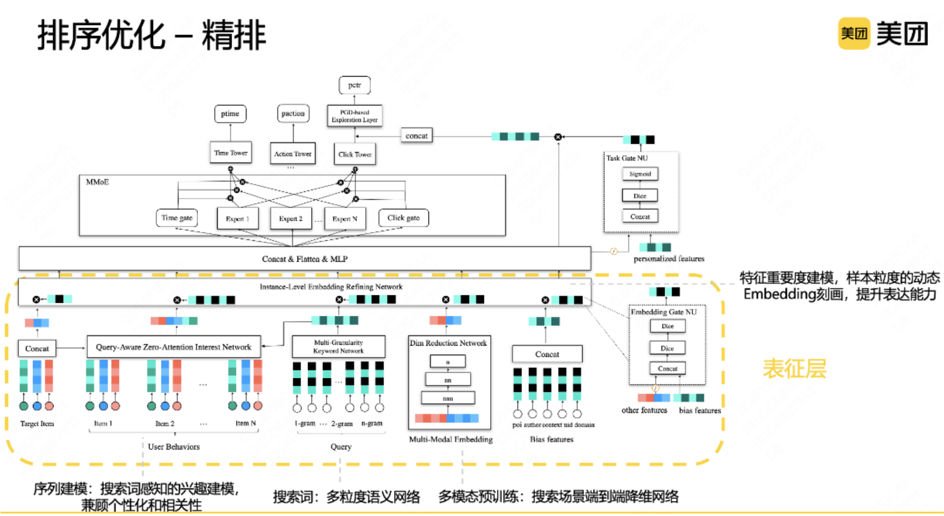
精排环节着重介绍在输入表征层、多目标建模层和输出层的相关工作。

首先是模型输入表征层，为了准确刻画 Query、用户、Doc、上下文等多种维度、各种粒度、各种来源的输入信息，我们从以下几个方面进行表征。

- **Query 语义表征**：搜索场景下 Query 是用户需求的直接表达，借鉴向量检索的工作，对 Query 进行了不同粒度的刻画，通过多粒度语义网络进行搜索词表征。
- **用户序列表征**：引入用户全站行为序列，捕捉用户长短期个性化偏好。搜索场景需要兼顾个性化和相关性，但用户历史行为和当前搜索词不一定存在关联，为此在主流建模方案 DIN 基础上，引入零向量注意力机制来权衡个性化和相关性。具体来说，引入了 Query 语义表征，对长尾低频 Item 做过滤，帮助模型决策哪些历史行为和当次搜索词相关，且在历史行为和搜索词无关时不引入额外的噪声。
- **多模态表征**：图像、摘要等创意维度信息，对于用户决策至关重要，也是内容高效分发的基石。为此引入高维的多模态预训练向量，并结合场景进行端到端

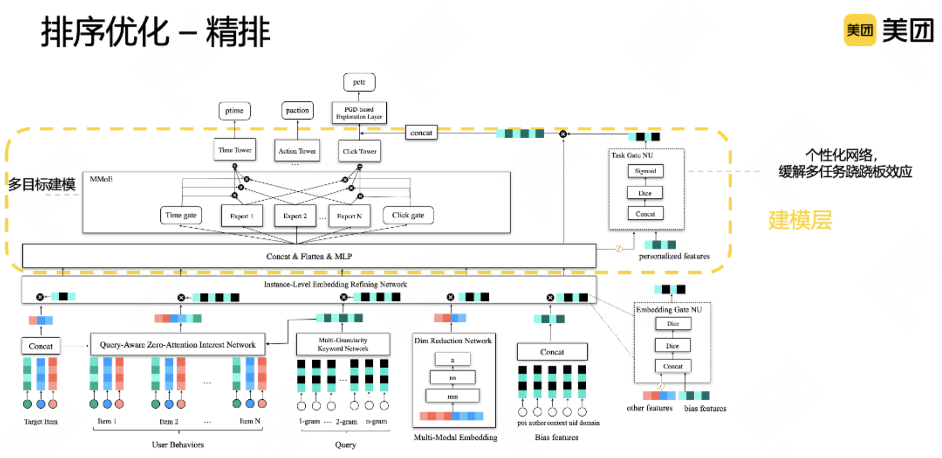
降维，既引入了丰富的多模态语义信息，又能够兼顾线上时延，对于刻画用户的多模偏好、提升新内容高效分发至关重要。

- **特征重要度建模**：通过动态权重的建模范式，捕捉样本粒度的动态表征，可以有效增强模型的表达能力。通过在 EPNet、MaskNet 等模型结构基础上，结合场景特点设计域感知的多门控网络、并联结构，实现了特征重要度的动态建模。



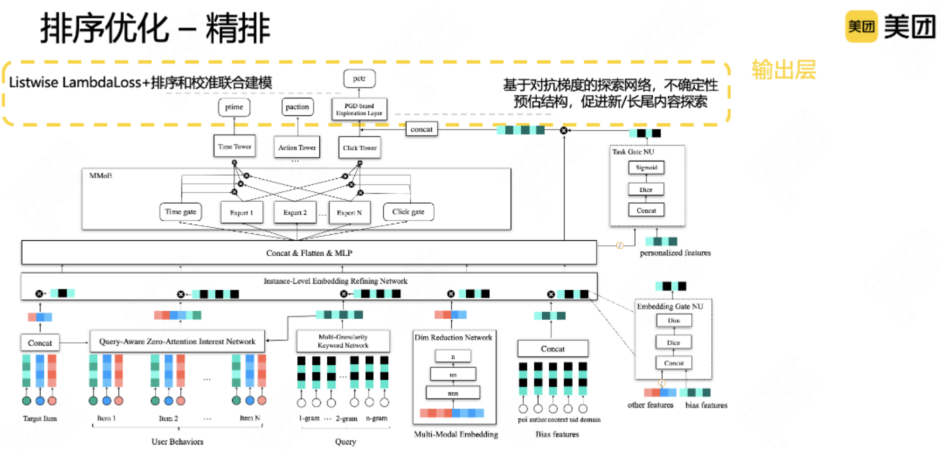
接下来是多目标建模层，由于点击、时长、交互等各个目标行为量级不同，导致优化过程中很容易出现跷跷板问题，为此在模型结构、优化方式等方面进行相关探索。

- **模型结构**：我们采用 MMoE 和 PPNet 融合的方案，为了防止 Gate 极化现象，对门控网络结构上进行 dropout、设计 skip connection 等；在各个任务上会引入个性化因子，通过个性化网络 PPNet 建模，MMoE 和 PPNet 的输出会拼接后传到预估输出层。
- **优化方式**：底层稀疏 Embedding 很容易受到各个多目标梯度反传的影响，造成梯度冲突，从而引起指标跷跷板问题。为此针对重要的表征增加参数量或新增任务特定表征，并对重要表征控制梯度反传，时长或交互目标不更新底层部分 Embedding 或更新时设置较小学习率。



最后是模型输出层，为促进新内容、长尾内容分发，并保证模型输出的预估分的稳定性和准确性，我们从探索结构和学习目标上进行了对应优化。

- **探索结构**：搜索场景消费内容个数比推荐少，马太效应问题也更加严重，对行为积累不够充足的新内容或长尾内容，预估不够准确。为此设计全链路冷启和探索通道，并基于不确定性预估范式，在模型中引入基于对抗梯度的探索网络，基于 CTR 预估的不确定性和对抗梯度在输入侧做扰动和探索。
- **学习目标**：之前搜索场景采用的学习目标是 Listwise 的 Lambdaloss，在排序能力上优于 Pointwise，但预估准确性上不足，会造成后续链路无法使用预估分。业界有不少研究关于 Listwise 损失如何做预估校准，例如 KDD 2023 中阿里巴巴校准工作 JRC、CIKM2023 中 Google 校准工作等。参考相关工作并结合场景特点，在原有的 LambdaLoss 基础上增加用于校准的 Logloss，在梯度更新上控制校准 Loss 不影响底层的 Embedding 更新，只更新多目标建模层和输出塔的参数，提高预估分数的稳定性和准确性，方便后续融合、混排等环节使用。



2.4 满意度优化

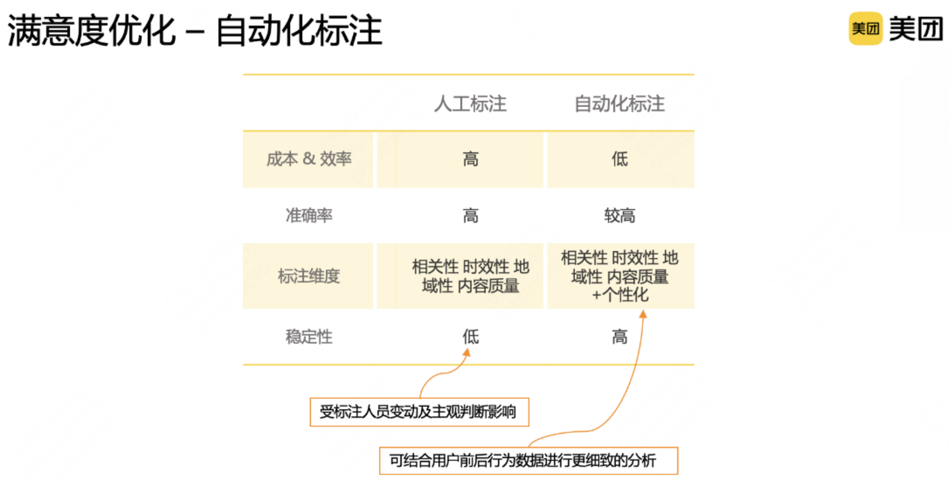
除了优化内容消费指标如点击、交互、时长等，搜索场景还很重视满意度优化。用户对搜索结果是否满意，可以从结果是否相关、是否足够新鲜、是否是对应地域、内容质量高低等显式维度进行刻画。

相关性是搜索满意度中最基本、最重要的维度。大众点评的很多内容有关联商户，可以比较方便地获取很多明确的结构化信息，比如商户类目、区域等，可用于辅助判断相关性。但也可能由于内容误关联商户带来噪音，为此需要综合从图片、文本、商户信息进行关键信息抽取，作为相关性模型的输入。

除了相关性，搜索结果的时效性也很影响用户体感。比如迪士尼疯狂动物城园区开始对外开放，属于突发性热点，通过敏锐捕获到突发热点，在搜索“迪士尼”时优先呈现对应的结果，可以带给用户惊喜。另一类查询词如“平安夜”属于周期性时效性热点，每年到这个时间段都会有这样的热点。为了更好地对时效性进行建模，从多个来源挖掘建立了热点事件库，接入商家自己提报的新鲜事，建立独立召回通道进行承接，并结合线上点击反馈进行误识别纠正。

以上满意度的评测通常较为依赖人工标注，近期开始探索自动化标注，对比分析如下：

- 在成本和效率上，人工标注需要准确理解搜索诉求，并对结果进行精确评判，从相关性、地域性、时效性、内容质量等维度进行刻画，成本非常高，通过自动化标注可以极大降低成本。
- 在标注准确率上，虽然还没有完全达到人工标注的水平，但自动化标注也达到了可用标准。
- 在标注维度上，自动化标注可以比较方便地对原有标注维度进行扩充，成本变化可控，比如在 Prompt 中提供用户的历史行为和偏好，就可以综合判断个性化需求是否得到了满足。
- 在标注稳定性上，人工标注质量可能会受标注人员主观判断甚至心情影响，但自动化标注不会有这样的问题。



	人工标注	自动化标注
成本 & 效率	高	低
准确率	高	较高
标注维度	相关性 时效性 地域性 内容质量	相关性 时效性 地域性 内容质量 +个性化
稳定性	低	高

在具体实现上，我们通过分步推理来实现自动化标注，首先分析用户当前意图，再结合当次搜索 Query、搜索意图、搜索结果等信息，从几个维度对搜索结果进行分析，最终综合判定当前搜索结果对需求的满足程度。

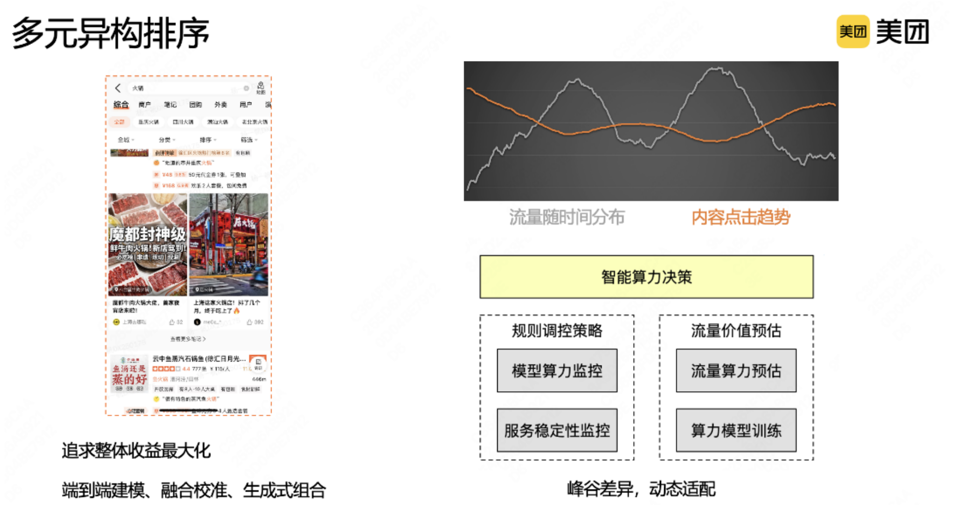
2.5 多目标融合

在得到内容点击、交互、时长、满意度等多维度的预估分数后，多目标融合层负责融合各个维度分数并排序。

- **精准预估**：多目标融合的前提是保证各个因子的打分稳定性和精准性，这也是前文提到做排序和校准联合建模的原因。
- **融合搜参**：通过 AutoML 方式进行自动搜参，寻找帕累托最优解，针对细分流量进行单独搜参，更加精准地刻画不同场景下对于各个目标之间的不同需求。
- **分发调控**：将生态或调控导向的因子引入融合公式，进行分发调控，比如对于新内容的扶持、更老内容的分发治理、近距离和特殊供给扶持等。

2.6 异构混排

前面各环节动作集中在内容搜索自身链路上，而最终内容是作为搜索结果的一部分和商户、团单等不同类型结果混排，追求整体搜索收益的最大化，为此需要进行多元异构混排。业界常见的混排建模方式包括端到端建模、价值融合公式、序列生成和评估等。



此外，本地生活领域流量分布有独有特点，在用户快决策和慢决策的场景下，对内容的需求存在差异，午餐和晚餐流量高峰期对内容的点击偏低，下午茶和夜宵等时段内

容消费意愿更强。结合内容和商户峰谷差异，依托工程能力如流量价值预估、模型算力和服务稳定性监控等，进行算力动态适配，从而保证整体搜索结果更能满足用户需求。

3 总结与展望

综上所述，大众点评内容搜索通过优化用户体验持续提升渗透率，进入快速增长阶段。在商户体系之外构建了基于内容的搜索分发能力，同时针对站内需求和供给特点进行了专项建设。

在后续工作中，希望建立体验问题的自动发现机制，帮助产运促进供给生产，并推动大模型在各个环节扎实落地、提升全链路的时效与性能，让内容得到高效准确及时的分发，进而在本地生活信息领域形成体验优势，助力建设本地吃喝玩乐社区。

4 参考文献

[1] Li S, Lv F, Jin T, et al. Embedding-based product retrieval in taobao search[C]. Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining. 2021: 3181-3189.

[2] Ai Q, Hill D N, Vishwanathan S V N, et al. A zero attention model for personalized product search[C]. Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 379-388.

[3] Chang J, Zhang C, Hui Y, et al. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information[C]. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 3795-3804.

[4] Wang Z, She Q, Zhang J. MaskNet: Introducing feature-wise multiplication to CTR ranking models by instance-guided mask[J]. arXiv:2102.07619, 2021.

[5] Chang J, Zhang C, Hui Y, et al. Pepnet: Parameter and embedding personalized network for infusing with personalized prior information[C]. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 3795-3804.

[6] Burges C J C. From RankNet to LambdaRank to LambdaMART: An Overview; 2010[R]. MSR-TR-2010-82. Available from: https://www.microsoft.com/en-us/research/publication/from-ranknet-to-lambdarank-to-lambdamart-an-overview, 2010.

[7] Sheng X R, Gao J, Cheng Y, et al. Joint optimization of ranking and calibration with contextualized hybrid model[C]. Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. 2023: 4813-4822.

[8] Bai A, Jagerman R, Qin Z, et al. Regression Compatible Listwise Objectives for Calibrated Ranking with Binary Relevance[C]. Proceedings of the 32nd ACM International Conference on Information and Knowledge Management. 2023: 4502-4508.

搜索广告召回技术在美团的实践

本文整理自美团技术沙龙第 81 期《美团在广告算法领域的探索及实践》(B 站视频)。首先介绍了美团搜索广告的三个阶段：多策略关键词挖掘、分层召回体系、生成式召回；然后重点介绍了生成式关键词召回、多模态生成式向量召回、生成式相关性判断在美团的实践。最后是一些经验分享及总结，希望能对大家有所帮助或启发。

美团搜索广告介绍

从美团流量场景角度来看，美团搜索广告分为两大类，一是列表推荐广告；二是搜索广告。推荐广告以展现商家模式为主，通常叫商家流。搜索广告的展现形式比较丰富，有商家模式，即以商家展现为主，会挂上菜品 / 商品；还有商品模式，即以商品展现为主，以呈现商品大图、商品标题等核心商品信息为主。

美团搜索广告流量有以下几个典型特点：

- 搜商品意图占据绝大多数份额，搜索商家只占较小的一部分；因此检索以商品为主，看候选规模的话，美团有百万量级的商家和十亿级别的商品，供给规模较庞大。
- 从商家特点来看，它有一个和业界传统电商场景不太一致的特点是很多是中小商家 / 夫妻店，他们的线上运营能力较弱，导致美团商家的内容质量没有其他电商平台好，所以在内容质量处理上，花费了很多时间。
- 美团的 O2O 场景特点是 LBS 属性，供给相对不那么充分，一个蜂窝内的几百个、上千个商家，搜索场景里有相关性约束，供给队列更短，有很多位置受限于供给没有填上。因此，美团搜索广告对召回率的要求更高。

上图展示了美团广告和传统广告之间一些的差异。下面介绍围绕着召回率提升我们做的一些工作。美团的搜索广告从 2019 年开始建设，主要经历了三个发展阶段：

第一阶段是美团**搜索广告启动阶段**，我们叫多策略关键词挖掘阶段。这时的工程基建能力处于起步阶段，也缺乏线上反馈数据，另外考虑落地节奏，希望尽可能快的把整个系统从 0 到 1 搭建起来，并希望在数据有限的情况下，快速支持迭代效率。所以这个阶段的召回方式是 SPU 通过离线方式，挖掘核心关键词，在线与 Query 精确匹配。

- **特点**：一是只聚焦于通过离线方式覆盖高频流量；二是缺乏线上的行为数据，以 NLP 的挖词技术为主；三是为了追求更多的覆盖，采用了多策略并行的方式，不断叠加新的召回策略，以达到更高的流量覆盖。
- **缺点**：第一，它不是一个正向匹配过程，而是从商品反向挖掘，所以整体挖掘效率很低，挖出了大量无效关键词，放到线上后，又无法匹配；第二，由于它是一个离线策略，所以只能覆盖一些高频流量，20%-30% 的长尾流量无法覆盖；第三个是多策略并行，在后期，新通路会通过不断挤压旧召回通路，最终形成 10+ 的召回通路，这种模式的维护成本较高，而且如果一个算法同学优化一个召回通路，策略面覆盖有限，整体的 ROI 在后期较低；第四个是缺乏个性化技术。

第二阶段是**分层召回体系**，它是基于流量和供给特点，按照业务类型，聚焦在几个象限内，每个象限里采用更聚焦的针对性召回策略，进行优化。

- **特点**：第一，在一个业务范畴内，通过把技术做深能够取得业务效果的极大提升；第二是随着基建能力的提升，更多的是把召回由离线切换成在线，以此覆盖更多的流量；第三是在单通路的召回能力上，我们突破了传统单一 NLP 技术瓶颈，开始大规模使用个性化 / 图 / 多模态等新的召回技术。在 2022 年底，整个分层召回体系取得了不少成效。
- **缺点**：第一是整个召回体系还是以判别式召回模式为主，决策空间不够，倾向于学习历史数据行为，马太效应现象变得越来越严重，而且整个探索空间在这种判别式模型下面，局限性也越来越明显；第二是整个模型规模和容量相对不足，天花板很容易逼近；第三是采用多通道独立优化的方式，每个通道都有自己的样本特征，很难做到通道之间的融合，难以形成 1+1>2 的效果。

第三个阶段是**生成式召回**。核心思路是借鉴生成式大模型的思路和能力，改造现有的召回技术体系，长期上来看，我们会探索 DSI 新召回范式。

大模型在 C 端流量的落地，会遇到很多算力瓶颈。经过一年的探索，我们形成了大模型落地的方式和原则，分为三类。第一是离线用能力构建领域微调大模型；第二是在线用大模型技术思想，结合传统模型改造现有模型能力；第三是通过蒸馏方式，在线尽可能学习离线大模型能力，通过蒸馏方式把大模型通用知识蒸馏到在线规模相对较小的模型上。

面临的挑战包括三个方面：第一是有算力焦虑；第二是在模型规模变得越来越大的情况下，如何保证模型迭代效率；第三模型的变化不能发挥模型能力本身的优势，我们希望构建以大模型核心能力基础为核心的架构，拿到更好的效果，但改造成本较大。

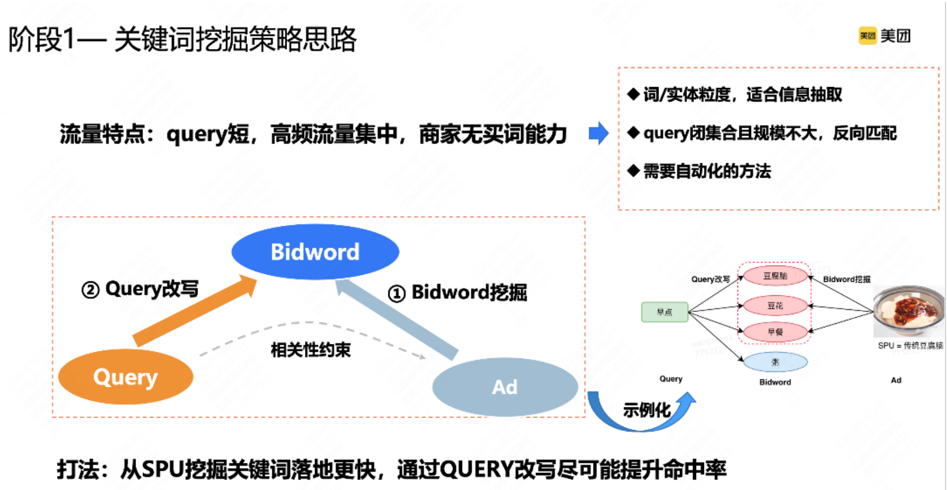
美团搜索广告召回发展阶段

阶段一：多策略关键词挖掘

对于多策略关键词挖掘阶段，美团搜索广告的特点一是 Query 较短，平均长度也就两三个字，因为很多人在美团 App 搜索比如烧烤、西餐这种很泛但又很短的 Query；二是流量分布比较集中，高频、Top 几万的 Query 就占了大约 70% ~ 80% 的流量，头部效应比较明显；三是区别于业界传统的搜索广告，美团搜索广告商家没有买词能力，通常以整个店铺的投放模式为主。

基于这三个特点，我们设计了关键词挖掘策略思路。一由于 Query 很短，我们很容易通过信息抽取，把词或实体核心信息抽取出来；二是因为头部效应比较明显，Top2 万的 Query 覆盖了很多流量，采用这种离线方式能快速拿到大部分收益；三是由于商家没有买词能力，如果用 Query 直接匹配商品，会涉及到传导文本匹配问题，匹配难度会更高，所以我们最后采用模型从商家商品里挖掘核心词，在线做短串匹配的方式。

如左下图所示的召回模式是离线，我们从广告或 SPU 里通过关键词挖掘的方式挖掘出关键词，在线通过 Query 改写的方式尽可能提升在线匹配效率。



具体来说，我们的关键词挖掘策略经历过三版迭代，按照技术由浅入深的方式做的。

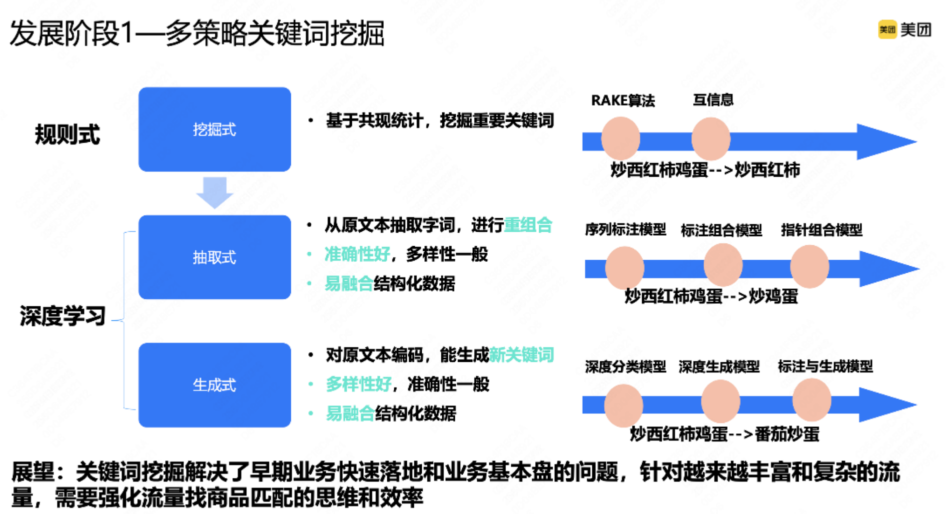
- 早期第一版创建时，我们更多采用基于规则的挖掘式策略，把流量分成了商家词、商品词和品类词。商品词通过分词和词频贡献的算法，挖掘核心关键词，由于品类字面没有完全匹配的信息，我们通过互信息，构建词之间的权重去挖掘。但问题一是规则能力较弱；第二是只能挖掘出连续的短差，比如“炒西红柿鸡蛋”，它只能挖掘出“炒西红柿”，挖掘不出“炒鸡蛋”。

所以在快速落地了规则式挖掘策略后，我们开始用模型方法自动挖掘关键词。模型通常有两种，抽取式和生成式。

- 从准确性和数据局限考虑，先采用的是抽取式挖掘方式挖掘关键词，这经过了三个阶段的策略迭代。第一版将规则式升级为了模型挖掘方式，传统上叫序列标注模型，这种模型只能挖掘出连续短串，好处是挖掘效率比基于规则的挖掘模式高，但会导致很多关键词受限于连续短串的方式而挖掘不出来；后面做了两版突破连续短串的挖掘方式，分别是标注组合模型和指针组合模型。标注组合模型能够跨越连续短串挖掘，但它有一个顺序概念在里面；指针组合模型可以在原有短串里随机组合词，突破顺序和连续的局限。但抽取式模型的准确率较高，探索空间不足。

- 在迭代了三版抽取式策略后，我们将策略重心聚焦在生成式挖掘方向上，希望突破字面极限，探索更大的流量空间，最后做了三个阶段的迭代。第一个阶段是深度分类模型，它能够突破字面限制，Top2 万的 Query 能够覆盖大部分流量，那将 SPU 商品文本直接分类到这 2 万个 Query 标签里，做词和 Query 间的匹配，但这种多分类模型较难优化，也不能泛化出更多的 Query，时效性和更新频率也有限；所以后来我们采用了深度生成模型，实现了相对广阔空间的挖掘，但受限于模型规模和样本丰富度，准确性不太好，所以我们在后面加了标注和生成模型，在具备生成泛化性的同时，尽量控制 Query 质量，以上所有模型都是传统 NLP 里的基础模型，我们只是把模型数据、业务特点做了适配。

未来，我们期望在关键词挖掘阶段，较好地解决了早期业务落地和基本盘问题，但是面对美团比较复杂的流量场景，还需要通过新方式强化流量，提高商品匹配效率。



阶段二：分层召回体系

2022 年，我们开始正式规划第二代召回体系即分层召回体系，核心思路是按照流量和供给特点分类，强意图是直接搜索一个商品；泛意图比如搜索“烧烤”这个品类，

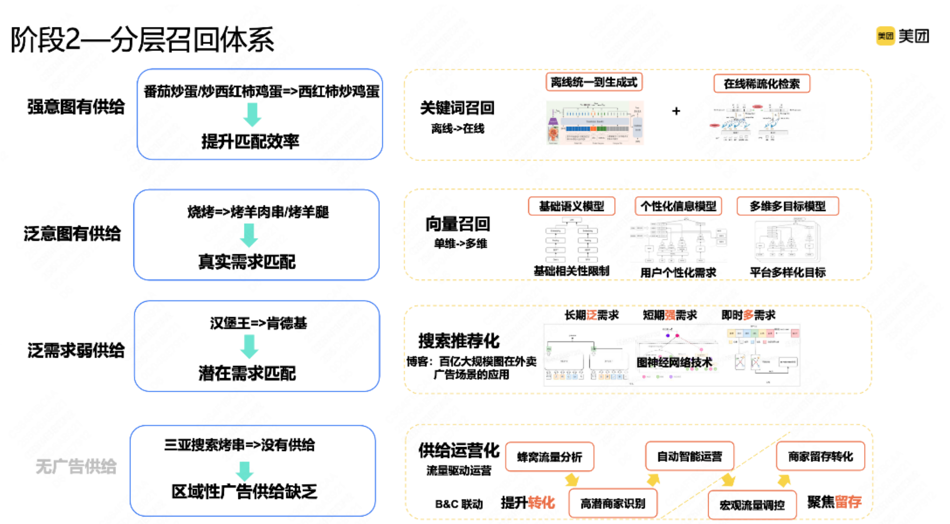
泛意图用户虽然表达了需求，但满足需求的候选可以很广，甚至可以替代；供给层面分为有供给、弱供给和没有供给三个象限。我们找到核心象限聚焦优化，最终找到以下四类场景。

- 第一是强意图有供给，通过关键词就能较好满足，因此在这个象限里，我们更多是在迭代关键词召回技术。一是通过离线统一到生成式的方式。前面介绍离线关键词挖掘策略可能会有十几个通道，不管迭代哪个通道，策略召回的覆盖面都是有限的，而且团队也没那么多人迭代，但这种情况下，我们把整个离线关键词十多路的挖掘策略通过规模较大的生成式模型做了统一，引入了多模态信息，做到了数据更多、模型更多以及召回目标更多的情况，后期只需要通过优化模型能力，就能取得线上全流量覆盖的效果；二是通过离线关键词的方式做到了在线。我们并没有采用业界传统的布尔检索，这种方式有两个局限，一是 Query 改写以及商品分词基于较浅层的模型，整体效果会受限于模型效果。二是它没有做到检索和最终目标的匹配。在线系数化检索方式类似于双塔向量检索，但每个模型出来不是一个稠密的向量，而是一个几万维稀疏的 term 粒度，通过端到端的建模方式，把 Query 和商品映射到一个稀疏的几万维槽位的几个槽位里，离线训练时通过槽位端到端的建模，实现目标检索和目标一致性，在线检索时，基于槽位构建倒排检索，具备一定的可解释性。
- 第二个是泛意图有供给，体现了用户的个性化偏好，通过迭代向量召回模型覆盖这个场景。向量召回经过了三版迭代。第一版是基于传统语义相关性约束的双塔模型，和业界的做法类似；我们的向量召回目标最终要让建模用户个性化，第二版将用户个性化提上了日程，但如果只把用户个性化特征和传统语义特征融合在一起，黑盒式学习很容易被用户个性化信息带偏，最后我们让用户个性化信息和语义个性化信息分别学习，通过显式叠加的方式做端到端的建模。这种检索方式能够兼顾个性化和语义相关性信息；第三版是基于平台的多样化目标，我们需要对齐后链路的精排目标，在召回阶段考虑整体商业价值。
- 第三个是泛需求弱供给，比如搜索“汉堡王”，但给 TA 一个“肯德基”，TA 也会下单，通过搜索推荐化的方式覆盖和解决。这个场景比较复杂，从业务来

看，它需要做引导和推荐，在结果页里也做偏泛结果的推荐，涉及到搜索前和搜索中，搜索中既有商家也有菜品，既涉及要推荐什么样的菜品，也涉及推荐什么样的商家；另外推荐本身是一个关系建模。我们最后选择基于图模型的迭代，因为图模型首先是一个基于关系的建模，而且图模型具备多场景海量信息的容纳能力，在图建模里，一是构建了异构的多节点百亿规模图，通过图预训练加微调的方式识别多个场景，我们最近也在尝试做图和大模型训练相结合的方式；二是我们把整个图检索搬到在线，因为在搜索场景中，用户需求是即时需求，属性较强，只有把检索搬到在线，通过图在线的实时检索聚合到用户当前最有可能的潜在兴趣情况下，才能实现收益最大化。

- 第四个是没有供给的场景，通过流量结合供给运营化的方式解决。

阶段 2 通过划分象限和场景聚焦迭代的方式，拿到了不错的收益，但很快也遇到了瓶颈。



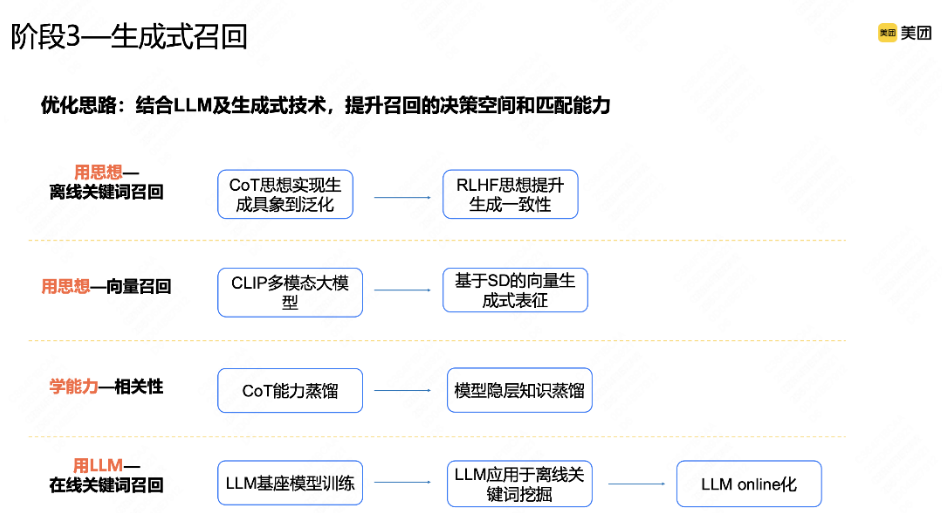
阶段三：生成式召回

2023 年，我们开始探索新生成式召回方式，核心思路是结合大模型或生成式技术思想，提高召回算法的决策空间，提升模型的匹配能力。经过一段时间迭代，我们抽象

出广告子模块结合 LLM 落地的三类思想及方式，分别是用思想、学能力、用 LLM。具体和子模块结合的一些探索如下：

- 一是离线关键词召回方向。如刚才介绍，我们已经把整个离线关键词召回技术方式统一到了规模不错的生成式模型方式上。大模型出来后，直接用大模型其实还存在着算力及效果的 2 个挑战。但我们认为大模型的两个核心技术思想：Cot（Chain-of-thought，能使大型语言模型能够更好地理解人类的语言请求）推理和 RLHF（Reinforcement Learning from Human Feedback，一种基于人类偏好的强化学习方法）对齐人类反馈思想，对我们现有模型的优化也是有帮助的，因此我们使用大模型的这些技术思想来改造离线生成式召回模型。
- 二是在向量召回方向。我们已经将向量表征升级为多模态模型，进一步我们思考，LLM 语言大模型对于离散 Token 的信息归纳及表征是有比较大的提升的，但是在稠密表征领域，一个值得借鉴的方法是扩散模型，因为扩散模型也是通过多步去噪的方式来生成目标，通过扩散多步过程，在其中引入多元信息多步融合的思路，提升整个向量召回的向量表征能力。
- 三是随着我们探索的深入及对应算法能力的提升，我们构建了美团领域广告大模型，尝试直接把大模型用到美团实际场景里做关键词召回，将离线中等规模的生成式模型直接替换成大模型，并探索大模型在线化。
- 第四个是蒸馏大模型能力，主要在相关性场景落地，目前蒸馏了两块能力，Cot 推理能力和模型隐层知识能力蒸馏。

下面我主要介绍下结合 LLM 的能力，在召回场景下已经全量的一些技术探索。

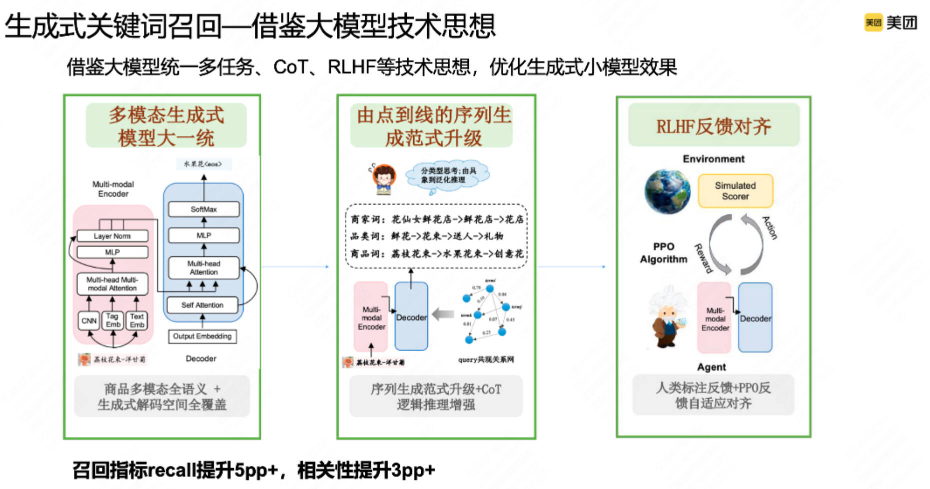


生成式关键词召回

- 生成式召回主要借鉴大模型思想，我们已经升级为统一的生成式模型，它的工作方式是基于 beamsearch 的方式，一次生成多个结果，但结果之间是互相看不到的，我们认为这种方式会存在问题，另外，从线上和实际生成结果来看，词之间是有关系的，按照概率方式来看，如果一个关键词能够推理出另一个关键词，大概率前面这个关键词要比下一个关键词的信息含量多，那能否借鉴大模型推理思想，按照序列生成方式逐步推理出多个关键词。
- 我们通过构建概率贡献图的方式，采样得到关键词之间的导出关系，在一次生成时，直接生成多个关键词，这多个关键词之间有推理关系，比如要给“花仙女鲜花店”商家生成关键词，第一个关键词就是相对具象的“鲜花店”，它的含义和商家的商品描述是确定的，在生成“鲜花店”时，可以推理成“花店”，进一步可能会生成新关键词，通过这种序列推理方式，能够很好地利用关键词之间的关系。
- 在序列推理生成关键词时，比如生成了 5 个关键词，有一个关键词不相关，剩下的 4 个关键词是相关的，那如何通过模型识别出这种不一致现象，能否借助人类反馈方式，实现模型序列好坏端到端的判断。模型生成的关键词序列与人

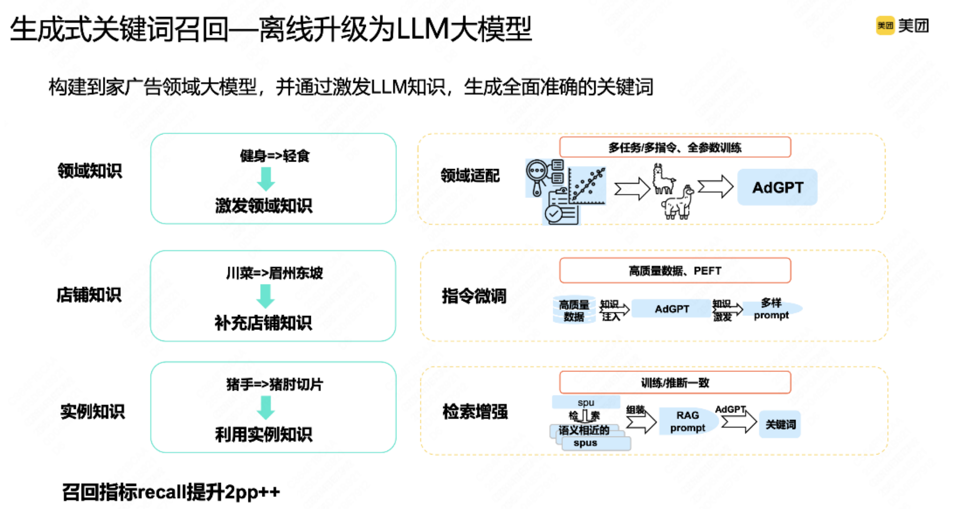
工标注是否一致，通过这种反馈对齐的方式喂给模型，提升整个序列生成结果的一致性。

通过这种方式，召回得到明显提升，而且生成相关性的准确度也得到明显提升。



对于离线关键词，前面是中等规模的模型，我们最近把整个离线关键词替换成大模型，之前没有替换是因为开源通用大模型能力在领域场景里，挖掘词的准确性和通用性有限，我们一直在构建美团广告领域的大模型，通过激发大模型知识，生成更全面准确的模型，我们做了 3 个阶段的优化。

第一是融合领域知识，比如健身和轻食相关，这是领域知识，通过领域全参数训练得到一个基础的广告领域模型。第二是融入场景知识，美团有很多店铺和商品名，比如川菜和眉州东坡在店铺里有很多相关数据。通过这种指令微调的方式学习店铺知识，在实际应用时，再学习偏实际的知识，比如搜索“猪手”时，发现他之前检索过很多“猪肘切片”，通过这种检索方式增强大模型当前推理知识能力。最后通过构建领域大模型和检索增强范式，在一些场景里替换传统大模型，这样，我们发现召回效率明显提升。



多模态生成式向量召回——结合扩散模型，多阶段生成向量表征

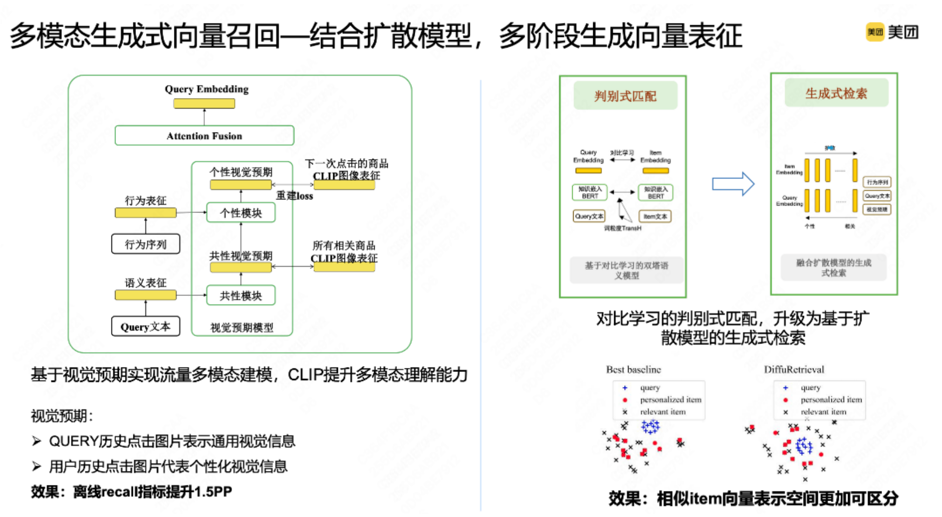
我们改造或优化多模态向量召回，在表征里结合扩散模型做了优化，如下图左边所示，传统的多模态向量召回更多是在 item 侧表征里，将商品图片和文本模态信息融合在一起，得到一个表征，那能否通过一些方式在 Query 侧也实现多模态表征。一个用户在美团场景里搜索一个 Query 时，大概率他的脑海里已经有关于这个 Query 所对应菜品图片的大致印象。那我们如何通过模型建模的方式还原图片的印象，核心在于还原用户的潜在意识。我们的做法是，一是把 Query 历史点击的图片信息汇集在一起，表征 Query 所代表的通用视觉信息；二是将用户历史点击图片代表用户个性化视觉信息，把这两类视觉信息叠加在一起，可以在一定程度上反映用户在当前搜索框架下，想要得到的流量侧多模态信息，最后通过多模态表征匹配技术，整个离线召回效率也有提升。

但这种方式也是基于传统的判别式表征，比如现在大家都在做个性化向量召回，相关性和个性化之间有递进关系，最浅层的需要保证相关性，第二层才需要在相关性里挑选更个性化、更符合用户偏好的候选集，给到下游链路。但传统的判别式方式一般在特征阶段叠加不同特征，通过建模、多目标落实反向迁移方式，不能很好的显式学习到不同目标间的递进关系，但 SD 生成模型比较适合这种稠密向量生成，通过多步还

原过程，本质上也是一个不断推理的生成式过程。

我们希望向量表征具备不同信息的推理能力，SD 的多步加噪去噪过程类似于推理过程，可以相结合，在不同步骤中引入不同维度的信息，做到多维信息的显式理解及融合。在正向编码过程中，先将 item 通过编码器编码成向量后，逐渐加噪还原成白噪声，在反向去噪还原过程中，在噪声里分阶段添加用户 Query 以及 side info 信息，通过多步还原的方式，还原出 Query 所代表的信息。并有两个对比的操作，一是传统的样本 Paiwise 学习，通过对比学习方式拉近 Query 与相似 Item 的表征；二是我们认为相似 item 有类似的标准过程，通过对比学习拉近相似 item 之间在扩散中间过程的表征，这是整个建模过程。

在还原阶段，我们会显式还原中间步骤叠加相关性信息、个性化信息，通过对比方式让模型在还原过程中显式相关性和个性化信息，最后在模型结果里能看到，如下图左边是传统的判别式模型里最好的一个 Baseline，它能够较好区分 Query 和正样本信息，但它在个性化样本和相关性样本里基本是混在一起的，通过这种扩散模型方式，相关性样本和个性化样本就有一定程度区分开来的能力。



总结

生成式算法相比判别式，能够有效的拓展整个召回的策略空间，2023 年我们基于大模型的技术思想赋能现有的召回模型拿到了一些效果，但远未达到新技术方式的上限。看未来，一方面随着算力的逐渐提升，我们可以探索更大规模的生成式模型直接落地，另一方面可以探索在线的端到端生成式召回，来优化多级漏斗带来的样本偏差和漏斗效率问题。

全域用户建模在美团首页推荐的探索与实践

一、背景

1.1 产品形态及业务特点

美团首页推荐展位，如下图 1 所示，是用户打开美团 App 后，触达美团各业务的流量入口，每天服务数千万用户。首页推荐的核心能力体现在差异化地承载并快速响应用户个性化需求，需要支持高效分发外卖、餐饮、休闲娱乐、酒店旅游、优选、买菜、电商、超市闪购等各种业务供给。业务之间在履约特性、供给特点之间存在巨大的差异性，如外卖业务通常为用户“饭点儿”的即时随性消费，受到配送距离的强限制；酒店旅游业务需较长周期种草规划，偏好相对稳定，但集中于特定节假日时段；电商业务随着不同品类不同场景对于履约周期有所取舍。这些特点对个性化推荐建模提出了较高的要求。

图 1. 首页推荐展位及其业务特点

1.2 全域用户建模的必要性

全域用户建模是指在多个平台、多个应用或多个领域中，整合用户在不同环境下的行为数据，构建一个统一的用户画像或模型。这种建模方法旨在更全面地了解用户的兴趣、偏好和行为模式，增强建模的准确性和多样性，从而提供更加精准的个性化服务和推荐。

具体到美团首页推荐的业务场景下，全域用户建模旨在将用户的源域兴趣迁移到目标域，利用其他源域的用户行为数据来增强目标域即首页推荐的召回排序链路，解决目标域中的数据稀疏问题[1]，以提高推荐效果。本文中的源域表示美团大搜、金刚区等除首页推荐外的其他渠道或展位，目标域即指首页推荐。推荐场景下的全域用户建模也可称为跨域推荐。

过去美团首页推荐各模块建模所用的行为数据过于依赖特定来源（仅首页猜喜），缺乏来自其他重要渠道的用户交互数据，这限制了首页猜喜推荐系统在全面理解用户行为模式上的能力。

美团 App 展位众多，相比于金刚区、首页搜索等展位，首页推荐展位用户行为较为稀疏。较多用户养成了通过频道区内页、主动搜索等方式来获取感兴趣供给的心智习惯，不利于猜喜展位释放更多的引流潜力。这一现状对猜喜侧用户兴趣预估、CTR/CVR 排序预估等模型类建模方式造成挑战，主要体现在两方面：1）用户行为稀疏导致训练不充分；2）用户行为有偏导致分布预估不准。为解决这一挑战，本文通过引入全美团多源数据信号（包括多展位、多应用渠道），对猜喜各链路算法建模进行数据增广、纠偏、领域迁移等优化，旨在提升用户兴趣建模能力与人货匹配能力，实现召回排序技术升级，以带动核心业务目标提升、改善用户体验。

1.3 落地难点

由于展位之间显著的数据分布差异、多业务间的可迁移性差异以及本地生活业务时空场景的强相关性三个方面的原因，使得将外域信号引入推荐各链路中存在严重的**负迁移**挑战。其中，展位差异是行业内落地全域建模的共有挑战，时空场景强相关性是美

团本地生活业务特点带来的特有挑战，而业务间差异则是美团首页推荐展位分发多业务供给带来的展位特有挑战。

1.3.1 展位之间显著的数据分布差异

由于不同展位承载差异化的需求表达，美团搜索和业务金刚区较多承载用户意图更明确的主动需求，而首页推荐更多是用户有模糊性需求时的自然浏览，且不同展位的物料展示形式也不同，使得不同展位流量在用户分布、行为习惯等方面存在显著差异。美团 App 不同展位之间的业务占比具有显著差异，体现用户心智在不同渠道存在一定差异，在扩展用户信号覆盖时，需要考虑不同展位间用户行为的差异性。

例如，在美团的首页猜喜，向用户推荐药品可能会显得不合时宜，因为用户可能没有明确的购药意图。相反，在美团医药频道，用户的行为预示着他们有明确的相关意图，因此在该频道内搜索和推荐药品是用户预期之内的行为。如果直接将其他数据源应用于猜喜模型，不考虑上下文的差异，可能导致负迁移现象，即模型性能由于数据整合不当而降低。应对这一挑战的有效方法是采取精细化的数据集成方法和周到的模型训练策略，以确保数据的适当应用并优化模型性能。

1.3.2 多业务间的可迁移性差异

美团本地生活服务业务类型众多，与以内容、商品为主体的推荐场景下业务相比，特性显著。

- 行业内，服务电商相比实物电商具有更大的业务差异性。比如餐饮外卖、酒店预订、到店桌游、生鲜自提之间的差异相较于实体电商的服饰、书籍、3C 等品类有更大的差异性，后者可通过用户网络下单、商家发货、快递到手的统一流程满足用户需求；而美团的“外卖”、“KTV”等业务涉及实时配送和到店消费，业务逻辑间有所不同。
- 美团内部，首页推荐业务交杂情况更为显著。典型推荐场景还有美团外卖推荐和大众点评内容推荐等。美团外卖推荐更聚焦于外卖的不同品类，需平衡商户的复购和用户尝鲜探索，目标层面更关注短期的用户体验和转化效率；大众点

评推荐是内容的分发，更多关注“点击”、“停留时长”、“收藏”等信息消费目标，不同类型内容之间的差异相比服务之间较小。

首页推荐承接的外卖、餐饮、休闲娱乐、酒店旅游、优选、小象、电商、超市闪购等多个业务之间的履约特性、供给特点之间巨大的差异性导致全域建模时不同业务的可迁移性具有显著差异。如医药、闪购等大部分是在用户有明确需求时的即时需要；优选、小象超市等则表现为用户下单时会包含多件商品，其中可能包含用户为凑单而购买的商品，而并非用户对这些商品真正地感兴趣。这类业务往往在全域用户建模过程中更容易导致负迁移问题，而到店餐饮、休闲娱乐等业务的下单行为更能体现用户常态化的兴趣，因此负迁移问题相对较弱。

1.3.3 时空场景的强相关性

与纯电商推荐不同，美团推荐中的生活服务对时空限制非常敏感，所以时空场景信息对推荐结果十分重要。举几个时空相关的例子，用户在午间 11 点前后想订外卖吃饭，节假日前可能订酒店去旅游；用户在家需要零售买菜类商品，用户在商场需要到店餐饮类卡券。下图 2 展示了外卖和门票业务在首页推荐的点击量级随星期和小时的变化，可以发现用户对外卖的兴趣往往集中在饭点，而对门票的兴趣集中在周末，和大众的认知相一致。时空场景决定了用户的行为动机，用户与业务的交互依赖所处场景。

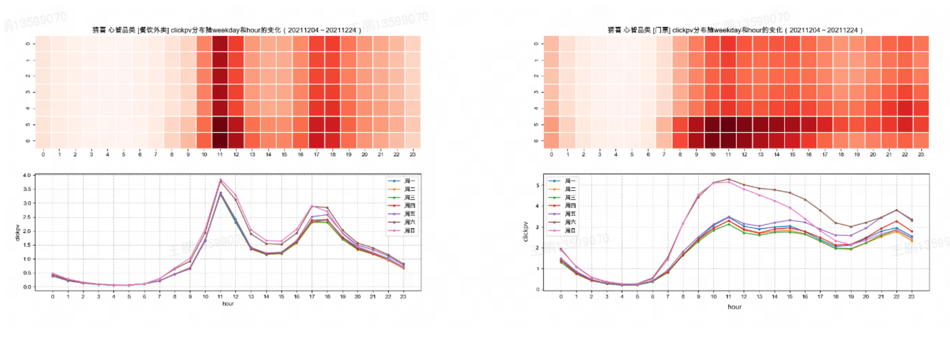


图 2. 外卖和门票业务在首页推荐的点击量级随星期和小时的变化

正如场景信息对首页推荐的模型提效非常重要，全域用户建模考虑其他展位信号向推荐展位迁移时也需要强调时空场景信息，以实现在不同的时空场景下对各业务信号的差异化迁移。例如在饭点，我们希望其他展位的外卖兴趣能够迁移到推荐展位，而在周末则希望用户过往在其他展位表达的对门票业务的兴趣更多地迁移到推荐展位。而如果在非饭点将大量餐饮外卖的兴趣迁移到推荐域，使得推荐系统在非饭点大量曝光餐饮外卖，反而会降低推荐系统效率。

基于上述几个方面的原因，为解决负迁移问题，应对不同展位之间显著的数据分布差异，需要对即时需求类的兴趣进行筛选而仅迁移常态化的用户兴趣；而生活服务供给的时空场景强相关性，则要求基于场景信息差异化地迁移适合当前场景的兴趣信号，给予其他展位不同类型的兴趣信号以不同的迁移权重，从而实现仅将对推荐域有益的其他域兴趣信号迁移到推荐域，排除干扰信号及噪声信号。

二、全域用户建模的探索与落地

综上所述，全域用户建模有助于扩展用户理解和个性化服务的空间，但由于不同展位、不同业务之间用户行为和需求的显著差异，存在较大挑战。业界现有跨域推荐方法按照知识跨域迁移方式可以划分为基于 Content 的迁移 [2][3]、基于 Rating Patterns 的迁移 [4][5] 和基于 Embedding 的迁移 [6][7][8]，其中基于 Embedding 的方式应用最为广泛，即利用机器学习方法首先学习出不同域的 User 或 Item 的 Embedding 表征，然后将 Embedding 作为知识载体进行域间的信息迁移。

业界现有方法尽管已经取得了不错的进展，但当应用于展位和业务之间都存在显著差异的美团场景下，均效果不佳。原因在于现有方法往往研究在多个相似领域之间进行知识转移（大多为不同场景下的多个推荐域 [5][6][8][9][10][11]），相似域间的负迁移问题往往比较小，但在美团场景下不同展位和不同业务带来跨领域之间显著的差异性。

另外，现有的跨域推荐方法主要关注隐式建模范式，学习目标是去拟合目标域或源域兴趣的 Ground Truth，而没有对兴趣信号的跨领域迁移进行直接监督。因此，从源域到目标域的兴趣迁移是隐式的和不可控的。这种隐式建模范式使得区分源域的有用

信号和噪声信号变成一大挑战，很可能将不适当或干扰的兴趣迁移到目标域，从而导致负转移问题。

因此，基于多展位、多业务及时空场景强相关性带来的落地挑战，我们计划对不同域的数据在细粒度时空场景下的可迁移性进行精准建模，强调时空场景信息的重要性，仅迁移其他域中对推荐域有用的兴趣信号，以实现兴趣信号的有效迁移并优化模型效果。

全域用户建模的落地过程中，我们采取递进式的迭代策略，在首页推荐召排模块分阶段逐步摸索与落地。全域用户建模召排升级的整体思路及落地路径如下：

图 3. 全域用户建模召排升级的落地路径

- **第一阶段**：优化全域行为召回策略。首先将全域信号引入召回阶段，将行为触

发的 i2i 类、热单类等多路召回子策略中的数据源由猜喜行为扩展到多展位行为，并进一步扩展到多渠道行为，快速验证全域用户建模在首页推荐落地的可行性。

- **第二阶段**：模型训练样本引入全域信号，并缓解负迁移问题。在首页推荐召回和排序模型引入金刚区、首页搜索中的支付行为数据，作为训练点击、支付目标的正样本，对猜喜原训练样本 Label 进行改写，以融入全域兴趣信号。然而由于不同域数据分布的显著差异，在实验过程中，我们发现大量引入优选、医药等业务的外域正样本导致模型在这些业务上猜喜离在线 AUC 大幅下降，导致模型“学偏”。为此本文根据不同业务导致负迁移的严重程度设置了阶梯式的训练权重，来缓解负迁移问题。
- **第三阶段**：提出显式兴趣迁移框架解决负迁移问题。上一阶段我们对不同类型（本域与外域）监督信号分业务赋予阶梯式权重来缓解跨域负迁移问题。此种方案虽然取得了较好的业务效果，但对训练权重的区分仅限于业务粒度，且阶梯式权重是由人工拍定的硬参数，仍难以精准地建模细粒度场景（如 < 工作日、中午、雨天、青年、白领 > 构成用户上下文场景）下源域兴趣向目标域的迁移。为了实现仅对适合推荐域的兴趣信号进行迁移，本文创新性地提出了一种显式兴趣迁移跨域推荐框架，解决现有方法面临的跨展位、多业务、强时空相关性场景下的负迁移挑战。无需复杂的网络结构及繁琐的模型训练过程，本文提出的显式框架能快速在工业推荐系统落地，为跨域推荐提供了一种简单而有效的解决方案。
- **第四阶段**：全域全链路统一建模及感知增强。前述阶段通过召回策略、样本数据及模型结构方面的升级，在召回及排序模块初步构建起了全域感知能力，但召排全域体系建设在扩展全域样本空间、构建全域特征、统一召排全域样本等方面尚存优化空间。为全方面增强全域感知能力，我们进行了全域、全链路、全供给统一样本建设，在全域信号更完备的新样本基础上实现全域全链路统一建模。
- **第五阶段（理想态）**：基于用户全域行为的生成式范式统一召排。前面 4 个阶

段已经基本完成并全量上线，取得了预期成果及业务收益。现今，随着大模型与推荐系统结合的日益深入，生成式推荐范式展现出了巨大的应用潜力。生成式模型的序列化 Token 输入形式与全域信号构建的用户全域行为序列可以完美结合。我们计划基于生成式推荐范式进一步统一召排全域建模方案，构建样本、特征和模型范式统一且可复用的召排全域体系。

2.1 优化全域行为召回策略

美团拥有众多场景展位和消费渠道以满足用户不同类型的需求，同一用户在不同展位 / 渠道上都可能有行为反馈，体现了用户细分的、多样的兴趣。为提升召回层对用户全域兴趣的感知能力，我们按照【首页猜喜 -> 多展位 -> 全渠道】的路径逐步扩展召回行为数据源，对行为触发的多路召回子策略进行了升级改造。

我们将重定向、热单、i2i 类的多路召回子策略计算逻辑及召回 trigger 的行为数据源由仅首页猜喜行为扩展为全渠道行为（包括搜索、点击、支付行为等），在召回阶段补齐首页推荐系统对本展位外行为信号的建模及兴趣感知能力。通过在召回模块引入全渠道的多源（多展位、多应用渠道）兴趣信号，提升用户兴趣建模能力与人货匹配能力，带动大盘点击、支付等核心业务指标增长。

作为全域信号引入美团首页猜喜的初步尝试，优化全域行为召回策略的快速落地，并取得了较好的核心业务指标收益，验证了“全域用户建模”在首页猜喜的可行性，为后续我们围绕全域建模的更多模型侧的深入探索提供了基础验证和强大信心。

2.2 召排模型训练引入全域信号

全域行为召回策略优化取得业务收益后，我们认识到了全域信号具有较大的应用潜力。因此我们进一步在模型训练中引入全域信号，通过在召回 u2i 模型和排序模型训练样本中引入金刚区、首页搜索中的支付行为数据，作为训练点击、支付目标的正样本，对原训练样本 Label 进行改写，Label 改写方式如下图 4 所示。通过引入全域正向用户兴趣信号大幅扩充了模型训练所使用的正样本，克服了推荐域用户行为稀疏导致的训练不充分问题。

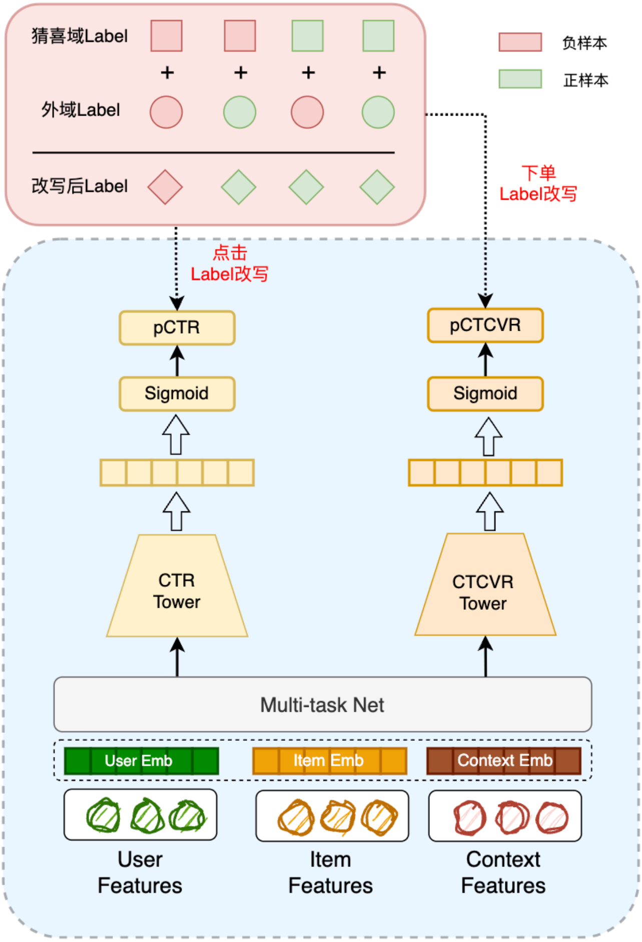


图 4. 引入外域支付数据扩充模型训练正样本示意图

尽管通过 Label 改写方式大幅扩充了模型训练的正样本，在一定程度上缓解了数据稀疏问题，但囿于不同域数据分布和用户习惯上的较大差异，跨域迁移建模中存在的外域信号干扰本域信号的负迁移成为一大挑战。针对这种跨域信号引入过程中部分业务存在的“水土不服”问题，本文采用分业务赋予不同类型（本域与外域）监督信号阶梯式权重的方式，有效缓解了全域用户建模中的负迁移问题。

在实践过程中，针对首页推荐源域信号，赋予正常权重；针对金刚区、首页搜索的外域监督信号，赋予较低训练权重；针对易导致严重负迁移问题的场景强相关性业务

（如优选、医药等），赋予极低训练权重。具体权重可根据模型训练时离线指标的变化调整。此外，扩充的正样本 Label 源于首页推荐原始负样本的改写，故不会明显增加离线存储与模型大小。

2.3 显式兴趣迁移框架解决负迁移问题

为缓解外域信号干扰本域信号的负迁移问题，我们最初的解决方案是对不同类型（本域与外域）监督信号分业务赋予阶梯式权重。此种方案虽然取得了较好的业务效果，但对训练权重的区分仅限于业务粒度，且阶梯式权重是由人工拍定的硬参数，仍难以精准地建模细粒度场景（如 < 工作日、中午、雨天、青年、白领 > 构成用户上下文场景）下源域兴趣向目标域的迁移。

我们希望将业务粒度的硬参数权重升级为精细时空场景粒度的自适应软权重，以实现在美团首页推荐场景下更精准地建模源域兴趣向目标域的迁移，即在不同的时空场景下精确地排除他域的噪声兴趣，保留适合推荐域的有用兴趣。为此，本文创新性地提出了一种显式兴趣迁移跨域推荐框架，解决现有方法面临的跨展位、多业务、强时空相关性场景下的负迁移挑战。无需复杂的网络结构及繁琐的模型训练过程，我们提出的显式框架能快速在工业推荐系统落地，为跨域推荐提供了一种简单而有效的解决方案。

我们也将这一创新思路整理并投稿顶会论文《EXIT: An EXplicit Interest Transfer Framework for Cross-Domain Recommendation》，文章已被 CIKM 2024 接受，arXiv 下载链接为：PDF。

2.3.1 显式建模范式

基于对业界方案的充分调研，我们认为当前业界跨域推荐方案无法应对美团首页推荐场景下全域用户建模突出的负迁移问题。在无业界适用方案的情况下，我们基于对业务的深刻理解，创新性地提出显式兴趣迁移跨域推荐框架 EXIT（EXplicit Interest Transfer framework），来显式建模其他域适合向推荐域迁移的兴趣，解决传统隐式跨域推荐方法用于美团这类跨展位、多业务场景下广泛存在的负迁移问题。EXIT 方

案与传统跨域推荐方案的区别如下图 5 所示，和传统跨域推荐方法不同的是，EXIT 框架能够基于用户所处的时空场景仅从源域迁移那些对目标域有益的兴趣信号，从而防止负迁移。

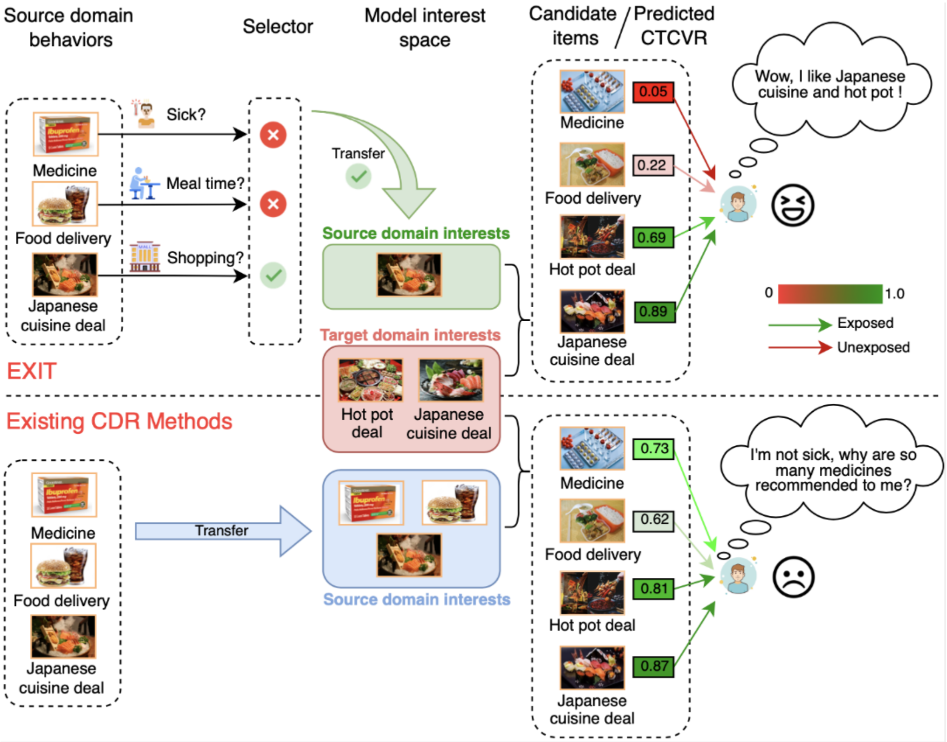


图 5. EXIT 框架与传统跨域推荐方案的区别

EXIT 框架由协同训练的 multi-task 兴趣建模网络、兴趣组合标签和场景选择网络（SSN Net）组成，框架图如下图 6 所示。Multi-task 兴趣建模网络分别构建了目标域 tower 建模用户的目标域（推荐域）兴趣、源域聚合 tower 统一建模推荐域之外的其他源域兴趣（搜索、金刚区等外域），避免多域兴趣单独建模导致模型参数和在线推理耗时增加，便于模型在线部署。兴趣组合标签作为兴趣迁移过程中监督学习的标签，代表了全域空间下用户在目标域中完整兴趣的 ground truth，显式提供了跨域兴趣迁移的监督信号。场景选择门控网络（SSN Net）建模了细粒度场景下外域兴趣向推荐域的迁移力度，从众多的外域兴趣信号中筛选出适合推荐域的有用兴趣信号。以

上三个模块协同建模了适合推荐域的用户完整兴趣信号。

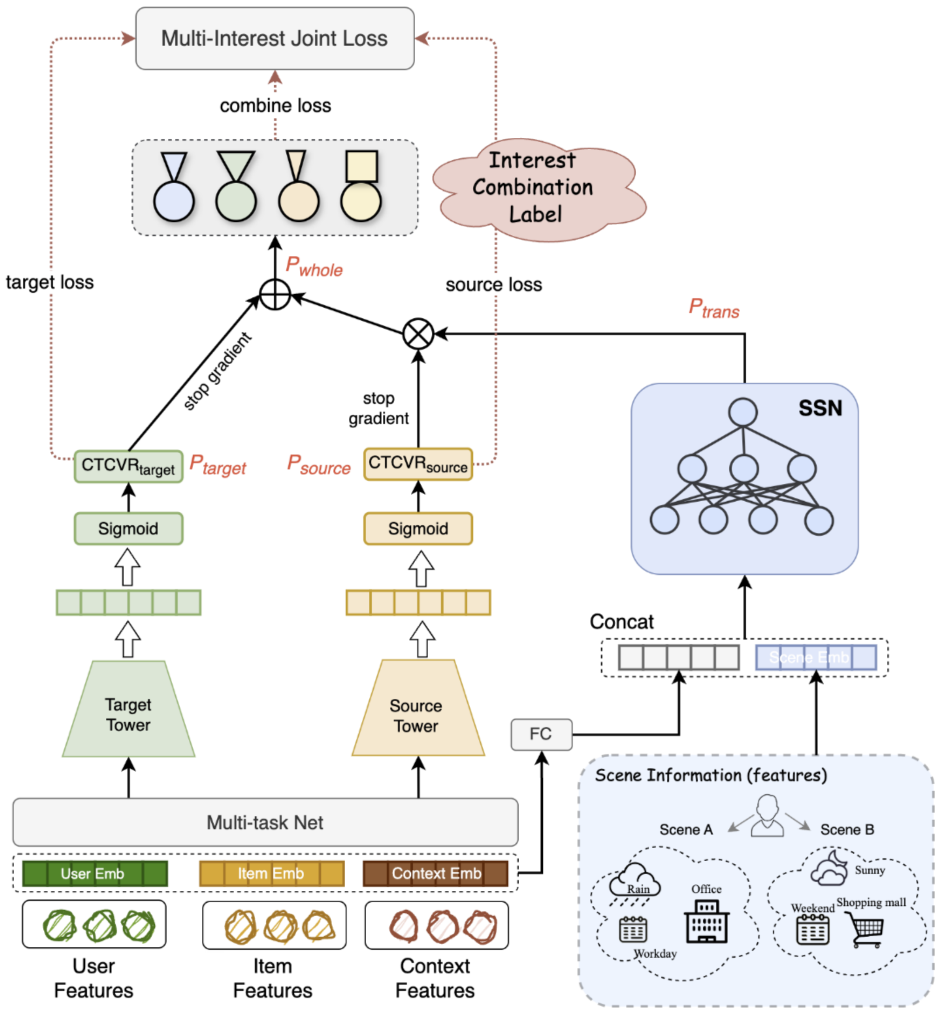


图 6. EXIT 框架图

从全局视角来看，显式兴趣迁移跨域推荐框架EXIT将适合推荐域的用户完整兴趣P_{whole}表示为：

$$P_{whole} = P_{target} + P_{source} * P_{trans}$$

其中P_{target}, P_{source}和P_{trans}均由模型建模输出，P_{target}, P_{source}分别表示兴趣建模网络学习到的用户目标域兴趣和源域兴趣，P_{trans}表示场景选择门控网络输出的当前场景下的兴趣迁移概率，是源域兴趣是否适合迁移到目标域的量化表示。

2.3.2 兴趣组合标签

我们使用监督学习来建模跨域兴趣迁移过程，监督学习需要显式的标签。P_{target}, P_{source}表示的目标域兴趣和源域兴趣的监督信号可直接由用户是否购买商品的目标域标签y^t和源域标签y^s给出。我们构建了兴趣组合标签（Interest Combination Label, ICL）作为兴趣迁移概率P_{trans}的监督信号，在模型训练过程中，通过最小化兴趣组合标签对应的损失，P_{trans}可以达到"当源域兴趣适合目标域时趋于1，而在不适合时趋于0"的效果，从而实现对源域兴趣的筛选，只将有用的兴趣信号迁移到目标域。根据用户在具体时空场景下是否购买商品的目标域标签y^t和源域标签y^s，兴趣组合标签y^{icl}的构造方式如下表1所示：

Table 1: Interest Combination Label

y^t	y^s	y^{icl}
0	0	0
1	0	1
0	1	η
1	1	2

举例来说，用户在推荐域的场景 A 中购买了一件商品，而在场景 B 中曝光了这件商品但用户没有购买（如用户在中午饭点购买了餐饮外卖，但下午对餐饮外卖曝光未点），兴趣组合标签表达的含义是只有场景 A 的情况下用户在源域表达的对该商品的兴趣可以迁移到目标域，使得模型可以学习到源域兴趣在不同的时空场景下具有不同可迁移性，并泛化到相似用户或相似商品。

针对上述场景B中用户在推荐域对一件商品曝光未点或曝光未买，而在其他域有购买的情况（即$y^t=0, y^s=1$），初始的兴趣组合标签构造方案是$y^{icl}=0$，即场景B中不考虑源域兴趣向推荐域迁移。考虑如下场景：有大量用户在源域和目标域中都购买了同一件商品，即使当前用户在目标域没有购买该商品，也不一定代表对该商品没有兴趣，可能是该商品曝光位置靠后或者用户浏览时长过短的原因。且大量用户在多个展位均有表达的兴趣信号往往是热门商品，此类商品造成负迁移的可能性较低，因此我们希望向当前用户"种草"这一商品。在这种情况下，通过挖掘协同过滤信息得到的群体一致性兴趣可以更准确地反映跨领域兴趣的可转移性，因而此种情况下我们使用群体一致性兴趣η表示源域适合转移到目标域的兴趣，兴趣组合标签则为$y^{icl}=\eta$。商品i的群体一致性兴趣η_i使用以下公式计算：

$$\eta_i = \frac{U^t_{pay_i} \cap U^s_{pay_i}}{U^t_{pay_i} \cup U^s_{pay_i}}$$

其中，$U^t_{pay_i}$和$U^s_{pay_i}$分别表示在目标域和源域购买了商品i的用户集合。值得注意的是，上述公式可以通过Spark工具高效计算，并且在实际应用部署时只需天级更新，不会带来额外的计算和时延压力。我们也通过消融实验验证了使用群体一致性兴趣具有更好的离在线效果。

2.3.3 场景选择网络

根据前述的落地难点分析，用户在美团的兴趣偏好与特定的时空场景具有强相关性，因此场景信息在用户兴趣预估过程中发挥着重要作用，也是影响跨域兴趣迁移的重要因素。我们在 EXIT 框架中提出了场景选择网络（Scene Selector Network，SSN），来与兴趣组合标签相配合，使模型学习到在不同的细粒度场景下对不同业务输出差异化的兴趣转移强度，以确保兴趣迁移概率与用户兴趣的实际变化相匹配。例如，在 < 工作日，中午，写字楼，白领 > 的场景下，我们期望模型预估的外卖业务的兴趣迁移概率较高，而该场景下到店服务的迁移概率则应相对较低。相反，在 < 周末，下午，购物中心，白领 > 的场景下，模型预估的到店服务的兴趣迁移概率则应该较高。

SSN首先将用户Embedding向量E^U、商品Embedding向量E^M和上下文Embedding向量E^C进行拼接，随后经过一个全连接层得到压缩后的隐向量E^{hid}。SSN将压缩得到的隐向量E^{hid}和时空场景Embedding向量E^{scene}作为输入来增强场景信息的利用，并最终输出兴趣转移概率P_{trans}，其公式如下：

$$\mathbf{E}^{hid} = FC([\mathbf{E}^U||\mathbf{E}^M||\mathbf{E}^C])$$

$$\mathbf{E}^{scene} = [\mathbf{E}(F_{age})||\ldots||\mathbf{E}(F_{type})||\ldots||\mathbf{E}(F_{hour})])$$

$$\mathbf{H}^{SSN} = MLP([\mathbf{E}^{hid}||\mathbf{E}^{scene}])$$

$$P_{trans} = Sigmoid(FC(\mathbf{H}^{SSN}))$$

其中E^{scene}是由用户Profile、商品类型和当前时间等各种场景特征的Embedding拼接而成，F_{age}, F_{type}和F_{hour}表示不同的场景特征。

2.3.4 离在线实验结果

为了验证所提出的显式兴趣迁移框架的效果，我们选取了多个基准模型进行了大量的离在线对比实验。基准模型包括经典推荐方法和跨域推荐方法两类。对比实验结果如下表所示，结果表明我们所提出的显式框架不仅能更好地建模用户兴趣偏好（离线 AUC 和在线 CTCVR 均有提升），也能防止负迁移问题（负反馈率 NFR 下降，负反馈率表示负反馈 PV/ 总曝光 PV，衡量用户对推荐结果的满意度）。EXIT 框架在美团首页推荐系统的全量上线取得了全域用户建模单次迭代的最大线上收益。

Table 2: Offline and online experimental results on different methods. "↑": the larger the better. "↓": the smaller the better. Underline: runner-up.

Method	Offline Metric		Online Metric		
	AUC↑	Logloss↓	CTCVR↑	GTV↑	NFR↓
LR [12]	0.8984	0.06861	-	-	-
DNN [13]	0.9052	0.06677	-	-	-
DeepFM [14]	0.9069	0.06685	+0.278%	+1.374%	-1.245%
DCN [15]	0.9067	0.06675	+0.448%	+0.896%	-0.921%
MV-DNN [16]	0.9008	0.06820	-0.024%	+0.858%	+2.025%
CoNet [6]	0.9073	0.06662	+0.540%	+1.616%	+0.272%
MiNet [7]	0.9078	0.06655	+0.650%	+2.174%	-0.251%
STAR [9]	0.9069	0.06672	+0.520%	+1.199%	-0.781%
UniCDR [11]	0.9075	0.06668	+0.775%	+2.718%	-1.211%
EXIT(ours)	**0.9098**	**0.06637**	**+1.336%**	**+4.109%**	**-6.937%**

我们对 EXIT 框架中的主要模块进行了消融实验，如下表 3 所示，实验结果表明框架中的各个模块都是有效的。兴趣组合标签 ICL 发挥了最显著的作用，不仅有助于更精准地建模用户兴趣，而且在防止负迁移方面起着关键作用。

我们也通过 ICL 的两种变体研究了群体一致性兴趣的作用：（1）$ICL(\eta=0)$ 表示 ICL 完全基于用户的个性化兴趣构建，而不考虑群体一致性兴趣；（2）$ICL(y^t+\eta)$ 表示在所有场景下直接使用群体一致性兴趣来代表跨域兴趣的可迁移性。表 3 的实验结果表明，忽视群体一致性兴趣或仅依赖于群体一致性兴趣来建模跨域兴趣迁移都会导致模型性能下降。实验结果证实了 ICL 构建方法的合理性，为了准确表示跨域兴趣的可迁移性，在用户的源域和目标域兴趣不一致时使用群体一致性兴趣；而当两个领域的兴趣相一致时，基于用户的个性化兴趣来决定是否进行跨越兴趣迁移。

Table 3: The ablation study for the components in EXIT.

Ablations	Offline Metrics		Online Metrics		
	AUC↑	Logloss↓	CTCVR↑	GTV↑	NFR↓
w/o ICL	0.9055	0.06749	+0.581%	+1.226%	+0.779%
w/o SSN	0.9075	0.06661	+1.092%	+1.971%	-4.159%
w/o joint loss	not converged		-	-	-
ICL($\eta = 0$)	0.9081	0.06643	+0.874%	+3.053%	-5.231%
ICL($y^t + \eta$)	0.9074	0.06658	+0.544%	+1.814%	-3.670%
EXIT(ours)	**0.9098**	**0.06637**	**+1.336%**	**+4.109%**	**-6.937%**

为了更好地理解兴趣迁移概率的作用，我们可视化了模型在线 Serving 时外卖业务的平均兴趣迁移概率在一天 24 小时内的变化，如下图 7 所示。可以发现外卖业务的兴趣迁移概率在早餐、午餐和晚餐时间出现了三个峰值，这表明模型已经学会在不同的场景中自适应地调节跨领域兴趣迁移的强度。我们的显式建模框架能够基于细粒度的场景选择性地筛选出适当的源域兴趣信号，从而实现我们的建模目标。

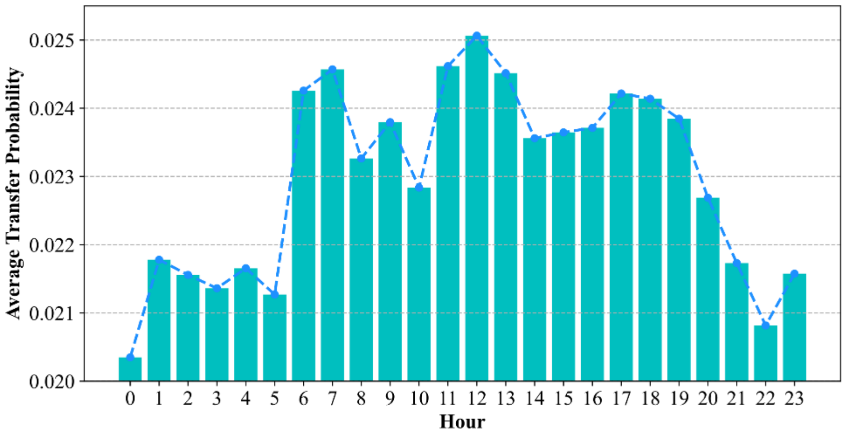


图 7. 外卖业务的兴趣迁移概率在一天 24 小时内的变化情况

2.4 全域全链路统一建模及感知增强

召回和排序模型虽已通过「样本 Label 改写」及「显式兴趣迁移框架」等多期全域建模升级具备了全域感知能力，但召排全域体系建设仍然具有**外域信号不全、全域特征缺乏、链路一致性不足**的问题：

- **外域信号不全**：一方面，全域建模仅在推荐曝光样本空间中进行，对于未在推荐域曝光而在其他域有点击或支付的用户行为无法感知；另一方面，过往全域建模遵循业界做法均仅引入了外域正向信号，但负向信号也能反映用户的兴趣偏好。这在一定程度上造成了部分有效外域信号的丢失。
- **全域特征缺乏**：召排模型已从样本、模型结构方面增强了全域感知能力，但作为模型迭代三件套之一的特征工程未有全域角度的建设和迭代。

- **链路一致性不足**：当前召回和排序的全域样本割裂，在外域信号引入、全域特征建设等方面难以同步及统一，这种链路不一致性可能会成为单模块效果的掣肘（如根据全域兴趣召回的 item 不能在排序阶段透出），难以完全发挥全域建模的作用。

为全方面增强推荐整体链路的全域感知能力，构建统一的召排全域体系，我们进行了全域、全链路、全供给统一样本建设，并基于新的全域样本继续升级召排模型，通过更全面信号和链路一致性进一步发挥全域建模的效果。

2.4.1 全域、全链路、全供给统一样本建设

基于当前召回全域体系在样本层面存在的不足，我们对召回及排序模型样本进行统一升级，具体动作如下：

- **正样本扩充**：过往全域信号引入对于不在曝光样本空间但在源域有支付的行为数据未进行考虑。为引入更充分的源域兴趣信号，对于与猜喜曝光样本空间无法关联的全域支付行为数据，通过限制一定范围的时间窗口（－δ_time=< 用户其他展位的下单时间 − 猜喜请求时间 <= δ_time）方式与用户在猜喜的请求相关联，从而扩充对应的支付正样本，并且该样本的 User 侧特征与用户下单时的真实特征相差不大，有助于进一步抑制全域建模中的负迁移问题。
- **负样本扩充**：业界在全域建模领域的通常做法是只引入用户正向行为数据，但负向信号也能反映用户的兴趣偏好，知道用户不喜欢买什么可以帮助模型做排除选项，因此我们对其他展位负样本做随机采样添加到粗排模型训练样本中。增加其他展位的随机负样本也可以帮助解决粗排模型训练和预测样本空间不一致的样本选择偏差（Sample Selection Bias，SSB）问题。实践中为解决 SSB 问题，我们一方面继续增加未加载样本（粗排出口 Item）作为负样本，另一方面在全体样本空间下通过城市下带降热的随机负采样方式进一步扩充负样本。

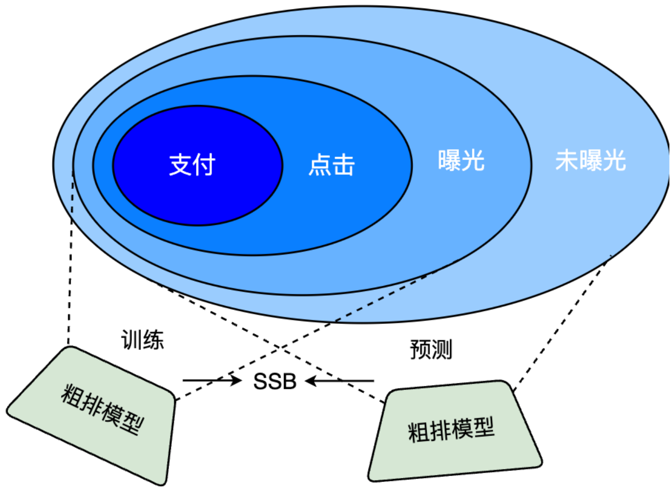


图 8. 粗排样本选择偏差 SSB 问题

- **全域特征体系建设**：在特征工程方面进行全域升级，构建搜索、频道区等其他展位相关特征，基于全域行为的 UI 统计特征等，在特征侧增强模型的全域感知能力。例如从全域行为数据中提取用户交互特征，如点击率、下单率、停留时间等，基于全域行为构造 UI 统计特征如频次统计、时序特征等。
- **召排样本统一**：统一召回及排序样本，使得召排模型在外域信号引入、全域特征建设等方面保持同步，召排全域体系的链路一致性更高，更好发挥全域建模的效果。

全域、全链路、全供给统一样本的示意图如下：

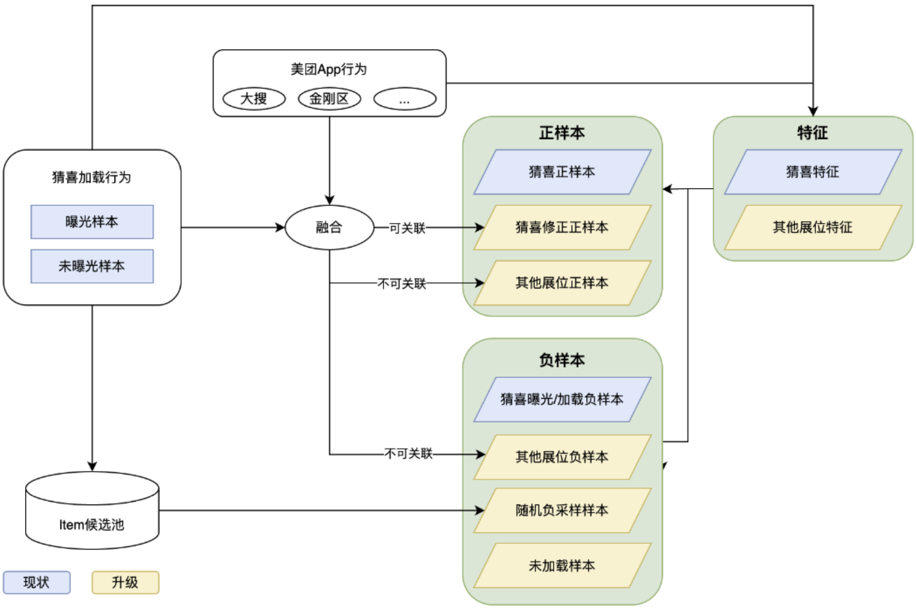


图 9. 全域、全链路、全供给统一样本

2.4.2 全域感知增强建模

升级后的召排统一样本具有更完备的全域正负信号及全域特征体系，为适配新样本增强链路全域感知能力，同时增强链路一致性以更大程度发挥全域建模效果，我们基于新样本对粗排模型进行了重构，以统筹兼顾全域兴趣预估与链路一致性任务。我们通过对全域兴趣预估与一致性联合建模，在模型中同时建模曝光、点击、支付、时长、互动和一致性（Pointwise+Pairwise）等目标，使模型同时具有校准能力和排序能力，且两类能力能互相补充。全域兴趣预估与一致性联合建模的多目标逻辑示意图如下：

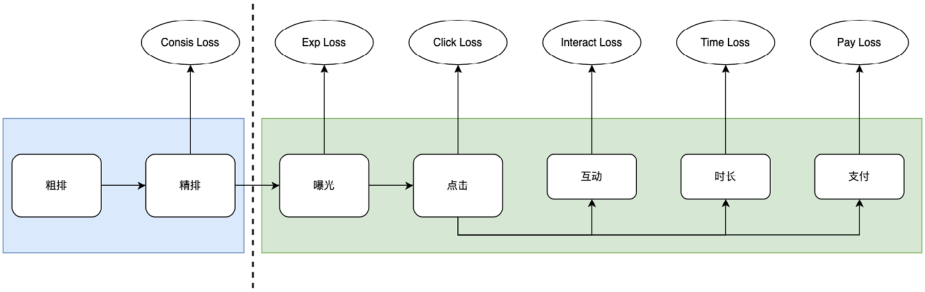


图 10. 全域兴趣预估与一致性联合建模的粗排多目标逻辑

粗排全域兴趣预估与一致性联合建模升级保证了全域信号在推荐系统各链路透出的一致性，起到了承“召回”启“精排”的作用，进一步增强了推荐系统各链路的全域感知能力。基于构建好的全域、全链路、全供给统一样本，我们也在同步升级召回模型及召回全域体系。

三、总结与展望

3.1 总结

总的来说，我们通过多阶段递进式探索验证的方式，在美团首页推荐召排模块引入多展位、多应用渠道的多源用户交互数据，并在落地过程中解决美团多展位、多业务、时空场景强相关性的特点导致的严重跨域信号负迁移挑战。全域用户建模经过多期算法落地已经取得显著业务收益，缓解了首页推荐用户行为稀疏导致的模型训练不充分及用户兴趣预估有偏问题，大幅提升了首页猜喜推荐系统在全面理解用户行为模式上的能力。此外，我们在排序模块的部分创新成果也已在 CIKM2024 会议上发表。

全域用户建模在美团首页推荐的成功，验证了全域感知的重要性，增强推荐系统的全域感知能力是一条可行的路径。我们将进一步结合美团业务特点与业界先进技术，探索并创新更多全域用户建模在推荐系统落地的有效方案。

3.2 后续计划

- **引入外域点击信号**。当前我们主要引入外域支付信号到首页猜喜来进行算法建

模，这是由于支付信号相比点击信号来说兴趣表达更强烈，且平台的目标以支付为主。后续我们进一步考虑引入外域点击信号，点击信号相比支付信号更为丰富，但噪声也更大，对数据清洗的要求更高，也更容易导致负迁移问题。这对我们的算法建模提出了更高的要求，团队也会持续在解决跨域负迁移问题上进行更多的探索和创新。

- **升级显式兴趣迁移范式**。我们提出的用于 CTCVR 预测的显式兴趣迁移框架，在 Item 粒度考虑兴趣的跨域迁移，要求源域和目标域具有重叠的 Item。我们计划对这一范式进行升级以支持源域和目标域不具有重叠 Item 的场景，扩大显式兴趣迁移框架的应用范围，并继续发表创新性成果。
- **探索基于全域行为的生成式推荐范式**。受 LLM 取得巨大成功的启发，最近 Meta 的相关同行重新审视了现代推荐系统中的基本设计选择，将整个推荐问题建模成类似 GPT 的生成式问题，并拥有和 LLM 一样的 Scaling Law 规律。我们也在基于用户全域行为数据构造用户超长行为序列，进行生成式推荐范式的探索与落地。

四、招聘信息

团队招聘火热进行中，以上讨论的全域用户建模和生成式推荐都在进一步升级迭代中，诚邀各路英才加入。简历请发送至：wangjinpeng04@meituan.com。

五、参考文献

[1] FengZhu, YanWang, ChaochaoChen, JunZhou, LongfeiLi, and GuanfengLiu. 2021. Cross-Domain Recommendation: Challenges, Progress, and Prospects. InProceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021. ijcai.org, 4721 - 4728.

[2] Shlomo Berkovsky, Tsvi Kuflik, and Francesco Ricci. 2007. Cross-domain mediation in collaborative filtering. In User Modeling 2007: 11th International Conference, UM 2007, Corfu, Greece, July 25-29, 2007. Proceedings 11. Springer, 355 - 359.

[3] Shulong Tan, Jiajun Bu, Xuzhen Qin, Chun Chen, and Deng Cai. 2014. Cross

domain recommendation based on multi-type media fusion. Neurocomputing 127 (2014), 124 – 134.

[4] Ming He, Jiuling Zhang, Peng Yang, and Kaisheng Yao. 2018. Robust Transfer Learning for Cross-domain Collaborative Filtering Using Multiple Rating Patterns Approximation. In Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining, WSDM 2018, Marina Del Rey, CA, USA, February 5-9, 2018. ACM, 225 – 233.

[5] Feng Yuan, Lina Yao, and Boualem Benatallah. 2019. DARec: Deep Domain Adaptation for Cross-Domain Recommendation via Transferring Rating Patterns. In Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019. ijcai.org, 4227 – 4233.

[6] Guangneng Hu, Yu Zhang, and Qiang Yang. 2018. CoNet: Collaborative Cross Networks for Cross-Domain Recommendation. In Proceedings of the 27th ACM International Conference on Information and Knowledge Management, CIKM 2018, Torino, Italy, October 22-26, 2018. ACM, 667 – 676.

[7] Wentao Ouyang, Xiuwu Zhang, Lei Zhao, Jinmei Luo, Yu Zhang, Heng Zou, Zhaojie Liu, and Yanlong Du. 2020. MiNet: Mixed Interest Network for Cross-Domain Click-Through Rate Prediction. In CIKM '20: The 29th ACM International Conference on Information and Knowledge Management, Virtual Event, Ireland, October 19-23, 2020. ACM, 2669 – 2676.

[8] FengZhu,ChaochaoChen,YanWang,GuanfengLiu,andXiaolinZheng.2019. DTCDR: A Framework for Dual-Target Cross-Domain Recommendation. In Proceedings of the 28th ACM International Conference on Information and Knowledge Management, CIKM 2019, Beijing, China, November 3-7, 2019. ACM, 1533 – 1542.

[9] Xiang-RongSheng,LiqinZhao,GuoruiZhou,XinyaoDing,BindingDai,Qiang Luo, Siran Yang, Jingshan Lv, Chi Zhang, Hongbo Deng, and Xiaoqiang Zhu. 2021. One Model to Serve All: Star Topology Adaptive Recommender for Multi- Domain CTR Prediction. In CIKM '21: The 30th ACM International Conference on Information and Knowledge Management, Virtual Event, Queensland, Australia, November 1 – 5, 2021. ACM, 4104 – 4113.

[10] Pan Li and Alexander Tuzhilin. 2020. DDTCDR: Deep Dual Transfer Cross Domain Recommendation. In WSDM '20: The Thirteenth ACM International Conference on Web Search and Data Mining, Houston, TX, USA, February 3-7, 2020. ACM, 331 – 339.

[11] Jiangxia Cao, Shaoshuai Li, Bowen Yu, Xiaobo Guo, Tingwen Liu, and Bin Wang. 2023. Towards universal cross-domain recommendation. In Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. 78 – 86.

[12] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In Proceedings of the 16th International Conference on World Wide Web, WWW 2007, Banff, Alberta, Canada, May 8-12, 2007. ACM, 521 – 530.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In Proceedings of the 26th international conference on world wide web. 173–182.

[14] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: A Factorization-Machine based Neural Network for CTR Prediction. In Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017. ijcai.org, 1725–1731.

[15] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In Proceedings of the ADKDD' 17, Halifax, NS, Canada, August 13 – 17, 2017. ACM, 12:1–12:7.

[16] Ali Mamdouh Elkahky, Yang Song, and Xiaodong He. 2015. A Multi-View Deep Learning Approach for Cross Domain User Modeling in Recommendation Systems. In Proceedings of the 24th International Conference on World Wide Web, WWW 2015, Florence, Italy, May 18-22, 2015. ACM, 278–288.

信息流广告预估技术在美团外卖的实践

1 信息流广告业务及预估技术现状

1.1 信息流广告业务特点

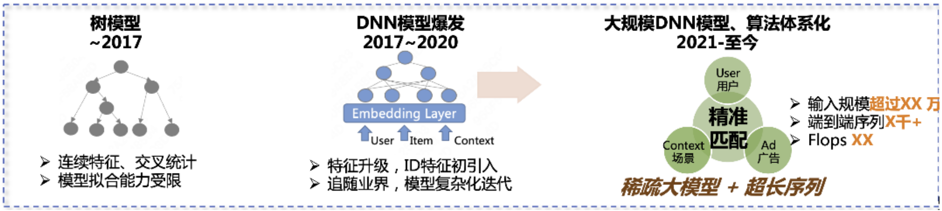
目前，美团外卖的广告主要包括信息流广告、搜索广告、营销广告、展示广告等等。外卖业务都有着典型的业务特点：

- **用户行为连贯性强**：用户用餐意图明确，一般在 10 分钟内完成，UV 成单率较高。
- **展示信息丰富**：卡片信息覆盖了评分、评价、优惠、配送等多种信息，对用户的决策影响较强。
- **文本信息多**：在电商场景中，商品作为候选图片往往占据很大的决定因素，而在外卖场景下，商家作为候选更加复杂，商户名称、评价、热销菜品等文本信息能够影响用户做出决策。

1.2 技术概况及演进阶段

这里先来介绍一下预估技术的现状。从技术层面，下图展示了广告投放系统的整体的流程：

总的来说，外卖广告系统跟我们在业界的搜推广系统是比较相似的，召回、粗排、精排以及各种机制。但外卖广告和业界场景区别较大的地方在于召回，因为它是基于位置服务（LBS）的，这个过程本身带有一定的约束。因此，我们会在精排和机制层面投入更多的算力和资源，以期给整体链路带来最大化的提升。

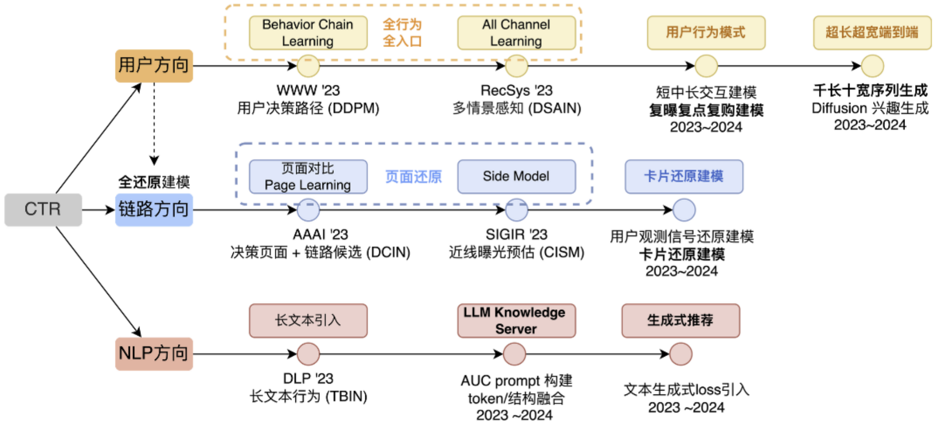


在过去六七年中，到家广告预估算法历经了三个发展阶段，第一个阶段就是树模型，包括连续特征、交叉统计等等，当时的模型拟合能力是比较受限的。第二个阶段是从 2017 年开始到 2020 年，在这个阶段 DNN 模型开始爆发，我们进行了特征升级，也开始紧随业界步伐，引入更加复杂的模型，不断提升业务效果。第三个阶段就是 2021 年至今，我们主要方向是稀疏大模型 + 超长序列，进一步实现业务效果的提升。

1.3 预估技术现状

在信息流广告预估技术层面，主要探索方向为用户方向、链路方向以及 NLP 方向（如下图所示）。当然，如果这张图更全面一些，还会包括交叉方向，还有多场景多目标等等。而没有选择其他方向，主要是因为就交叉方向而言，我们发现随着互联网行业的不断发展，用户的行为会越来越多且更加复杂，而交叉方向仅仅能带来 Context 级别的深度学习能力，也就无法持续的成为效果的来源。另一个方面，虽然交叉技术也在发展，但发展方向上也是从 ID matching 到 Sequence Matching，单纯平铺类

类别特征的交叉模型能力发展有限。综合多方面的因素，我们并没有将交叉作为一个长期的方向进行迭代。



还有一个是多场景方向，其实此前我们在这个方向也做过一些迭代带来了一波效果，但是后来我们发现，这种技术更加适合多个小场景的链接。如果你所服务的业务仅仅只有 1~2 个比较大的场景，这些场景本身的用户需求差异性、展示形态、候选供给差别又不大，就不太能发挥这个方向的技术能力及作用。

我们整体的思路是从用户的元素匹配、页面匹配，到路径匹配，最后到长期的兴趣匹配。本质上都是在做不同层面的用户匹配相关的工作。其中，元素匹配、页面匹配归到了链路方向。原因是链路方向更多的都是在解决“看不见的问题”，然后再通过这些“被看见”的信息去做相应建模，所以我们将链路方向单独列了出来。

- 在用户方向，我们也大概经历了三个阶段，第一个阶段，要从原始单点、单入口的行为，向全行为、全入口进行扩张；第二个阶段，是在已有的输入的情况下，去探索更多的行为模式；第三个阶段，我们主要是做一些自动化模式提取，或者说网络自动拟合行为的能力做到更强。
- 在链路方向，主要关注两件事，一个是页面还原，一个是卡片还原，通过算法和工程能力来还原用户“所见”到模型决策中。
- 在 NLP 方向，过去我们还有一个方向叫多模态，但是客观来讲，随着 LLM 的

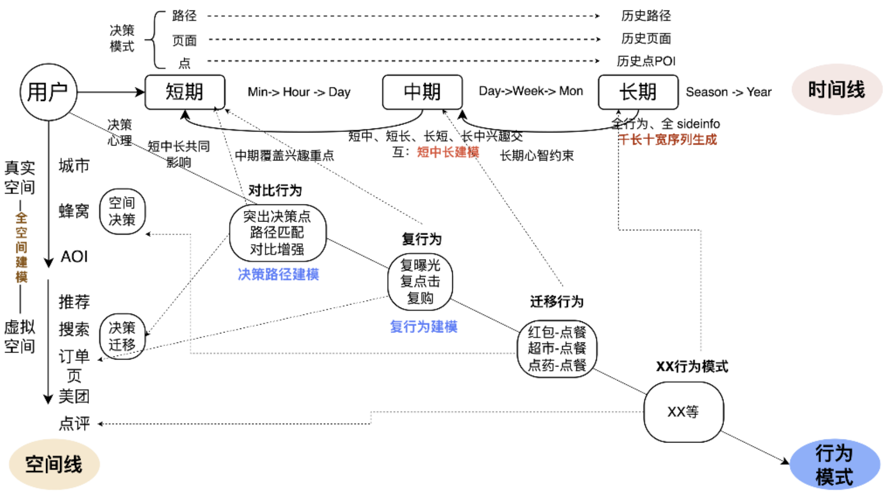
火爆，外界的技术也给我们更多的输入，因此我们将 LLM IN CTR 单独列出来作为一个主要的技术方向。

2 信息流广告在美团的实践

2.1 用户建模思路概览

用户方向整体又拆解为三个反向，第一个是时间线，第二个是空间线，第三个是时间和空间共同作用下的行为模式。当时我们在拆解的时候，也参考了业界学界主要迭代的方法，包括 Session 建模、超长行为建模、多行为建模、长短期建模等。以学界和业界为基础，结合业务问题、特点，将技术和业务结合的更好，有了以下的技术拆解。

在时间线上，我们认为长短期的多 Level 融合更加重要。一方面是用户兴趣在不同级别“片段”上关注点有显著差异，比如页面倾向比较、路径上的兴趣更连续、用户会连续吃一段时间轻食等，我们需要将这种在不同级别片段上的用户行为模式提取出来。因此，一方面我们通过更多页面和路径的方式将短期和长期进行联合；同时，我们通过增加日、周级别的中期兴趣，将短中、短长进行交互，增强时间线行为上的连接。另一方面，在模型上增加一些端到端的方式，自动化的将行为规律挖掘出来。这是时间线要解决的关键问题。



在空间线上，真实物理空间维度下，我们面对的问题比较明确，在不同的位置下，比如上班的时候和在家的时候，人的兴趣其实是不完全一样的，我们根据空间的位置为大家进行推荐。在虚拟空间下，比如用户在使用美团 App 和大众点评 App 等不同入口，人的兴趣和意图也会发生较大变化。一个显著的例子是，用户在首页和会员入口上对优惠的关注区别较大。空间线解决的问题是结合真实空间、虚拟空间，去判断用户的真实的意图或者行为模式。

第三条线，就是跟业务进行结合，比如用户在 App 上进行了一些操作（领取了红包），那么这个行为会对点餐有什么影响。本质上是模型在理解用户进行了一些操作行为后，会对接下来的行为产生哪些影响，进而模型能学习到不同的用户行为模式、更好的预测用户的行为。以上就是我们用户建模的整体思路。

2.1.1 决策路径建模

本部分会介绍一下决策路径建模，第一个核心的问题就是，DIN 单点匹配忽略了什么？单点匹配，我们认为是忽略了前序行为对用户后续行为的一个影响。对绝大多数电商业务来说，用户在一段时间内的行为是具备一定的连贯性的，我们可以根据用户的历史行为数据，对接下来的行为做出预测。这里有两个挑战，第一，如何构建核心路径；第二，如何解决路径本身的噪声、稀疏性、匹配等问题。我们的解法主要有三点：

第一，Path Enhance Module（PEM）提取核心路径。

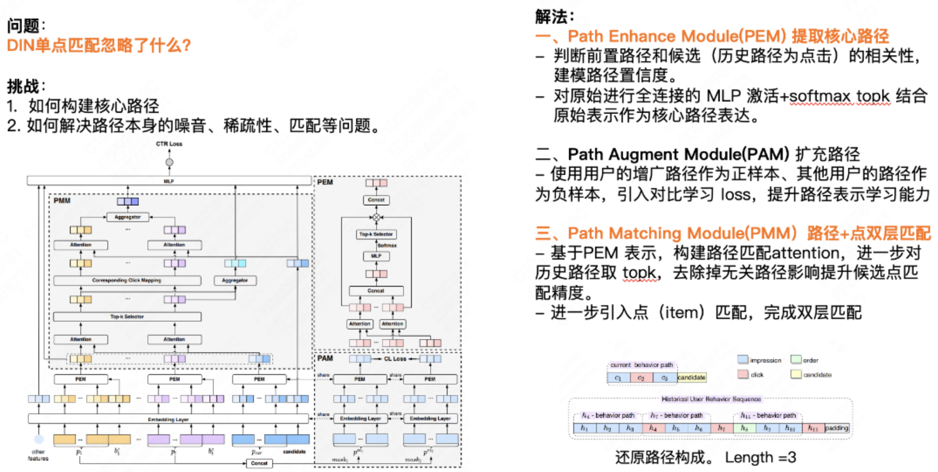
- 判断前置路径和候选（历史路径为点击）的相关性，建模路径置信度。
- 对原始进行全连接的 MLP 激活 +Softmax Top K 结合原始表示作为核心路径表达。

第二，Path Augment Module（PAM）扩充路径。

- 使用用户的增广路径作为正样本、其他用户的路径作为负样本，引入对比学习 Loss，提升路径表示学习能力。

第三，Path Matching Module（PMM）路径 + 点双层匹配。

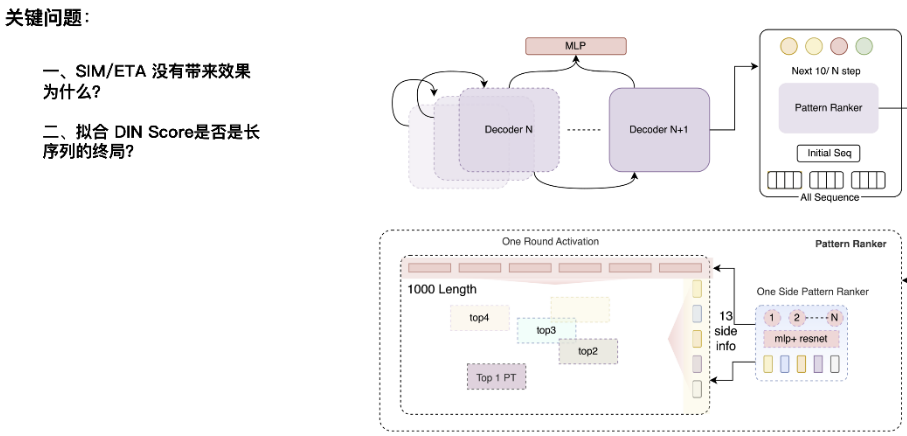
- 基于 PEM 表示，构建路径匹配 Attention，进一步对历史路径取 Top K，去除掉无关路径影响提升候选点匹配精度。
- 进一步引入点（item）匹配，完成双层匹配。



2.1.2 用户行为超长超宽建模

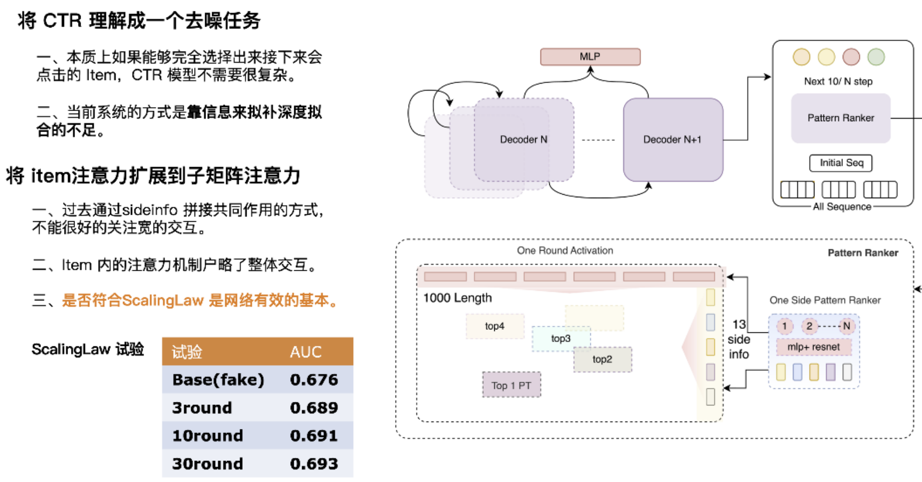
相信很多同学都知道超长建模，基本上是通过聚类、局部 hash 等近似技术来进行实现。要引入超宽建模，本质上是因为我们需要将所有的输入都放到一起，所以需要一个更大更复杂的模型去把事情给处理了。不过，在实际中我们没有完全实现，因为目前的算力还是无法支持的。我们做了一个折中，长度是 1000（Length）的级别，宽度我们目前做到 10+ 这个级别，离线可以支持到更大的规模、到万级别效果也有较大提升，但是迭代效率、线上压力都会有较大限制。

这里也面临 2 个问题：第一个问题就是 SIM/ETA 为什么没有带来效果？这个方向，最早是电商平台提出来的，SIM 主要是 Hard 过滤，比如通过用户在网站浏览鞋相关的行为时，他们也会看其他各种各样的东西，它通过硬过滤，能够把跟「鞋」相关的产品能过滤出来，过滤掉和鞋无关的噪音，学习到用户在鞋上的偏好。点外卖相对不太一样，对一个候选汉堡来讲，通过汉堡品类过滤掉和汉堡无关的行为，这会损失较多用户口味信息。这是业务差异性带来的。



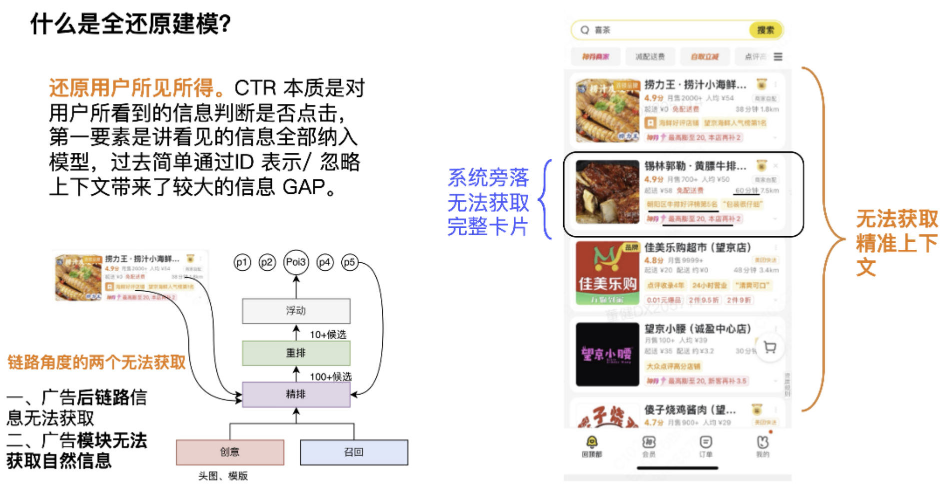
第二个问题是，拟合 DIN Score 是否是长序列的终局？之前业界有一篇文章认为 DIN Score 是基准，把它线性扩展就能带来效果，进一步扩展到万级别或者十万级别把效果推到最大化。但是通过实验，我们的 CTR 场景线性扩展到超长级别并没有持续带来效果，反而到一定长度有所下降。我们认为 DIN 网络本身的去噪能力不是很强，或者说它去提取出 Label 结果的结构能力并不够强。如果它不是一个特别强的网络，做更大的扩展的时候，它所能容纳的信息是比较有限的。

我们可以把 CTR 理解成一个去噪任务，本质上是根据用户历史和当前场景 Match 用户和候选的过程。我们发现，如果能预测出精准或者去除掉所有噪声，比如拿穿越信息 Label POI 与 Target 进行匹配，使用简单的网络也可以有很高的 AUC。因此，我们认为完美的 CTR 网络应该是一个强预测网络 + 弱匹配网络的组合。预测网络应该是一个能力非常强的网络，能够进行多层的叠加，把信息进行萃取得到一个预测更加精准的结果，来与 Target 进行匹配。所以我们设计了一个多层的 Decoder，每一层的 Decoder 都能做信息的整合。通过不断的选取有效矩阵、反复叠加有效信息，来使得信息更加精准。这里我们做了一组 Scaling Law 实验，通过叠加多轮网络来验证结果的有效性。可以看到随着轮数的增加，网络学习用户行为的能力（AUC 逐层提升）也有所增加。



2.2 全还原建模

首先，什么是全还原建模？我们给的一个定义是还原用户所见所得。CTR 任务是根据用户看到的信息来判断用户是否点击，最重要的一点是将看见的信息全部纳入到模型之中，过往简单的通过 ID 表示建模忽略了上下文及展示的信息，带来了较大的信息 GAP。



第一个视角，上下文卡片无法获得。上下文信息对当前候选、当前卡片的 CTR 相当重要。有的同学可能会认为重排可以搞定这个事情，但我们始终认为，上下文信息属于链路信息，我们认为每个模块都需要去学习上下文信息，当然每个模块可能学习的重点不一样，而且实际上都能带来一定的效果。第二个视角，我们从算力的角度，因为预估侧的算力比较高，其影响的范围会更大，实际上也能够带来更多的效果空间。

再看左下角的图，从链路视角来说，对于预估模块、广告模块模块存在两个无法获取。第一个是广告的后链路信息无法获取，这里包括了展示的配送信息、配送费、准确的优惠信息等；第二个是自然的信息无法获取，这里包括了自然的上下文。因此，还原从另一个角度说是，如何打破链路的束缚来使用穿越信息。

这就是全还原建模所面临的一些问题，实际上可以概括成两个方向，一个是卡片还原，一个是页面还原。在早期我们做了个空间判断，我们把卡片、页面信息完完整整的放进来，观测 AUC 的提升来判断整体空间，结果表明页面信息有百分点级别、卡片信息有大几个千分点。

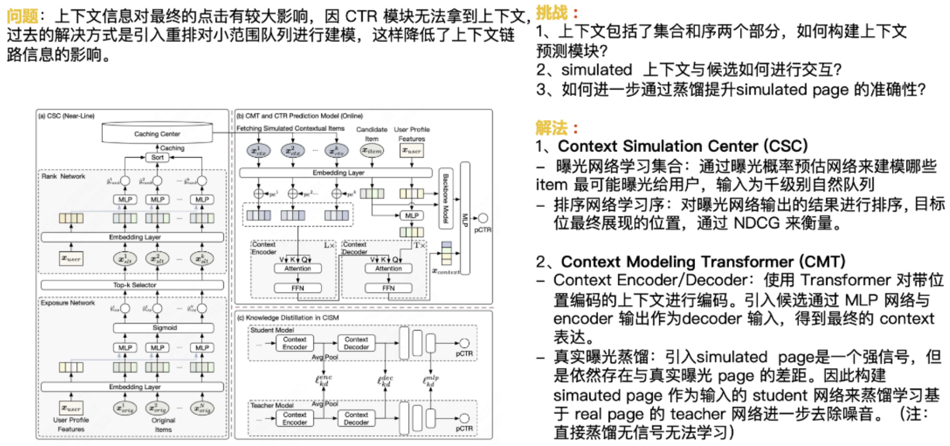
整体解决思路

这里把页面还原和卡片还原展开。首先，从思路上讲，我们主要从算法、工程两个维

度去解。在算法层面，第一个就是去猜页面，最大化利用前链路信息猜页面；第二个是猜元素，创意链路前置，创意优选结果输出给精排。在工程层面，第一个是存页面，引入近线系统，基于旁路系统的 Side Model 端到端预测最终展现信息，最大化利用后链路视野。第二个是存元素，引入近线系统 + 高维 KV，提升元素获取覆盖率至 +100%、准确率 70%+。

前文也提到，上下文信息对最终的点击有较大影响，因 CTR 模块无法拿到上下文过去的解决方式是引入重排对小范围队列进行建模，这样降低了上下文链路信息的影响。接下来，我们面临的挑战就是：

- 上下文包括了集合和序两个部分，如何构建上下文预测模块？
- Simulated 上下文与候选如何进行交互？
- 如何进一步通过蒸馏提升 Simulated Page 的准确性？



我们的解法是：

Context Simulation Center（CSC）

- 曝光网络学习集合：通过曝光概率预估网络来建模哪些 –item 最可能曝光给用户，输入为千级别自然队列。
- 排序网络学习序：对曝光网络输出的结果进行排序，目标位最终展现的位置，通过 NDCG 来衡量。

Context Modeling Transformer（CMT）

- Context Encoder/Decoder：使用 Transformer 对带位置编码的上下文进行编码。引入候选通过 MLP 网络与 Encoder 输出作为 Decoder 输入，得到最终的 Context 表达。
- 真实曝光蒸馏：引入 Simulated Page 是一个强信号，但是依然存在与真实曝光 page 的差距。因此构建 Simauted Page 作为输入的 Student 网络来蒸馏学习基于 Real Page 的 Teacher 网络进一步去除噪音。（注：直接蒸馏无信号无法学习）

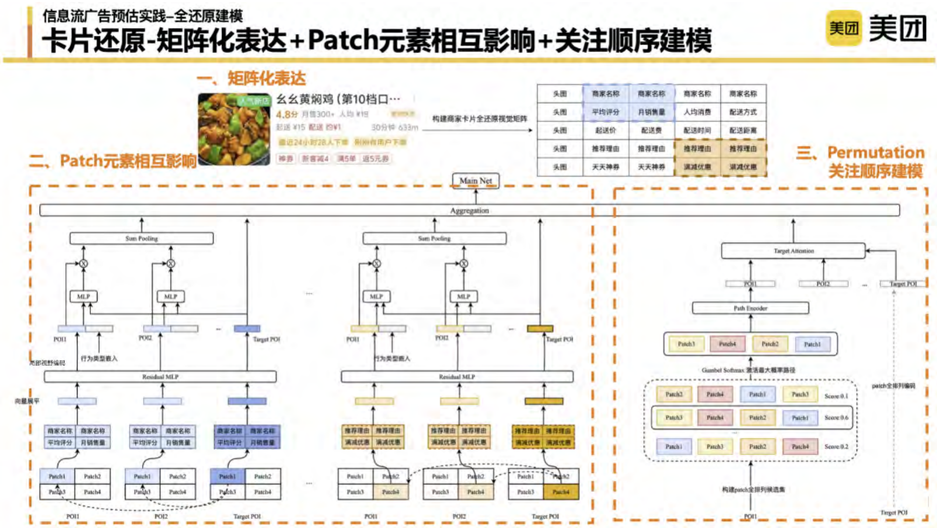
我们引入了缓存的和预测配置策略，加上真实的蒸馏，就帮助我们进一步地去提升效果，这属于页面还原的部分。

卡片还原这部分整体的思路分为三部分：第一个部分，能够拿到卡片信息；第二部分，组成用户看到的卡片；第三部分，通过卡片来做历史兴趣匹配。

卡片方案思考：由于搜推广系统的模块串并行原因，有一部分数据我们是无法获取的，前期我们一直在思考有没有“终极方案”。早期我们就是纯平铺（纯 ID）的方式来还原用户所看到的信息，但是有没有更好的方式呢？比如直接将用户看到的图片引入进来。但是，当前的技术能力不太支持将整个图片完成的记录下来、更不支持图片信息完整准确的建模表达。最终，我们选择了通过矩阵的方式来组成卡片，模拟用户看到的信息。

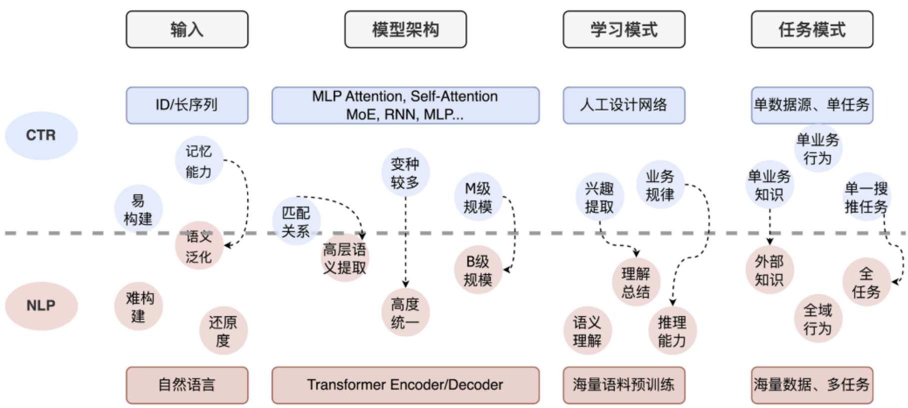
矩阵化表达、Patch 级别建模：首先，我们使用了矩阵化的表达来组成卡片的形状，并构建和用户所见所得的上下元素关系。表示层面，不同的矩阵构建方式对结果会有一定的影响，具体不在这里展开。第二个方面，我们也借鉴了图像领域的一些思想，引入了 Patch 的概念，来帮助我们将图片化作 Token 进一步学好不同展示元素之间的相互影响。在实践的过程中，我们也需要调整一些参数，比如 Patch 到底是 2*2 的，还是 3*3。包括 stride，我们发现 stride 设置的越短，确实能够带来更好的效果。我们在整个 patch 级别匹配的过程中，也做了很多次的实验，初步的结论是，单位置 Patch 和全局 Patch 的匹配，最终的效果比较好。

关注顺序建模：顺序建模是在用户关注哪些元素基础上，进一步模拟用户浏览顺序。按道理说，我们没有眼部监测实际上是拿不到这部分数据的。这里，我们做了一个小 Trick，将这 4 个 Patch 的矩阵进行了全排列，将用户的所有 Patch 级别的路径都列了出来，让模型自己来学习不同排列组合的隐式分数。激活分数最高的的 Patch 顺序组合，通过 Encoder 聚合成关注顺序表达来进一步和 Target 的 POI 的关注顺序组合来进行匹配。



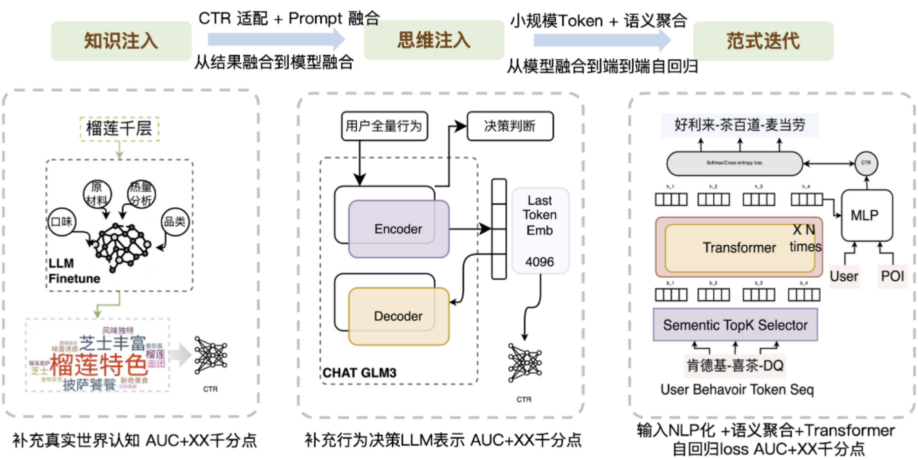
2.3 LLM in CTR

最后分享一下大模型在 CTR 的应用。我们做了一些初步的调研，发现目前很多技术团队整体的思路是差不多的。下图展示了 CTR 任务跟 NLP 任务的对比，可以看到从输入到模型架构，再到学习模式和任务模式，都有较大区别。NLP 任务是自然语言做 Token + 大规模 Transformer + 理解及推理能力，CTR 任务是 ID 输入 + 人工设计网络 + 强记忆能力。同时对于 CTR，大部分业务只用了自身业务数据，是缺乏外部知识和全任务的理解能力的。



所以基于以上几个方面，我们做了三方面的工作：

- 第一层，知识注入，就是要把外部的、真实的、当前 CTR 缺乏哪些知识放到模型中，这部分工作很多公司都在做，这块主要要求的是 Prompt 工程能力。因为生成的结果，不一定是 CTR 需要的，我们就需要做好适配工作。根据 CTR 的特点，就可以将高频词和低频词区分出来，同时也需要做一些 Prompt 融合相关的后处理工作来提升和 CTR 任务的匹配度。
- 第二层，思维注入，就是要把大模型的结构能力给引入进来，或者说将大模型判断过程的引入进来。
- 第三层，范式迭代，最近，Meta 似乎给生成式推荐指出了一条路。我们去年在探索这个方向的时候，主要的思路是把输入的形式进行变化，换成了更小规模的 Token，大概可能只有几万的规模，来解决大规模 Softmax 问题。然后通过 Transform 叠加的方式，结合聚合语义，从模型融合到端到端自回归，让数据能够跑通。我们发现如果噪音特别高的输入，Transformer 并不能处理得很好，但一个相对来讲语义比较明确的信息，Transformer 对上下文理解的性能还不错，因此我们先做了一层语义聚合来降低输入 Token 序列的噪音。总的来说，我们通过小规模 Token，加上语义聚合，结合 Transformer 的架构，给业务效果带来了一波提升。



总结一下，本质上是要把 CTR 不具备的能力通过大模型进行补齐。我们将 CTR 目前不具备的能力，划分成了知识能力、泛化能力和推理能力。对应的，我们也列举了一些我们尝试的结果如下图所示：

03 总结和展望

总的来说，预估的本质上还是要发掘用户的真实需求，我们一方面参考业界，另一方面深入业务，去挖掘更多的用户行为模式，也在探索有没有更自动化的方式将各种用户问题解决掉。还原建模是算法和工程的联合聚力带来的提升，归根结底算法工程的相互结合才能带来更大的改变。

大模型与推荐的结合越来越得到大家的关注，但是客观地讲，这依然是属于一个偏长期的工作，这个时候还是要找到一条可行的路径，不断去优化和提升，如果完全指望用一个“大招”去解决掉所有的问题，会非常困难。端到端推荐大模型是大家共同的期望，但是在这个基础上，我们认为输入规模是效果的保障，算力是以上两者的保障。只有软件和硬件的强强联合，才能赢得未来。

基于多模态信息抽取的菜品知识图谱构建

1. 背景

中国有句古话："民以食为天"。对食物的分析和理解，特别是识别菜肴的食材，在健康管理、卡路里计算、烹饪艺术、食物搜索等领域具有重要意义。但是，算法技术尽管在目标检测 [1]–[3]、通用场景理解 [4][5] 和跨模态检索 [6]–[8] 方面取得了很大进展，却没有在食物相关的场景中取得好的表现，尤其是对烹饪菜肴的相关场景。其核心原因是缺乏细粒度食材的基准，这已经成为该领域发展的瓶颈。

以往的研究主要集中在食物层面的表征学习，如 Food2K 上的食物识别 [9]–[12]，UNIMIB2016 上的食物检测 [13]–[15]。然而，这些方法忽视了菜肴中的食材组成，也不理解食材之间的上下文关系。相比之下，一系列的方法 [16]–[18] 运用 Recipe1M 的"食谱 – 图像"对，实现了跨模态的食谱检索 [16]。

然而，由于缺乏食材边界框的标注，这种类型的研究只能通过三元组建模出整个食物图像和食谱文本之间的关联 [16],[19],[20]。这种限制导致图像区域与食物的一系列食材之间存在模糊的匹配关系，产生虚假相关性 [21]。综上，目前迫切需要一个细粒度的食材级基准，促进复杂的食品场景理解算法的发展，并支持细粒度的任务，如食材检测和跨模态食材检索。

在本研究中提出对于中餐进行理解这一新任务，旨在捕捉中餐图像中食材之间的语义关系，并建立了有关中国菜品理解的新基准。我们大致设定了中餐理解的两个任务：食材检测和食材检索。对于食材检测，目标是确定图像中特定食材的存在并提供精确的定位。对于食材检索，目标是探索不同食材组合与食品图像之间的细粒度对应关系。对中餐的理解扩展了食品相关任务的范围，在食品领域开辟了更广泛的应用。同时，食材的多样外观和它们错综复杂的语境关系，对中餐的理解提出了一个更大的难题。

为了进行中餐理解这一新任务，我们需要构建一个包含食材粒度标注的数据集。然而，由于中餐种类繁多、风格独特，因此在食材标注上面临着巨大的挑战。构建含中餐食材的细粒度跨模态数据集主要有三个难点。

- 首先，相同的食材有不同的名称。图 1.1(a) 说明了这种情况："圣女果"和"小番茄"都是广泛使用的食材名称，它们是同一食材的不同名称，这样的情况使得我们需要花费更多的精力来清除数据集中的模糊标签以及其他噪声。
- 其次，同一植物类食材之间的图像存在细微差异，如"青菜"和"油菜"，"香菇"和"冬菇"，如图 1.1(b) 所示。这些情况对标注人员来说是相当具有挑战性的，他们需要从文本部分获得一些提示。此外，对于下游任务来说，基于视觉特征来区分它们也是相当具有挑战性的。
- 第三，由于烹饪方法的原因，中国菜肴的食材通常分散在图像中。如图 1.1 所示，碎片化食材通常缺乏清晰的轮廓边界。此外，从图 1.1(d) 中可以看出，食品图像中的主要食材往往占据显著区域，这不可避免地削弱了辅助食材的语义信息。这使得在提取食材特征的同时，对辅助食材之间的上下文关系进行建模成为一个关键问题。

为了应对上述挑战并促进对中餐理解的研究，我们开发了一个名为 CMIngre (Cross-Modal Ingredient-level Dataset) 的跨模态食材级数据集。该数据集旨在通过提供对食材及其关系的有价值的见解来增强对中国烹饪的理解。该数据集由来自三个不同来源的 8,001 张图像组成，即菜肴，食谱和用户生成内容（UGC）。该数据集包含 429 种不同的中国食材和 95,290 种食材边界框。

为了对广泛的食材进行全面的语义分类，我们根据中华人民共和国健康行业标准对食品食材数据表达的规定 [23]，将其划分为更高级的层次。这些层次关系也可以作为先验信息，以促进在后续研究中探索不同食材之间的上下文关系。此外，我们评估了传统的基于 CNN 的检测算法和基于 Transformer 的预训练模型在 CMIngre 上食材检测任务的性能。我们还提出了食材检索任务的基线方法，该方法捕获单个食材的语义信息以及各种食材组合之间的关系，并进一步采用 pooling 策略来研究跨模态图像 - 食

材之间的匹配关系。在 CMIngre 数据集上进行的深入实验评估证实了我们提出的方法在提高食材检测和检索性能方面的有效性。

图 1.1 菜品中不同尺寸的食材

本文的贡献可以概括为以下几点：

- 本文提出了一种新的基于“图像 - 文本”对的中餐理解任务，该任务扩展了细粒度对象检测和检索的范围，对中餐烹饪领域的理解提供进一步的帮助。
- 为了支持对中餐理解的研究，我们建立了一个名为 CMIngre 的跨模态食材级别的数据集，该数据集由来自三个不同来源的 8,001 组图像食材组成，涵盖了 429 种不同的中国食材和 95,290 个边界框。
- 我们评估了不同的目标检测算法在 CMIngre 数据集上的性能，并提出了跨模态食材检索任务的基线方法。
- 我们在 CMIngre 上对两个食材级的食品理解任务进行了广泛的实验，以评估我们提出的方法的有效性。

2. 数据集

在本节中，我们将讨论如何构造 CMIngre 数据集。我们将在第一部分中介绍我们如何收集和标注数据。在第二部分中，我们对数据进行了后处理，提升原始数据的质量。在第三部分中进行了 CMIngre 数据集的统计和分析。

2.1 数据收集和标注

数据收集：为了收集全面的食物图像，我们探索了三种类型的图像 - 文本对：

- **菜肴图片**：如图 2.1 第二行所示，这一类别包括与其名称配对的菜肴图像。与其他类型相比，这种类型的文本提供了最简洁的描述。
- **菜谱图片**：如图 2.1 第三行所示，这些数据由菜谱图像和详细的食谱文本组成。这些图像的质量更高，并且比其他两个类别的图像描述的信息更丰富。
- **用户 UGC 图片**：如图 2.1 的最后一行所示，这种类型数据主要包含用户拍摄的图像及其附带的评论。由于用户生成的内容缺乏约束限制，图像和文本描述经常包含与食物无关的元素，例如餐厅氛围或餐具。为了将该数据集细化为专注于食物，我们使用菜肴名称识别算法 [45] 来识别带有菜肴名称的文本。具体来说，我们会选择评论中包含三个以上菜名的照片，减少与食物无关的内容。

这三种类型的数据在线上平台很流行，并且提供了食品相关数据的多样化表示。我们

总共收集了 11,300 个图像 – 文本对用于标注。

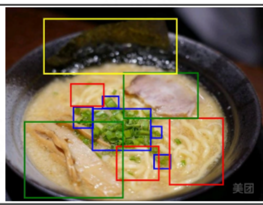
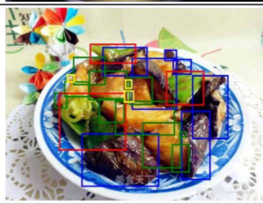
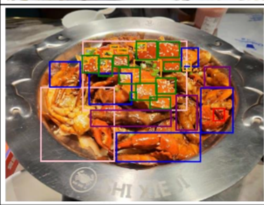
source	text	text annotation	image	image annotation
Dish	Hakata ramen	ramen		green onion ramen pork seaweed
Recipe	1. Prepare eggplant, potatoes and green peppers. 2. Wash the eggplant, cut into hob pieces, and marinate with salt for 5 minutes. 3. Peel the potatoes and cut them into hob pieces the same size as the eggplants, remove the stems of the green peppers and cut them into pieces. Take a bowl and add soy sauce, salt, sugar, vinegar, starch, and a small amount of water to make a sauce.	eggplant potato green pepper soy sauce salt sugar vinegar starch water garlic		eggplant potato green pepper garlic
UGC	"#Set meal: big meat and crab satisfying meal for 3 people""#Crab Casserole" is delicious for the first time\\n[Taste] The taste is good and delicious\\n[Environment] Hygienic, clean and tidy\\n[Service] The service attitude is very good and enthusiastic\\n[Value for money] The price is high and not expensive\\nThere are many shrimps, chicken feet, potatoes, etc. in the meat and crab pot. The location is also very good.	shrimp meat crab chicken feet potato		crab shrimp green onion sesame chicken feet chili

图 2.1　不同数据来源的图像 – 文本对，其中 UGC 表示用户生成的内容

数据标注：这里将详细介绍收集到的“图像 – 文本”对的标注过程。我们首先雇佣了 8 名母语为中文的工作人员，分别对文字描述和图片进行标注。然后，使用另外两名工作人员进行双重检查过程。

1. **文字描述标注**：标注人员的任务是识别文本描述中提到的所有食材。该标注的结果如图 2.1 第三列所示。
2. **图片标注**：如图 2.1 最后一列所示，图像标注遵循两个关键原则：1) 要求标注人员标注文本中提到的和图像中可见的食材。2) 文本中没有提及但在图像中可以识别的食材也需要标注。在这个过程中，标注人员遇到了几个挑战：1) 一个图像包含相同食材的多个实例。在这种情况下，标注人员需要用多个边界框标注所有实例。但是，如果同一食材的多个实例紧密聚集在一起，则可以将它们分组在一个边界框中。2) 多种食材被其他食材覆盖。在这种情况下，

标注人员需要标注出所有可识别的部分。本质上，食材中任何可以被辨别和识别的部分都应该被标注。

经过标注过程后，最终的数据集包含 11,300 个图像 - 文本对，用 4,492 个不同的食材标签和 199,853 个边界框进行了标注。

2.2 标注数据后处理

由于缺乏对标注人员关于每个图像的边界框的大小和数量的限制，最终的标注结果中存在边界框大小的显著变化和相当多的冗余边界框。为了解决这个问题，我们分别对图像和文本进行了进一步的后处理。

- **图像标注清洗**：为了提高数据集中边界框的质量，我们基于两个关键策略实现了清理过程：1) 边界框融合：我们通过将相同标签（重叠，相互包含或临近）合并到单个边界框中来解决冗余边界框的问题。具体来说，融合是基于边界框的面积，计算每个边界框内的像素数。如果融合前后的面积比大于一个特定的阈值，我们将这些边界框整合成一个新的边界框。这个阈值的设置是一个关键问题。我们注意到，过高的阈值将使融合策略无效，而过低的阈值将导致可能包含多种食材的过大的边界框。因此，我们根据经验将其设置为 0.6 作为平衡。2) 较小边界框移除：我们通过两个过程来移除数据集中的小边界框。首先，为了去除只有小框的图像，我们去除所有框的总面积小于整个图像面积 3% 的图像 - 文本对。其次，如果图像中有超过三个相同类别的边界框，我们只保留面积至少为该类别中最大边界框面积 0.8 倍的边界框。在这些清理步骤之后，我们的精细化数据集包含 8,001 个图像 - 文本对，共有 95,290 个边界框。
- **文本标注清洗**：为了改进数据集中的食材标注，我们实现了两个步骤：1) 为了保留足够的数据用于训练和测试，我们删除出现在少于五张图像中的食材。由于原始数据集中存在显著的长尾问题，这一步使得食材标签总数减少到 510。2) 在这 510 种食材中，我们发现了不同名称指代同一种食材的情况，例如“松花蛋 - 皮蛋”。为了解决这个问题，我们利用中华人民共和国健康行业标准

[23] 中的食物成分数据表达规范，对目前 510 种食材进行比较和组合。具体而言，两个标注人员最初将 510 个食材中的每一个分类到分层本体的适当叶节点中。随后，另一个标注人员在同一父节点下审查并合并具有相同语义的食材。合并操作进一步将食材标签减少到 429 个。

综上所述，清理后的数据集包括 8,001 张图像，95,290 个边界框和 429 个食材标签。

2.3 数据统计和分析

在 CMIngre 中，有 1,719 对来自菜肴的图像 – 文本，2,330 对来自食谱，3,952 对来自 UGC。如 2.1 所述，UGC 的图像质量比菜肴和食谱的图像质量差，这给我们在接下来的食物理解任务中处理低质量数据带来了更多的工作量，因为 UGC 覆盖了近一半的数据集。

数据集中每个食材上的图像数量如图 2.2 所示，少量食材在我们的数据集中出现了很多次。例如，“葱 – scallion”在 1,961 张图片中出现次数最多，约占图片总数的 24.51%。此外，有 138 种食材出现在不到 10 张图片中。例如，只有 5 张图片包含“西柚 – grapefruit”，8 张图片包含“桃 – Peach”。图 2.3 显示了我们数据集中每个食材的边界框数量。如图 2.3 所示，每种食材对应的边界框数量分布与图 2.2 中包含该食材的图像数量分布大致相似，均为长尾。为了说明边界框尺寸的差异，图 2.4 给出了不同尺寸边界框的比例。我们观察到小尺寸的边界框（面积比在 0.0025 ~ 0.01 之间）的比例最大。同时，有超过 50% 的边界框的面积比小于 0.01，说明数据集中有很多小物体。

表 2.1 显示了与食品相关数据集的统计比较。我们可以看到，现有的食品相关数据集主要集中在食品识别任务上，其目的是识别图像内的食品类别。很少有数据集为食物边界框提供标注，这是由于它们的目标是定位整个菜肴，而不是各种类型的食材。相比之下，Recipe 1M 为每个食物图像提供食材标注。然而，由于缺乏对这些细粒度食材的位置标注，它们只能隐式地建模整个食物图像与相应食材之间的关联，从而限制了模型的性能。因此，我们引入了 CMIngre，旨在通过食材检测和检索任务增强

对中餐的理解。

Dataset	Task	image number	Annotation Category	The number of annotation category	BBox
ChileanFood64 [42]	Food Recognition	11,504	Food	64	✓
UECFood256 [46]	Food Recognition	29,774	Food	256	✓
UNIMIB2016 [15]	Food Recognition	1027	Food	73	✓
ISIA Food-500	Food Recognition	1027	Food	73	✕
Food2K [12]	Food Recognition	1,036,564	Food	2000	✕
Recipe 1M [32]	Recipe Retrieval	1,029,720	Recipe	1047	✕
CMIngre	Ingredient Detection & Retrieval	8001	Ingredient	429	✓

表 2.1　现有食品相关数据集之间的统计比较

最后，我们将 CMIngre 数据集与广泛使用的目标检测数据集 COCO 进行了比较分析。在图 2.5 中，横轴表示每张图像中标签种类的数量（在 CMIngre 中标签为食材，在 COCO 中标签为物体）纵轴表示每种图像的比例。很明显，CMIngre 图像通常包含更多的对象（在我们的例子中是食材）。具体来说，CMIngre 中包含三个以上标签的图像的占比高于 MS COCO 数据集。这一趋势在边界框的数量上也很明显。如图 2.6 所示，与 MS COCO 相比，我们的数据集中超过 5 个边界框的图像比例更大。综上所述，CMIngre 中的图像比其他现有数据集具有更丰富的语义和更密集的边界框，这对图像理解提出了更艰巨的挑战。

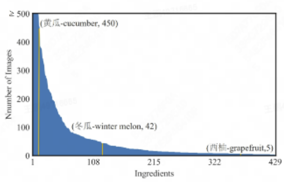


图2.2 包含每个食材的图像数量

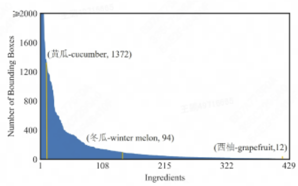


图2.3 每个食材的边框数量

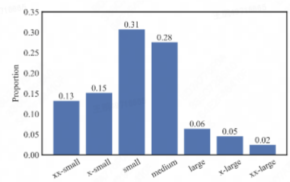


图2.4 不同大小的边界框的分布

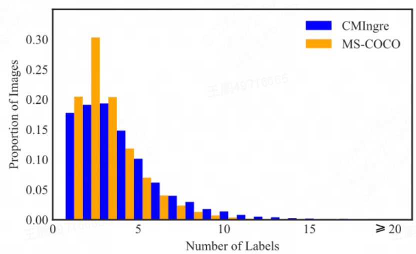


图2.5 具有不同数量标签的图像的比例

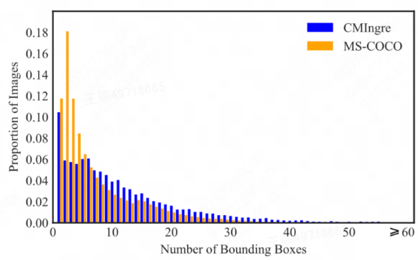


图2.6 具有不同数量边框的图像的比例

3. 方法

在本研究中，我们引入了两项从食材层面理解中国菜食材的任务，即食材检测（任务1）和跨模态食材检索（任务2）。任务1的重点是识别食材并在图像中标注准确的位置信息，任务2旨在研究图像与食材组成之间的复杂关系。对于任务1，我们使用现有目标检测模型在CMIngre数据集上进行微调，构建有关中国菜品理解的新基准；对于任务2，我们在现有跨模态检索方法的基础上，提出了一些创新性的做法，填补了有关中国菜品食材粒度理解的空白。

3.1 食材检测

与传统的目标检测数据集相比，CMIngre数据集具有极其详细的食材分类和密集的边界框注释，因此直接利用现存的目标检测算法进行拟合是一件非常具有挑战的事情。直接对现有的大规模目标检测模型[1]在原始边界框注释上进行微调的效果并不让人满意，因此我们采用融合和过滤策略来缓解边界框密集和尺寸较小带来的问题。

具体而言，我们首先按照融合前后的边界框面积百分比 不低于阈值 τ 的规则，对同一类别的多个边界框进行融合，在实验中这个阈值被设置为0.6。接下来，我们对融合后的边界框进行排序，并将边界框的三个最大区域保留为真值。此外，我们将食材树层级结构的最低级标签都转换为第三级标签，例如“紫菜”和“海带”都融合为“藻类”，“冬笋”和“酸笋”都融合为“笋”，这样可以避免模型无法识别同一分支中高度相似的类别的问题。根据这种转换，类别总数从429减少到67个。在这种设置下，我们使用如下的两种不同的基线方法进行实验。

3.1.1 基于CNN的方法：Faster R-CNN[47]和YOLO v5[48]

Faster R-CNN是一种经典的基于卷积神经网络（CNN）的两阶段目标检测框架。在第一阶段，Faster R-CNN利用CNN提取输入图像的特征映射，然后利用区域提名网络（RPN）生成候选目标区域。在第二阶段，基于候选目标区域，利用图像区域边界框回归以及区域食材识别两个约束进行网络参数的整体更新。相比之下，YOLO（You Only Look Once）是一种单阶段目标检测算法，以其速度和效率而闻名。与

Faster R-CNN 不同，YOLO 在一次评估中处理整个图像，同时预测多个对象的分类概率和边界框。

3.1.2 DINO[1]

DINO（DETR with Improved deNoising anchOr boxes）是一个融合对比降噪训练（contrastive way for denoising training），混合查询选择锚点初始化（mixed query selection method for anchr initialization），前向两次预测（look forward twice scheme for box prediction）的端到端 Transformer 框架。相比于 Faster R-CNN，DINO 是一个参数量更大且更高效的目标检测模型。

评估方案：使用平均精度（AP）来评估基线模型的检测性能。对于 Faster R-CNN，YOLO 和 DINO，分别评估了不同 IoU 阈值（0.5、0.75 和 0.5:0.95）下的标准平均精度结果。

3.2 跨模态食材检索

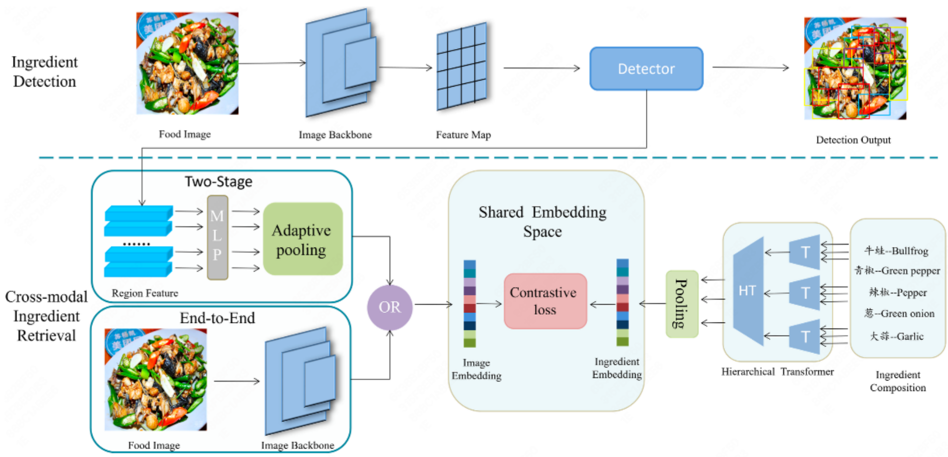


图 3.1 中餐理解框架

跨模态食材检索旨在揭示食品图像与食材之间复杂的对应关系。给定N个训练样本对$I=\{m_i,g_i\}_{i=1}^{N}$，其中m_i表示第i个食物图像，g_i表示含有列表$g_i=[ing_1,ing_2,...,ing_O]$第$i$个食材的组合，其中$O$为食材数量。与使用三级食材标签标注的食材检测不同，这里使用最精细的食材标注（429种）进行跨模态食材检索。

如图所示，使用两个独立的特征提取器提取图像特征和食材特征。然后，应用对比约束以端到端的方式来缩小匹配的图像和食材之间的嵌入距离。考虑到食材检测能够学习不同图像区域中食材的语义嵌入，我们进一步研究了两阶段的检索模型的有效性，该模型首先使用食材检测算法提取区域特征，然后使用区域特征和食材来训练一个联合嵌入模型。

3.2.1 方法 1– 端到端训练

在端到端设置中，我们首先将食品图像和食材组合投影到公共的嵌入空间中，然后使用对比损失来约束跨模态特征对齐。对于图像编码器，受视觉 – 语言 Transformer 在各种下游任务中取得成功的启发，我们采用预训练的 [49]–[51]CLIP ViT B/16 作为图像特征提取器对图像特征进行编码，然后利用线性全连接层将原始图像特征投影到公共的嵌入空间中：

$$f_M = ||Fc(E_M(M))||_2$$

其中，f_M表示图像嵌入，E_M表示图像编码器，$||\cdot||_2$表示l_2归一化。对于食材编码器，与直接编码文本特征的方式不同，我们利用分层Transformer来捕获食材语义和食材间的关系。具体来说，给定一个食材组$g_i = [ing_1, ing_2, \ldots, ing_O]$，首先将每个食材视为一个单独的文本，并将其投影到特征向量$f_{ing_i} \in R^{t\times d}$，其中$t$表示第$i$种食材中的token，$d$表示特征的维度。为了捕获不同食材之间的组合关系，我们首先使用平均池化来获得每种食材的聚合表示，然后将多个食材的特征序列输入到额外的Transformer中，来建模食材之间的关系。此外，我们进一步利用平均池化来获得食材组的原始特征。最后，我们使用一个全连接层，然后进行l_2归一化，从而得到食材组的嵌入：

$$f_G = ||F_c(T_2(T_1(ing_1), \ldots, T_1(ing_O)))||_2$$

其中，T_1和T_2是分层Transformer编码器。根据已有的跨模态图像-文本检索研究[6],[52],[53]，我们利用对比损失来最大化正样本对之间的相似性，最小化负样本之间的相似性。具体来说，给定图像嵌入f_M和食材组合嵌入f_G，利用余弦相似度计算匹配分数：

$$sim = \frac{f_M \cdot f_G}{|f_M||f_G|} \in R^{N\times N}$$

其中，N是样本对的个数。然后，在一个固定的阈值τ下，限制正样本对之间的相似性应该超越所有其他负样本对之间的相似性。

$$L = \sum_{i=1}^{N} \sum_{j=1, i\neq 1}^{N} max\{sim_{i,j} - sim_{i,i} + \tau, 0\}$$

其中，$sim_{i,i}$表示正样本对的相似度，$sim_{i,j}$表示负样本对的相似度。

3.2.2 方法 2- 二阶段训练

与图像编码器直接提取的全局图像特征相比，从食材检测模型中提取的局部特征包含了特定的食材语义信息，为跨模态食材检索提供了更有利的初始化状态。为了利用这一优势，我们首先使用食材检测模型提取 Z 个区域特征。然后，我们提出了一个自适应式池化策略来自动融合多区域特征和多食材特征。

在这种情况下，假设我们可以访问样本对中的Z个区域特征和O个食材。对于食材编码器来说，我们使用与端到端训练阶段相同的分层Transformer编码器来获取食材组成的原始特征。值得注意的是，我们没有使用平均池化来融合食材特征，从而保留了K个食材特征$\{f_{ing_1}, \ldots, f_{ing_O}\} \in R^{O \times d}$。关于区域特征处理，我们使用了一个具有两个隐藏层的多层感知机对区域特征进行编码：

$$f^z_{region} = BN(W_2 * BN(W_1 * f^z_{region}))$$

其中，f^z_{region}是第z个区域的特征，W_1和W_2是可学习的映射矩阵，BN是批归一化。

受Chen等人[53]的启发，我们使用序列模型BiGRU学习池化系数θ，自适应地融合多区域特征和多食材特征。特别地，给定Z个区域的原始特征$\left\{f^z_{region}\right\}^Z_{z=1}$和$f^z_{region} = \{e^1_z, \ldots, e^d_z\}$，我们的目标是通过使用长度$Z$的系数来衡量图像的原始区域特征，从而获得图像$f_M = \{\omega_1, \ldots, \omega_d\} \in R^d$的整体嵌入：

$$\omega_i = \sum_{z=1}^{Z} \theta_z \cdot max_z(\{e^i_z\}^Z_{z=1}), where \sum_{z=1}^{Z} \theta_z = 1$$

为了自适应学习不同长度区域对池化系数，我们将每个区域的位置索引编码为一个向量$p_z \in R^{d_2}$。接下来，我们将它们打包到固定长度的位置嵌入中，并将它们输入序列模型和多层感知机中，从而输出池化系数$\theta = \{\theta_z\}^Z_{z=1}$的序列：

$$\theta = \{\theta_z\}^Z_{z=1} = MLP(BiGRU(\{p_z\}^Z_{z=1}))$$

因此，通过对区域特征进行自适应聚合，得到图像嵌入。采用自适应池化策略融合多个食材特征，得到食材组的嵌入。

评估方案：使用两个评估指标来评估跨模态食材检索的性能：medR和Recall@K。medR表示每个查询检索到的样本的中位数索引，Recall@K表示正确样本的索引位于检索到的前K个样本中的百分比。在实验中，分别将K设置为1、5和10。此外，还记录了R@1、R@5和R@10的综合Rsum来评估模型的整体性能。medR越小，性能越好；R@K和Rsum越高，性能越好。

4. 实验

4.1 算法实现细节

CMIngre 数据集在本次实验中被随机划分为 6,001 个训练样本，1,000 个验证样本和 1,000 个测试样本。所有的实验都使用了 PyTorch 框架，在 2 张 NVIDIA GTX 3090 GPU 上进行实验。

- **食材检测**：对于 Faster R-CNN 框架，与方法 [47],[54] 保持一致，利用 ResNet-101 作为特征提取器，设置 batch size 为 2，学习率为 0.001，并利用 SGD 优化器进行端到端检测优化。对于 YOLO 算法，遵循官方报告 [48] 使用 yolov5x6 进行检测实验。对于 DINO 框架，与官方设置 [1] 保持一致，然后选用 Vision Transformer 作为特征提取器 fine-tune 整个模型。
- **跨模态食材检索**：选用 Adam 优化器训练整个模型并且设置 batch size 为 128，最终映射层维度为 1024。对于双层自注意力编码机制，选用包含有 2 层、4 个头部的 Transformer 作为每层编码器，并且设置隐藏层维度为 512。对于图像食材区域特征预提取，在 Faster R-CNN 框架中提取 36 个维度为 2048 的区域特征，在 DINO 框架中提取 128 个维度为 256 的区域特征。为了增加模型泛化能力，随机消去 20% 的图像区域，并且设置位置编码向量维度 d_2 为 32。

4.2 实验结果

4.2.1 食材检测

为了验证现有的检测框架在 CMIngre 食材数据集上的有效性，我们利用基于 CNN 以及基于 Transformer 的端到端框架。实验结果如表 4.1 所示，可以发现 YOLO v5，Faster R-CNN 和 DINO 在 CMIngre 数据集上性能一般。这一结果表明，目前的目标检测方法为明确的目标边界而设计，很难直接检测到自由形式的食材。这也表明，在食品相关领域开发更多细粒度食材理解算法仍有很大的性能提升空间。与 Faster R-CNN 相比，DINO 在不同的 IoU 阈值下的检测性能更好，这说明大规模预训练模型在食物领域依然存在着较强的理解能力。

此外，为了验证微调目标检测模型实验的有效性，我们找到了 CMIngre 数据集和 MS COCO 数据集中的七个公共类别：蛋糕、西兰花、苹果、胡萝卜、橙子、香蕉、甜甜圈。接下来，我们选取 CMIngre 数据集中包含这七类食材的数据，对预训练模型和使用 CMIngre 中数据微调后的模型进行了对比验证。表 4.2 展示了 Faster

R-CNN 和 DINO 在 CMIngre 数据集中公共 7 类食材上的检测结果。与 Faster R-CNN 相比，预训练的 DINO 和微调后的 DINO 都表现出了更优的性能，突出了大规模预训练模型的泛化能力。此外，在 CMIngre 数据集上对 DINO 进行微调后，模型对常见类别的检测性能有了很大的提高。具体而言，微调后的 DINO 在 7 个公共类别上 AP50:95、AP50 和 AP75 方面分别比预训练的 DINO 提高了 18.3%、25.2% 和 21%，这证明了在 CMIngre 数据集上进行模型调优的有效性。

数据集	方法	AP	AP50	AP75
MS COCO	YOLO v5	50.5	68.1	54.7
	Faster R-CNN	42.0	62.4	42.2
	DINO	58.0	76.8	63.4
CMIngre	YOLO v5	2.9(-47.6)	5.5(-62.6)	2.6(-52.1)
	Faster R-CNN	3.5(-38.5)	7.0(-55.4)	3.11(-39.1)
	DINO	11.1(-46.9)	17.5(-59.3)	11.4(-52.0)

表 4.1 CMIngre 和 MS COCO 的检测结果 (%)，“()”表示检测方法在 MS COCO 和 CMIngre 上的性能差异

方法	设置	AP	AP50	AP75
Faster R-CNN	Pretrain	1.0	3.2	0.7
	Finetune	6.0(+5.0)	10.8(+7.6)	6.3(+5.6)
DINO	Pretrain	13.6	22.9	13.1
	Finetune	16.8(+3.2)	26.9(+4.0)	16.0(+2.9)

表 4.2 Faster R-CNN 和 DINO 在 MS COCO 和 CMIngre 的共有类别上的检测性能

4.2.2 跨模态食材检索

在这一节中，我们重新实现了几个图像 backbone（ResNet-50, ViT B/16 和 CLIP ViT B/16）和食材 backbone（分层 Transformer 和分层 LSTM）进行性能对比。此外，还进行了两阶段实验设置，验证了食材对象和跨模态食材检索相结合的有效性。实验结果如表 4.3 所示，其中 APS 表示自适应池化策略。最后，在表 4.4 中，我们

重新实现了两种最先进的跨模式食谱检索方法（TFood[19] 和 VLPCook[56]），来比较我们提出的 CMIngre 和 Recipe 1M[32]。

方法	设置	图像到食材				食材到图像				Rsum↑
		medR↓	R@1↑	R@5↑	R@10↑	medR↓	R@1↑	R@5↑	R@10↑	
ResNet + H-LSTM	端到端	62.0	3.4	13.3	20.2	66.0	3.6	12.1	19.4	72.0
ResNet + H-Transformer		40.0	5.1	18.0	26.7	42.0	4.3	17.8	26.6	98.5
ViT + H-Transformer		14.0	13.3	32.8	43.8	14.0	15.1	33.6	44.1	182.7
CLIP + H-Transformer		7.0	21.0	45.8	57.5	7.0	21.5	44.9	57.3	248.0
Faster R-CNN + H-Transformer + APS	两阶段	**2.0**	48.4	73.0	79.4	**2.0**	48.3	71.5	79.5	400.1
DINO + H-Transformer + APS		**2.0**	**48.9**	**74.9**	**83.4**	2.0	**50.4**	**75.5**	**84.0**	**417.1**

表 4.3 CMIngre 中跨模态食材检索性能

结果表明，ResNet+H-LSTM 的性能并不令人满意。我们认为这是因为卷积神经网络的接受域有限，ResNet-50 只能捕获整体图像的粗粒度语义，而忽略了细粒度的食材特征。这个结果突出了在跨模态食材检索中对于图像进行细粒度分析的重要性。通过利用 Transformer 中的自注意力机制对不同食材之间的语义关联进行建模，ResNet+H-Transformer 增强了食材组合的表现力，从而提高了检索性能。

具体来说，在图像到食材的设置中，medR 从 62.0 降低到 40.0。当使用视觉 Transformer[58] 作为图像 backbone 时，检索性能显著提升。这证明了视觉 Transformer 通过利用不同图像区域之间的关系来提取细粒度食材表示的能力。受视觉 – 语言基础模型在各种下游任务中获得成功的启发，我们采用 CLIP[49] 作为图像 backbone 进行实验，与其他端到端设置相比，CLIP 具有最佳的检索性能。这些实验结果表明，当采用更深和更先进的 backbone 时，检索性能得到了一致的改善。

除此之外，我们还探索了结合食材检测和跨模态食材检索的两阶段模型的检索性能。首先，我们使用 Faster R-CNN 和 DINO 提取固定长度的区域特征。然后，引入自适应池化策略（APS）来融合多区域特征。如表 4.3 所示，在所有的评估指标中，两阶段的方法明显优于端到端的方法，这表明当前的图像编码器很难直接从图像中提取细粒度食材的判别特征。

在这种情况下，更有效的方法是下训练一个专门针对食材图像的检测模型，然后使用经过训练的检测模型提取的细粒度食材特征进行检索任务。此外，可以观察到，与 Faster R-CNN 相比，使用 DINO 的区域特征可以进一步提高检索性能。这表明食材检索模型的性能提升可以同步体现在跨模态食材检索中。

数据集	方法	设置	图像到食材				食材到图像			
			medR↓	R@1↑	R@5↑	R@10↑	medR↓	R@1↑	R@5↑	R@10↑
Recipe 1M	VLPCook[56]	端到端	1.0	73.6	90.5	93.3	1.0	74.7	90.7	93.2
	TFood[19]		1.0	72.3	90.7	93.4	1.0	72.6	90.6	93.4
CMIngre	VLPCook[56]		6.0	21.6	47.2	60.2	6.0	22.5	48.8	61.5
	TFood[19]		6.0	23.3	49.5	58.8	6.0	22.3	49.6	60.1
	Faster R-CNN + H-Transformer + APS	两阶段	**2.0**	48.4	73.0	79.4	**2.0**	48.3	71.5	79.5
	DINO + H-Transformer + APS		**2.0**	**48.9**	**74.9**	**83.4**	**2.0**	**50.4**	**75.5**	**84.0**

表 4.4　CMIngre 和 Recipe 1M 的跨模态检索性能

为了进一步将所提出数据集与其他跨模态食品检索数据集的复杂性进行对比，我们在 Recipe 1M 中重新实现了两种最先进的方法 [32]，并对比了这些方法在 CMIngre 数据集上的检索性能。根据表 4.4 所示，CMIngre 数据集上的检索效率大约是 Recipe 1M 上的一半，这一显著差异凸显了中国食材面临的更大挑战。具体来说，Recipe 1M 提供了一套全面的食谱细节（包括配料、标题和说明），它丰富了图像和食谱之间的上下文关系，从而促进了跨模态检索。相比之下，CMIngre 数据集仅局限于食材信息，这对有效的跨模态检索提出了更大的挑战。值得注意的是，我们的两阶段方法明显优于这些对比方法，这进一步凸显了两阶段方法的优势，即训练食材检测方法提取细粒度食材特征可以显著增强图像的表示能力。

4.3 可视化

我们从三种类型的数据（菜名，菜谱，用户生成内容）中随机采样一个查询样本，执行跨模态检索任务，并可视化该查询样本的 Top-5 检索结果。如图 4.1 所示，查询图像所对应的正确食材组合成功的以最高相似度出现在第一个检索结果中，验证了我们图像搜索食材的有效性。此外，我们观察到查询样本和 Top-5 检索结果有着一定

程度上的关联，例如在菜谱（recipe）查询图像的检索结果中，Top-5 的食材组合都包含有鸡蛋和蔬菜（油菜、蔬菜、西兰花），并且第一个检索结果和第二个检索结果仅仅是“蔬菜”和“油菜”的细微区别，这说明我们的方法可以有效挖掘到图像和食材间的匹配关系。

如图 4.2 所示，上述相同的现象也出现在三类查询食材的 Top-5 检索结果中。我们也在图 4.3 中可视化了一些最佳匹配失败的案例，发现当图像中所包含的食材不能被清晰认知时，模型会倾向于给出一个相似的具体食材。例如在菜品名称查询图像中，其中的一个绿色食材由于无法被清晰的辨识所以被标注为更高级的“蔬菜”标签。然而当模型执行跨模态检索时，会更倾向于将其认知为更细粒度“芥菜”和“秋葵”而不是“蔬菜”。另外一个观察是相比于最佳匹配案例，错误案例中 Top-5 检索结果的相似度往往倾向于更低且更平均，表示出了模型很难分辨菜品图像中模糊食材的具体分类。

图 4.1 使用图像检索食材组合，三种不同来源查询图的 top-5 检索结果

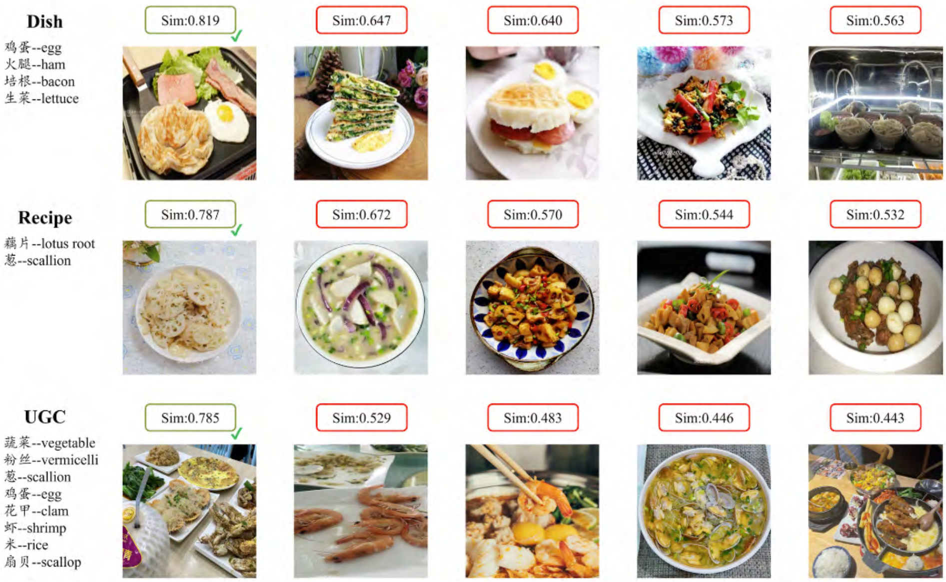


图 4.2　使用食材组合检索图像，三种不同来源查询食材组合的 top-5 检索结果

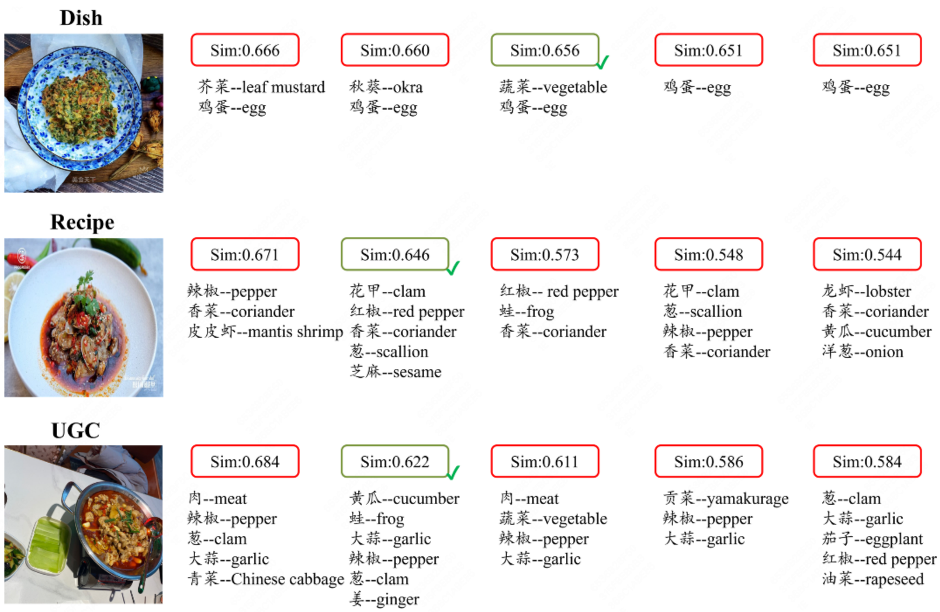


图 4.3　三种不同来源查询图像最佳匹配失败示例

此外，按照 [59] 中描述的方法，我们可视化了单个食材的匹配下降分数（MDS）。具体

来说，我们将单个食材的 MDS 定义为当从食材组合中删除特定食材时，图像与其相应食材组合之间的相似性变化。如图 4.4 所示，具有明显视觉特征的食材往往具有更高的 MDS。例如，在第一张图像中，删除“米”导致了 0.1216 的相似度显著下降，这个下降明显高于土豆、胡萝卜、肉。另一个值得注意的是，具有模糊视觉外观的食材会对跨模态检索产生负面影响。例如，在第三张图中，由于煮熟的青菜缺乏鲜明的视觉特征，导致图像与缺乏青菜的食材组合匹配相似度增加。

图 4.4　单个食材在 CMIngre 上的 MDS。MDS 最高的食材用红色表示，MDS 为负的食材用蓝色表示

5. 业务应用

菜品作为餐饮业务的最基本单元，在供给策略运营、用户需求洞察、业务经营分析等场景都必要依赖。2020 年至 2021 年，到餐研发团队基于业务菜品数据，进行了标准统一和知识融合，整体菜品知识准确率达到 94.51%、覆盖率达到 87.01%。但在局部视角，部分菜品知识属性受限于获取信源单一、挖掘技术难度大等原因导致知识覆盖不足，例如烧烤 / 火锅品类准确率仅 63.6%，食材属性覆盖率 67.5%，口味属性覆盖率 11.9%，影响支持业务精细化、智能化的运营需求。

为了提升菜品知识的覆盖，我们提出一套构建多模态知识图谱的流程，分别从文本和图像两个模态获取菜品知识。

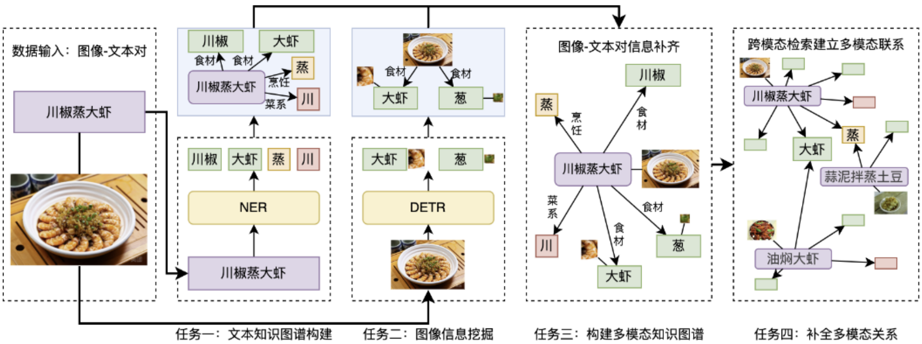


图 5.1 多模态知识图谱构建流程

对于文本模态，使用命名实体识别提取文本中的食材、口味、口感、菜系、烹饪方法；对于图像模态，使用目标检测提取图像中的食材信息和对应区域对文本信息进行补充。在对单个图像 – 文本对构建多模态知识图谱对基础上，通过相同食材、口味等信息对不同的图像 – 文本对进行关联，进而构建完整的菜品多模态知识图谱，从而提升菜品知识覆盖率。

6. 结论

在本研究中，我们将重点放在中餐食材理解上，它扩展了细粒度对象检测和检索的范围，在中餐领域提供了更广泛的应用。为了支持新任务的研究，我们设计了第一个跨模态食材级数据集 CMIngre，该数据集由来自菜肴、食谱和 UGC 三种不同来源的 8,001 对图像食材组成，涵盖了 429 种不同的中国食材和超过 95,290 个边界框。我们在 CMIngre 数据集上评估了不同目标检测算法的有效性，表明开发更高级的细粒度食材检测算法仍然有足够的性能提升空间。此外，在 CMIngre 上进行的广泛的跨模态食材检索实验验证了我们提出的基线的有效性。此外，我们希望这个基准可以激发更多新颖的细粒度食材理解算法的发展，从而促进食品相关领域的进步。

利用以上技术能力，在多模态数据集上建设菜品知识图谱。对比文本单模态（知识准确率 95%、覆盖率达到 80%)，通过在评测数据上进行验证，该项目提升菜品知识图谱的属性知识的质量，知识准确率 96.52%、覆盖率达到 87.01%。将菜品知识图谱

的能力应用于相同商品识别的业务场景，通过提供商品理解的关键信息，识别的错误率从 20.38% 降低至 2.3%，提升美团精细化运营的效率。

7. 致谢

本课题是在到店研发平台和天津大学共同参与下完成。在课题推进过程中，感谢天津大学刘安安教授、王岚君研究员的悉心指导，以及天津大学张晨宇、张国楷、李秋静、杨博、胡明望等同学的积极参与，助力课题的顺利完成，并在美团餐饮美食场景带来实际的业务价值。本课题也获得了 2023 年度美团科研合作「卓越实践奖」。

8. 参考文献

[1] H. Zhang, F. Li, S. Liu, L. Zhang, H. Su, J. Zhu, L. M. Ni, and H.-Y. Shum, "Dino: Detr with improved denoising anchor boxes for end-to- end object detection," arXiv preprint arXiv:2203.03605, 2022, doi:10. 48550/arXiv.2203.03605.

[2] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 10 012 – 10 022, doi:10.1109/ICCV48922. 2021.00986.

[3] X. Dai, Y. Chen, B. Xiao, D. Chen, M. Liu, L. Yuan, and L. Zhang, "Dynamic head: Unifying object detection heads with attentions," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2021, pp. 7373 – 7382, doi:10.1109/CVPR46437. 2021.00729.

[4] A.-A. Liu, H. Tian, N. Xu, W. Nie, Y. Zhang, and M. Kankanhalli, "Toward region-aware attention learning for scene graph generation," IEEE Transactions on Neural Networks and Learning Systems, vol. 33, no. 12, pp. 7655 – 7666, 2021, doi:10.1109/TNNLS.2021.3086066.

[5] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh, "Graph r-cnn for scene graph generation," in Proceedings of the European confer- ence on computer vision (ECCV), 2018, pp. 670 – 685, doi:10.1007/ 978- 3- 030- 01246- 5 41.

[6] C. Liu, Z. Mao, T. Zhang, H. Xie, B. Wang, and Y. Zhang, "Graph structured network for image-text matching," in Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, 2020, pp. 10 921 – 10 930, doi:10.1109/ CVPR42600.2020.01093.

[7] H.Diao,Y.Zhang,L.Ma,andH.Lu, "Similarityreasoningandfiltration for image-text matching," in Proceedings of the AAAI conference on artificial intelligence, vol. 35, no. 2, 2021, pp. 1218 – 1226, doi:10.1609/ aaai.v35i2.16209.

[8] Y. Wang, Y. Su, W. Li, J. Xiao, X. Li, and A.-A. Liu, "Dual-path rare content enhancement network for image and text matching," IEEE Transactions on Circuits and Systems for Video Technology, 2023, doi:10.1109/TCSVT.2023.3254530.

[9] L. Bossard, M. Guillaumin, and L. Van Gool, "Food-101-mining discriminative components with random forests," in Computer Vision- ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part VI 13. Springer, 2014, pp. 446-461, doi:10.1007/978- 3- 319- 10599- 429.

[10] J.ChenandC.-W.Ngo, "Deep-based ingredient recognition for cooking recipe retrieval," in Proceedings of the 24th ACM international confer-ence on Multimedia, 2016, pp. 32-41, doi:10.1145/2964284.2964315.

[11] W. Min, L. Liu, Z. Wang, Z. Luo, X. Wei, X. Wei, and S. Jiang, "Isia food-500: A dataset for large-scale food recognition via stacked global-local attention network," in Proceedings of the 28th ACM International Conference on Multimedia, 2020, pp. 393-401, doi:10.1145/3394171. 3414031.

[12] W. Min, Z. Wang, Y. Liu, M. Luo, L. Kang, X. Wei, X. Wei, and S. Jiang, "Large scale visual food recognition," IEEE Transactions on Pattern Analysis and Machine Intelligence, 2023, doi:10.1109/TPAMI.2023.3237871.

[13] E. Aguilar, B. Remeseiro, M. Bolan-os, and P. Radeva, "Grab, pay, and eat: Semantic food detection for smart restaurants," IEEE Transactions on Multimedia, vol. 20, no. 12, pp. 3266-3275, 2018, doi:10.1109/TMM.2018.2831627.

[14] R. Morales, J. Quispe, and E. Aguilar, "Exploring multi-food detection using deep learning-based algorithms," in 2023 IEEE 13th International Conference on Pattern Recognition Systems (ICPRS), 2023, pp. 1-7, doi:10.1109/ICPRS58416.2023.10179037.

[15] G. Ciocca, P. Napoletano, and R. Schettini, "Food recognition: a new dataset, experiments, and results," IEEE journal of biomedical and health informatics, vol. 21, no. 3, pp. 588-598, 2016, doi:10.1109/JBHI. 2016.2636441.

[16] A. Salvador, N. Hynes, Y. Aytar, J. Marin, F. Ofli, I. Weber, and A. Tor-ralba, "Learning cross-modal embeddings for cooking recipes and food images," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3020-3028, doi:10.1109/CVPR.2017.327.

[17] A. Salvador, E. Gundogdu, L. Bazzani, and M. Donoser, "Revamping cross-modal recipe retrieval with hierarchical transformers and self-supervised learning," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 15 475-15 484, do: 10.1109/CVPR46437.2021.01522.

[18] M. Carvalho, R. Cad è ne, D. Picard, L. Soulier, N. Thome, and M. Cord, "Cross-modal retrieval in the cooking context: Learning semantic text-image embeddings," in The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval, 2018, pp. 35-44, doi: 10.1145/3209978.3210036.

[19] M. Shukor, G. Couairon, A. Grechka, and M. Cord, *Transformer decoders with multimodal regularization for cross-modal food retrieval," in Proceedings of the

IEEE/CV Conference on Computer Vision and Pattern Recognition, 2022, pp. 4567–4578, doi: 10.1109/CVPRW56347.2022.00503.

[20] H. Wang, D. Sahoo, C. Liu, K. Shu, P. Achananuparp, E.-p. Lim, and S. C. Hoi, "Cross-modal food retrieval: learning a joint embedding of food images and recipes with semantic consistency and attention mechanism," IEEE Transactions on Multimedia, vol. 24, pp. 2515–2525, 2021, doi: 10.1 109/TMM.2021.3083109.

[21] M. Li, P.-Y. Huang, X. Chang, J. Hu, Y. Yang, and A. Hauptmann, "Video pivoting unsupervised multi-modal machine translation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 45, no. 3, pp. 3918–3932, 2023, doi: 10.1109/TPAMI.2022.3181116.

[22] Chinese cuisine culture, Last accessed on June 23, 2023.

[23] "Regulation of food composition data expression," https://www.chinanutri.cn/fgbz/fgbzhybz/201707/P020170721479798369359.pdf,Last accessed on June 23, 2023.

[24] T. Joutou and K. Yanai, *A food image recognition system with multiple kernel learning," in 2009 16th IEEE International Conference on Image Processing (ICIP). IEEE, 2009, pp. 285–288, doi: 10.1109/ICIP.2009.5413400.

[25] Y. Kawano and K. Yanai, "Food image recognition with deep con-volutional features," in Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, 2014, pp. 589–593, doi: 10.1145/2638728.2641339.

[26] K. Yanai and Y. Kawano, "Food image recognition using deep con-volutional network with pre-training and fine-tuning," in 2015 IEEE International Conference on Multimedia & Expo Workshops (ICMEW). IEBE, 2015, p. 1–6, doi: 10.1109/ICMEW.2015.7169816.

[27] M. T. Turan and E. Erzin, "Domain adaptation for food intake classification with teacher/student learning," IEEE Transactions on Multimedia, vol. 23, pp. 4220 4231, 2020, doi: 10.1109/TMM.2020.3038315.

[28] H. Liang, G. Wen, Y. Hu, M. Luo, P. Yang, and Y. Xu, "Mvanet: Multitask guided multi-view attention network for chinese food recognition," IEEE Transactions on Multimedia, vol. 23, pp. 3551–3561, 2020, doi: 10.1109/TMM.2020.3028478.

[29] J. He, L. Lin, H. A. Eicher-Miller, and F. Zhu, "Long-tailed food clas-sification," Nutrients, vol. 15, no. 12, 2023, doi: 10.3390/nu15122751.

[30] K. Aizawa, Y. Maruyama, H. Li, and C. Morikawa, "Food balance estimation by using personal dietary tendencies in a multimedia food log," IEEE Transactions on multimedia, vol. 15, no. 8, pp. 2176–2185, 2013, doi: 10.1109/TMM.2013.2271474.

[31] J.-J. Chen, C.-W. Ngo, F.-L. Feng, and T.-S. Chua, "Deep understanding of cooking procedure for cross-modal recipe retrieval," in Proceedings of the 26th ACM international conference on Multimedia, 2018, pp.1020–1028, do: 10.1145/3240508.3240627.

[32] Y.-C. Lien, H. Zamani, and W. B. Croft, "Recipe retrieval with visual query of

ingredients," in Proceedings of the 43rd International ACM SI-GIR Conference on Research and Development in Information Retrieval, 2020, pp. 1565-1568, do: 10.1145/3397271.3401244.

[33] W. Min, B.-K. Bao, S. Mei, Y. Zhu, Y. Rui, and S. Jiang. "You are what you eat: Exploring rich recipe information for cross-region food analysis," IEEE Transactions on Multimedia, vol. 20, no. 4, pp. 950-964, 2017, doi: 10.1109/TMM.2017.2759499.

[34] G. Ciocca, P. Napoletano, and R. Schettini, *Learning cnn-based features for retrieval of food images," in New Trends in Image Analysis and Processing-ICIAP 2017: ICIAP International Workshops, WBICV, SSPandBE, 3AS, RGBD, NIVAR, IWBAAS, and MADiMa 2017, Catania, Italy, September 11-15, 2017, Revised Selected Papers 19. Springer, 2017, pp. 426 434, doi: 10.1007/978-3-319-70742-6_41.

[35] X. Chen, Y. Zhu, H. Zhou, L. Diao, and D. Wang, "Chinesefoodnet: A large-scale image dataset for chinese food recognition," arXiv preprint arXiv: 1705.02743, 2017, doi: 10.48550/arXiv. 1705.02743.

[36] S. Hou, Y. Feng, and Z. Wang, "Vegfru: A domain-specific dataset for fine-grained visual categorization," in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 541-549, doi:10.1109/ICCV.2017.66.

[37] J. Qiu, F. P.-W. Lo, Y. Sun, S. Wang, and B. Lo, "Mining discriminative food regions for accurate food recognition," arXiv preprint arXiv:2207.03692, 2022, doi: 10.48550/arXiv.2207.03692.

[38] J. Wang, X. Ding, and B. Guo, "High precision food detection method based on deep object detection network," in 2021 IEEE Sth Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), vol. 5. IEEE, 2021, pp. 646-650, doi: 10.1109/ITNEC52019. 2021.9587189.

[39] $. Akti, M. Qarage, and H. K. Ekenel, "A mobile food recognition system for dietary assessment," in International Conference on Image Analysis and Processing. Springer, 2022, pp. 71-81, doi: 10.1007/978-3-031-13321-3_7.

[40] Y. Matsuda, H. Hoashi, and K. Yanai, "Recognition of multiple-food images by detecting candidate regions," in 2012 IEEE International Conference on Multimedia and Expo.IEEE, 2012, pp. 25-30, doi: 10.1109/ICME.2012.157.

[41] Y. Kawano and K. Yanai, "Foodcam-256: a large-scale real-time mobile food recognifionsystem employing high-dimensional features and compression of classifier weights," in Proceedings of the 22nd ACM international conference on Multimedia, 2014, pp. 761-762, doi:10.1145/2647868.2654869.

[42] B. Muñoz, I. Chirino, and E. Aguilar, "Can deep learning models recognize chilean diet," IEEE Latin America Transactions, vol. 20, no. 9, pp. 2131-2138, 2022, doi:10.1109 TLA.2022.9878168.

[43] Y. Kawano and K. Yanai, "Automatic expansion of a food image dataset leveraging existing categories with domain adaptation," in Computer Vision - ECCV 2014

Workshops, 2015, pp. 3–17, doi:10.1007/ 978–3–319–16199–0_1.
[44] J. Chen, L. Pang, and C.–W. Ngo, “Cross–modal recipe retrieval: How to cook this dish?” in MultiMedia Modeling: 23rd International Conference, MMM 2017, Reykiavil, Iceland, January 4–6, 2017, Pro–ceedings, Part I 23.978–3–319–51811–4_48. Springer, 2017, pp. 588–600, doi: 10.1007/
[45] X. Li, J. Feng, Y. Meng, Q. Han, F. Wu, and J. Li, “A unified MRC framework for named entity recognition,” in Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5–10, 2020, 2020, pp. 5849–5859, doi: 10.18653/V1/ 2020.ACL–MAIN.519.
[46] Y. Kawano and K. Yanai, “Automatic expansion of a food image dataset leveraging existing categories with domain adaptation,” in Computer Vision–ECCV 2014 Workshops: Zurich, Switzerland, September 6–7 and 12, 2014, Proceedings, Part III 13. Springer, 2015, pp. 3–17, doi: 10.1007/978–3–319–16199–0_1.
[47] 1 S. Ren, K. He, R. Girshick, and J. Sun, “Faster I–cnn: Towards real–time object detection with region proposal networks, “Advances in neural information processing systems, vol. 28, 2015, do: 10.1109/TPAMI. 2016.2577031.
[48] G. Jocher, “Yolov5 by ultralytics,* 2020, doi: 10.5281/zenodo.3908559.[Online]. Available: https://github.com/ultralytics/yolov5
[49] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark et al., *Learning transferable visual models from natural language supervision,” in International conference on machine learning. PMLR, 2021, pp. 8748–8763, doi: 10.48550/arXiv.2103.00020.
[50] L. H. Li, P. Zhang, H. Zhang, J. Yang, C. Li, Y. Zhong, L. Wang,L. Yuan, L. Zhang, J.–N. Hwang et al., “Grounded language–image pre–training,” in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 10 965–10 975, doi: 10.48550/ arXiv.2112.03857.
[51] H. Zhang, P. Zhang, x. Hu, Y–C. Chen, L. Li, x. Dai, L. Wang, L. Yuan, J.–N. Hwang, and J. Gao, “Glipv2: Unifying localization and vision–language understanding,” Advances in Neural Information Processing Systems, vol. 35, pp. 36 067–36080, 2022, doi: 10.48550/arXiv.2206.05836.
[52] K.–H. Lee, X. Chen, G. Hua, H. Hu, and X. He, “Stacked cross attention for image–text matching,” in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 201–216, doi: 10.1007/ 978–3–030–01225–0_13.
[53] J. Chen, H. Hu, H. Wu, Y. Jiang, and C. Wang, “Learning the best pooling strategy for visual semantic embedding,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2021, doi: 10.1109/ CVPR46437.2021.01553.
[54] J. Yang, J. Lu, D. Batra, and D. Parikh, “A faster pytorch implementation of faster r–cnn. “https://github.com/jwyang/faster–renn.pytorch, 2017.
[55] P. Anderson, X. He, C. Buehler, D. Teney, M. Johnson, S. Gould, and L. Zhang,*Bottom–up and top–down attention for image captioning and visual question answering,” in CVPR, 2018, doi: 10.1109/CVPR.2018.00636.

[56] M. Shukor, N. Thome, and M. Cord, “Vision and structured-language pretraining for cross-modal food retrieval,” Available at SSRN 4511116, 2023, doi: 10.48550/arXiv.2212.04267

[57] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan et al., “Microsoft coco: Common objects in context,” in European conference on computer vision. Springer, 2014, pp.1740-755, doi: 10.1007/978-3-319-10602-1_48.

[58] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly et al., “An image is worth 16x16 words: Transformers for image recognition at scale,” arXiv preprint arXiv:2010.11929, 2020, doi:10.48550/arXiv.2010.11929.

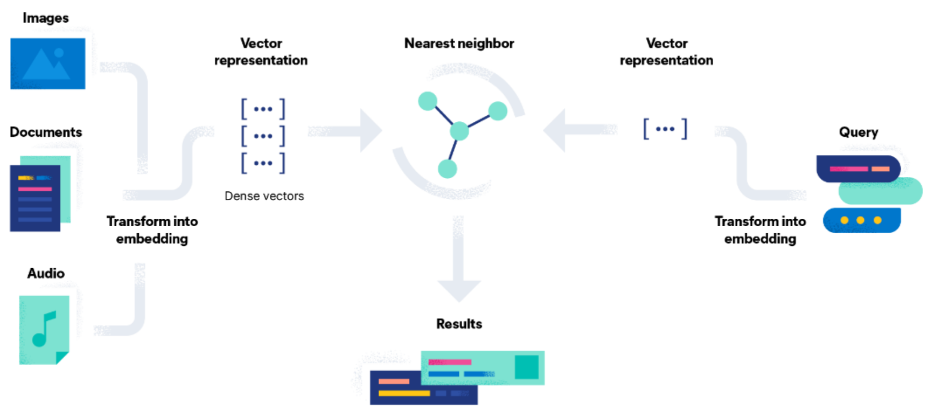
[59] Z. Zheng, L. Zheng, M. Garrett, Y. Yang, M. Xu, and Y.-D. Shen, “Dual-path convolutional image-text embeddings with instance loss,” ACM Trans. Multimedia Comput. Commun. Appl., vol. 16, no. 2, 2020, doi: 10.1145/3383184.

美团外卖基于 GPU 的向量检索系统实践

1 背景

随着大数据和人工智能时代的到来，向量检索的应用场景越来越广泛。在信息检索领域，向量检索可以用于检索系统、推荐系统、问答系统等，通过计算文档和查询向量之间的相似度，快速地找到与用户需求相关的信息。此外，在大语言模型和生成式 AI 场景，向量索引做为向量数据的底层存储，也得到了广泛的应用。

如下图所示，向量检索主要分为三个步骤：（1）将文本、图像、语音等原始数据经过特征抽取，模型预估，最终表征为向量集合；（2）对输入 Query 采用类似的方式表征为向量；（3）在向量索引中找到与查询向量最相似的 K 个结果。一种简单直接的检索方式是与向量集合进行逐一比较，找到与查询向量最相似的向量。这种方法也被称为暴力检索。在大数据量或者高维度场景中，暴力检索的耗时和计算资源消耗巨大，无法在现实场景中直接使用。



为了解决上述问题，业界提出 ANN（Approximate Nearest Neighbor）近邻检索方案：通过构建有效索引，减少向量计算量，牺牲一定的召回精度以换取更高的检索速

率。另一方面，研究如何通过 GPU 的并行计算能力，加速向量相似计算，也是一个比较热门的发展方向之一。Facebook 开源的向量检索库 Faiss 在 GPU 上实现了多种索引方式，与 CPU 版性能相比，检索速率提升 5 到 10 倍。开源的向量检索引擎 Milvus 基于 GPU 加速的方案使得检索提高 10+ 倍。

目前，向量检索已经广泛应用在美团外卖搜推业务各场景中。相较于其他业务场景，美团外卖业务特点具有较强的 Location Based Service（LBS）依赖，即商家的配送范围，决定了用户所能点餐的商家列表。以商品向量检索场景为例：向量检索结果集需要经过“可配送商家列表”过滤。

此外，在不同的业务场景使用过程中，还需要根据商家商品的品类、标签等标量属性进行过滤。当前，美团外卖向量检索基于 Elasticsearch+FAISS 进行搭建，实现了 10 亿级别 + 高维向量集的标量 + 向量混合检索的能力。为了在保证业务高召回率的同时进一步减少检索时间，我们探索基于 GPU 的向量检索，并实现了一套通用的检索系统。

2 美团外卖向量索引的发展历程

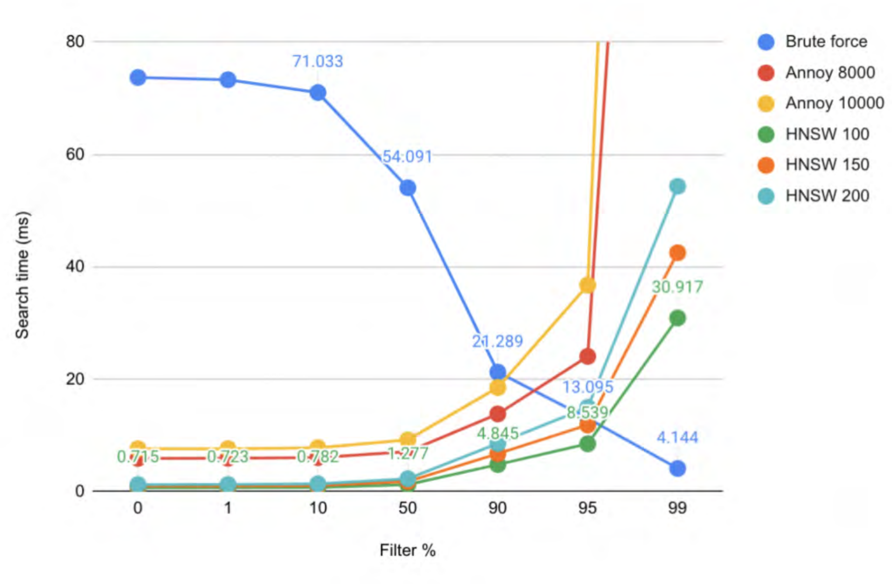
在美团外卖向量检索系统的建设过程中，我们相继使用了 HNSW（Hierarchical Navigable Small World），IVF（Inverted File），IVF-PQ（Inverted File with Product Quantization）以及 IVF-PQ+Refine 等算法，基于 CPU 实现了向量检索能力。在过去的几年间，我们对 Elasticsearch 进行定制，实现了相关的向量检索算法，在复用 Elasticsearch 检索能力的情况下支持了标量 - 向量混合检索。下面是这四种技术的简介及演进历程。

2.1 HNSW（Hierarchical Navigable Small World）

HNSW 是一种用于大规模高维数据近似最近邻搜索的算法，它的基本思想是使用一种层次化的图结构，每一层都是一个导航小世界图，从而实现了在高维空间中的高效搜索。导航小世界图是一种有着特殊拓扑结构的图，它的特点是任意两点之间的路径

长度都很短，而且可以快速找到。

在 HNSW 算法中，这种导航小世界图的层次结构使得搜索过程可以从图的高层开始，快速定位到目标点的大致位置，然后逐层向下精细化搜索，最终在底层找到最近邻，在通用检索场景上有显著的优势。然而该算法在高过滤比下性能会有折损，从而导致在到家搜推这种强 LBS 过滤场景下会暴露其性能的劣势。业界有较多相关的 benchmark 可以参考，以 Yahoo 的向量检索系统 Vespa 相关博客为例，性能与召回率的趋势如下：



2.2 IVF（Inverted File）

IVF 是一种基于倒排索引的方法，它将高维向量空间分为多个簇（Cluster），每个簇对应一个倒排列表，存储了属于该簇的向量索引。这种方法大大减少了搜索时需要比较的向量数量，从而提高了检索速度。它的缺点是需要存储原始的向量数据，同时为了保证检索性能需要将其全量加载到内存中，从而占用了大量的内存空间，容易造成内存资源瓶颈。

2.3 IVF-PQ（Inverted File with Product Quantization）

在候选集数量巨大的场景下，比如商品向量检索场景下，IVF 带来的内存空间大的问题很快就显现出来，为了解决内存空间的问题，开始尝试使用了 IVF-PQ 方法。该方法在 IVF 的基础上，使用了乘积量化（Product Quantization，PQ）的方法来压缩向量数据。PQ 将高维向量分为多个子向量，然后对每个子向量进行量化，从而大大减少了对内存空间的需求。

然而，由于量化过程会引入误差，因此 IVF-PQ 的检索精度会低于 IVF，从而导致召回率无法满足线上要求，对召回率要求相对较低的场景可以使用 IVF-PQ，对召回率有一定要求的场景需要其他解决方案。

2.4 IVF-PQ+Refine

为了提高 IVF-PQ 的检索精度，进一步采用了 IVF-PQ+Refine 的方案，在 IVF-PQ 的基础上，在 SSD 磁盘上保存了未经压缩的原始向量数据。检索时，通过 IVF-PQ 召回数量更大的候选向量集合，然后获取对应的原始向量数据进行精确计算，从而提高检索精度。这种方法既保留了 IVF-PQ 的存储优势，解决了内存资源瓶颈，又保证了召回率，因此在实际应用中得到了广泛的使用。

2.5 基于地理位置的向量检索

美团外卖业务有一个区别于普通电商的明显特征——LBS 特征，用户和商家的距离在很大程度上影响着用户的最终选择。因此可以考虑在向量检索过程中增加地理位置因素，使距离用户更近的商品可以优先被检索到。通过将经纬度编码为向量，优化具体做法是将用户或商家的经纬度以加权的方式加入查询 Query 和候选向量中，在计算 Query 和候选向量的相似度时，距离因素就可以在不同程度上影响最终的检索结果，从而达到让向量索引具备 LBS 属性的目标。在加入地理位置信息后，向量检索的召回率有较大提升。

除了以上几种检索方式，常见的向量检索方式还有 Flat（即暴力计算），可以实现 100% 的召回率，但是由于计算量大，其性能较差，一般仅用于小规模的数据场景。

3 目标与挑战

3.1 目标

在以上几个方案落地后，向量 + 标量混合检索、前置过滤、支持海量数据检索几个挑战都得到了解决，但是检索性能及召回率与理想目标仍有一定差距，需要探索其他可能的解决方案。考虑到美团外卖的业务场景，目标方案应该满足以下要求：

- **支持向量 + 标量混合检索**：在向量检索的基础上，支持复杂的标量过滤条件。
- **高过滤比**：标量作为过滤条件，有较高的过滤比（大于 99%），过滤后候选集大（以外卖商品为例，符合 LBS 过滤的商品向量候选集仍然超过百万）。
- **高召回率**：召回率需要在 95%+ 水平。
- **高性能**：在满足高召回率的前提下，检索耗时 Tp99 控制在 20ms 以内。
- **数据量**：需要支持上亿级别的候选集规模。

在调研业界向量检索方案后，我们考虑利用 GPU 的强大算力来实现高性能检索的目标。当前业界大部分基于 GPU 的向量检索方案的目标都是为了追求极致的性能，使用 GPU 来加速向量检索，如 Faiss、Raft、Milvus 等，然而它们都是面向全库检索，不直接提供向量 + 标量混合检索的能力，需要在已有方案的基础上进行改造。

3.2 解决方案探索

实现向量 + 标量混合检索，一般有两种方式：前置过滤（pre-filter）和后置过滤（post-filter）。前置过滤指先对全体数据进行标量过滤，得到候选结果集，然后在候选结果集中进行向量检索，得到 TopK 结果。后置过滤指先进行向量检索，得到 TopK*N 个检索结果，再对这些结果进行标量过滤，得到最终的 TopK 结果。其中 N 为扩召回倍数，主要是为了缓解向量检索结果被标量检索条件过滤，导致最终结果数不足 K 个的问题。

业界已有较多的成熟的全库检索的方案，后置过滤方案可以尽量复用现有框架，开发量小、风险低，因此我们优先考虑后置过滤方案。我们基于 GPU 的后置过滤方案快速实现了一版向量检索引擎，并验证其召回率与检索性能。GPU 中成熟的检索算法

有 Flat、IVFFlat 和 IVFPQ 等，在不做扩召回的情况下，召回率偏低，因此我们在 benchmark 上选择了较大的扩召回倍数以提高召回率。

测试数据集选取了线上真实的商品数据，据统计，符合标量过滤条件的候选向量数量平均为 250 万，在单 GPU 上验证后置过滤检索性能与召回率如下：

检索类型	TopK	扩召回倍数	时延（ms）			召回率(%)
			Avg	TP99	TP999	
Flat	100	480	58.8	80.5	91.0	92.7
Flat	100	720	64.1	94.5	105.2	94.7
IVF	100	480	11.9	21.2	23.5	82.4
IVF	100	720	14.9	22.1	40.2	84.4
IVFPQ	100	480	7.9	16.8	22.1	73.7
IVFPQ	100	720	10.0	23.6	34.3	75.3

测试结果表面，以上三种算法均无法同时满足我们对检索性能和召回率的需求。其中 IVF 与 IVFPQ 召回率较低，Flat 算法虽然召回率较高，但是与全体候选集计算向量相似度导致其性能较差。

举个例子，候选向量数据规模为 1000 万，向量维度为 D。

（1）Flat 是纯暴力计算的算法，精度最高，但需要在全体候选集上计算相似度，单条查询向量的计算量为 1000 万 *D 次浮点运算。

（2）IVF 在 Flat 的基础上通过 IVF 倒排索引，将候选集划分成多个簇（Cluster），然后选取部分离查询向量较近的簇计算相似度，这样可以按比例降低计算量，如果将候选集分成 n_list=1024 个簇，每次查询只选取 n_probe=64 个簇，则单条向量的计算量为 Flat 的 1/16，即 62.5 万 *D 次浮点运算。

（3）IVFPQ 对比 IVF 算法，使用了乘积量化，将 D 维向量切分成 M 组子向量，每组子向量训练出 K 个聚类中心，如果 M=8，K=256，则单条查询的计算量为 8*256*D 次浮点计算 +1000 万 *8 次查表 +1000 万 *8 次加法运算。

在 Flat 算法的基础上，我们考虑通过向量子空间划分的方式，将全量候选集划分为多个向量子空间，每次检索时选取其中的一部分向量子空间，从而减少不必要的计算量，提高检索性能。

考虑到外卖搜索的强 LBS 属性，可以基于 GeoHash 来进行向量子空间划分。构建索引时，根据商家的地理位置（经纬度）计算 GeoHash 值，将全量商品数据划分为多个向量子空间。检索时，根据用户的地理位置信息计算其 GeoHash 值，并扩展至附近 9 个或 25 个 GeoHash 块，在这些 GeoHash 块内采用 Flat 算法进行向量检索，可以有效减少计算量。这种向量子空间划分方式有效地提高了检索性能，但是存在某些距离稍远的商家无法被召回的情况，最终测得的召回率只有 80% 左右，无法满足要求。

综上，后置过滤方案无法同时满足检索性能和召回率的需求，而 GPU 版本的 Faiss 无法实现前置过滤功能，考虑到美团外卖的业务场景，向量 + 标量混合检索能力是最基本的要求，因此我们决定自研 GPU 向量检索引擎。

4 GPU 向量检索系统

4.1 前置过滤实现方案选择

基于 GPU 的向量检索，要想实现前置过滤，一般有三种实现方案：

1. 所有原始数据都保存在 GPU 显存中，由 GPU 完成前置过滤，再进行向量计算。
2. 所有原始数据都保存在 CPU 内存中，在 CPU 内存中完成前置过滤，将过滤后的原始向量数据传给 GPU 进行向量计算。
3. 原始向量数据保存在 GPU 显存中，其他标量数据保存在 CPU 内存中，在

CPU 内存完成标量过滤后，将过滤结果的下标传给 GPU，GPU 根据下标从显存中获取向量数据进行计算。

由于 GPU 与 CPU 结构与功能上的差异性，使用 GPU 完成前置过滤，显存资源占用量更大，过滤性能较差，且无法充分利用过滤比大的业务特点，因此不考虑方案 1。

方案 2 与方案 3 性能对比与各自的优点如下所示：

方案	检索类型	TopK	时延（ms）			召回率(%)	优点
			Avg	TP99	TP999		
方案2	Flat	100	31.5	59.8	62.9	100	• 数据在显存中是连续存储的，访存友好 • 内存一般比显存大，可以支持更大规模的数据集
方案3	Flat	100	12.9	25.4	26.2	100	• 只需要在CPU和GPU之间传输下标，相比传输原始向量数据，更节省带宽

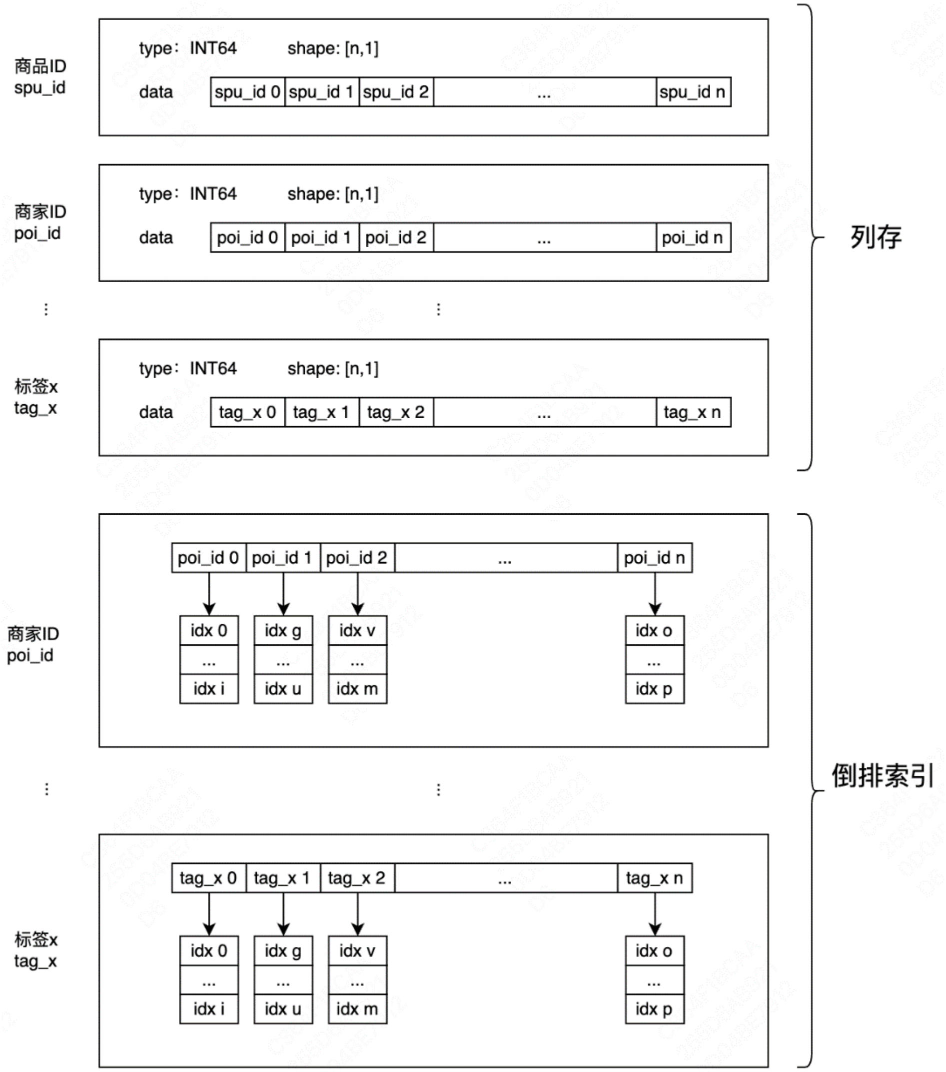
实验结果表明，方案 2 在数据拷贝阶段耗时严重，时延无法达到要求。因为在美团外卖的场景下，过滤后的数据集仍然很大，这对 CPU 到 GPU 之间的数据传输带宽（A30 显卡带宽数据如下 CPU-GPU：PCIe Gen4: 64GB/s；GPU-GPU：933GB/s）提出了很高的要求，因此我们最终选择了方案 3。

4.2 GPU 向量检索引擎

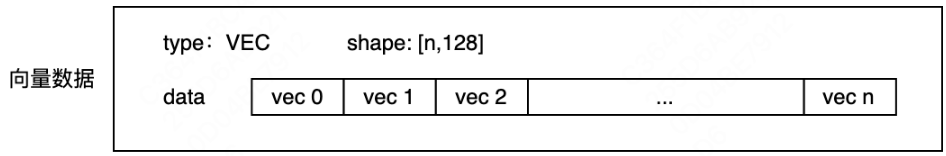
4.2.1 数据结构

考虑到显存的价格远高于内存，因此我们在设计方案的过程中，尽可能将数据存储在内存当中，仅将需要 GPU 计算的数据存储在显存当中。

内存中保存了所有的标量数据，数据按列存储，通过位置索引可以快速找到某条数据的所有字段信息，数据按列存储具备较高的灵活性和可扩展性，同时也更容易进行数据压缩和计算加速。针对需要用于过滤的标量字段，在内存中构造了倒排索引，倒排链中保存了对应的原始数据位置索引信息，内存数据结构如下图所示：



显存中保存了所有的向量数据，数据位置索引与内存中的数据一一对应，可以通过位置索引快速获取某条数据的向量信息，如下图所示：



4.2.2 检索流程

Flat 暴力检索

初始化阶段，在内存中构建用于标量过滤的倒排索引，同时，将向量数据从 CPU 内存拷贝到 GPU 显存，通过位置索引进行关联。

1. 标量过滤

标量过滤过程在 CPU 内存中进行，通过内存中的倒排索引，可以快速得到符合某个标量过滤条件的原始数据位置索引列表，通过倒排索引的求交、求并等逻辑，可以支持多个标量过滤条件的与、或关系组合，最终，得到所有符合条件的位置索引列表。

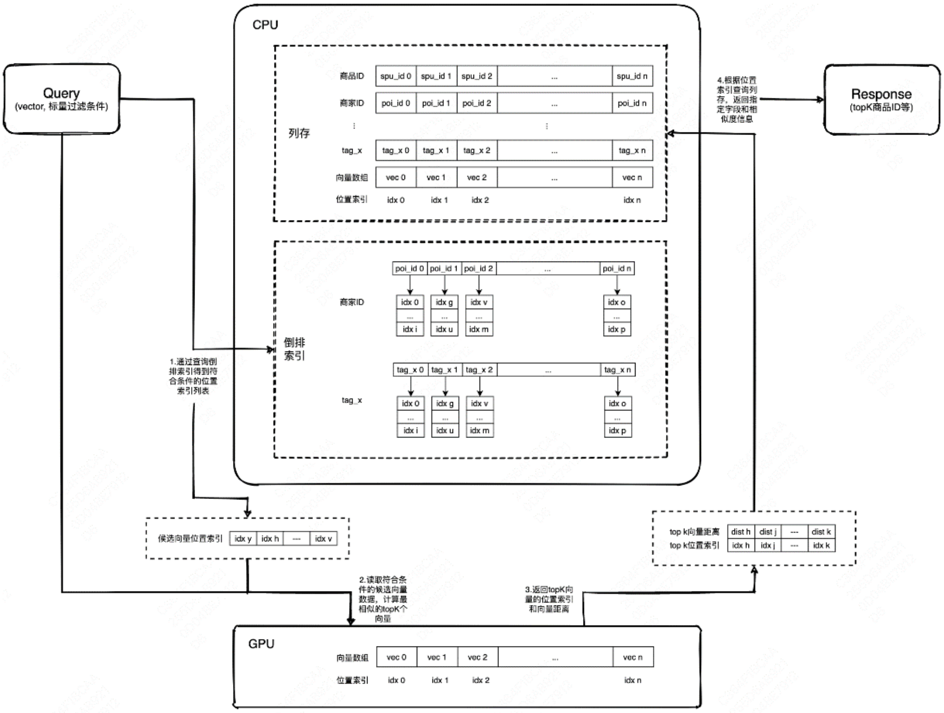
2. 相似度计算

相似度计算在 GPU 中进行，通过上一步标量过滤得到的位置索引列表，从 GPU 显存中读取符合条件的候选向量数据，然后使用常见的向量距离算法计算最相似的 TopK 个向量，将检索结果下表列表回传给 CPU。

3. 检索结果生成

通过上一步的检索结果下表列表，在 CPU 内存中获取对应 record 记录并返回。

整体检索流程如下：



IVF 近似检索

在某些场景下，我们对检索性能有更高的要求，同时对召回率的要求可以适当放宽，因此我们在 GPU 向量检索引擎上支持了 IVF 近似检索。

在初始化阶段，使用向量数据训练出 P 个聚类中心，并针对每个聚类中心构建局部的倒排索引，倒排索引结构与 Flat 方案类似，区别在于位置索引信息只保存在最近的聚类中心下。

1. 标量过滤

标量过滤过程在 CPU 内存中进行，先找到与 query 向量最近的 N 个聚类中心点，在这些聚类中心点下进行标量过滤，得到 N 个候选位置索引列表，再 merge 成最终的候选位置索引列表。与 Flat 方案相比，IVF 近似检索减少了计算量，因此能获得更好的检索性能。

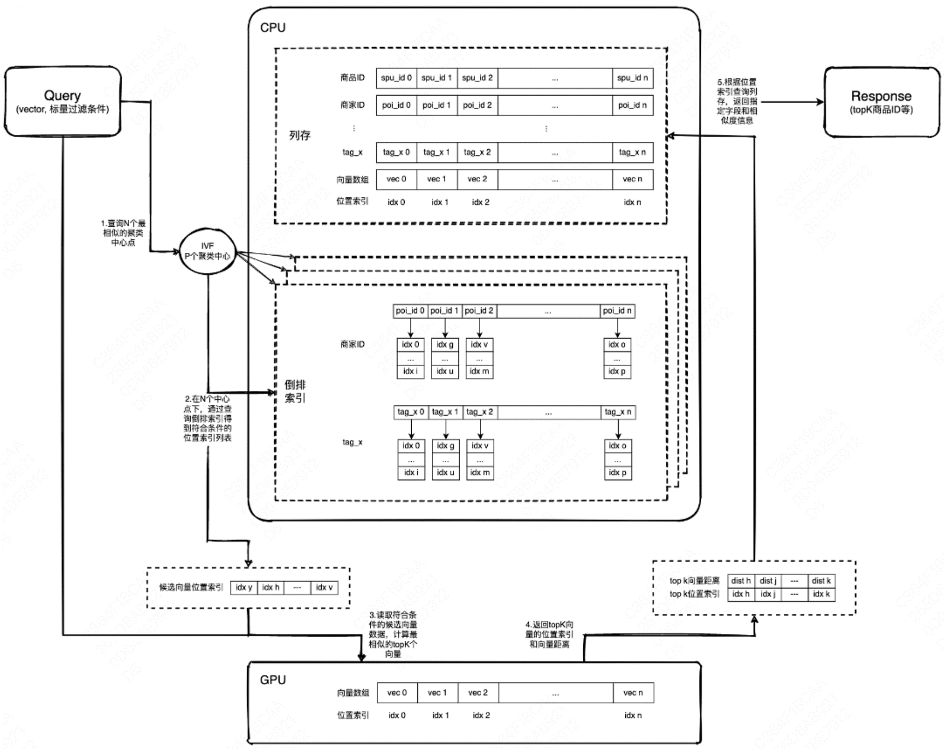
2. 相似度计算

相似度计算阶段与 Flat 方案相同。

3. 检索结果生成

检索结果生成阶段也与 Flat 方案相同。

整体检索流程如下：



在单 GPU 上验证检索性能与召回率如下（测试数据集同后置过滤）：

检索类型	TopK	选取的中心点个数	时延（ms）			召回率(%)
			Avg	TP99	TP999	
Flat	100	-	31.5	59.8	62.9	100
IVF	100	64	3.6	9.9	12.2	87
	100	128	6.0	15.2	18.8	93
	100	256	10.8	25.0	30.5	97

可见，无论是 Flat 还是 IVF，在相同的召回率下，使用前置过滤的性能都要明显好于后置过滤。

4.2.3 性能优化

完成前置过滤的向量检索功能之后，我们对向量检索引擎做了一系列优化。

1. 单 GPU 性能优化

- 高并发支持，通过 Cuda Stream，GPU 可以并行处理多个查询请求，高并发压测下，GPU 利用率可以达到 100%。
- 通过 GPU 实现部分标量过滤功能，支持在 GPU 上实现部分标量过滤功能，向量计算与标量过滤同处一个 Kernel，充分利用 GPU 并行计算能力（标量过滤本身是一个无状态操作，天然支持并行处理，CPU 并发能力受限于 CPU 核数，但 GPU 可以支持上千个线程的并发，所以在性能上体现出明显优势）。
- 资源管理优化，支持句柄机制，资源预先分配，重复利用。每个句柄持有一部分私有资源，包含保存向量检索中间计算结果的可读写内存、显存，以及单独的 Cuda Stream 执行流；共享一份全局只读公有资源。在初始化阶段，创建句柄对象池，可以通过控制句柄数量，来调整服务端并发能力，避免服务被打爆。在检索阶段，每次向量检索需从句柄对象池中申请一个空闲的句柄，然后进行后续的计算流程，并在执行完后释放响应的句柄，达到资源回收和重复利用的目的。

在单 GPU 上性能优化后的检索性能与召回率如下(测试数据集同后置过滤):

检索类型	TopK	选取的中心点个数	时延(ms)			召回率(%)
			Avg	TP99	TP999	
Flat	100	-	12.9	25.4	26.2	100
IVF	100	64	1.5	4.2	5.4	87
	100	128	2.4	6.4	7.9	93
	100	256	4.7	11.1	13.3	97

2. 多 GPU 并行检索

除了以上优化方案，还可以考虑将数据分片，通过多 GPU 并行检索，减少单卡计算量来提升检索性能；同时，多卡架构也能支持更大规模的向量数据检索。

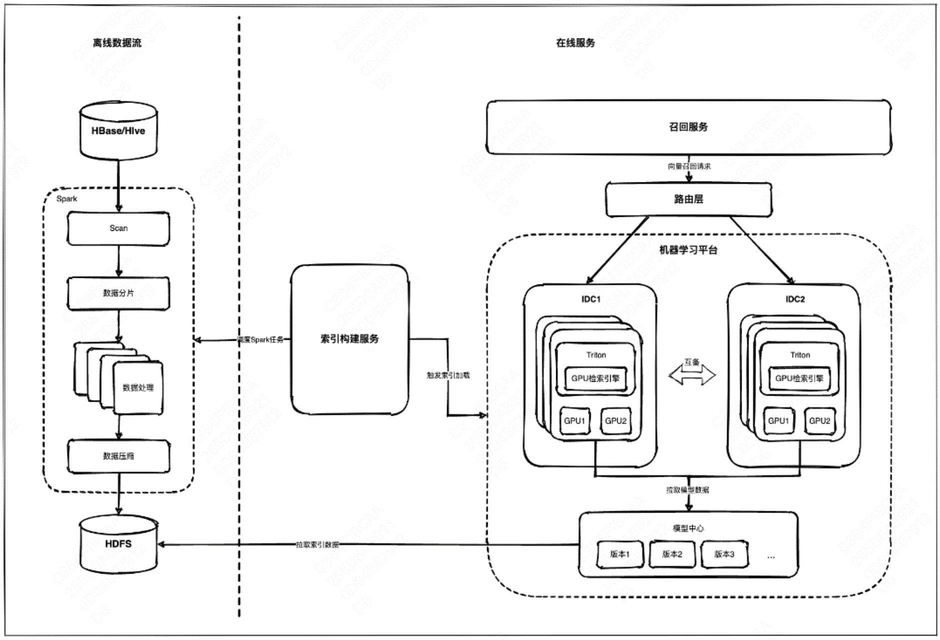
相比多机多卡的分 shard 架构，单机多卡可以有效减少网络传输开销，并且具有较低的索引加载复杂度，因此我们最终选择了单机多卡的数据分片方案，单台服务器部署多张 GPU，检索时并行从本地多张 GPU 中检索数据，在 CPU 内存中进行数据合并。

3. FP16 精度支持

为了支持更大规模的向量数据检索，我们还在 GPU 检索引擎上支持了半精度计算，使用 FP16 替换原来的 FP32 进行计算，可以节省一半的 GPU 显存占用，经验证 Flat 召回率由 100% 下降到 99.4%，依然满足需求。使用半精度之后，单机可以加载近 10 亿数据，足够支撑较长时间的业务数据增长。

4.3 向量检索系统工程实现

向量检索系统的工程化实现包括在线服务和离线数据流两部分，总体架构图如下：



GPU 检索系统上线后实际性能数据如下（数据量 1 亿 +）：

检索类型	TopK	选取的中心点个数	时延（ms）			召回率(%)
			Avg	TP99	TP999	
Flat	100	-	2.1	5.1	10.1	99.4

5 收益

到家搜索团队面向在线服务场景实现的 GPU 向量检索系统，目前已经应用于外卖商品向量检索，向量召回链路的检索性能、召回率均有显著的提升，满足策略对召回扩量和策略迭代的需求，具体提升如下：

1. 向量索引召回率由 85% 提升至 99.4%。
2. 向量索引检索时延 TP99 降低 89%，TP999 降低 88%。

6 展望

- GPU 向量检索系统目前只支持 T+1 全量构建索引，后续计划支持实时索引。
- GPU 向量检索当前支持 FLAT 和 IVF 检索算法，后续计划支持 HNSW 算法，在过滤比较低的场景下可提供更高的检索性能。
- 除了 GPU，后续还会在 NPU 等新硬件上做更多的尝试。

分布式因果推断在美团履约平台的探索与实践

1. 业务背景

近年来，因果推断在商品定价、补贴、营销等领域得到广泛应用并取得了显著的业务效果提升，例如用户增长、活动营销等业务场景。这些领域的共性是需要“反事实推断能力”，传统机器学习算法更关注预测问题，而因果推断提供了更佳的反事实推断能力。以营销活动为例，我们不仅需要知道当前优惠券金额下，订单数是多少（预测问题），还要知道在改变金额的情况下，订单数会发生怎样的变化（反事实问题）。

常见的因果建模方法主要包含 Meta-Learner、深度表征学习和 Tree-Base 算法三大类。其中以因果树为代表的 Tree-Base 算法泛化性强，适用于多种业务场景。相较于 Meta-Learner，树模型建模流程简单；相较于深度表征学习，树模型特征处理和调参过程简单并且具备极强的可解释性。

开源社区涌现出了微软的 EconML 和 DoWhy，Uber 的 CausalML，以及因果森林作者的 grf-lab 等等众多优秀开源项目，但这些项目均为单机实现，不能满足工业场景下亿级样本的模型训练、评估、解释分析。Meta-Learner 和深度表征学习可以轻松借助 XGBoost、LGBM、Spark MLlib、Tensorflow 等开源工具支持海量数据，但是这些项目都不支持因果树相关的 Tree-Base 算法的分布式训练。

具体来说，XGBoost、LGBM、Spark Random Forest 等树模型是为解决预测问题而提出的经典算法实现，而因果树算法引入了新的训练理论以及因果理论独有的干预变量、工具变量等概念。这意味着我们并不能通过对现有分布式树模型的简单改造，来实现因果理论下树模型的分布式训练，而是需要充分理解各类单机因果树算法的原理之后，选择合适的分布式编程范式高效地实现出来。

为了解决上述问题，美团履约平台技术部对开源项目进行了精细梳理，集各家之所长

实现了一套高性能的分布式因果森林框架，在半小时内即可完成亿级样本 100 棵树的训练，突破了单机开源项目仅支持百万级样本的瓶颈。并经过复杂的抽象设计，最终实现通过自定义损失函数即可支持各类因果森林算法的能力，极大提升了框架的扩展性。

除此之外，美团履约平台技术部还在因果效应评估、观测数据去偏等方面建设了大量高效实用的分布式工具。本文将重点为大家分享如何设计实现一个分布式的因果森林算法，以及因果效应评估方面的经验技巧，将我们在分布式因果推断领域的一些探索和内部的实践经验分享给大家。

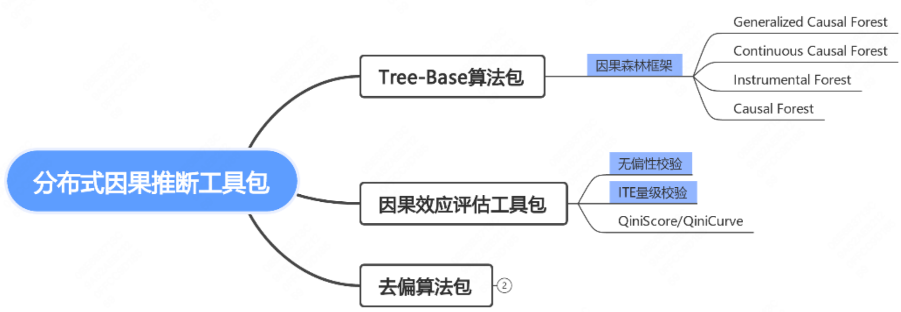


图 1 美团履约因果推断工具包

2. 分布式因果森林框架

因果森林算法的提出引发了 Tree-Base 算法应用于因果建模的研究热潮，众多学者相继在因果森林的基础上提出了多种多样的改进算法。监督学习领域的树模型有众多优秀的开源分布式实现，例如 Xgboost、LightGBM、Spark Random Forest 等等。

但是开源的因果树模型分布式实现基本处于空白状态。因果树算法引入了新的训练理论（比如 Honesty Tree）并且因果树的分裂还依赖于干预变量、工具变量，这导致我们无法通过对现有分布式树实现做简单来更改来实现。因此，我们立足于论文，充分调研并借鉴业内优秀的开源实现，最终设计实现了一套高性能的分布式框架，并能提

供统一的 Serving 方案。

借助这套框架，新增因果森林类算法只需要专注于损失函数设计即可，完全不必考虑分布式的工程实现。截止到目前，我们已经实现了四种因果森林算法，能够灵活支持多维连续 treatment 和及工具变量，半小时内即可完成亿级样本 100 棵树的训练。下面我们将从技术选型与框架设计、性能优化、Serving 实现这几个方面为大家介绍这套框架。

2.1 技术选型与框架设计

单机树模型的工程实现可以概括为：遍历所有潜在的切分点并计算分裂指标（损失函数），取指标指标最佳的分裂点分裂，不断分裂树节点直到满足退出条件。而分布式环境下每台机器只包含部分样本，分布式环境下任何全局指标计算都会带来极大的通讯成本，因此需要选择合适的分布式架构帮助我们计算分裂指标。

因此，对于分布式因果森林框架，我们关心三个问题：第一，如何计算因果树的分裂指标（损失函数）；第二，如何求潜在分裂点；第三，选用何种分布式编程架构。在此基础上进一步抽象整合，就可以实现不同树模型共用一套分布式框架的目标。

从论文出发

为了深入了解因果森林类算法，我们仔细阅读了因果森林论文以及其作者 Susan Athey 的另一篇在因果领域有重要影响力的《Generalized Random Forests》论文。Susan Athey 认为随机森林本质上是一种自适应的最近邻算法（KNN），也就是通过对样本空间的递归划分从而找到距离该样本点最近的 K 个点（落入同一个叶子节点）来表示该点的值。而因果森林算法本质上是随机森林算法在因果推断领域的一种特殊应用。

因果森林和传统分类、回归森林一样采用了二叉的 CART 树（Classification And Regression Tree）作为基模型。与分类和归回问题相同，特征值仅用于样本划分而不参与分裂指标的计算。不同之处在于，分类和回归问题仅研究预测观测值 Y，而因果建模需要研究 treatment、instrumental variable 等变量与观测值 Y 之间的关联。

此外，多维连续 treatment 是学界的热门研究方向。因此，相较于分类和回归问题，因果推断需要在样本表示上做出相应调整。

因果森林论文提出 honestyTree 的概念：将样本分成 growSet 和 predictionSet 两个部分，growSet 用于树的生长，predictionSet 用于 prediction 值的计算。在论文《Generalized Random Forests》中证明了最小化子节点评估值与真实值之间的误差等价于最大化左右节点间的异质性，并对 CART 树的生长过程做了更加广义的抽象，将其分解成 labeling step 和 regression step 两步。Susan Athey 的单机 C++ 开源项目 grf-lab 中将这两种观点融合在一起，把树的生长定义为 relabeling/splitting/prediction 三个步骤。

综上，我们可以得出一些指导方案设计的结论：

1. 因果森林本质上是 CART 树 Bagging 算法在因果建模领域的特殊应用。因此 CART 树相关的论文和开源项目都可以广泛借鉴。
2. 不同于 CART 树，因果树的样本表示需要做相应抽象，根据不同算法灵活支持单维 treatment 多维 treatment 和工具变量。
3. 因果树的支持 honestyTree，可以将树的生长拆分为 relabeling/splitting/prediction 三个步骤，根据不同算法灵活实现。

Pre-sorted Algorithm Or Histogram-based Algorithm？

主流 CART 树模型求分裂点的实现有两种方式，以早期 XGBoost 为代表的预排序算法，以 LightGBM 和 SparkRandomForest 为代表的直方图算法（目前 XGBoost 也提供了直方图算法的实现）。

1. **预排序算法**：对每一个特征的所有取值排序，依次遍历这些值计算分裂指标，取指标最佳的分裂点将节点分裂为左右子节点。
2. **直方图算法**：直方图的主要思想是将连续特征离散化到最大 k 个桶中，同时构造一个宽度为 k 的直方图。在遍历样本时，以离散化值为索引在直方图中累积统计量。遍历每个特征的每个分桶计算分裂指标，取指标最佳的分裂点将

节点分裂为左右子节点。

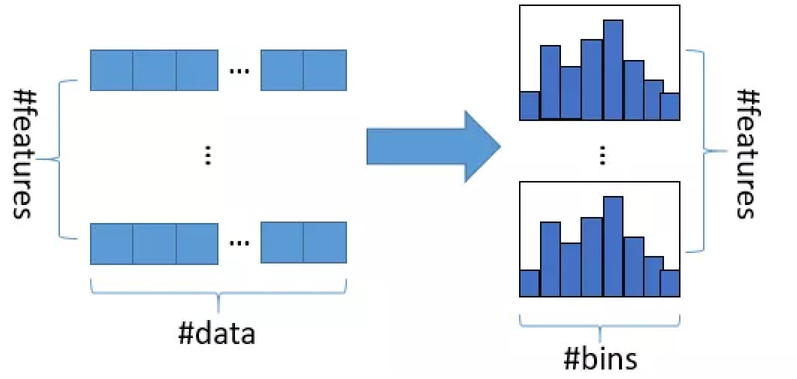


图 2　离散化分桶

图 3　直方图作差

相较于预排序的实现，直方图算法的时间复杂度由 $O(data*features)$ 降低为 $O(bin*features)$，同时离散化后的特征内存占用更低，并且可以通过直方图作差的方式（父节点直方图减去左节点直方图）进一步降低计算量。受限于篇幅，预排序算法与直方图算法的差异这里不再赘述。最终我们选择了直方图算法方案，这也意味着需要在框架中采样计算直方图和特征离散化的环节。

AllReduce Or MapReduce？

工业界主流的分布式机器学习架构有 AllReduce、ParameterServer、MapReduce 三种，其中 AllReduce 性能最高（ParameterServer 架构也可以和 AllReduce 结合，为了方便讨论，这里不再细究）。

架构	实现	性能	代表框架
AllReduce	C++	最优	XGBoost、微软 LightGBM、谷歌 Tensorflow
Parameter-Server	C++	居中	谷歌 Tensorflow (PS 模式)、Tencent Angel，主要应用在深度学习领域
MapReduce	Java/Scala	一般	Spark MLlib、H2O（Uplift Random Forest）

因为 XGBoost 内建了一个 AllReduce 框架 RABIT 可以直接复用，因此我们迅速拟定了两个调研方向——复用 XGBoost 的 AllReduce 高性能实现和 Spark MapReduce 实现。

方案	架构	明细	性能	技术栈	开发难度	测试难度	支持的样本量级
方案 1	AllReduce	XGB RABIT + Spark	高	C++ 和 Scala	高	高	百亿
方案 2	MapReduce	Spark	一般	Scala/Java	较高	较高	十亿

由于履约使用的样本量在几千万级别，综合考虑开发测试成本和训练性能后，我们最终选择了 MapReduce 方案。

框架设计

综合上文的分析，我们为分布式因果森林框架设计了 4 个模块：

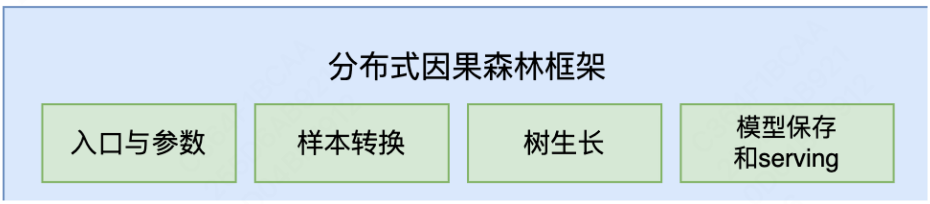


图 4 分布式因果森林框架

1. **训练入口与参数模块**：抽象出 Abstract CFEstimator 用来整合树模型的通用参数，新增算法继承此类后添加专属参数即可作为对应算法的训练入口。
2. **样本转换模块**：负责采样构建直方图与特征离散化，上文中单维 treatment

多维 treatment、工具变量、观测值 y 的转换也封装在此模块中。

3. **森林生长模块**：框架的核心模块，使用 MapReduce 实现。包含随机森林需要的树采样、特征采样，同时实现 honesty。抽象出 relabeling/splitting/predcition 这几个接口，不同的算法按需实现树的生长逻辑，并以此为基石抽象损失函数接口。
4. **模型保存和 serving 模块**：抽象出统一的树模型保存和加载方案。

2.2 性能优化

在选定 MapReduce+ 直方图的方案后，我们迅速将目光锁定在同样使用直方图算法的 Spark RandomForest 算法上（以下简称 SparkRF）。我们在 SparkRF 上快速实现了一版分布式因果森林框架，并进一步实现了 Generalized Causal Forests 算法。

但是测试过程中我们发现，随着总节点数的增加，跨节点通信量（也就是 Shuffle）剧增，同时还非常容易溢出。为了支持更大规模的模型训练，我们从跨节点通信、内存占用、计算复杂度、剪枝以及 CPU 缓存命中优化等多个方面优化了整个框架。为了讲清楚我们优化逻辑，大家先来看看 SparkRF 是如何实现的。

SparkRF 的实现

SparkRF 整个实现过程可以概括为如下几个步骤：

（1）将全量样本离散化并 cache 到内存，这一步包含三部分：

- 采样 collect 到 driver 为每个特征等距分桶，得到潜在切分点 split。
- 使用潜在切分点 split，将每个样本的特征离散化，此时特征值从 double 被转换成 int。
- 根据树采样比例，为每条样本生成标记数组（由 int 数组实现），标记这条样本用于哪棵树的生长。

（2）树的生长

- 将整个森林看做一张图，采用深度优先搜索待分裂的节点，一次迭代一组节点，由 maxMemoryInMB 参数控制节点数。
- 根据样本的标记数组，计算每个样本在每个节点的每个 split 下的直方图（统计信息）。
- 通过 reduceByKey 算子，将同一个待分裂节点的所有 split 下的直方图汇总到同一个 worker 中。
- 将待分裂节点的每个切分点直方图积分，例如 feature0 有 3 个切分点 [a,b,c]，积分后为 [a, a+b, a+b+c]，使用直方图作差，计算左右子节点增益，获取最佳切分点。
- 将待分裂节点的最佳切分点 collect 回 driver，完成森林的生长。
- 使用 rdd cache 记录样本所属节点 id（由 useNodeIdCache 参数控制）或广播模型。
- 持续迭代直到达成退出条件。

可以看到，Spark 的实现除了直方图，还有不少精妙的地方。例如在每次可训练的总结点数有限的情况下，深度优先搜索相较于广度优先搜索更倾向于快速完成单棵树的训练，从而减少后续训练需要广播的树模型。篇幅所限，下面将主要为大家介绍分布式因果森林框架在内存占用方面的优化。

减少 Cache 体积

从上文可以看出，SparkRF 使用 int 来表示最大分桶个数，而 lightGBM 使用无符号 byte 来存储，支持最多 256 个分桶。我们认为 128 个分桶足以支撑因果森林的业务需要，所以使用了有符号 byte 来表示分桶，相比 int 内存占用减少至 1/4。

前文中提到，SparkRF 为每个样本创建了一个标记数组。例如训练一个 2 棵树的森林，这个标记数组为 [4,0]，这表示此样本在 tree0 有放回采样 4 次，在 tree1 未被使用。此外，框架需要支持 honestyTree，也就意味着需要另一个标记数组记录样

本在 growSet 还是 predictionSet。考虑到无放回采样足以覆盖绝大部分场景，并且为了不引入第二个标记数组，我们最终选择了 BitSet 实现。每棵树最多使用 2 个 bit，1 个 bit 表示是否是该树的样本，1 个 bit 表示是否是 honesty 样本。当关闭 honesty 或者不使用下采样时，每棵树只需要 1 个 bit，内存占用最多减少至 1/32。

支持更大模型广播

上文中提到，SparkRF 每一轮迭代调用 reduceByKey 之前都需要计算出哪些样本属于待分裂的节点，Spark 通过 useNodeIdCache 参数提供了两种策略：

- **策略一**：每次迭代将树模型跟随闭包广播到各个 worker 节点通过 predict 获取节点 id。
- **策略二**：使用 RDD[Array[Int]] 类型来缓存当前样本隶属于每棵树的哪个节点（例如训练 100 棵树，则创建长度为 100 的 int 数组，每一个元素记录了此条样本在对应下标的树模型中的叶节点编号）。

从源代码中我们发现，策略二每一轮迭代都会卸载上一轮持久化的 nodeIdCache，再创建一个新的 nodeIdCache 持久化到内存。以 1 亿条样本 100 棵树的森林举例，每一轮迭代就是 1 亿个长度为 100 的 int 数组的创建与垃圾回收。实际测试中我们也发现策略二的效率不如方案一高。那么策略一又如何呢?

SparkRF 在每一轮迭代中能够训练的最大节点数由 maxMemoryInMB 控制，我们希望通过增大这个参数来减少迭代次数。但随着树或树深的增加，往往陷入增大该参数就导致树模型广播到 worker 溢出的尴尬境地。经过对 SparkRF 源码分析，我们发现每个 LearningNode 都会存储当前节点、左子节点、右子节点的直方图，最终实现在一套通用框架下计算出每个节点的增益、纯度、预测值等等属性，但这导致了 3 倍的内存占用。

考虑到因果森林 honestyTree 原则，叶节点 prediction 值的计算使用 predictionSet，因此生长过程中每个节点全都带着 growSet 的直方图是完全没有意义的。因此我们优化了树的生长逻辑，每个节点仅保留自身的直方图，对于已分裂的节点则

清除直方图。以二叉满树为例，叶节点约占整棵树节点的 1/2，结合直方图从 3 倍冗余到 1 倍存储，这一优化使树模型直方图的内存占用下降到原本的 1/6，极大降低了模型体积。

BenchMark

经过一系列优化，最终实现了百棵树亿级样本小时级训练的目标。

样本量	特征数量	树棵树	最大树深	资源配置	Generalized Causal Forest算法	Continuous Causal Forest算法
1 亿	127	100	8	400*(7core16g)	29min	17min

备注：不同森林算法的复杂度不同，跨节点通讯量不同，总耗时会存在明显的差异。

2.3 Serving 实现

因果森林本质上是随机森林算法的变种，由一棵棵彼此独立的二叉因果树构成，每棵树由 innerNode 和 leafNode 构成。其 prediction 的逻辑非常简单，每棵因果树单独 predict 获取 leafOutput 向量，森林中所有树预估的 leafOutput 向量求均值即可得到森林的输出值。因此，整个树模型的结构其实非常清晰，innerNode 存储特征 split 信息，leafNode 存储输出向量。除此之外还包含 gain、impurity、count 等属性用于计算特征重要性。

模型 serving 除了性能还需要考虑模型离线存储体积、模型的内存占用、模型字段的扩展性。结合因果树的特点，就需要特别注意 leafOutput 向量的实现。以下表中的场景为例，使用 float 数组大约就需要 500*4096*40* 4 byte / 1024/ 1024 = 312.5mb，而 List 则需要约 4 倍内存，正因如此我们快速放弃了简单快捷的 Protobuff 方案。

树	树深	满树节点数	满树叶节点数	叶节点统计指标长度
500	12	8191	4096	40（例如 ccf 算法 20 维 treatment 下的输出）

为什么要重视模型字段的扩展性呢？这是因为离线模型训练追求快速迭代而在线

Serving 追求稳定性。模型的扩展性好，不仅可以轻松做到新版本服务向下兼容老模型，还可以做到在不使用新特性的情况下，老版本服务向上兼容新模型，从而减少在线服务更新发版的次数。综合考虑以上因素以及对 Spark 的兼容性和对 java serving 生态的兼容性，我们设计了如下方案。

1. 使用 parquet 文件格式存储模型文件。
    - 字段扩展性：好，读取类似 KV，模型文件可以随意扩展而不影响线上服务
    - 模型内存体积：好，相较于 protobuf，可以逐行读取转换为 float 数组而非 Float List
    - 模型存储体积：好，采用 snappy 算法压缩
2. 字段平铺的方式存储树模型。相较于 SparkRF 的采用 tree-node 嵌套的方式，更利于字段扩展。虽然会带 treeId 等个别字段的冗余存储，但是列存储的压缩效率非常高，影响很小。
3. 提供独立 jar 包 cos-serving 实现模型加载和 prediction 的功能，实现了离线模型训练升级而在线服务可以不升级的目标。

我们将离线模型的保存和加载逻辑抽象封装到了因果森林框架中，进一步增强了因果森林框架的扩展性，开发新森林算法时专注于将论文中树的生长逻辑实现即可。

3. 分布式因果效应评估

业内常见的因果效应评估手段主要评估 ITE 的序关系，例如 qini score 和 auuc。但是存在如下三方面不足：

1. 缺乏对数据和模型无偏性的校验
2. 缺乏因果效应量级关系的评估，qini-score 和 auuc 只能反应弹性的序关系
3. 开源因果评估工具都是单机实现，仅支持百万级样本的计算

下文将为大家一一进行说明。

3.1 无偏性校验

无偏性校验分为数据无偏性和模型无偏性。

数据无偏性校验可以通过 X ⊥ T 验证。首先可以训练一个 X->T 的倾向性得分模型，如果倾向性得分模型的 auc 在 0.5 附近则说明 X 无法正确地预测 T，也就是说 X ⊥ T，此时数据无偏。例如，使用了 post-treatmen 特征会导致特征穿越，最终导致数据是有偏的，这时候使用 X ⊥ T 的校验工具可以快速帮我们排查出这一类问题。

模型无偏性校验使用 ITE ⊥ T 验证。首先用训练好的弹性模型在随机实验数据上预测 ITE，接着对样本按照 ITE 升序排列后等频分桶，计算每个 ITE 分桶下实验组样本占比（下图的 trtRatio 曲线）。理想情况下，每个 ITE 分桶中实验组样本占比应该和随机试验中实验组样本占比一致，此时 ITE 正交于 treatment。比如，随机实验中实验组比对照组为 1 比 1，那么 trtRatio 就应该在 1/2 附近浮动。如果 trtRatio 比例不符合预期，我们就可以进一步去排查模型结构的问题。这项工具更是作为标准测试组件融入到分布式因果森林早期的开发过程中。

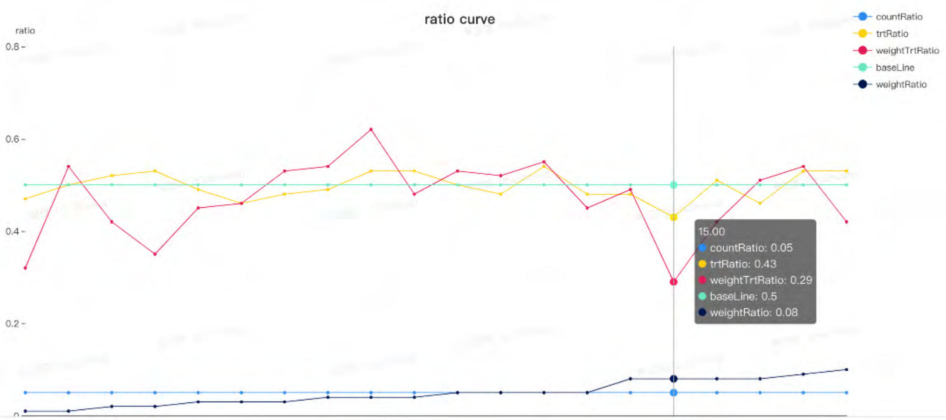


图 5 模型偏差大

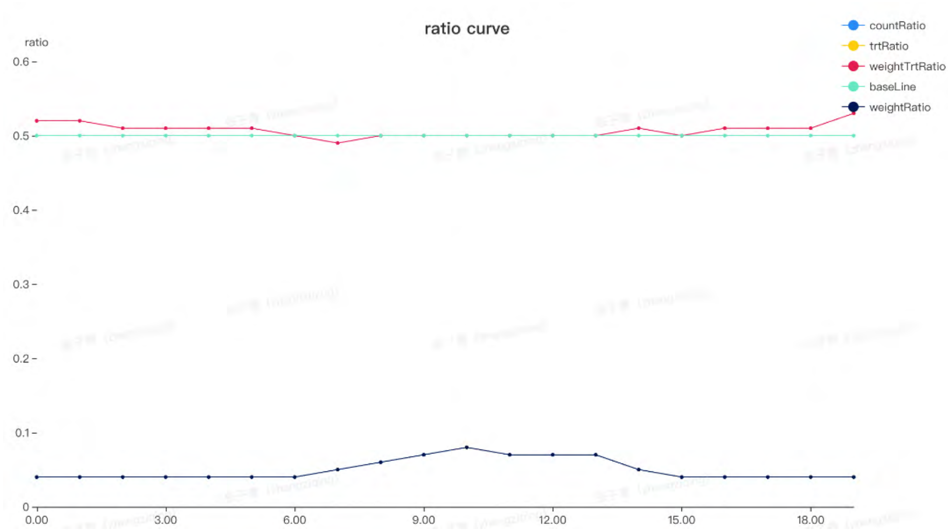


图 6　模型偏差小

3.2 因果效应量级关系评估

因果效应的序关系和量级关系同样重要，只是将弹性的序关系学习准确而没有将弹性的量级关系学习准确，决策者无法预估该 treatment 对用户的影响程度。例如，将量级错误的弹性应用到运筹优化决策中，可能会导致无法满足重要约束从而无法求得可行解。针对弹性量级无法评估的问题，我们在原有的 qini_curve 基础上增加了 qini_pred_curve_counterfactual 和 qini_pred_curve。

qini_curve 及其扩展

qini_pred_curve_counterfactual：将每个样本按照模型预测的 ITE 降序排列，按照如下公式依次计算前 t 个样本的反事实 qini_pred 即可得到曲线。

$$qini\_pred\_counter(t) = pred\_ite_t \cdot \frac{N_t^T}{N_t^C + N_t^T}$$

- $pred_ite_t$ 代表前 t 个的样本 ITE 累加。

- N_t^T 代表前 t 个样本中 treatment 组样本数量。
- N_t^C 代表前 t 个样本中 control 组样本数量。

通过比较 qini_pred_curve_counterfactual 和 qini_curve 这两条曲线的重合程度和右端点纵坐标，我们可以观察出 ITE 的预估量级和真实量级是否一致。

qini_pred_curve：每个样本按照模型预测的 ITE 降序排列，按照如下公式依次计算前 t 个样本的 qini_pred 即可得到曲线。

$$qini\_pred = pred_t^T - \frac{pred_t^C \cdot N_t^T}{N_t^C}$$

- $pred_t^T$ 代表前 t 个的样本中 treatment 组样本预估 outcome 的累加。
- $pred_t^C$ 代表前 t 个样本中 control 组样本预估 outcome 的累加。

qini_pred_curve 和 qini_pred_curve_counterfactual 差异越大，模型偏差越大，也就是 ITE 与 T 不正交。我们以下图的案例来说明这三条曲线。

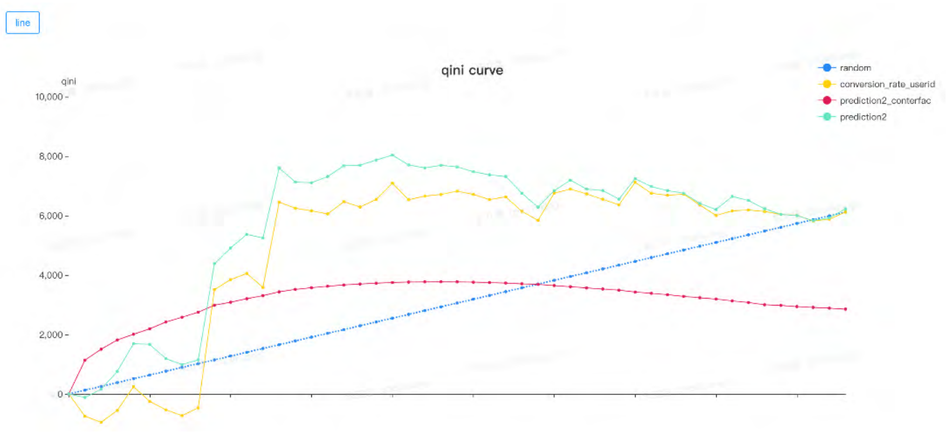


图 7　模型偏差大

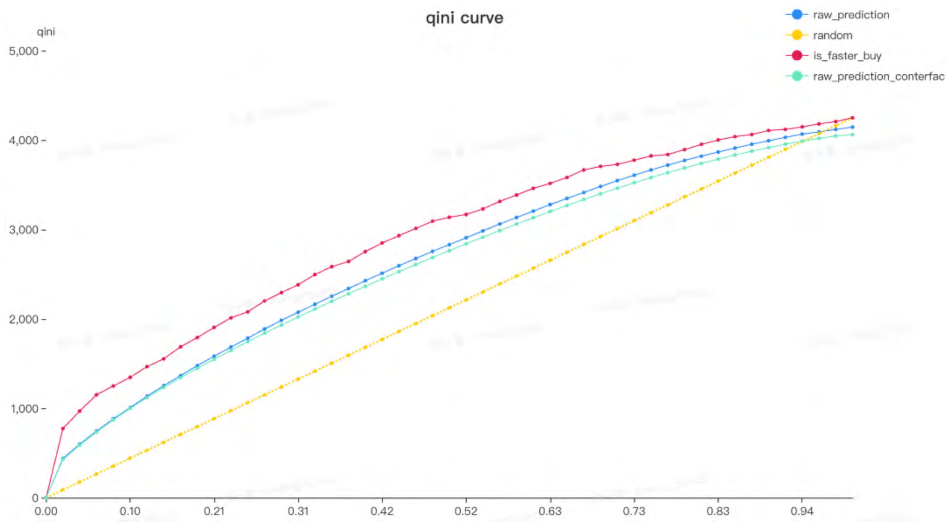


图 8　模型偏差小

根据这些曲线的形状、覆盖面积、重合程度，我们可以得到如下的判断：

1. 如果数据无偏，那么 qini_pred_curve_counterfactual 会和 qini_pred_curve 重合，反之则表示数据有偏，即 ITE 不独立于 T。
2. qini_pred_curve_counterfactual 和 qini_curve 的右端点纵轴的差距，代表了弹性预估的量级和弹性真实的量级存的差距。
3. label 曲线的 qini score>0.5，也就是 label 曲线有明显向下的趋势时，存在过拟合现象，即学到了负弹性。
4. 如果弹性模型对于弹性序关系和弹性量级关系学习得非常准确，那么三条曲线会几乎重合在一起。

avgITE 和 ATE 的对比

上文中提到的三项指标都是累计因果效应的评估，我们还想更有针对性地观察每个弹性分桶下预估因果效应和真实因果效应量级的差异，所以开发了 avgITE 和 CATE 的对比工具。

同样将样本按照模型预测的 ITE 降序排列，然后等频分桶，统计每个分桶内预估 ITE

的均值（下图的 avgITE 曲线）和 CATE 值（下图的 cate 曲线）。对比 avgITE 和 CATE，可以评估出真实因果效应和预估因果效应量级的差异。

$$avgITE = E(pred_y(X_i, T_i = 1) - pred_y(X_i, T_i = 0))$$
$$CATE = E(Y_i|T = 1) - E(Y_i|T = 0)$$

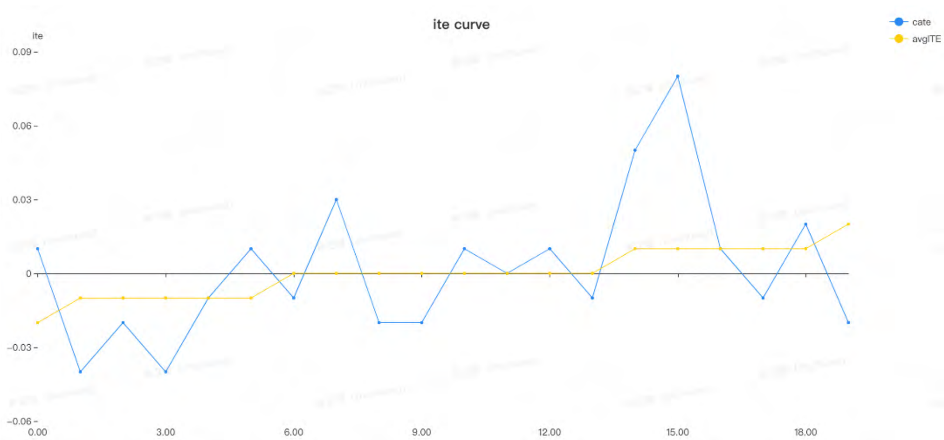


图 9　预测与真实 ITE 量级偏差大

3.3 分布式评估体系

早期我们也使用了 pandas 实现的单机评估算法，当样本量增加到 400w 条以上时遇到了严重的单机瓶颈。为此，我们对上述评估指标全部做了分布式改造。排序类指标的实现有分桶积分和逐条积分两种实现思路。考虑到逐条积分会有更高的精度，最终选择了分布式环境下逐条积分的方案。

不仅如此，我们还使用 Spark 实现了带权重的分布式的因果效应评估，能够支持十亿样本的评估。此外我们还融入了评估预估 y 与观测值 Y 之间的差异的指标，包括 mae/mse/rmse，并将这些指标封装到二元因果效应评估组件中。由于我们实现的部分因果森林算法能够输出多元 treatment 下预估的 y，因此我们还进一步封装了多元因果效应（拆分成多个二元因果效应）评估功能。

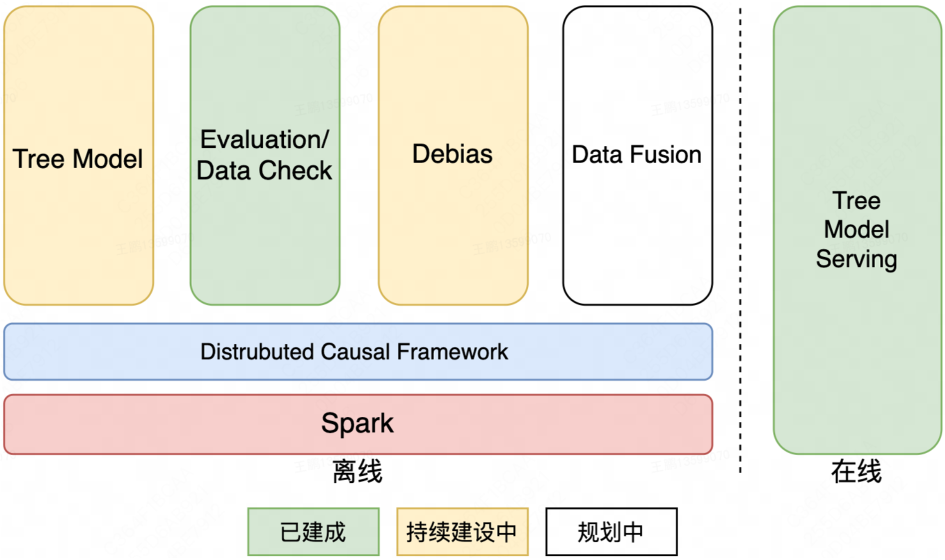


图 10　Causal On Spark

4. 总结

经过两年持续迭代，我们实现的分布式因果推断工具包已经发展成集模型训练、评估、去偏、Serving 于一身的综合型因果工具包。我们内部为这个项目命名为 Causal On Spark，简称 COS。目前这个项目也已经全部集成到图灵机器学习平台中。将来有机会我们会再次为大家分享美团履约技术团队在分布式因果推断领域的探索和实践经验。

5. 参考资料

[1] Wager S, Athey S. Estimation and inference of heterogeneous treatment effects using random forests[J]. Journal of the American Statistical Association, 2018, 113(523): 1228-1242.

[2] Athey S, Tibshirani J, Wager S. Generalized random forests[J]. The Annals of Statistics, 2019, 47(2): 1148-1178.

[3] Li, G., Chen, Q., & Usunier, N. (2017). LightGBM: A Highly Efficient Gradient Boosting Decision Tree. Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS 2017), 3146-3154.

[4] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System.

Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16), 785–794.
[5] 微软亚洲研究院:《开源 | LightGBM:三天内收获 GitHub 1000 星》.
[6] https://grf-labs.github.io/grf/index.html.
[7] https://github.com/uber/causalml.
[8] https://github.com/apache/spark.

测试 & 数据 & 安全

AutoConsis | UI 内容一致性智能检测

美团到店研发平台质量工程部与复旦大学计算机学院周扬帆教授团队展开了大前端智能化测试领域的科研合作，从 UI 界面内容一致性校验入手，并实现了一套自动化智能检测流程，相关论文被软件工程领域具有国际影响力的会议 ICSE 2024（CCF-A 类会议）的 Software In Practice Track（软件工程实践应用）收录。

1. 背景

目前，移动 App 上的业务页面愈发复杂，技术团队常会以页面为单位来拆解团队开发分工，同一类业务元素信息分散在不同团队负责的页面内。在具体的实践中，存在一类不易检出但又影响用户体验的异常：页面中的 UI 信息相互矛盾（如下图中同一个商品在多个页面上的实际价格不一致）。此类 UI 内容不一致的异常，没有固定的出现位置和表现规律，长期以来主要依赖测试人员对于 UI 的熟悉度，主要靠手工测试执行时来随机进行发现。

图：界面间 UI 内容不一致举例（示意图）

美团 App 中的众多业务具备内容繁多的多种页面布局，以及多技术栈共存，如何低成本地在多类业务、多样化布局、多技术栈的 UI 页面间检测内容一致性，是终端测试领域中的一项重要挑战。

图：UI 和 API 服务对应关系（以价格计算为例）

为解决该问题，美团到店研发平台质量工程部与复旦大学计算机学院周扬帆教授团队展开了大前端智能化测试领域的科研合作，在 UI 认知与校验方面积累了多项智能技术。在应用方面，我们选择从 UI 界面内容一致性校验入手，对多个页面状态信息的合理性与一致性进行自动化检查，并在美团 App 中的多类营销业务场景中进行落地。

在方法上，我们针对 UI 层面设计并实现了一套自动化智能检测流程，取名为 AutoConsis，在 UI 内容一致性检测上做到了低成本、高泛化性、高置信度。该工作对于大前端 UI 的质量保障多个领域都具有可借鉴意义，并产出了一篇学术论文《AutoConsis: Automatic GUI-driven Data Inconsistency Detection of Mobile Apps》，该论文被软件工程领域具有国际影响力的会议 ICSE 2024（CCF-A 类会议）的 Software In Practice Track（软件工程实践应用）收录。

2. 实现原理与项目实践

本文以特价商品营销业务为例，来介绍智能化 UI 内容一致性检测所需要的能力。特价商品营销是一种大型促销应用，与之相关的一致性测试涉及数百个城市，单个城市内的多个商品品类，每个商品的多个所涉及页面，其状态空间非常复杂。传统的自动化测试方法需要对各个状态逐一适配，成本极高。理想情况下，我们需要提供一种低成本、易适配、可以覆盖所有状态的能力，同时还需要较强的泛化迁移能力，在不同页面模板、技术栈、App 上自动进行适配。

具体到特价商品营销的一致性校验，其包含营销会场页、详情页、提单页等多个页面，不同页面之间价格的一致性是检查的重点。对于每一件商品，我们希望从上述三个页面中识别出商品的价格，并判断其是否一致。需要注意的是，由于商品的价格受优惠活动的影响，而优惠活动有多种形式，因此如何准确分析出各页面中商品的价格是主要难点。

2.1 总体流程

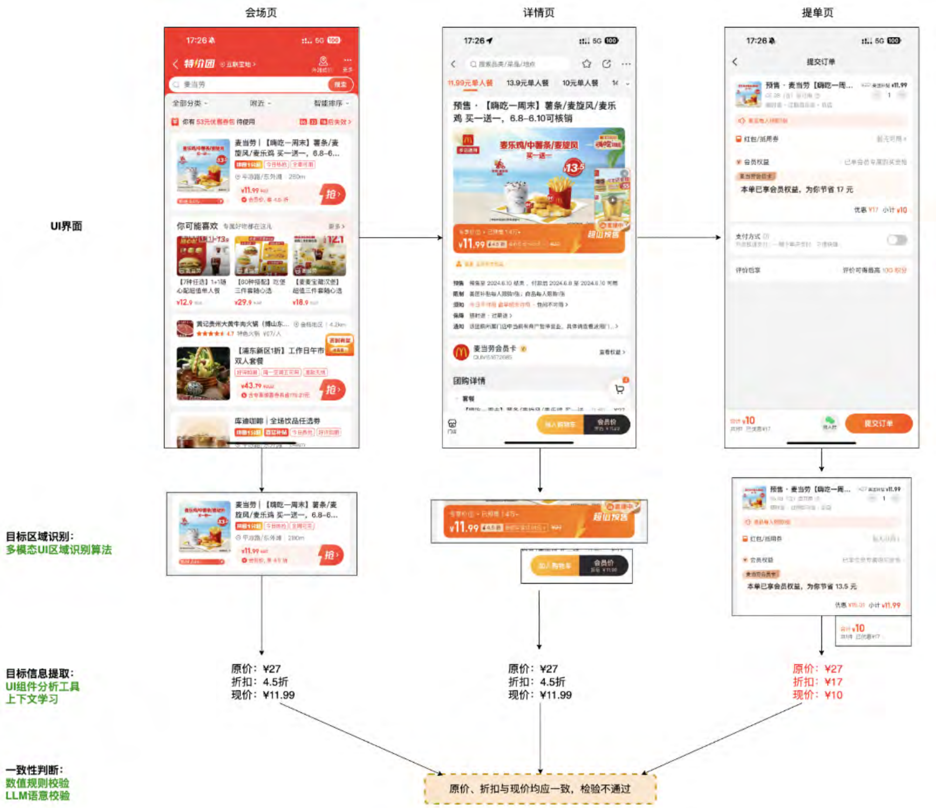
AutoConsis 的核心优势在于其在泛化性、准确性上的良好效果。电商平台中的内容

一致性校验，包括商品名称、描述、价格、库存方面的校验，本质是对 UI 页面的特定信息提取，价格一致性采用的也是信息提取流程。我们将 UI 页面分析任务转化为目标检测和内容理解的组合，利用了大语言模型的能力，实现对不同技术栈页面的适应能力。以下介绍工具的设计过程。

AutoConsis 有三个关键处理流程：

- 目标区域识别：AutoConsis 首先识别 UI 界面中与检测相关的关键区域。通过图像处理和模式识别，工具能够准确地定位到包含重要信息的 UI 部分。
- 目标信息抽取：在目标区域确认后，经过 OCR 和 UI 组件分析工具（使用自建的开源系统能力 Vision-UI）提取目标区域的文本和元素，填入预先设置好的 CoT Prompt（CoT，Chain of Thought），通过大模型推理提取一致性校验所需要的关键信息。
- 一致性判断：AutoConsis 对提取出的信息进行一致性校验，确保 UI 信息的准确性和一致性。

下面我们会以「购买特价商品」场景为例来介绍 AutoConsis 的工作流程。如下图所示：AutoConsis 首先从一个营销会场页开始分析，其主要包含一个商品列表，经过目标区域识别模块识别出各个商品卡片，再从各个商品卡片的截图中识别出原价和折扣等金额相关的信息。之后继续提取每个页面的相关价格信息，最后由一致性判断模块检查页面间的价格一致性。



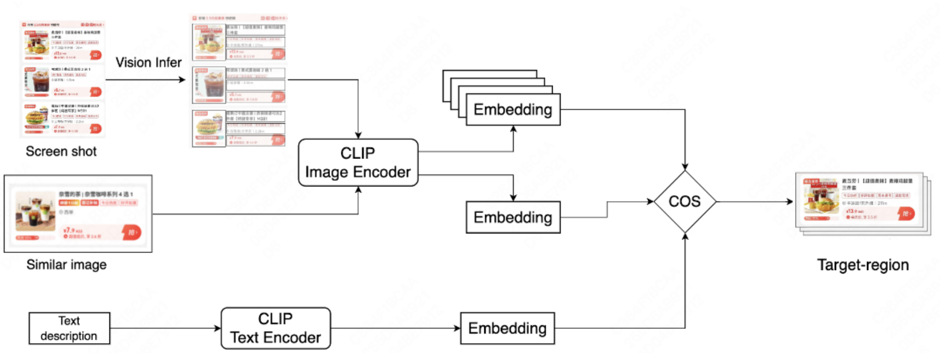
图：AutoConsis 的工作流程（以购买特价商品为例）

2.2 目标区域识别

为了提取关键的一致性信息，在实践中我们发现：将页面上所有的文字 OCR 识别后直接交给 LLM 分析，得到的分析结果并不准确。我们分析后认为，UI 界面包含大量的与当前检测需求无关的文字，噪声过大干扰了 LLM 的判断。因此，我们考虑对 UI 界面进行裁切，即通过目标区域识别的方式将无用的文字信息去掉，从而使 LLM 聚焦。

图像的目标识别是 CV 领域的传统方向，基于传统深度学习的目标检测模型对适配复杂多变的 UI 界面成本很高，需要进行大量的数据标注和训练工作才能够使用，同时泛化性也一般。为了使方法具备良好的泛化性，同时能够以较低的成本使

用，我们采用了基于视觉语义的识别模型 CLIP（Contrastive Language-Image Pre-training，由 OpenAI 提出的图像 - 文本语义匹配模型）。CLIP 可以将图像和文字的语义映射在同一个高维向量空间，且由于经过海量数据的训练其具备良好的通用性。针对 UI 的目标区域识别场景，我们对其原有设计进行了改进。具体来说，我们提供图像和文本两种检索词并设置权重进行多模态的匹配。



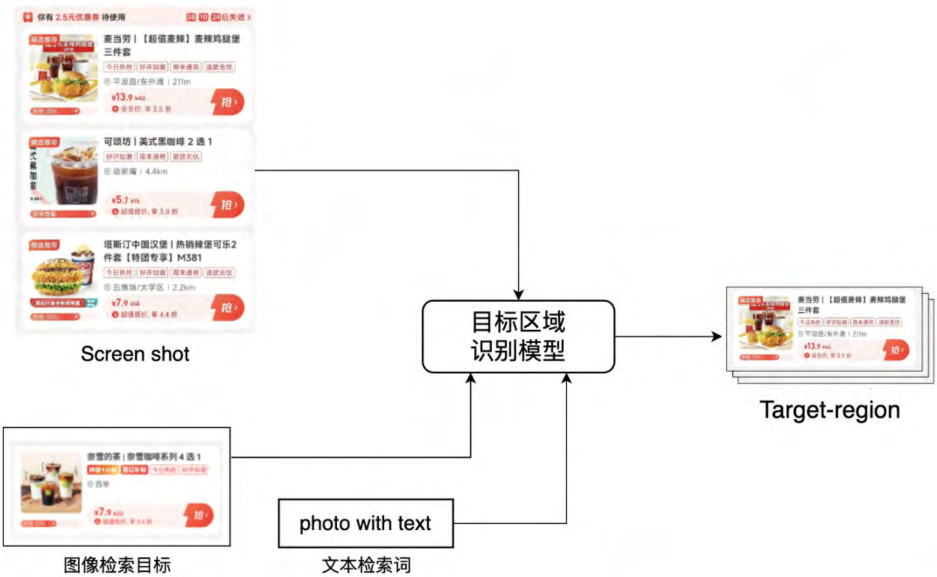
图：目标 UI 区域识别流程

为了验证上述多模态 UI 区域识别流程的有效性，我们设计了检索词仅包含图像和仅包含文本两个单模态的目标区域识别算法作为 CLIP 多模态算法的对照组。考虑到商品信息在线上购物应用中的核心作用，以及其用户界面通常较为复杂，我们决定选用商品信息界面作为 UI 区域识别算法的测试场景。具体来说，我们收集了 100 个商品列表页，测试多模态 UI 区域识别算法和两个对照方法从中识别商品卡片的效果。实验收集到了如下数据：

Input Type	Precision	Recall	F1-Score
Text Only	0.266	0.108	0.154
Image Only	0.714	0.414	0.524
Multi-modal	0.981	1.000	0.991

由该实验数据可见，经图像和文本两种信息相互补充（Multi-Modal），AutoConsis 采用的识别算法能有效识别出目标 UI 区域。

对于营销会场页而言，如 UI 区域识别流程图所示：我们将一个会场页的 UI 截图送入识别模型，并提供一个商品卡片和对应的文本描述作为检索词，该多模态模型会根据检索词从经 UI 组件分析处理过的会场页中筛选出近似的商品卡片。



图：营销会场页商品卡片识别示例

2.3 目标信息提取

目标区域的识别可以减少 UI 中大部分无关信息，但判断 UI 内容是否一致还需要关键的校验点信息（例如商品价格一致性的关键信息是价格和折扣），因此还需要对提取到的 UI 目标区域做关键信息提取。由于 UI 的样式多样，同类业务上的关键信息（如商品详情卡片上的折扣）往往有多种表达形式，难以通过通用性规则准确提取。对此，AutoConsis 借助了大语言模型的理解能力对页面进行理解分析，同时针对大语言模型常见的“幻觉”问题，我们按照上下文学习（即 In-Context Learning，指一种让 LLM 在提示词中进行类比学习的增强手段。）的思路参考“思维链 CoT”的范式设计

了信息提取 Prompt。

在流程上，对于每一个 UI 目标区域，AutoConsis 利用 OCR 提取所有可识别的字符，随后将分词的结果与 CoT 示例进行拼合构成 Prompt，最后从 LLM（AutoConsis 的实验部分调用 GPT-3.5-Turbo 完成）的输出中获取一致性检验所需的关键信息。同样，为了定量地探究 AutoConsis 所采用 Prompt 的有效性，我们设计了两种 Prompt 进行了消融实验：去除了推理过程的 Standard ICL（Standard In-Context Learning）和直接去除示例的 Zero-shot。

Standard ICL Prompt

```
Here is an examples for your reference:
Here is a list of phrases from commercial mobile apps: ['和平路','花园蛋糕', '香草蛋糕','已售66%','甜点', '随时退', '¥20', '4折', '速抢', '¥50', '会员价']

Q: Given the text list from commercial mobile apps, analyze and extract the following data:

- original price
- discount:
- price after discount:

A:
- original price: "50",
- discount: "30",
- price after discount: "20"

My question is as follows, please ensure that your answer is in the same format rasthe examples I provide:
Here is a list of phrases from commercial mobile apps: [$Phrases]

Q: Given the text list from commercial mobile apps, analyze and extract the following data:
- original price:
- discount:
- price after discount:

A: [To be completed]
```

Zero-shot Prompt

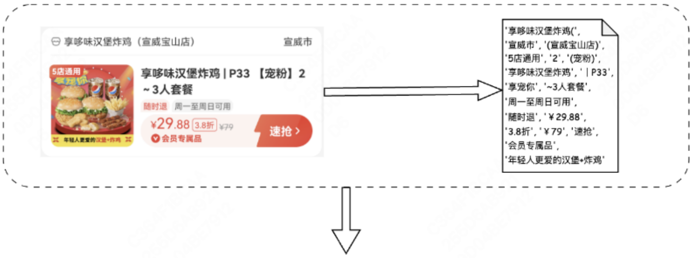
```
My question is as follows, please ensure that your answer is in the same format rasthe examples I provide:
Here is a list of phrases from commercial mobile apps: [$Phrases]

Q: Given the text list from commercial mobile apps, analyze and extract the following data:
- original price:
- discount:
- price after discount:

A: [To be completed]
```

图：两种对照 Prompt 示例

选取的实验场景为从商品卡片中提取价态信息，实验数据见下表：

Prompt Type	Acc.	Precision	Recall	F1-Score
Zero-Shot	0.507	0.696	0.708	0.702
Standard ICL	0.796	0.927	0.927	0.927
CoT	0.939	0.971	0.971	0.971

在特价营销的价格一致性检查场景下，我们希望从商品卡片中识别出与价格相关的所有信息，并分析出原价、优惠和现价。AutoConsis 首先利用 OCR 从前述商品卡片中获取所有文本信息，再利用包含商品价格推理例子的 Prompt 从中解析出上述三个价态信息。对于其他营销模式如满减，可以对应修改 Prompt 中的示例和回答指令。



Question Prompt

```
Here is an examples for your reference:
Here is a list of phrases from commercial mobile apps: ['和平路', '花园蛋糕',
'香草蛋糕', '已售66%', '甜点', '随时退', '¥20', '4折', '速抢', '¥50', '会员价']

Q: Given the text list from commercial mobile apps, analyze and extract the
 following data:

 - original price:
 - discount:
 - price after discount:

A: price-related phrases are ['¥20', '4折', '¥50'].
 '¥50' is likely the original price as it is the highest of these mentioned.
 According to '4折', which indicates
 the after discount price is '¥50'*'0.4'= '¥20',
 and the discount is '¥50'-'¥20'='¥30'.
 Therefore, the three target data are:

 - original price: "50",
 - discount: "30",
 - price after discount: "20"

My question is as follows, please ensure that your answer is in the same
 format as the examples I provided:

 Here is a list of phrases from commercial mobile apps: ['享哆味汉堡炸鸡(',
'宣威市', '(宣威宝山店)', '5店通用', '2', '(宠粉)', '享哆味汉堡炸鸡', '| P33',
'享宠你', '~3人套餐', '周一至周日可用', '随时退', '¥29.88', '3.8折', '¥79',
'速抢', '会员专属品', '年轻人更爱的汉堡+炸鸡'].

 Given the text list from commercial mobile apps, analyze and extract the
  following data:

 - original price:
 - discount:
 - price after discount:
```

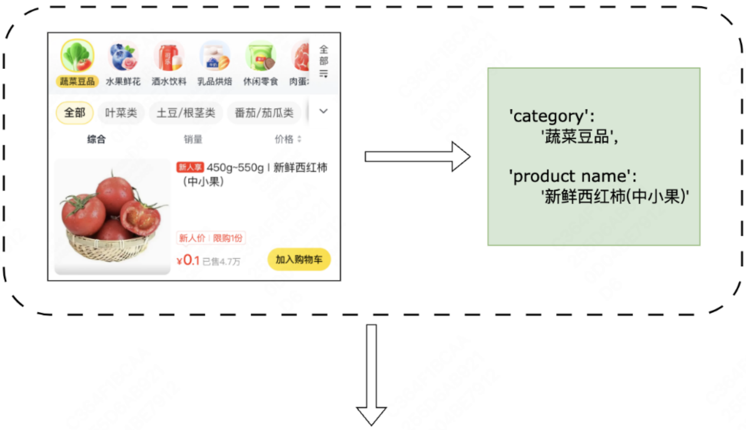
Answer

```
A: price-related phrases are ['¥29.88', '3.8折', '¥79'].
 '¥79' is likely the original price as it is the highest
 of these mentioned. According to '3.8折' which indicates
 the after discount price is '¥79'*'0.38' = '¥29.88',
 the discount is '¥79'-'¥29.88'='¥49.12'.

 Therefore, the three target data are:
 - original price: "79",
 - discount: "49.12",
 - price after discount: "29.88"
```

图：从营销会场页商品卡片中提取价态信息示例

2.4 一致性校验

一致性问题的校验设计简单直接，AutoConsis 会依据预先定义的校验规则判断前述提取出的 UI 内容的一致性。具体来说，从校验的规则有两种类型：数值逻辑类型和语义类型。对于数值逻辑类型的一致性问题（如检查商品价格在不同页面的一致性）AutoConsis 使用规则直接检查其一致性；对于相对复杂的语义规则（如校验商品与所属类别是否一致），则设计 CoT Prompt、借助 LLM 对自然语言的理解能力实现最终校验（见下图所示）。



Question Prompt

```
Please help me understand the natural language on the modern
commercial apps' pages.

Here is an example for your reference:
There are some textual phrases from the GUI of a commercial
mobile apps: ['category':'蔬菜豆品','product name':'现采大红番茄'].

Q: Please help me verify whether they are consistent, and
explain why. Fill in the judgement and explanation blanks:
 - judgement:
 - explanation:

 A: -judgement: Consistent.
 -explanation: According to the product name '现采大红番茄',
   the product can be '番茄'. The category is '蔬菜豆品',
   and '番茄' belongs to '蔬菜'. So the textual phrases from
   the GUI are consistent.

My question is as follows, please fill in the judgement and
explanation blanks:

 There are some textual phrases from the GUI of a commercial
 mobile apps: ['category':'蔬菜豆品',
               'product name':'新鲜西红柿(中小果)'].

 Please help me verify whether they are consistent, and explain
 why. Fill in the judgement and explanation blanks:
 - judgement:
 - explanation:
```

Answer

```
Judgement: Consistent
Explanation: The product name '新鲜西红柿' suggests that
the product is a type of tomato ('番茄'). The category
is '蔬菜豆品', which implies a category encompassing
vegetables and beans. Tomatoes are indeed vegetables,
the specific term used in the product name ('番茄')
aligns closely with the subcategory within the broader
category of '蔬菜豆品'.
```

图：利用 LLM 实现的品类一致性校验示例

对于营销场景的价态检查，AutoConsis 会对于每个商品检查会场页、详情页和订单页的现价是否一致（如下图所示）。

图：对营销场景价态的一致性校验示例

2.5 MLLM 相关方案讨论

在探索过程中，随着多模态大语言模型（MLLM）的发展，我们也尝试了用 GPT-4V 解决 UI 一致性检查的问题的可行性和效果。我们选取营销会场的商品列表页作为测试场景，对比两种方法在提取商品价格上的效果和成本。验证时，我们将营销会场的完整 UI 界面截图作为输入，让 GPT-4V 返回每种商品的原价、优惠 / 折扣以及现价，如果三个信息都与截图一致，则认为识别正确。

方法	准确性	单页面推理速度	单页面成本	1000页面综合成本
AutoConsis	0.94	2s-3s	0.018元 – text token	1小时 2元
GPT-4V	约0.85	45s	2.5元 – image token	15小时 2500元

AutoConsis 作为一整套一致性检测流程，相比于直接应用 MLLM 作为整体 UI 图像输入来判断，适配的复杂度会更高一些。但从结果看，AutoConsis 在执行效率、检测结果的可靠性、执行成本三个方面均更具优势。当前业务一次巡检需在几千量级的

页面规模上使用，所以我们选择 AutoConsis 来批量进行业务应用落地。

3. 应用效果

目前，该能力在美团特价团购营销会场场景上覆盖了 700 城市、4000 多页面，在项目开城与后续迭代过程中，持续以巡检形式执行，期间共发现几十个有效的业务问题。

图：业务缺陷示例 – 左侧为详情页，右侧为提单页，总价不一致

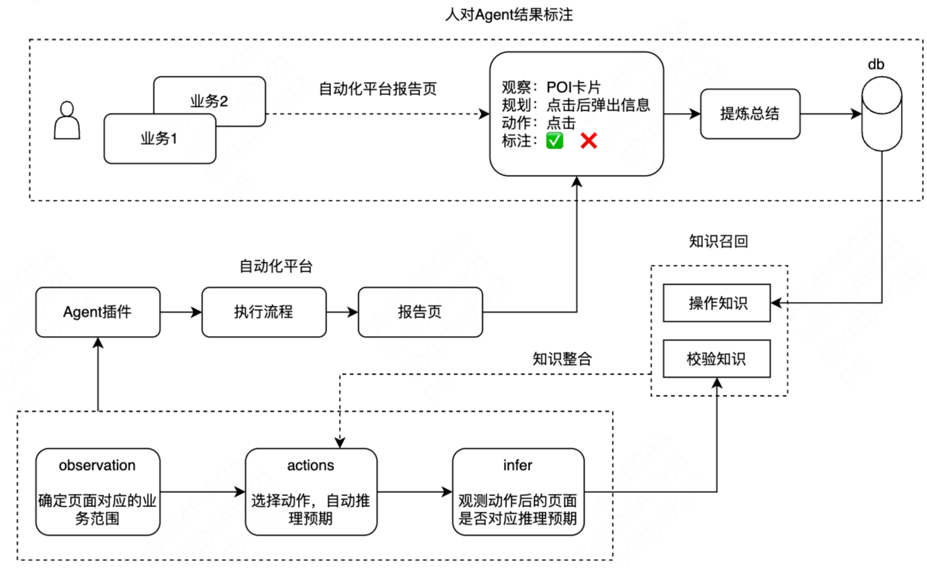
同时我们通过在营销会场上的落地应用形成了一套标准化工程流程，在其他有类似校验需求的业务场景快速完成了落地。下面是我们基于这套标准流程已经实现并应用的另一些项目：

门票运营活动巡检：分析页面所有可能有运营的位置，点击后判断内容和活动是否一致。目前已在 12 个城市的 200 个页面上进行使用，共计发现了 8 个有效问题。

优选品类巡检：分析优选业务中的各个商品与所属类别是否一致。

4. 认知与展望

将一个问题拆解为一连串的步骤，在每个步骤更加针对性地应用智能化能力，在现阶段往往能取得更好的效果。AutoConsis 所具备的 UI 分析和数据检查能力并不局限于内容一致性检测，目标领域识别 -> 目标信息抽取 -> 检测判断，这实际上是一个大前端通用的校验流程，相比传统的自动化方案有低成本、泛化性强等优势，相信未来可以在更多前端测试场景中进行推广。



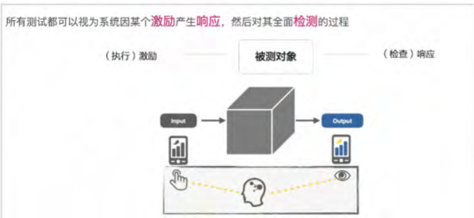
图：Agent 学习人类偏好经验流程

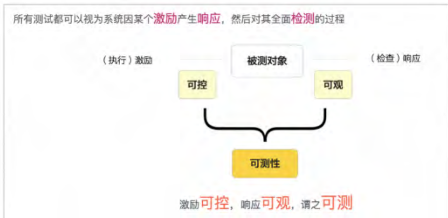
目前，我们正在尝试将 AutoConsis 的校验流程和 Agent 能力相融合，让 Agent 能根据人类反馈指令自我迭代，以更接近人的水平去操作和校验 UI，智能化的帮助节省人工时间，相关能力已经在美团的直播视频化等业务领域进行落地。后续，我们将结合更多业务需求对 AutoConsis 进行持续改进与维护，也欢迎业界同行们的反馈与交流。

小程序可测性能力建设与实践

1. 引言

测试活动从本质上可以视为被测系统因为某个激励产生相应的响应，并对这些响应进行全面检测的过程。这个过程（激励 -> 响应 -> 检查）涉及到两个角色：测试者以及测试对象，测试者执行激励与检查响应，由机器（程序）或者人来完成；被测对象接受激励，产生响应。从这个过程来看：**激励可控，响应可观，称之为可测**。以实际业务测试为例，修改缓存、网络请求 MCOK、页面跳转、用户登录态设置等都属于可测性能力。





在未经过任何可测性改进的终端产品中，测试人员只能通过 UI 交互，从 UI 界面观察来完成最基本的质量保障。然而应用内部存在各种各样复杂的逻辑、状态，要进行更加深入的测试则需要对这些信息进行介入与观测。例如，在进行打点测试时，操作页面后，需确认打点信息是否被正常上报，这一过程通常依赖网络代理调试工具来完成校验。同样，在用户登录测试环节中，登录完成后，需要检查缓存是否已正确记录登录信息，这要求具备缓存查看的能力，这些体现了实际业务测试场景对可测性能力的需求。

整体而言，完备地构造出目标场景进行测试涉及到多个复杂的方面，同时观测它是否符合预期也比较困难，如下图所示。终端测试长期面临着挑战。为应对这些挑战，我们以增强可测性为基础，将其贯穿测试活动的始终，使得测试能更细粒度地检查系统，提高测试深度和效率。

作为终端产品的一种形态，小程序是运行在宿主应用（如微信、快手、百度等）之上的“轻应用”，在 2017 年由微信推出后发展迅速。由于小程序非常依赖于宿主应用环境，因此在测试过程中，除了面临终端测试固有的难点外，它还存在一些特殊的影响因素。

从运行机制的角度来看，小程序的代码逻辑运行在宿主应用提供的容器环境内，它无法直接控制宿主应用本身和手机系统，这在一定程度上增大了测试与可测性改进的难度。

在目前的实践中，针对小程序的测试主要存在以下几种工具和策略：

1. 采用如 Charles、Fiddler 等网络代理工具进行 HTTP/HTTPS 请求和响应的代理分析与校验。虽然这类工具适合进行数据包的抓取和分析，但它们通常无法深入小程序的内部架构，因此无法全方位控制或感知应用的内部状态。
2. 运用图像处理技术的自动化测试工具如 Airtest 进行测试，它们主要关注于界面层面的操作，未能触及应用程序背后的逻辑处理，因此仍属于“黑盒测试”的范畴。
3. 利用微信官方提供的 Minium 小程序测试工具来执行更为精细的测试操作，能够进行诸如 API Mocking 等内部控制。然而，该方法操作复杂，并依赖于微信开发者工具，而后者与真机环境之间存在一定差异，可能影响测试结果的准确性。
4. 开发专用的自研调试面板用以验证程序逻辑和测试特定场景，但这些工具设计时常常专注于特定小程序，不易迁移至其他应用，而且它们通常不支持自动化测试流程。

综上所述，尽管存在多种测试工具和方法，但目前尚缺乏一套综合性的、易于使用的测试工具集，能够全面提升小程序的可测性。

2. 小程序可测性介绍

终端可测性能力全景图

小程序可测性的目标在于构建一套全方位的通用小程序可测性能力集合。该体系无缝支持真机和模拟器环境，兼容多端、多平台，并允许不同应用以低成本轻松接入。它能深入核心，为小程序提供全面而多元的可观测性与可控性，覆盖应用界面、内部状态、存储等关键领域。这一体系旨在赋能测试者更便捷地应对复杂测试场景，显著提

高测试的效率与深入度。

经过了长期的建设积累，目前我们已经构建了一套比较全面的终端可测性能力集，包含 Android、iOS、小程序、Web 等技术栈。其中小程序由于系统的结构特殊性，可测性能力相对其它端会有一些不同。小程序可测性主要包括**业务逻辑可测性、应用可测性、系统 & 设备可测性**三个层级，在每个层级中包含多个垂直的细分方向，除了支持多技术栈的公共可测性能力，还提供了如 AppData、宿主应用信息可观可控等特有能力。下面以几个典型能力说明小程序可测性使用方式与效果。

2.1 使用方式与效果

在实际的手工以及自动化测试工作中，小程序可测性能力能够很方便的使用，并在多个场景下发挥了重要的作用。

2.1.1 手工测试

下面将以缓存管理、页面跳转功能为例介绍小程序在手工测试中的使用方式以及效果。

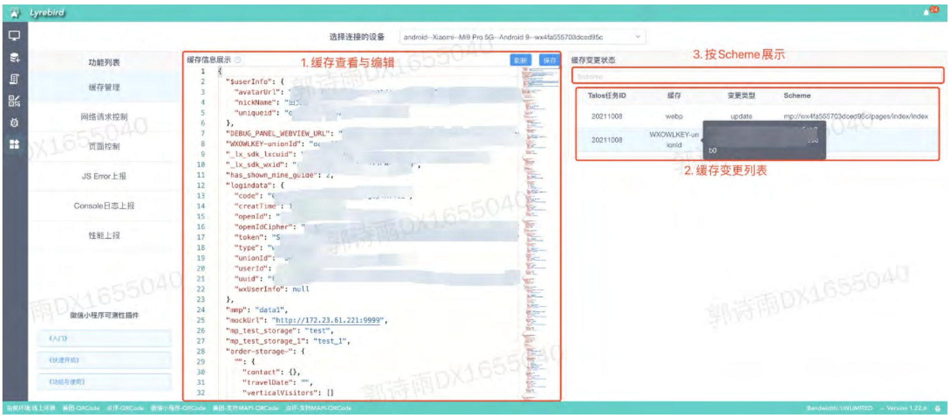
在实际的测试工作中，会结合 Lyrebird 使用小程序可测性，Lyrebird 是美团到店研发平台自研的终端测试工作台，包含终端状态数据管理、网络请求代理与 Mock、缺陷记录、自定义插件扩展等能力。同时它还提供了图形化操作界面，是手工与自动化测试中使用可测性能力的入口。

在小程序接入可测性能力 SDK 之后，可以通过可测性 SDK 提供的扫码功能与 Lyrebird 建立连接，后续就可以通过 Lyrebird 在 PC 端利用可测性对小程序进行控制以及观测。

缓存管理

我们可以通过缓存管理功能验证依赖缓存的业务逻辑正确性，如表单信息 \ 用户信息暂存到缓存功能等。

- 如下图所示，1 处为缓存编辑框，展示当前选择设备上的小程序所有的缓存信息，并对这些缓存进行管理，支持批量的增删改。
- 2 处展示目标小程序的缓存变更事件信息，包括在该页面对缓存的编辑以及小程序自身内部对缓存的增删改操作事件，会随着事件的触发实时更新。



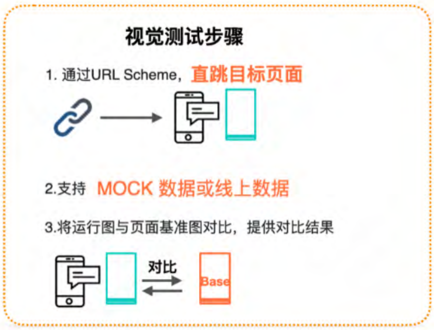
页面跳转

页面跳转是小程序业务测试中重度使用的能力，可以利用该功能跳转到如表单页，商品详情页等中间页面，不再需要从首页一步一步操作进入目标被测页面，减少测试前置准备工作，具体可以在该 Lyrebird 页面中输入页面路径进行跳转。

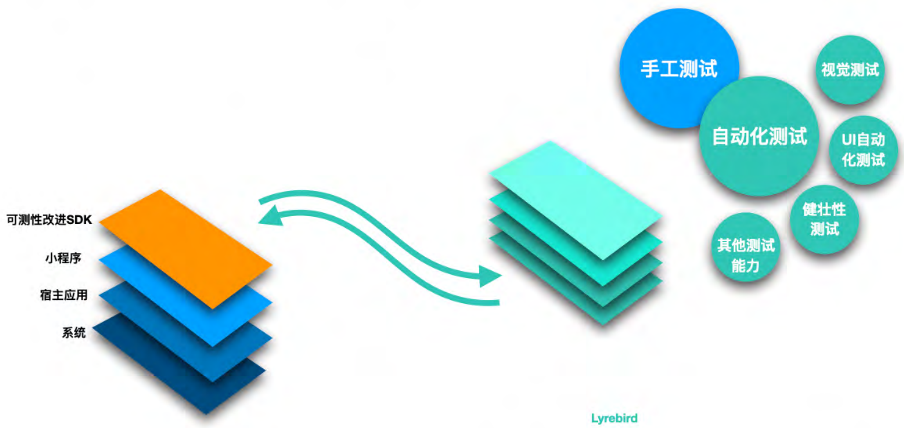
2.1.2 自动化测试

将可测性能力结合 Lyrebird 应用于自动化测试。如通过页面跳转能力直达测试场景，然后利用通过可测性录制的页面状态数据进行场景状态还原后进行页面渲染，获取页面上的数据 / 布局展示，最后将实际运行图和预先设置好的页面基准图进行对比，提供渲染的差异结果，进行视觉 DIFF 测试。

这类“视觉测试”以页面为单位，通过深度链接跳转技术配合一系列终端应用本身的可测性改进，直达测试场景，并通过图像处理技术如长图融合、图像增量对比和文本识别能力进行视觉 DIFF 测试。



可测性建设的是对应用内部状态的可观可控能力，对于任何测试方法，只要涉及应用内部，可测性都能发挥重要作用。比如在健壮性测试中通过可测性构造破坏性异常场景，或者在功能测试中模拟小程序不同的进入方式（如二维码、视频号、搜索等）来测试所有可能的使用场景下小程序的运行情况。



2.2 接入方式

小程序可测性能力 SDK 被封装为一个 NPM 包，在小程序源代码或者编译产物项目中引入此 NPM 包，便可实现可测性能力的接入，无需进行额外适配工作。

跨平台运行

除了对微信小程序的支持之外，小程序可测性能力 SDK 通过集成一个适配器（Adapter）将能力扩展到多个宿主应用，包括美团、支付宝、快手、百度等平台的支持。这些平台的基础库 API 与微信类似，适配器会根据不同平台的特点，对代码进行相应的调整，包括基础库 API、前端语法或文件类型等，以保证在各个平台上的兼容性和一致性，实现跨平台运行。

2.3 实现原理

小程序可测性实现的核心思路是通过 JavaScript Hook 的方式，在小程序 JavaScript Runtime 中对如微信小程序 JS 基础库、业务公共基础组件等目标模块进行透明化介入，实现对其内部的可观可控。在此之后，通过可测性 SDK 内的中控与外部建立网络链接，从而实现在远端对小程序内部状态与功能的可观可控。

JavaScript Hook 介绍

JavaScript Hook 基于 JavaScript 的动态特性，有以下方法：

函数 Hook：直接覆盖或修改原函数：

```
let _originAlert = alert;  // 保存原函数
alert = function () {
  console.log('alert 执行开始');
  _originAlert.apply(this, arguments); // 执行原函数
  console.log('alert 执行结束');
}
```

对象属性 Hook：通过 Object defineProperty 定义新的或直接修改某个对象的属性，如修改 Getter/Setter 方法，控制对某个对象的获取 / 设置流程。

```
Object.defineProperty(document, 'cookie', {
  set: function(val) {  // 控制 cookie 的设置流程
    console.log('获得 cookie: ', val);
    currentCookie = val;
    return val;
  },
  get: function() {  // 控制 cookie 的获取流程
```

```
    return null;
  }
});
```

原型链 Hook：修改原型链上的数据，如 String、Date。

```
let _originalGetTime = Date.prototype.getTime;  // 保存原型链原方法
Date.prototype.getTime = function() {
  console.log('getTime has been called');
  return originalGetTime.apply(this, arguments); // 执行原方法
};
```

Proxy 对象：创建代理模式替代原始对象，可以重新定义获取、设置和定义属性等基本对象操作。

```
// 创建 Proxy 有两个参数：
// target：要代理的原始对象
// handler：定义哪些操作将被拦截以及如何重新定义被拦截操作的对象
let handler = {
  get: function(target, prop) {
    console.log(`获取 ${prop}`);
    return target[prop];
  },
  set: function(target, prop, val) {
    console.log(`设置 ${prop} 值为 ${val}`);
    target[prop] = val;
    return true;
  }
};

let proxy = new Proxy(window, handler);

proxy.test = 'test';      // 输出 : Setting test to test
console.log(proxy.test); // 输出 : Getting test
                         //       test
```

静态 Hook：小程序构建时在特定文件中直接修改其 JavaScript 源代码。

其他方式这里就不详细展开了。

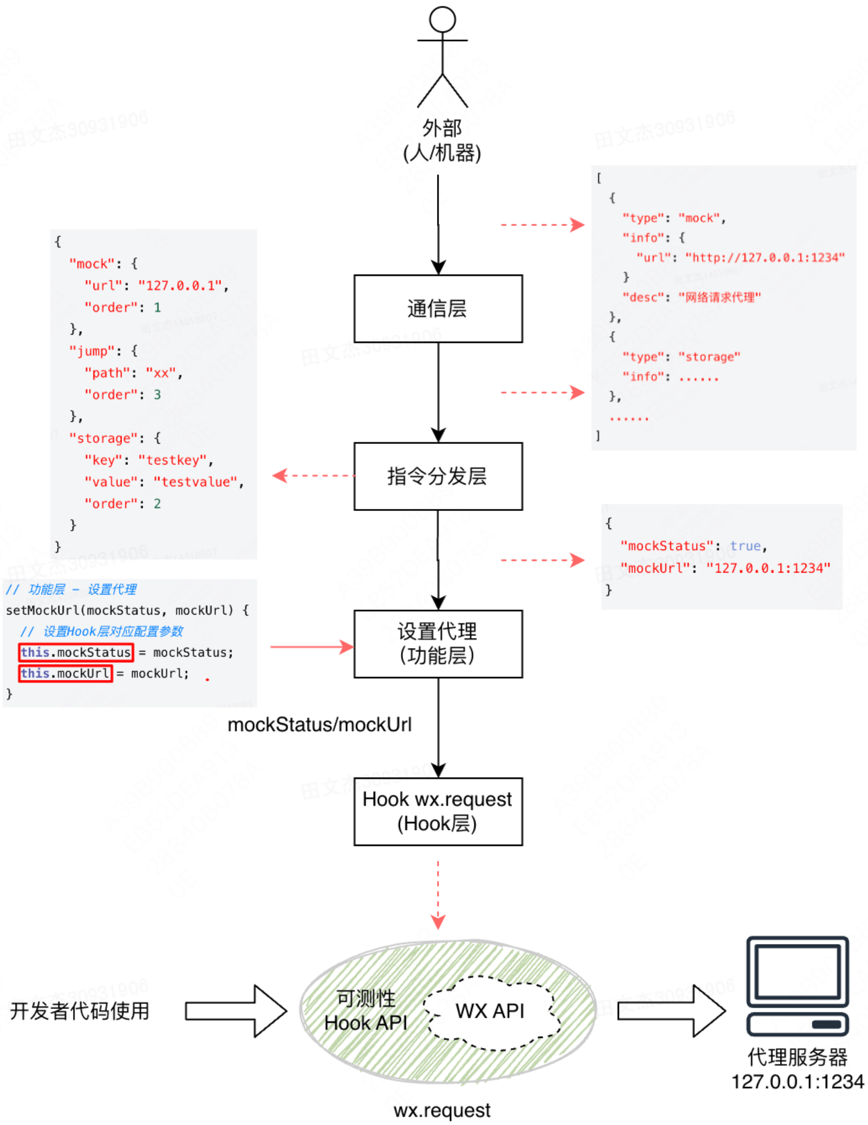
可测性 SDK 的大体可分为四层：

- **通信层**：与外部进行通信，负责指令和数据与远端（如 Lyrebird）的双向流动。
- **指令分发层**：对通信层接收到的参数指令进行解析，依次调用控制小程序相关状态的功能层模块。
- **功能层**：实现小程序特定功能可观可控的业务逻辑，包括 UI、网络请求、存储、应用状态等模块，实现如请求代理与修改、切换登录态或者控制缓存可测性功能。
- **Hook 层**：实现对实际逻辑模块状态和方法的透明化介入。由于小程序应用内部的状态 / 数据与开发者代码相关联，Hook 层通过 JavaScript Hook 对宿主应用基础库、公共组件、业务特定逻辑三种类型的功能模块进行拦截介入，使得其状态 / 数据可观和可控，为功能层提供实现基础。Hook 层一般需要先于业务代码加载，保证拦截的有效性。
    - 宿主应用基础库。通用性改造，对小程序容器提供的系统级接口进行介入，如网络请求、地理信息等。
    - 公共组件。组件级通用，如美团的公共登录组件，对其进行改造后，接入登录组件的小程序都能够使用相应的可测性能力，比如切换登录态 / 模拟登出等能力。
    - 业务特定逻辑。某个小程序特有的业务逻辑，通过可测性 SDK 提供的 API 对这些逻辑进行改造后以插件形式集成定制化能力。

下面将以网络请求可观可控为例介绍小程序可测性的实现原理。

网络请求代理

当外部希望控制小程序设置网络代理时，整体流程如下：



1. 外部（人 / 机器）首先通过 HTTP/WebSocket 方式传递包含设置小程序请求代理的指令，如图即拦截小程序发送的请求转发到 127.0.0.1:1234 代理服务器；
2. 可测性 SDK 在通信层接收相应的指令后。将其传递给指令分发层。在指令分发层中，收到指令后进行解析，并按预定规则对指令执行进行编排，确定执行顺序；

3. 指令分发层按编排顺序调用功能层设置网络代理并传入开启状态和代理服务器地址参数，功能层通过修改这两个变量，控制 Hook 层对请求 API 的拦截，从而改变请求代理的状态；
4. Hook 层拦截微信基础库里 wx 对象的 request 方法，如下图代码所示，分为以下流程：

   a. 保存 wx.request 原始方法的引用（3 行），并通过 Object.defineProperty 将 wx 对象设置为可写状态（4-8 行）；

   b. 将 wx.request 修改为 Hook 的新方法。新方法的入参与原始 wx.request 一致，包括请求头、请求地址、响应体等，因此可以对这些参数进行修改（12 行），比如替换请求域名、增加请求头、修改响应体数据等；

   c. 最后用修改后的参数使用原始方法进行执行（13 行）。

Hook 层通过 mockStatus 和 mockUrl 两个变量控制到小程序是否被代理以及代理服务器地址（19-22 行），当开发者代码中使用 wx.request 发起请求时，会先经过 Hook 指向的新方法。如果被设置代理，请求地址将会根据代理服务器协议进行修改，从而使得请求被代理。

3. 美团门票业务小程序测试实践

在到店众多应用了小程序可测性能力的业务中，美团门票业务从 2021 年开始即参与了小程序可测性建设，目前在门票质量保障工作中，可测性相关能力均深度应用在新需求测试、回归测试、线上巡检等各种类型的测试活动中。

3.1 可测性落地

下面通过门票业务一个具体的新需求测试例子来介绍可测性如何在测试活动中进行落地。

需求背景

用户从商品详情页进入到填单页，在选择日期、数量或填写游玩人等信息后，为了减少用户的操作路径，再次进入该填单页需要保持之前填写的这些信息不变。

操作路径划分

该过程需要经过以下步骤：进入填单页 —> 打开价格日历弹层，选择相应的日期 —> 添加数量 —> 填写或者选择游玩人 —> 点击返回退出填单页 —> 再次进入填单页，查看它当前的状态。我们选择对缓存进行可测性改进，依靠指令数据驱动 + 内部方法调用来达到同等 UI 操作的效果，保障此类场景测试的稳定性并提高执行效率。

技术实现

整体通过缓存实现。在进入填单页时，首先会读取小程序上的缓存并渲染；在选择日期、数量和游玩人时，分别对相关信息进行暂存；在退出填单页时，将这些暂存的数据写入缓存。

测试分析

由于进入填单页需要读取缓存进行渲染，因此测试过程中首先应从 UI 上进行验证，判断第二次进入的日期、数量和游玩人是否与上一次进入时选择的状态一致；其次还应从数据上进行验证，即进入填单页有“读”缓存的动作；在退出填单页时，需要将暂存的数据写入缓存，因此测试过程中应验证数据能正确地写入缓存，而且缓存里有正确的值。

可测性能力实践落地

- **通过可观校验“写”的正确性**。对于“写”，验证缓存的写入动作，并且写入缓存的数据是正确的。缓存的可观性改造能够将“写”的动作、“写”的当前值以及当前缓存具体信息，进行上报，这样就可以自动化校验当前操作后是否缓存值是否发生了正确的变化，以此完成对缓存“写”的校验。

- 通过可控校验“读”的正确性。对于“读”，首先验证 UI 能够正确展示，其次从数据上验证有缓存的“读”动作。由于测试缓存必须经历选择日期、选择数量、选择游玩人，返回退出填单页等多个步骤。测试路径较为繁琐，因此，对缓存的可控性改造后，传入相应的配置指令（如 2.2 部分介绍），控制缓存中的数据，直达被测页面和状态，并通过自动化测试比对当前运行的页面和页面基

准图，判断它是否正确被渲染，以此分别从数据和 UI 上完成对缓存“读”的校验。

门票业务在小程序测试上目前已经落地多种可测性能力，如下图所示，包括控制页面跳转、请求代理、控制登录、日志上报、隐私治理、前后端环境、录制回放、自动化交互控制等都在门票测试活动中有相应的落地，发挥着非常重要的作用。

3.2 业务实践总结

门票业务借助可测性改进使得测试的覆盖更加全面，目前 30%+ 的测试场景依赖于可测性能力进行构建。在美团小程序和点评小程序的门票频道以及门票独立小程序上均有上百个自动化测试用例，页面覆盖率已经达到 100%，场景覆盖程度达到 80%+。这些测试用例在门票新需求测试、回归测试等各个阶段都会触发自动执行，累计已辅助发现上百个有效问题。

4. 总结与展望

美团核心本地商业 / 到店研发平台从 2021 年开始系统化建设小程序可测性，到目前融入到店终端测试工具链以及质量保证体系之中，通过具备扩展性的通用能力框架，融合手工和自动化测试，贯穿测试活动始终。未来我们还将持续关注于基础可测性能力的稳定性，聚焦具备更多业务特性的可测性能力建设。

Q&A

Q：代理逻辑如果有 Bug 会不会影响比较大

A：代理逻辑本身很简单，出错概率不大。进行 Hook 时，会有异常监控能力以及相应的兜底策略，即使出问题，也尽量降低对业务实际使用的影响。

Q：可测性 SDK 需要对业务代码进行改造吗?

A：不需要，可测性 SDK 对于业务应用是透明的。

Q：Lyrebird 项目和小程序可测性 SDK 的关系是什么?

A：Lyrebird 与小程序可测性是两个独立的项目。小程序可测性 SDK 是以一个 NPM 包的形式实现的，在小程序里安装 NPM 包，即可使小程序具有可测性。Lyrebird 可以与小程序可测性 SDK 的通信接口进行连接，然后用户可通过 Lyrebird 中小程序可测性页面使用小程序可测性能力。

Q：针对小程序可测试性能力建设与实践，我想问下，如果我们要用你们的测试工具，

需要做什么适配吗?

A：不需要进行额外适配，最终的呈现会是 NPM 包形式，在产物里安装就可以接入我们的可测性能力，可以对它进行控制。

Q：生产环境会接入可观测 SDK 吗？如果接入对性能有多大影响?

A：首先是对它的性能的影响，我们实际上是对小程序里的基础库的 API 或者一些状态数据进行了拦截，会对性能产生一定的影响，但目前这个影响范围对业务来说比较小，是可接受的。生产环境的不会引入可测性 SDK，因此不会对线上质量造成影响。

Q：小程序可测性有不适合使用的场景?

小程序可测性主要针对小程序前端手工与自动化场景进行能力提升，它是具备一套通用可扩展框架，可以按照业务需求低成本进行可测性能力扩展，然而，存在特定情况下其适用性受限：首先，由于运行环境的约束，针对宿主应用如微信或支付宝自身的可测性需求，小程序的可测性无法支持。此外，小程序可测性专注于终端测试，因此对于那些需求后端服务链路验证的场景，并不适用，需配合针对性工具使用。

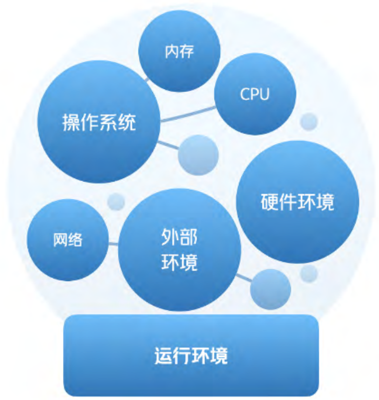
基于接口数据变异的 App 健壮性测试实践

01 什么是客户端健壮性

在维基百科的定义中，健壮性（Robustness）是指一个计算机系统在执行过程中处理错误，以及算法在遭遇输入、运算等异常时继续正常运行的能力。IEEE 中将健壮性定义为系统或组件在存在无效输入或压力环境条件下可以正常运行的程度。早在 1989 年，Barton Miller 首次提出了模糊测试的概念，通过向目标应用抛出随机字符串的方式来测试 UNIX 应用程序的健壮性；而在 1996 年的 Ballista 项目中，研究人员探索根据 API 定义的数据类型，对操作系统或软件接口进行自动化测试方法。两个项目均以“无应用程序崩溃或挂起”作为测试验证通过的标准。

在移动端 App 领域，健壮性可以理解为 App 运行时遭遇环境异常或者输入异常时客户端能够继续正常运行的能力。

其中，环境异常主要分为操作系统异常、外部环境异常、硬件环境异常三大类。比如内存不足、CPU 负载过高、线程池满载、内存分配失败、网络连接失败等。输入异常主要分为系统输入和用户输入。比如网络接口返回的数据异常、应用内缓存、数据库文件读写异常，这类的异常属于在系统输入异常；在电话号码输入框场景，用户输入的空格、富文本则属于用户输入异常。



对于这些风险，如果 App 没有处理，理论上都可能会产生展示异常、交互异常、性能、安全等问题，导致用户无法继续使用或在使用过程中产生不好的体验。比如用户操作 App 下单过程中，API 请求出现故障未返回状态码为 200 的响应，App 由于没有获取到预期接口响应的信息而发生崩溃，就会中断用户的使用流程。

异常类型	表现
展示异常	加载过程中出现Crash/ANR/JS Error
	功能/模块缺失
	界面错位
	字体/颜色/文本内容异常
交互异常	交互过程中出现Crash/ANR/JS Error
	交互无响应
性能问题	CPU、内存、线程池等资源消耗异常
	加载时间等时间维度性能指标劣化
安全问题	出现安全漏洞产生被攻击或信息泄露等风险

02 基于接口数据变异的 App 健壮性测试方案设计

在实际的客户端测试执行过程中，测试人员会考虑测试异常输入的场景，但由于成本无法做到无穷尽的测试，同时还存在人工执行遗漏的风险。

从美团 App 平台业务的历史故障分析中，我们发现：网络请求返回的数据与实现预期不符引发的 Crash 或核心功能缺失问题导致的故障占比最高，且影响面较广。比如接口返回非预期数据时，客户端处理数据类型转换异常导致闪退，即使 5 分钟内操作降级仍影响了百万量级的用户。因此美团平台业务 App 的健壮性测试探索优先从发现网络请求返回数据导致的异常开始。

针对于发现请求接口返回客户端非预期数据导致的 Crash，或者核心模块缺失问题这个诉求，我们调研后发现方案的基本原理都是相似的，即以网络请求的原始响应为基础，根据规则进行变异构造，使用代理工具改写响应体返回给客户端，在端上设备做异常检测。但是都存在一些问题不能满足诉求，比如测试变异数据是根据预置或者自定义规则随机生成组合，随机性过大，不能有效拦截健壮性问题；但如果不做随机，产生的用例组合量过大，测试不能在合理时间范围内结束；另外在检测能力方面，不具备发现业务异常或功能模块异常的能力。

因此，我们结合通用方案做了一些自定义改造，整体检测方案包含静态检测和动态检测两部分。

- 静态检测，主要是指静态代码扫描，将典型代码编写规范问题转化为自定义静态代码扫描规则，管控增量代码，同时长期治理存量风险。比如自定义了 PrimitiveParseDetector、ColorParseDetector，管控业务必须使用健壮性测试通过的工具类。
- 动态检测是指结合触发时机，构造并注入变异数据后，识别 App 运行时是否出现崩溃、挂起或业务功能模块异常。比如在集成事件 / 回归事件触发自动化测试运行，构造触发异常的数据进行动态测试，然后监测是否出现了异常。核心动作包含构造变异数据和完成检测两部分。比如将接口响应体中表示颜色含义的 Key 对应的 Value 值构造成非色值，然后检测客户端请求处理接口数据时是否出现崩溃或挂起。

下文重点介绍端到端的动态检测方案。

03 变异数据的构造和异常检测

对于美团 App 来说，首页有多种形态，对于某种特定形态，除了控制请求数据外还需要控制实验、策略等一系列因素，才能保证测试对象的唯一性。一个页面中包含多个异步请求，因此请求的构造也需要和页面路径关联。这些都是采集变异所需的基础数据时需要关注和控制的。

响应体由基本类型数据和复合类型数据组成，相同基本类型的数据可能具备不同的业务语义，需要根据语义的类型做变异规则的区分对待，才能保障业务场景覆盖。

因此，如何保障变异数据构造的全面性和准确性，是我们面临的首要挑战。

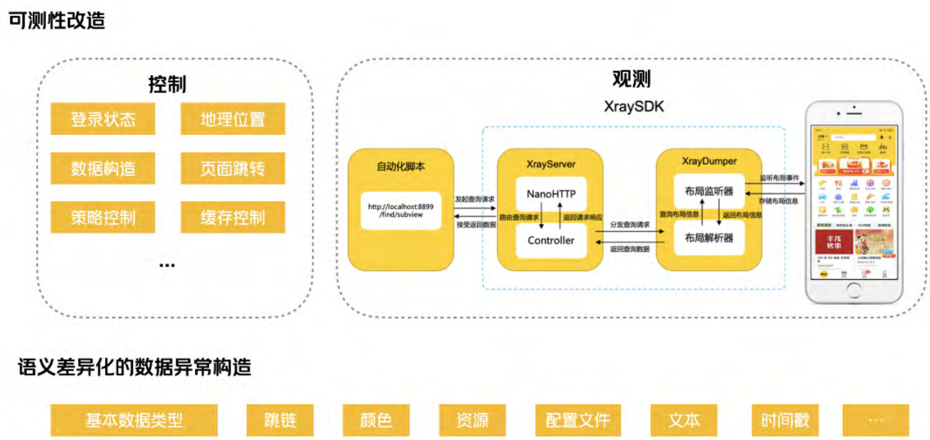
要解决数据构造全面性问题，首先要解决页面描述方案，这样才能控制获取基础数据的唯一性。在解决方案中，我们构建了页面描述的特征规则，解决用户视角的页面标识问题。需要的信息包含端信息、页面路由信息、实验策略账号信息、页面标识模块合集等。通过页面请求数据自动录制的方式，自动更新迭代请求数据和页面之间的绑定关系，使得基础数据能够随需求迭代更新，从而通过变异规则构造生成的用例也能够自动更新。

解决方案：

在用例变异生成构造上，对于响应体里的 Value 设置了语义匹配规则，比如字符串的语义可能代表颜色、页面跳转路由、动静态资源链接（即图片资源数据 / 视频文件 / GIF 文件），需要区分特征分别按语义构造异常数据。比如在图片的变异数据构造里，除了需要构造非图片链接情况外，还要考虑不同图片格式、非图片格式以及非合法的图片剪裁格式拼接等场景。

我们对接口返回数据使用脚本做了初步的语义分析，人工二次校正后建立了基本数据类型和语义的映射集合，结合基本数据类型边界值和语义定义了初始的变异规则。然后对历史的线上健壮性问题和线下测试发现的健壮性 Bug 的变异数据进行整理，作为增补的变异规则。

在自动化测试执行过程中，我们基于 App 可测性改造提供的能力，对测试场景进行了控制，同时基于布局视图的解析 SDK、App 异常上报 SDK 提供的能力，完成了对 App 异常的通用检测。



04 变异数据的精简方案

伴随着变异规则的丰富，自动生成的数据量级是巨大的，数据的变异组合如果按照全覆盖方式来生成组合数量就是指数级增长。比如对于 1 种有 7 种变异取值的变量，如果存在 n 个此类型变量，就会产生 7^n 种数据组合，并且在实际业务场景中很多组合情况是没有意义的。

如何在保障用例构造全面性的情况下精简变异构造的用例数，是我们面临的第二个挑战。解决方案包含 2 个策略：1）数组元素结构一致时，删减构造的用例数；2）结构不完全一致的数组元素，引入编辑距离和并查集算法判断节点相似性，节点不相似，可以在一次数据生成里做合并构造。

我们可以把请求响应的 JSON 理解成树，第一个解决思路是判断树中节点、路径的相似度，相似节点删减构造。

如果路径、节点相似，可以推测路径即业务逻辑也是一致的，比如页面上的一些列表元素，可能是数据结构对象完全一致数组，如果对每个数组对象中的每个元素进行全用例构造，生成的变异数据量极大，且对业务场景或代码逻辑的增量覆盖有限，因此我们决定将构造逻辑优化，进行删减构造。即假如数组中元素的结构完全一致，那么同含义的字段可以为他们分配不同的变异构造值，然后删减掉无效的构造情况。应用这种方法可以有效降低 28% 左右的用例构造数量。

如图数组的 3 个元素中均存在“resourceName”键值对，假如每个键值对有 3 种变异取值，按照全排列方式进行用例构造将会生成有 9 份变异数据，在删减构造情况下，可以分别为它们构造一个特定的变异值，这样变异生成用例数量可以从 9 减少为 1。

面临挑战

挑战二：如何精简用例降低测试运行时长

基本思路：

① 相似节点识别，利用相似节点的相同key取值不同的变异数据，删减构造用例数

在对业务接口返回数据的数据结构进行分析后，我们发现在层级越深的场景下，距离根节点越近的两个节点，业务逻辑耦合和结构相似程度越低，它可以进行合并构造，相互逻辑之间不会产生影响，比如有两个键值对，每个键值对的 Value 有 3 种变异取值，在合并构造情况下，可以从排列组合的 6 份数据减少到 3 份数据。

挑战二：用例精简降低测试耗时

基本思路：

② 结合响应体层级，量化计算不同层级间响应体中节点的相似度，距离根节点越近，相似度越低，需要合并构造

e.g.

data.modules[0].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName
data.modules[1].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName

```
data.modules[0].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为空串 -case 1
data.modules[1].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为空串 -case 1

data.modules[0].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为null -case 2
data.modules[1].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为null -case 2

data.modules[0].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为超长 -case 3
data.modules[1].proxyData.resourcesMap.utilInfoArea[0].materialMap.utilName 构造为超长 -case 3
```

基于这个个思路，我们在实践中引入了编辑距离和并查集算法，以节点路径为参照，对树的每一层的每两个节点计算编辑距离，生成一个 n*n 矩阵；同时以树的高度减去节点位于的层数作为权重，修正编辑距离。基于这样的计算，会产生多个编辑距离矩阵。

为了尝试最大化合并构造用例效果，我们把编辑距离做了 0,1 矩阵转化。其中，由于编辑距离为 1 的两个节点可能存在业务逻辑耦合关系，必须放在同一个组里分别构造，所以我们把编辑距离大于 1 的情况转化成了 0，最后得到了一个 0,1 的编辑距离矩阵。

挑战二：用例精简降低测试耗时

解决方案： 引入编辑距离和并查集算法，对于响应体中每一层构造N*N矩阵计算相似度，并通过距离根节点的距离赋予不同权重。

$$lev_{a,b}(i,j)=\begin{cases}\max\{i,j\}, \text{ if } \min\{i,j\}=0\\ \min\begin{cases}\lambda_{distance}\left(\mathrm{lev_{a,b}}(i-1,j)+1\right)\\ \lambda_{distance}\mathrm{lev_{a,b}}(i,j-1)+1\\ \lambda_{distance}\mathrm{lev_{a,b}}(i-1,j-1)+1_{(a_i\neq b_j)}\end{cases}\end{cases}$$

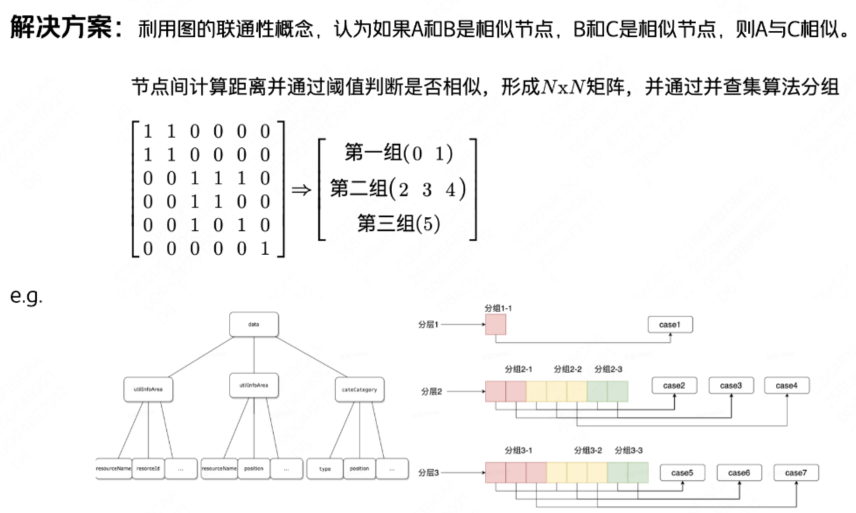
其中，a,b表示待构造节点的路径，Lamda代表距离权重，lev代表由a构造为b的最小距离。

为了合并最大用例，把同层节点的编辑距离矩阵做0,1转化，由于耦合关系的需要放在同一个组里分别构造，转化时把编辑距离大于1的转化为了0；

$$\begin{bmatrix}1&1&0&0&0&0\\1&1&0&0&0&0\\0&0&1&1&1&0\\0&0&1&1&0&0\\0&0&1&0&1&0\\0&0&0&0&0&1\end{bmatrix}$$

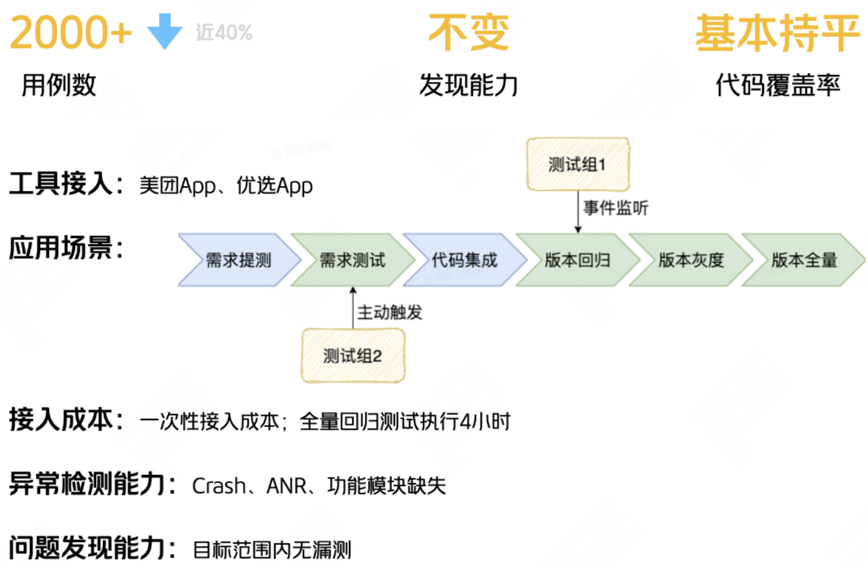
在 0,1 矩阵情况下，我们使用了图的连通性概念，如果 A 和 B 连通，B 和 C 连通，那我们认为 A 和 C 连通，转化到这里的概念就是 A 和 B 相似，B 和 C 相似，那么 A 和 C 相似，它们应该被放在同一个组里分开进行构造，那么在同层元素构造时，我们会从每个分组里取到一个节点，对这些规则进行变异组合构造。

挑战二：用例精简降低测试耗时



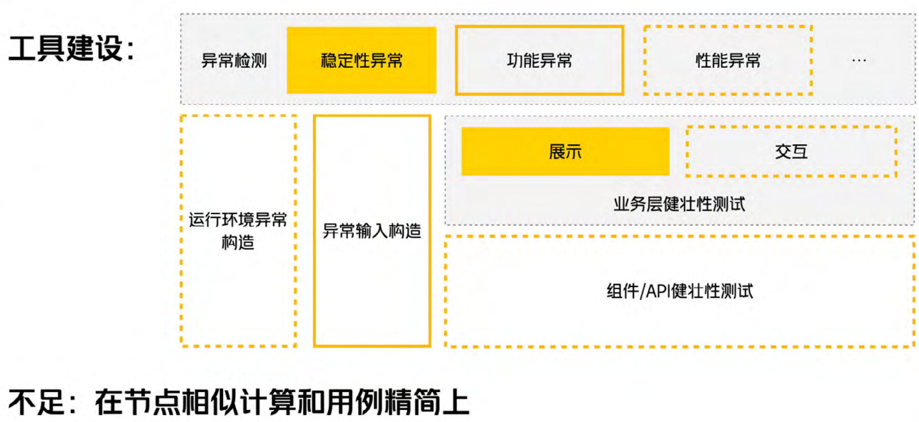
基于以上两个策略进行精简后生成的变异数据量较精简前降低了 40%，同时代码覆盖率没有明显变化，并且保持不变的健壮性问题发现能力。

美团 App 和优选 App 都接入了这个工具，在新需求阶段可以人工触发运行，还可以结合客户端组件集成事件和回归事件做自动触发。至今应用一年时间内，发现了几十个问题。

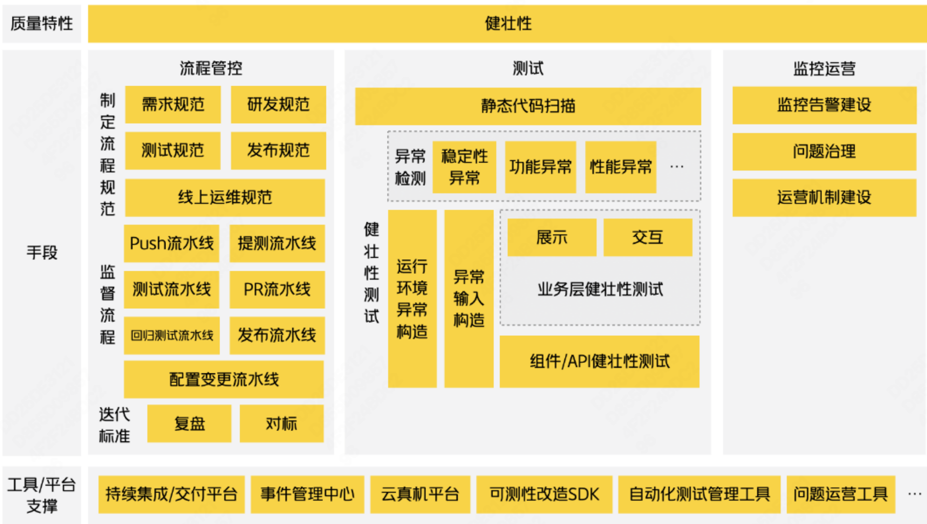


05 总结及展望

在健壮性工具建设一期里，我们实现了 App 页面加载展示场景的健壮性问题检测，支持崩溃、卡死和部分功能异常这三类异常检测。另外，基于节点相似性优化变异数据生成策略能够在保持效果不变的情况下有效控制测试时长，但是否有更优的合并算法和推荐算法，还需要更多的尝试。



在后续工具的迭代还会继续围绕异常构造和异常检测这两个方向，支持更丰富的构造能力和检测能力，以及更高效的构造效率。短期建设上，我们将会从业务视角出发丰富自动化变异数据生成建模，完善客户端异常通用异常检测能力，完成通用前后端交互的数据构造类型（比如：长连接消息）的覆盖；长期建设上，需要支持更丰富的数据和环境构造能力，通过智能化用例生成，提升测试效率。



进一步探索方向：

① 异常检测能力增强

② 异常构造能力增强：

  ① 千人千面场景下保证录制数据的全面性、有效性

  ② 消息、缓存、文件等数据环境构造能力增强

③ 用例生成智能化，减少无效构造

  ① 算法优化

  ② 智能推荐

06 Q&A

Q1：节点相似的判断依据是什么？

A：从实际的 response 分析来说，两个节点的路径完全相似就是从根节点到最终的叶子节点上，它们的路径命名完全相似，数组里两个对象的结构完全一样。

Q2：用例的生成能举个例子吗？

A：比如颜色色值的格式是 #+6 位字符，通常运营配置会出现的情况是忘记添加 #，或色值复制中少了一位。在这种情况下，我们会构造一个色值，比如没有返回 #、色值位数不对、色值添加透明度，把这种场景作为构造情况，在配置里添加上，最后用代码生成。

Q3：健壮性平时执行的频率是什么样的？

A：第一个基于需求维度，需求维度需要人工触发；第二个基于变更维度，当组件发生变更时，可以关联到这段代码或者组件变更的页面，然后触发页面对应的健壮性测试，执行频率会受到组件变更频率的影响；第三个在回归测试时，App 的回归测试两周一次，我们会把所有页面以及它关联的所有的用例都执行一次。

Q4：对于暴露给前端开发的接口，大部分是人为调用参数的变化，随机性相对比较高，对于必填和非必填参数如何确认用例的范围？

A：目前我们在实现的方案里，没有区分参数是必填参数还是非必填参数，所以对于整个数据接口返回里的所有结果都会进行构造，产生的问题是对于非必返回的参数可能产生的问题，到底是否是需要解决的问题，这部分目前通过运营手段做确认。

Q5：首页可能调用 10 个接口，然后针对每个字段都进行异常验证吗？

A：对于首页关联的接口，我们在接口请求、录制过程中和录制完数据后，会对接口进行确认到底有哪些接口是我们需要验证的，这是一次性的成本，录制完成后，会对每个字段都进行异常验证，当然会有一些黑白名单的设置。

Q6：对色号这种情况有一种生成规则嘛，这个规则是怎么制定?

A：刚刚我只是举了一个色号的例子，其实对于图片、请求的资源文件、配置文件、跳转链接，每一个对应到的业务语义，我们都有对应的用例生成规则，我们会根据参考依据，比如第一个是本身我们在通用的基础库里怎么处理这些问题，这里有一个基础的规则；第二个是我们积累了线上问题情况实际可能会产生的错误或者变异情况，生成第一版基础规则，在第一期工具里找相关研发达成共识，这样的话，数据变异是处于合理范围。

Q7：执行的时候，如何知道页面对应哪些规则提前配置?

A：执行时，在测试接入过程中有一个配置过程，它不是配置这个页面和接口的关联关系，而是配置我们要测试哪些页面，自动触发自动化录制过程，就是到这个页面时，会触发哪些接口请求，生成这个页面和这个接口请求的对应关系，给到对应的配置人做确认，保证哪些接口是真正可能想要构造的，哪些接口不需要构造，最后以这个为基准测试，基于录制过程，比如业务迭代里面产生了新接口，我们在录制中能够感知到它关联的接口发生了变化，在发生变化时发消息给对应的测试提交人 / 负责人，TA 确认这条规则放到黑名单里还是更新到需要构造的接口里。

Q8：是否有做页面显示的一个校验？怎么做的?

A：目前我们在页面里的模块做了“是否展示”校验，基于当前集成到美团的可测性 SDK，这个 SDK 会获取到当前页面是否渲染里是否展示了对应模块的信息，通过请求把对应模块描述传给 SDK，通过返回来校验是否展示。

07 参考资料

[1] 健壮性：https://en.wikipedia.org/wiki/Robustness

[2] IEEE 健壮性：https://ieeexplore.ieee.org/document/7438745

[3] Ballista：Carnegie Mellon 大学的研究项目，通过黑盒自动化测试的方式，发现导致系统崩溃或异常终止的系统调用或接口调用。

[4] 基于布局视图的解析 SDK：美团 App 页面视图可测性改造实践 –XraySDK

新一代实验分析引擎 | 驱动履约平台的数据决策

本文介绍了美团履约技术平台的新一代实验分析引擎，该引擎对核心实验框架进行了标准化，并融合了众多先进解决方案，有效解决小样本挑战。同时，提供了多样化的溢出效应应对策略，并针对不同业务场景提供了精准的方差和 P 值计算方法，以规避统计误差。希望对大家有所帮助或启发 ~~

1. 引言

自谷歌于 2000 年引入 AB 实验以来，这一方法已成为互联网公司优化产品的核心手段，使得基于数据的快速决策成为可能。但随着 O2O 平台经济的崛起，AB 实验需面对更复杂的多边市场和强 LBS 属性带来的挑战。在这些平台中，消费者、服务提供者和商家的互动极大增加了实验的复杂性，传统的实验平台在小样本和溢出效应面前显得力不从心。其根本原因在于用于接收实验数据和实验配置信息并输出实验报告的实验分析引擎，仅能正确支持非常有限的一组实验设计，难以满足不同用例的实验分析需求。我们的故事便从此开始，这是我们的故事，也可能是你的解答。

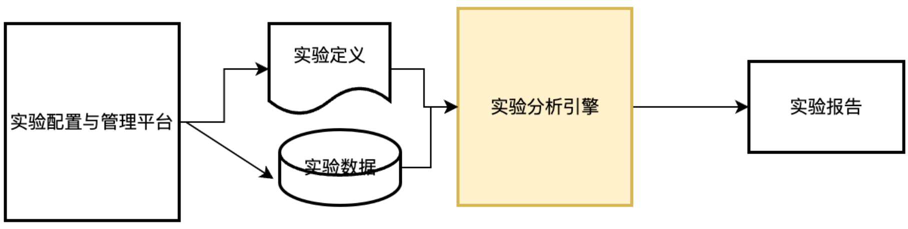


图 1. 实验分析引擎：接收实验配置与实验数据，输出实验报告

1.1 AB 实验的发展：从单边场景到单边与多边场景的共存

互联网经济的进步催生了商业模式的创新。我们看到了谷歌、阿里巴巴这样的公司通

过向消费者直接提供内容和商品而取得成功，同时，美团、滴滴这样的平台型经济（即时服务平台）代表了一种全新的商业模式，它们通过整合线上线下资源，创新性地提供即时配送和出行服务。在这一变革的背景下，AB 实验的应用已经超越了单一用户行为分析，扩展到包含多方市场参与者互动的更为复杂的多边场景中。这种从单边场景到多边场景的共存，使得多边实验相比于单边实验展现出新的特点：**多方参与带来的相互依赖性和溢出效应，LBS 特性引起的小样本挑战，以及更加复杂和多元化的指标体系**。

特点	单边场景实验（C端实验）	多边场景实验（物流实验）	多边带来的新变化
参与者与行为	单一群体参与且行为相对独立	多方参与者（消费者、商家、服务提供者如骑手或司机）	**溢出效应** （难以确定实验单位、需要多样分组分析方法应对）
样本规模	通常较大	受限（小样本）	**小样本** （不仅分组难同质而且难以检测出小提升）
指标体系	相对单一（仅关联单一用户行为或转化如点击率、转化率）	多元且复杂（需要反应多方互动效果）	**复杂多元的指标体系** （持续的指标建设）

1.2 传统实验分析引擎面临的不足：难以解决多边场景中的小样本与溢出效应挑战

单边场景实验，由于较少出现溢出效应和具有大样本的特性，基于用户的随机对照实验足以满足需求，从而简化实验设计。在分析方面，大样本的优势消除了因正态性假设不成立而导致的 P 值计算问题，而随机分组方式也避免了非随机分组情况下复杂的方差计算问题。多边场景实验面临小样本和溢出效应的挑战，导致实验设计和分析的复杂化。传统的随机分组和以用户为单位的实验方式往往不适应这些场景，需要采用多元分组方法和不同实验单位以提高准确性。分析时，必须谨慎处理小样本带来的统计误差和溢出效应解决策略可能引入的携带效应及方差计算误差。

传统实验分析引擎以单一实验单元、普通随机分组、大样本和个体独立性为基本设定，尽管在单边场景实验中表现出色，但在多边场景实验中则显得不足。这主要体现在大样本假设不再成立时，正态性假设失效引发 P 值计算的问题；小样本和溢出效应的存在使得单一实验单元和普通随机分组不再合适。**这不仅需要配备相应的分析方法来处理方差计算，并防止新方法可能引入的携带效应与大实验单位聚合所致的方差误**

差，还需要具备未来扩展的灵活性，以免在扩展时需要重新构建整体架构，确保能够快速迭代并满足交付时效。在这样的背景下，传统实验引擎在多边场景下由于方法单一、鲁棒性不足以及缺乏灵活性，往往难以胜任。

传统实验分析引擎在多边场景实验中的局限性		
问题分类	**具体问题**	**描述**
适配性问题	分组同质性要求	在小样本约束下，普通随机分组无法满足分组之间的同质性要求
	分析方法适用性和多样性	1. 传统的P值计算方法在小样本情况下可能不适用，因为正态性假设可能不成立 2. 传统实验引擎缺乏处理多边场景下小样本和溢出效应的多样化分析方法
灵活性问题	实验方法的鲁棒性	现有的实验方法在复杂场景下可能不够鲁棒，容易产生统计误差
	系统的扩展性	传统实验引擎可能缺乏快速迭代和扩展的灵活性，新场景适配可能需要重构架构

1.3 实现可信与高效：新一代 AB 实验分析引擎构建的动力

我们曾经有过这样的实验历程，实验者面临着效率低下和结论可信度不断受到质疑的双重困境，导致挫败感和无助，平台建设者，因实验平台的迭代效率和可信度，正在失去用户信任。

- 当手动调查和分析、重复实验和无休止的质疑成为了日常，这不仅影响实验的效率，更让每一次得出的结论都充满争议。

实验者困境	现象
实验效率低下	手动分析的漫长等待
	实验结果中的问题可能未被识别，使得实验得出的结论与策略实际上线后的效果出现偏差，重复实验
实验结论遭受质疑	标准化缺失而导致实验结论的连番质疑
	小样本和溢出效应束缚了经验不足的实验者，导致他们依赖于易受外界影响的准实验，这使得实验结论容易遭到质疑

- 实验平台的功能限制和不可信导致了用户信任危机。①功能局限和可信度问题频发，用户对平台失去信心。用户不得不线下手动设计和分析实验，实验平台的角色退化为提供基本的实验运行支持，而所有的设计和分析工作则需要用户自己承担；②该平台无法支持实验者遇到的各种用例和潜在的未来扩展。为支持新的实验用例，平台需要频繁进行迭代更新，不仅耗费时间和精力，甚至多次推迟了项目进度。

1.4 新一代实验引擎：对内实验分析的中心方法库，对外实验实践的有用指南

分流框架和实验方法是实验效率和可信度的关键，其中分流框架一经选定，其广泛适用性和稳定性意味着它很少需要变动。然而，为了适应不断演变和多样化的业务需求，实验方法必须不断完善和更新，这对于确保实验的可信度至关重要，同时也影响着实验效率。为此，我们构建了新一代实验分析引擎以标准化实验分析流程。并通过解耦和模块化集成，将引擎与实验操作流程的基础设施分离，以提高迭代的灵活性并保持平台的稳定性。也可能是你的解答：

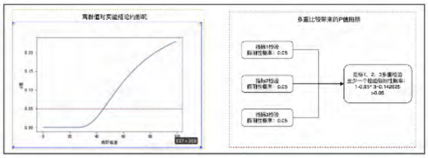
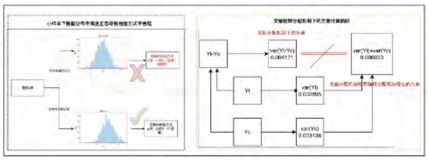
1. 该实验分析引擎集成了先进的实验技术（如协变量自适应分组来解决小样本问题、轮转实验和双重差分实验来应对溢出效应问题，以及降低方差的技术来增强实验功效），并作为核心方法库，面向司内全体成员开放，方便实验者按需取用。
2. 它应当是一个有用的指南，帮助您思考开发过程中面临的挑战，以及为了确保实验可信且高效所需采用的不同方法。

2. 实验者困境成因

2.1 在物流领域的多边实验场景下，构建一个正确和可靠的 A/B 测试平台依然是挑战

尽管单边场景实验（消费者实验）得益于其大样本量和单一业务流向，大多数情况下，除了个别场景的溢出效应外，基本的随机对照实验足以满足分组一致性、个体独立性的假设以及减少随机误差的理论标准，使得整个过程看起来相对简单。然而，在分析阶段，仍然潜伏着异常值、多重比较和方差估计盲点等陷阱。如果忽视这些问题，很容易导致分析偏差，使实验结果不可信性。多边场景实验，因其多边特点和 LBS 属性极大提升了实验的复杂性。除了需要应对单边场景实验中常见的统计问题，还面临由溢出效应和小样本引发的叠加挑战。溢出效应的解决策略不但可能遭遇方差计算和携带效应陷阱，有时还会因应对策略而产生小样本问题，小样本限制本身又增加了这

些问题的复杂度。为了解决小样本和溢出效应的问题，可能需要引入多种分组方法和不同的实验单位，这些都要求有相应的分析方法相匹配。一旦处理不当，很容易落入统计误区，造成分析上的偏差，进一步加剧了实验设计的复杂性。

类别	陷阱描述	陷阱举例
单边场景实验	**忽视异常值和多重比较陷阱，造成统计误差** 忽略离群值会导致偏差和高估方差，掩盖实际策略效应。模拟显示，离群值增多可使显著性检验从显著变为不显著。同时，多指标或多次实验观察若不考虑多重比较，可能增加假阳性风险。	
多边场景实验[1]	**实验设计需应对小样本和溢出效应对准确性的冲击，并权衡方差和偏差** 小样本可能引起分组差异和实验敏感度降低。不同质的对照和实验组难以提供有效比较，而样本量不足则更容易不同质。此外，相互作用引发的溢出效应则复杂化了结果解释，共同影响策略效果的准确性。实验设计也需权衡方差和偏差，例如在解决溢出效应时，轮转实验需要慎重选择实验地理单元以及时间片长度[2]。	
	实验分析需同时应对小样本约束以及解决方案可能新引入的统计误区 小样本约束下，中心极限定理可能失效，导致P值误判。为保持实验同质性，需探索配对或轮转分组等方法，但分组机制若在方差计算中被忽略，将面临错误估计方差的陷阱。	

2.2 实验服务团队构成与专业技能需求不匹配

获得高质量实验结果需要对实验和统计有专家级的理解，但许多实验服务团队主要由工程师组成，缺乏数学和统计学背景。尽管有些团队涉及数学、统计学人员，他们更多是以提出需求的方式参与平台建设，工程师在开发核心分析方法时面临专业知识限制，这影响了分析方法的质量和鲁棒性。实验服务团队缺乏数学科学背景，面对特定场景时难以有效应对。特别是在履约业务中，某些团队需采用更复杂、更高级的实验方法来解决场景限制。但受限于经验和平台功能，缺少能够确保高度可信性的先进实验方法，导致他们不得不选择信度较低的准实验或观察性研究。这些方法基于的假设易受外部因素影响，一旦环境发生变化，假设失效，实验就可能失败，需多次重做。

履约场景所需高级方法举例		
实验方法	**实验方法特点**	**实验难点及一些解决手段**
随机对照实验	1. 优先级最高，准确性最好。 2. 在样本量较少的场景下可能存在检验功效不足的情况。 3. 在某些业务场景中溢出效应较严重，例如调度场景下骑手会在实验组和对照组区域间来回跑动，采用区域随机分流会带来策略效果估计偏差。	1. 小样本场景：采用**CUPED降方差方法**，使实验检验灵敏度更高。 2. 存在溢出效应：考虑使用**随机轮转实验**或者**双重差分实验**来缓解溢出效应造成的估计偏差。
随机轮转实验	1. 优先级中等。 2. 能够规避溢出效应的影响。 3. 在某些业务场景下无法支持轮转实验，例如基础定价方向由于PR风险，不允许价格每日较大波动 4. 随机轮转需要确定合适的时间片，综合考虑携带效应带来的偏差和实验功效。虽然缩短时间片可以增加样本量，但携带效应可能引起偏差，小时间片也可能导致更大波动，因此增加的样本量未必提高实验功效。在确定时间片大小时需谨慎综合考虑。**以我们实际业务场景为例，虽然小时时间片能够增大样本量，但溢出效应不可忽略，且小时时间片增加的样本量带来了更大的方差，并未提升实验功效。综合评估后，我们最终确定以天为轮转周期。** 5. 按天轮转实验周期短时样本量较少，异常值影响不可忽略。	1. 业务上不支持轮转实验但又存在溢出效应：考虑使用**双重差分实验。** 2. 按天轮转样本量较少： a. 使用**Fisher非参数检验**方式，小样本下显著性检验稳定性更好； b. 使用**统合分析**整合多城的多次实验，提高实验灵敏度。 3. 异常值影响不可忽略：利用**异常值剔除**，摆脱异常值对结果影响。
配对随机实验	1. 优先级中等。 2. 能够更好保证指标同质性，尤其在小样本下。 3. 对于一些不可控因素难配对均衡，需权衡配对计算复杂度和配对偏差。 4. 较难适用于多个分组的情况。	1. 天气等不可控因素难均衡：（1）考虑对**地域相邻**的实验单位进行配对，使例如天气等不可控因素相似；（2）对差异较大的时段数据进行**异常值剔除**，进一步保证实验组对照组在不可控协变量上同质。
双重差分实验	1. 优先级最低，准确性相对较低。 • 半城双重差分实验的溢出效应仅发生在半城交界处，相对可忽略。 • 实验周期短时样本量少。 • 比较依赖平行趋势假设的成立。	1. 实验周期短时样本量少：回归模型中考虑**时间固定效应**与**个体固定效应**以提升检验功效。 2. 可能存在恶劣天气等剧烈波动破坏平行趋势：利用对实验前数据**异常值剔除**，进一步保证平行趋势成立。

2.3 实验分析引擎仅支持非常狭窄的一组实验设计，引发鲁棒性和扩展性挑战

实验分析引擎应确保提供准确无误的分析结论：无论用户选择何种实验设计，任何人都应能够信赖实验结果，无需进行定制验证、深入理解统计学，或详细了解平台。在评估了一些失败案例和其它平台的建设现状后，我们得出结论，建立在“单一实验单元、普通随机分组、大样本和个体独立性”基本设定下的传统实验引擎，只能正确支持非常狭窄的实验设计集合。面对复杂多变的实验需求，这种设计往往产生大量可疑数据，不仅威胁到实验结果的准确性，还导致了大量不必要的重复工作和劳动力浪费。

实验分析引擎不足	问题
实验分析引擎未内置指标方差处理，依赖ETL离线处理后导入，缺乏对实验上下文的动态适应能力	接入指标仅适用于特定分组或实验单位，变更条件即触发重新开发指标方差的离线流程
	面对异常数据剔除或降低方差等预处理需求，需重启指标方差的离线开发流程
分析方法单一，难以满足多变实验场景的适配需求	显著性分析依赖正态性假设，不适用于非正态分布的指标
	面对小样本和溢出效应，缺少高级分析手段，导致实验者依赖低可信度的准实验，使实验容易受环境干扰而失败，进而频繁导致重复实验的情况

3. 我们的方法：新一代实验分析引擎，在单边与多边实验场景中实现高效与可信

问题、失败和行不通的方案，这就是推动事情向前发展的意义所在。易出错的分析结果、问题域覆盖不足、数据科学家重复发明轮子的努力浪费、学习缓慢是驱动我们构建新一代实验分析引擎的动力，与传统实验分析引擎比，它应该具备一下显著特点：

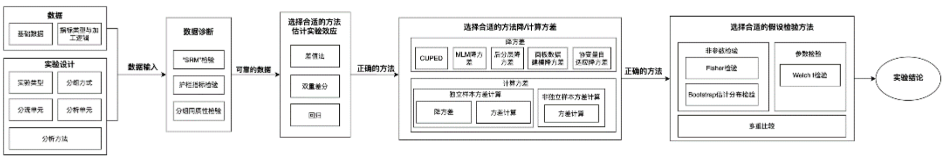
	传统实验引擎	新一代实验引擎
可信	仅支持非常有限[3] 的实验设计集合的可信	分析流程标准化，鲁棒、灵敏，适配多样实验用例和数据异常CASE，确保实验可信
问题域覆盖	单边实验场景	单边+多边实验场景，小样本、溢出效应解决方案丰富、前沿
开放易用	高度集成于实验平台，缺乏面向用户的交互界面，不易开放共享	开放易用的中心方法库：低门槛交互界面，与实验平台解耦
易扩展	基于固定设定[4]，紧耦合实验基础设施，难以在保持架构稳定的基础上扩展	解耦基础设施，简单的配置更改和在固定接口下的方法集轻松实现能力升级

3.1 标准化实验分析，确保结论可信：自适应实验上下文，通过标准化和自动化分析流程，消除低效率和结论争议

为确保任何实验结果的质量，它具有合理的准确性（一致性）、对模型误设的鲁棒性（鲁棒性），以及对业务或产品变化的敏感度（实验灵敏度）。实验分析缺乏标准化和自动化，使用 SQL 查询或临时脚本在本地环境中分析实验，这个过程既耗时又容易出错。由于每个人使用不同的分析方法，缺乏标准化，这可能会影响结果的质量和准确性（一致性）；统计陷阱防不胜防，因统计陷阱导致实验结果不可信，那么工程和分析的努力将会被浪费，减缓策略迭代速度。需要实验引擎对错误的模型假设要有鲁棒性，能够自动适配不同的场景和意外 case；在方差和 P 值计算方面，集成更多前沿功能以提高实验的灵敏度。

实验分析引擎无需预设，全面依据实验上下文自动选择分析方法，并通过标准化与自动化流程确保效率和一致性。每当一个新实验被输入到我们的实验平台时，它将经历一下流水线过程：数据可靠性验证、数据预处理、策略效应估计、方差计算（包括降低方差）、P 值计算（假设检验），最终得出分析结论，以实现流程的标准化。在这一固定流程下，我们设计了相应的方法选择器，并集成了一系列丰富的分析方法，以便分析引擎根据输入的数据和实验设计信息选择最合适的方法，从而得出可靠的结论，确保分析过程的鲁棒性。

1. **数据可靠性验证**：通过分组同质性检验、系统性偏差检验（SRM）和样本量检验来确保所依赖的数据质量。
2. **数据预处理**：通过异常值检测识别异常数据，在不改变数据分布的情况下剔除异常数据，提前探测数据分布，为后续流程选择合适方法提供关键输入。
3. **策略效应估计**：依据实验类型，选择恰当的策略效应估计方法，如差分法、双重差分法或回归分析，以估计策略效应。
4. **方差估计**：根据实验设计信息，选择合适的降方差和计算方差方法，进行方差估计。
5. **P 值计算**：根据实验设计和数据分布，选择合适的检验方法，如参数检验或非参数检验，计算 p 值。
6. **输出实验报告**：整合以上各阶段信息，输出详尽的实验报告。



实验分析引擎，标准化实验分析流程

3.2 中心方法库：整合优秀实践并加速共享；全面覆盖单边、多边实验场景实验问题域

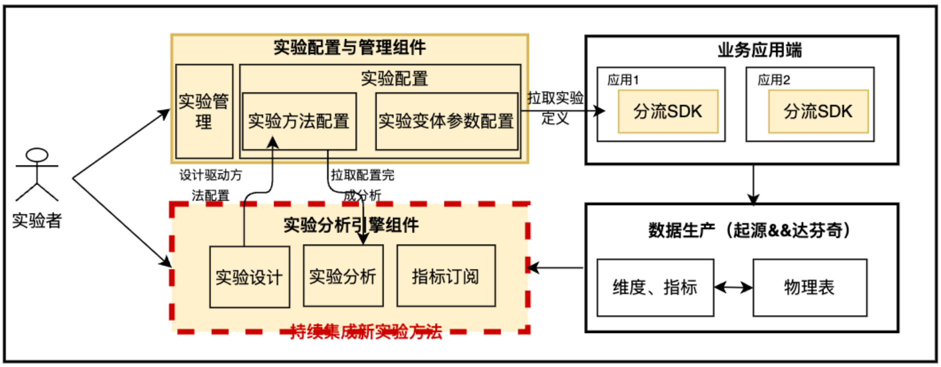
实验引擎作为核心方法库，整合了覆盖单边和多边场景实验的最佳实践，尤其包括填补行业和学术界空白的先进方法：①针对比率型指标减少方差的“二元降方差方法”；②针对小样本分组同质性的“协变量自适应分组技术”；还有医学界常用于应对小样本和数据隐私问题的统合分析方法；以及适应特殊场景的异常值处理和多样化的溢出效应解决方案。

作为一个开放的中心方法库，它不仅消除了重复开发相同解决方案的资源浪费和对业务响应的延迟，而且促进了跨团队的知识共享和能力提升，极大加速了实验学习过程，推动了实验能力的整体提升。

3.3 解耦实验基础设施：专业人员专注专业领域，打破实验平台能力和迭代的限制

在审视失败的实验案例和传统实验分析引擎后，我们认识到核心抽象只能正确支持有限的实验设计，微小的偏差就可能导致结果的不准确。实验和统计方法的选择取决于多种因素，而复杂实验的进行则需专家级的理解。因此，构建能支持广泛用例的实验平台不仅需要工程师，还需数据科学家和数据仓库专家的集体努力。新一代实验引擎在实验设计、配置、流量分配和数据处理等方面实现了与基础设施的解耦，使各专业团队能在各自领域内发挥专长，提高平台的迭代效率。工程团队负责构建实验平台的管道，提供配置管理和流量分配工具；数据科学家负责集成统计方法，确保方法的鲁棒性和问题域的全面覆盖；数据仓库专家则确保指标的一致性。通过角色分工，我们避免了单一专业造成的迭代瓶颈和平台鲁棒性不足。

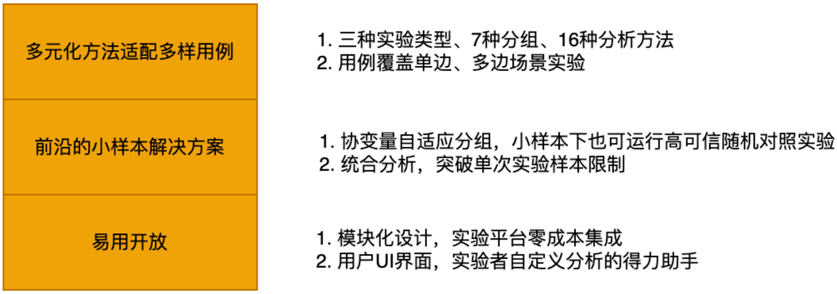
实验分析引擎与实验设计、配置、分流和数据生产流水线的基础架构解耦。实验日志和流水线生成包含实验单位、分组方法、分析方法、实验周期等元数据信息的统一数据集，以及详细的指标元数据。所有分析均在这些通用数据上完成，以产出最终的实验报告。这种流水线的通用设计**极大地简化了新方法的引入，仅通过简单的配置更改和在固定接口下的集成，就能够灵活适应不同的实验场景，并确保新能力的集成速度与业务的快速迭代保持同步。这种设计允许它像积木一样无缝集成到其他实验平台中，无需改变现有架构，从而帮助这些平台轻松应对各种实验场景的挑战。**



实验分析引擎：解耦实验基础设施

4. 实验分析引擎揭秘

在履约，运行着大量实验，我们希望赋能所有团队以速度、严谨和信心进行改进。为此，秉承着可信、开放、敏捷、易用的原则，打造了新一代实验引擎。该引擎不仅覆盖单边、多边实验场景多样实验用例，还提供了业界领先的小样本解决方案，并作为中心方法库向外界开放，加速实验能力共享和整体实验水平提升。



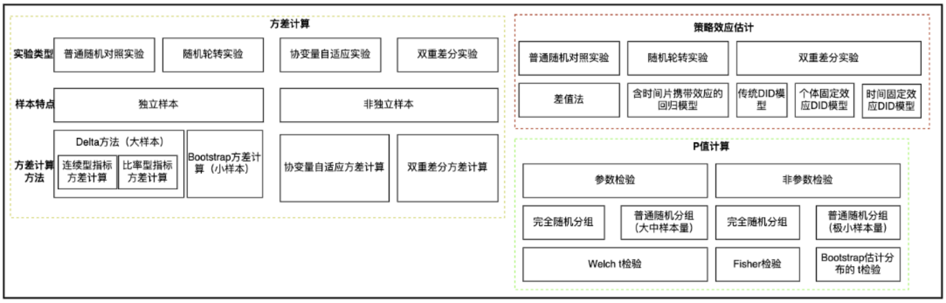
实验分析引擎揭秘

4.1 多元化方法适配多样实验用例

实验引擎集成了适用于随机对照实验、随机轮转实验、准实验的 7 种分组和 16 种分析方法，包括降方差（2 种）、效应估计（5 种）、P 值（3 种）和方差计算（4 种）、统合分析（2 种），以克服小样本和溢出效应带来的挑战，确保各种业务场景下统计结果的准确性和实验结论的可靠性。采纳业界先进的协变量自适应分组及二元 CUPED 技术，我们成功突破了小样本实验的约束，提高了结论的精确度。针对溢出效应，我们提供多元化的实验选择，包括分层随机轮转和多种双重差分模型，增强实验的适应性。系统的自动离群值排除能力和灵活的实验周期延长机制，进一步提升了实验结果的稳健性，降低了异常数据的干扰。

1. **自动化处理离群值和实验周期调整**：保证策略效果评估稳健性。自动排除离群值与调整实验周期增强了结果的稳健性。因素如天气、节假日可能引发离群值，大的离群值可能会使实验结果从显著变为不显著，影响统计显著性。尽管 AA 测试可预推实验周期，但实验前数据并不总能预见实际情况，故需灵活修正周期，以降低假阴性风险。

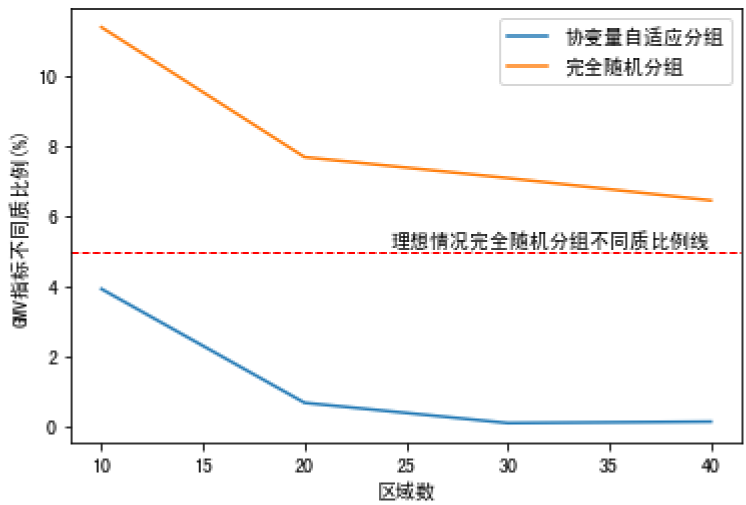
2. **精准实验分析**：业务适配与统计误差防范。实验引擎提供多元计算方法，满足不同实验和指标需求，确保分析的灵活性与准确性。针对以城市、区域、站点为实验单位的实验，自动调整标准误差以防数据聚集误差。方差计算根据样本独立性与规模，区分独立与非独立样本，大样本采用 Delta 法，小样本适用 Bootstrap。策略效应估计覆盖差值法至回归分析，适应多变实验环境。结合参数与非参数检验，适配样本量与分布差异，计算 P 值以确保实验结果的稳健性，避免统计误判。



实验分析引擎多元化分析方法适配多样用例

4.2 协变量自适应分组：小样本实验新解，实现从低可信准试验、观察性研究到高信度随机对照实验的进阶

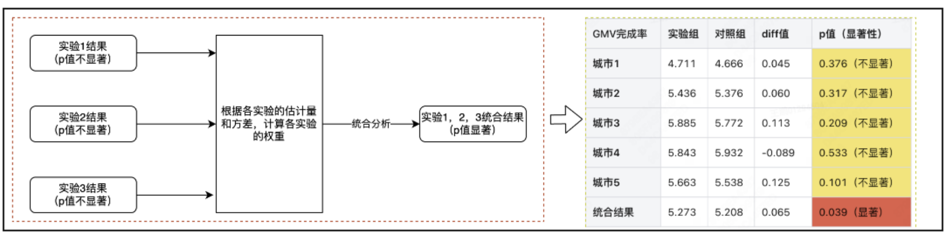
协变量自适应分组技术，起源于医学研究领域，能够在样本数量受限的情况下确保分组的同质性。经过数据验证，此技术在至少 10 个样本的条件下也能实现有效的分组同质性。这一方法已在物流、营销、应用程序测试等场景中得到应用，适应了小样本和资源受限的实验环境，满足了实验设计的同质性要求。通过这种技术，可以避免因缺乏高级方法而不得不依赖低可信度的准实验或观察性研究，从而解决了结果量化不准确、实验失败率高和重复实验等问题，有效地降低了实验成本。以我们的实际场景为例，针对区域分组，数据显示：随着样本量的减少，协变量自适应分组方法相比于业界常用的完全随机分组方法，能够稳定有效地将不同质比例控制在 5% 以下（这是大样本情况下的理想比例）。而在样本量少于 40 的情况下，完全随机分组方法的不同质比例已超过 6%，实际模拟如下图所示：



协变量自适应分组效果模拟验证

4.3 统合分析：突破单次实验样本量限制，可以利用历史实验结果得出更可信的结论

统合分析不仅是在小流量场景下提高统计功效的有效工具，还可以利用历史实验结果得出更可信的结论。它能整合对同一假设的多次实验结果，有助于在小流量场景下检测策略效果，且无需单个实验有大量样本就能获得可信结果。在小流量场景下，即使采用了各种提高统计功效的方法，实验的统计功效可能仍然不足。对此，可以运行两个或更多的重复实验，并通过统合分析整合结果以提高统计功效。对于意外的结果，可以通过运行重复实验并通过统合分析整合多个实验结果来进一步确认实验结论。

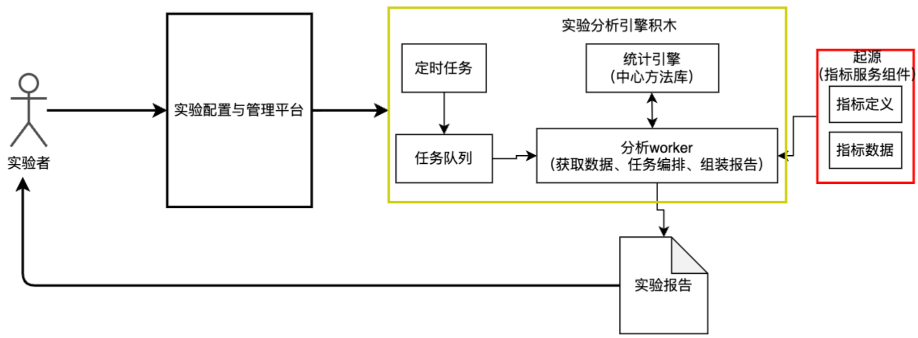


GMV完成率	实验组	对照组	diff值	p值（显著性）
城市1	4.711	4.666	0.045	0.376（不显著）
城市2	5.436	5.376	0.060	0.317（不显著）
城市3	5.885	5.772	0.113	0.209（不显著）
城市4	5.843	5.932	-0.089	0.533（不显著）
城市5	5.663	5.538	0.125	0.101（不显著）
统合结果	5.273	5.208	0.065	0.039（显著）

统合分析举例

4.4 零成本集成与自助分析

模块化设计：实现实验平台的零成本集成，推动实验平台发展的催化剂。实验引擎采用了一种通用流水线设计，能够通过识别实验单位、类型、分组方法、周期以及所需的指标数据和元数据，自动地进行实验分析并生成报告。这一设计摒弃了与具体实验操作流程的耦合，使得任何实验平台都能通过遵循标准协议与实验引擎交互，从而无缝复用分析引擎的功能，达到零成本集成。实验引擎内部，它通过分布式计算节点，引擎负责执行实验的分析工作。每日定时任务在各计算节点上启动，轮询待分析实验并激活分析流程。分析任务激活后，生成的分析 worker 会调用数据服务组件，提取所需的指标元数据和数据集。数据获取后，分析 worker 根据既定流程进行任务排程，随后启动统计引擎进行计算，最终整合分析结果，形成详尽的实验报告。



实验分析引擎的模块化设计

用户界面 (UI) 提供：自定义分析的得力助手。实验者和数据科学家在使用现有实验平台进行实验探索时，经常会因为指标覆盖不全、当前分析方法不适用以及需要即兴进行个性化分析而遇到难题。由于平台不能及时提供所需支持，他们不得不回到使用 SQL 查询或临时脚本在本地环境中分析实验。在这种情况下，实验引擎将成为您的得力助手，助您高效完成实验分析。实验者仅需在实验引擎提供的用户界面中完成实验分析配置，一键点击“实验分析”，即可立即生成实验报告。实验分析配置内容包括实验定义和指标定义：实验定义主要涵盖实验单位、实验类型、分组方式、分析方法（可选）等；而指标定义则囊括了指标名称、计算口径、类型等元数据定义以及数

据获取逻辑。

5. 总结与展望

问题、失败和行不通的方案，这就是推动事情向前发展的意义所在。基于固定设定[4]的传统实验分析引擎仅支持非常有限的实验设计集合，即便是微小的偏离也难以得出可信的实验结果，无法满足履约这种多样业务场景的实验诉求。基于此，我们打造了新一代实验分析引擎，该引擎对核心实验框架进行了标准化，并融合了众多先进解决方案，例如行业领先的协变量自适应分组、二元 CUPED 降低方差技术以及统合分析方法，有效解决小样本挑战。同时，提供了多样化的溢出效应应对策略，包括轮转实验、双重差分实验和半城随机配对实验等。此外，引擎还针对不同业务场景提供了精准的方差和 P 值计算方法，以规避统计误差。作为一个集成了履约数据科学团队所有最佳实践的中心方法库，它不仅强化了实验的标准化理念，还促进了知识共享。

将实验分析引擎作为中心方法库与基础设施解耦，对于自动化与新功能快速迭代至关重要。按照 Fabijan 等人（2017）对实验平台四个阶段[5]的描述，基础设施（如实验配置与管理组件、分流 SDK）一旦建立，迭代较少，而指标和实验方法则随业务演进持续增强。从用户角度看，常规计算流水线无法满足的高级实验分析功能需随业务场景不断演化。这种解耦，在保持基础架构稳定的同时，通过配置和统一接口集成实验方法，快速适应需求变化；新一代实验引擎通过与实验平台解耦，以实验分析平台的模式对外提供服务，不仅满足实验者和数据科学家的灵活探索需求，而且避免了重复造轮子带来的工作浪费，加速了实验能力的共享和提升。这一方案应能作为一个有用的指南，指导如何思考开发中的挑战，以及实验为了可靠和高效所需采用的不同方法。

未来，随着观察性研究方法和针对溢出效应的先进解决方案（如随机饱和实验、基于地理位置的溢出效应刻画、精确捕捉携带效应的小时级轮转方案）在履约场景的成功落地与实践，我们将进一步开放这些实践，促进更广泛的能力和知识共享。

注释

[1] 多边场景实验：多边场景实验同样存在单边场景实验的陷阱。

[2] 实验地理单元以及时间片长度：轮转实验通过地理单元和时间的随机化来减少 SUTVA 违规。较大地理单元可减少溢出，但会加剧小样本问题，增加方差；而短时间窗口可能带来携带效应，增加偏差。

[3] 非常有限：实验单位是用户，分流方式是随机分流。

[4] 固定设定：单一实验单元、普通随机分组、大样本和个体独立性。

[5] 四个阶段：

（1）爬行阶段：完成运行实验所必须的基础设施建设，包括：实验配置与管理组件、分流 SDK。

（2）步行阶段：定义更多标准指标，包括量化平台可信能力的指标和评估所需的业务指标。

（3）跑步阶段：更加丰富和完善的指标体系。

（4）飞行阶段：丰富的实验用例，满足更多实验域诉求。

美团 RASP 大规模研发部署实践总结

背景

RASP 是 Runtime Application Self-Protection（运行时应用自我保护）的缩写，是一种应用程序安全技术。RASP 技术能够在应用程序运行时检测并阻止应用级别的攻击。随着云计算和大数据的发展，应用程序安全越来越受到重视。RASP 技术作为一种新型的安全防护手段，正在逐渐被业界接受并广泛应用。其中 Java RASP 是一种针对 Java 应用程序的 RASP 技术。通过在 Java 虚拟机（JVM）级别进行监控和防护，能够有效防止对 Java 应用程序的攻击。

RASP 建设挑战

在业界，RASP 的部署形式一般有 `agentmain`、`premain` 两种方式，二者各有优劣。适合不同的业务场景，以及安全需求。

1. `agentmain`：业务无需改动，无需重启，热插拔，动态升级。有性能抖动，业务有感知。
2. `premain`：需要改动，需要重启，前置注入，升级需要重启。无性能抖动，业务无感知。

美团的 RASP 建设时，大部分业务都已经在线上运营，而且有多个发布平台，没有提供一个统一的方式来更改启动参数，也就是说无法通过 `premain` 方式是实现快速部署。为了抓住主要矛盾，快速解决大部分风险问题，我们选择了 `agentmain` 方式。

业务场景复杂

技术方案的设计，依赖于业务形态。美团内部的业务服务中，Java 语言占比 80% 以

上，是主要的风险所在。2010 年至今，有特别复杂的业务部署形态、业务依赖环境、繁多的 JDK 等等，这些都是 RASP 技术方案的挑战。

1. 业务部署方式：物理机、宿主机、富容器、轻容器等；
2. 发布环境：由于历史原因公司已知的发布系统至少有 3 个；
3. Web 中间件：Spring Boot、Jetty、Tomcat、WebLogic、自研框架等；
4. JDK 版本：Oracle、OpenJDK、MJDK、Kona、Dragonwell、毕昇等；
5. 进程数量：单个主机上进程数量和生命周期差异大，有的几千个进程，生命周期有分钟级、年级等；

问题的拆解思路依旧是抓住主要矛盾，以 JDK 版本为例，各个版本 JDK 的主机占比如下图 1 所示：

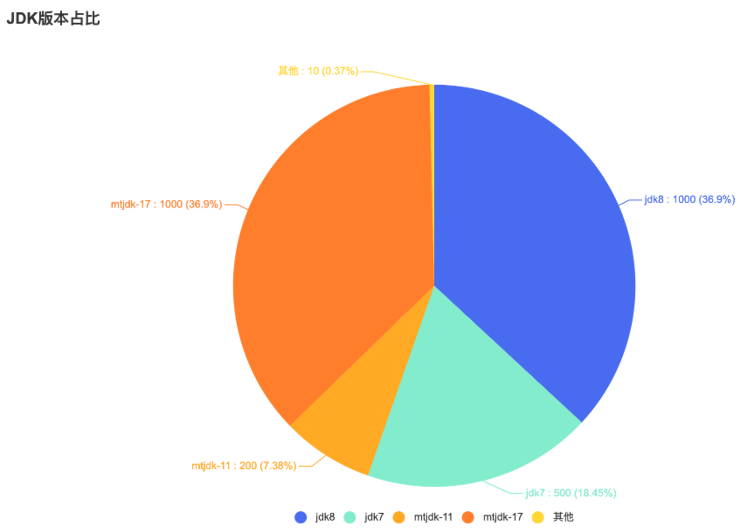


图 1　公司 JDK 版本分布占比

业务目标确定后，解决方案同样具体到某一类的 JDK 上。同样，在发布环境、Web 中间件的差异上，对 RASP 也有了更多的兼容要求。

对业务性能影响大

`agentmain` 的动态注入机制，对 JVM 的影响是不可规避的。影响大小可以从与其他安全防护产品的部署位置看出，下图 2 是常见的基础安全防护产品：WAF、HIDS 和 RASP，他们与业务的隔离方式有以下几类：

1. 主机隔离
2. 进程（容器）隔离
3. 无隔离（或者类加载器隔离）

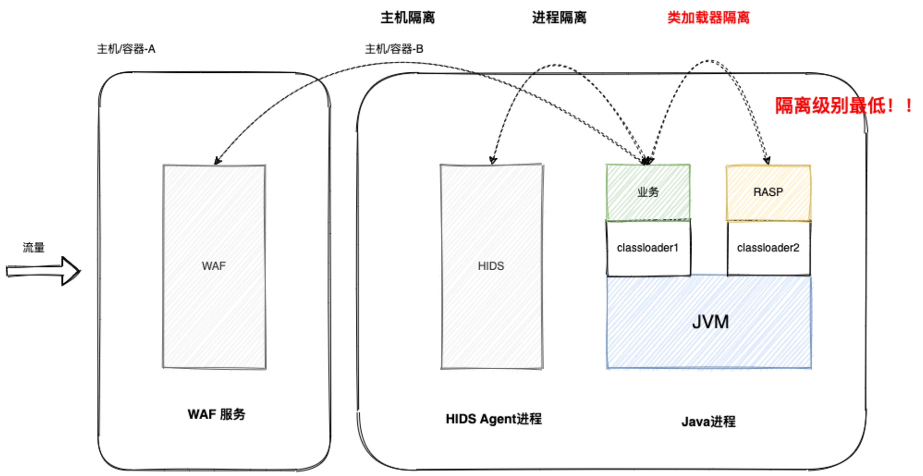


图 2 主机安全防护产品与业务的隔离等级

与其他的安全产品相比，如网络应用防火墙（WAF）和主机入侵检测系统（HIDS），RASP 与业务部署在同一 Java 虚拟机（JVM），其隔离级别是最低的。这就意味着，当 RASP 自身出现 BUG 或者与业务不兼容时，对业务造成直接影响。RASP 一旦出现故障那至少是 S4 级别（核心功能受影如资损、客诉，且预判 5 分钟无法恢复）。从业务指标上分为 cpu 和执行耗时，执行耗时方面主要是对服务的 TP9999 影响较大，而 CPU 方面出现 `cpu.busy` 指标抖动情况。对于业务的指标影响，有以下几种。

运行时注入 cpu.busy 指标突增

下图 3 为特殊情况下运行时注入 `cpu.busy` 指标抖动情况，在 RASP 注入时间内（CPU 分钟级别采样），Java 进程的 CPU 从 0% 飙升到 50%，然后又恢复。如果 RASP 注入之前 Java 进程的 CPU 已经很高了，注入时 CPU 会直接打满（注入前后 10 分钟）。

图 3　运行时注入 cpu.busy 指标抖动情况

运行时注入 TP9999 指标

下图 4 为运行时注入 `TP9999` 指标抖动情况。单机维度，注入时 `TP9999` 从 **5ms 飙升到 1000ms**，大幅度增加，`TP9999` 出现明显的尖刺，对响应时间敏感的服务影响特别大。

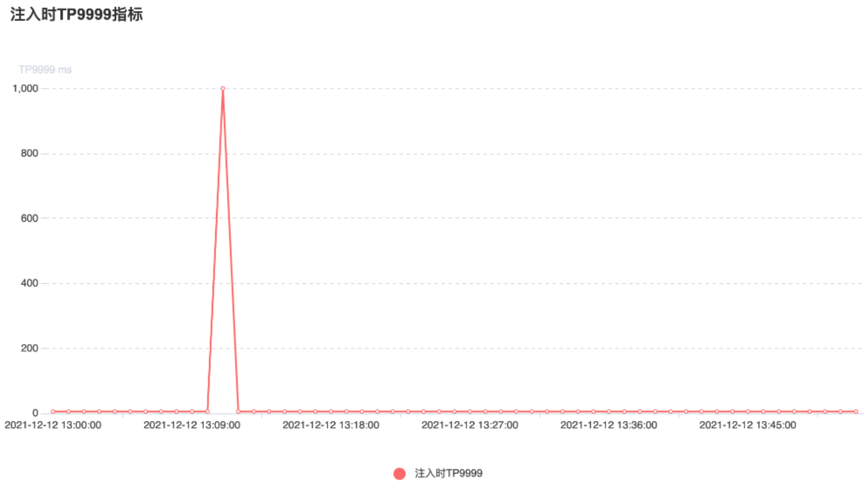


图 4　运行时注入 TP9999 指标抖动情况

启动时性能差与检测逻辑执行耗时长

在 RASP 启动时，大量请求进入到检测流程中，此时 RASP 检测代码没有完成预热，检测方法处于字节码解释运行模式，执行效率低，从而导致启动时 TP 线高。如果正常的请求检测耗时过长，将严重影响业务的 TP 线，甚至导致请求超时。 在 RASP 运行过程中，因为检测引擎执行耗时长也会导致业务超时。

升级变更难

由于原生 Java Agent 的限制问题，JVM 一旦加载了 Agent，就无法进行更新，只能等待 JVM 重启。

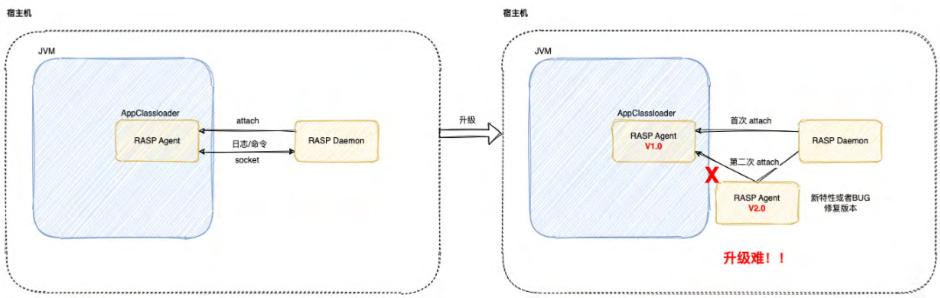


图 5　运行时 Java Agent 的实现原理与升级过程

图 5 左边的图展示了一个典型的**运行时 Java Agent** 的实现原理。在这个过程中，守护进程（这里指主动发起 Attach 的进程 RASP Daemon）会 attach 到目标 JVM 上，然后 RASP Agent 的 jar 包会被 JVM 的 `AppClassLoader` 加载，接着 Agent 就会初始化并开始运行。然而，由于 JVM 类加载机制的限制，同一个类（Agent 入口类）无法被 `AppClassLoader` 加载器加载两次。使用新的 Agent jar 包重新 attach，即使 attach 成功，也不会加载新的类。因此想要增加新的功能或者进行 bug 修复，就必须等待业务进程重启后才能实现。

这也就是说，RASP 功能的升级完全依赖于业务进程的重启时机。然而，我们发现线上有些业务，如大数据服务的核心节点，其重启时间可能长达半年甚至更长时间，这就使得 RASP 的功能升级过程变得异常漫长。由于服务长期未重启，RASP 版本无法进行更新。影响主要有 2 个方面，一方面长期未重启服务的 RASP 版本低于最新版本，RASP Daemon 需要兼容多种 RASP Agent 版本，这无疑提升了代码工程向下兼容的工作量和稳定性；另一方面，未重启的服务最新的 hook 点无法生效，也带来一定的安全风险。

热更新是强诉求

在美团内部，安全部门需要不对业务有过多打扰的前提下保障业务安全运行。大规模重启服务风险高，不具备可实施性。如果遇到紧急漏洞或者重大 bug 时，这种升级难的问题尤为突出。升级难的问题是 RASP 在部署中遇到的第一个重大问题。

监控难

当 JVM 加载 Java Agent 后，由于其运行在业务的同一层面，必然会对业务产生一定的影响。这些影响可能包括 CPU 使用率飙高、`TP9999` 线的波动，甚至可能出现故障如内存泄漏、磁盘打满、核心转储（Core Dump）、触发 JDK Bug、线程死锁、GC 时间变长等等各种问题。业务反馈的线上各类问题的占比如下图 6 所示：

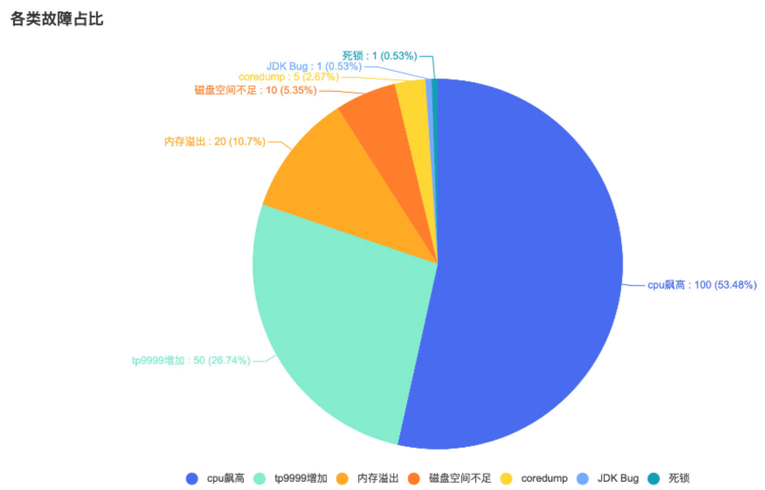


图 6 RASP 各类故障占比

由于 RASP 接入对用户无感知，一旦出现这些问题，业务方定位问题的源头往往耗费大量时间。业务需要对业务状态日志、GC 日志、系统变更日志等进行详细的排查，以确定问题的根因。在实际的运行过程中，往往是业务最先反馈 RASP 影响，而 RASP 不能做到对故障及时感知与处理。

RASP 架构介绍

美团 RASP 利用 Java agent 和 `instrumentation` 技术，通过 ASM 修改类字节码，实时分析检测命令执行、文件访问、反序列化、JNDI、SQL 注入等入侵行为。它最初是从开源项目 btrace 演化而来，后使用 Golang 重写了 btrace 的进程注入的

功能，即架构中的 RASP Daemon 部分，在 Java Agent 端也参考了一些开源项目和公司内部的性能诊断工具。经过多年的迭代，RASP 逐渐形成目前的架构。

通过 RASP 管理端进行主机维度的配置下发，将最新配置更新应用到 RASP Daemon。日志收集和 jar 包下载使用公司基础组件，通过这些组件的协同工作，实现对 RASP 部署过程的管理，包括支持灰度发布、配置回滚、降级和一键关闭操作。下图 7 为 RASP 的配置分发流程。

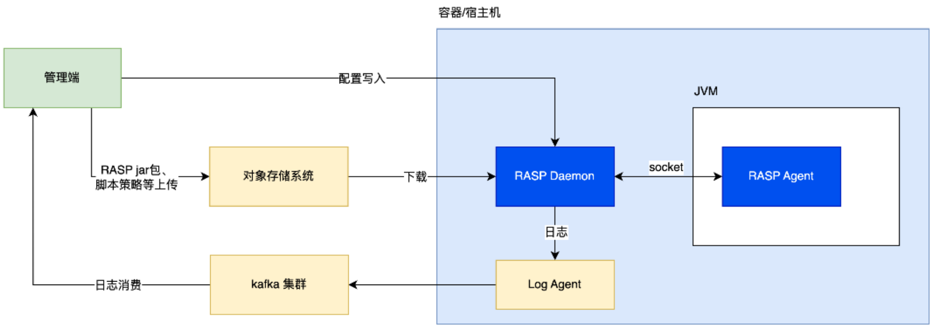


图 7　RASP 的配置分发流程

解决方案

灰度部署方式和复杂场景的兼容

RASP 启动方式

传统的 RASP 直接修改 JVM 启动参数增加 RASP 的 Java Agent 参数，即 `premain` 方式。而美团的 RASP 在最初只支持运行注入 `agentmain` 方式，不支持 `premain`。原因主要是下面的 2 个方面：

1. 在 RASP 项目建立时，公司的机器节点数量已经有几十万规模了，业务先行，安全补位。已经面临风险，需要尽快实现安全能力覆盖。
2. 早期公司内部服务发布平台不完善，有多个发布系统，并且每个业务线的发布脚本不统一，统一控制的力度弱。

综合业务现状与安全诉求，比较符合技术选型的是 `agentmain` 机制。无需业务改动，也不依赖统一的代码发布平台，做到安全部门可控的能力覆盖。

经过多年部署，RASP 已经覆盖大部分业务，具备相应安全能力。但也逐步遇到业务抱怨 RASP 注入带来的性能抖动问题。随着公司基础组件建设，也逐步统一了代码发布系统，在 JAVA 类服务的管控上有了统一的控制入口。同时，IDC 内服务形态逐渐从 VM 虚拟机演化到容器，RASP 的服务环境也与以往不一样。

当下主要矛盾发生变化，业务形态发生变化，支持 premain 的技术方案迫在眉睫。RASP 联合服务发布与镜像团队在拉起服务之前将 RASP 的 Java Agent 以环境变量的方式设置到服务启动脚本的上下文中。下面为部署脚本中关于 RASP 环境变量的设置片段。

```
// 前置检查 ...

// 增加环境变量
if [[ $RASP_SWITCH=="ON" ]];then
     JAVA_TOOL_OPTIONS="$JAVA_TOOL_OPTIONS -javaagent:rasp-premain.jar"
&& export JAVA_TOOL_OPTIONS
fi

// 启动 Java 进程 ...
```

配置分发方案

在 RASP 升级新版本时，为尽可能地提高稳定性，需要按照一定策略进行灰度升级。

1. 公司内部分为测试和生产等多种环境，并且测试环境服务器数量为万级别，RASP 需先在线下环境稳定运行足够时候后再开始线上环境灰度；
2. 服务按照重要性（或者环境复杂性）从低到高划分为普通服务、重要服务、高优服务 3 个类别，依次进行。
3. 每一个服务需要再按照主机数量的百分比进行灰度，一个服务下的主机不能同时进行 RASP 的升级，需按照 10%、30%、50%、100% 的比例灰度。

Web/JDK 版本识别与准入

RASP Daemon（golang 语言）通过内核进程事件，感知新进程。再识别进程的 cmdLine、JDK、Tomcat、Jetty、Spring Boot 等的关键 jar 包，解析出 JDK 版本、Web 类型和版本。对于已经兼容的服务可以开启注入，对于无法识别或者与 RASP 不兼容的服务关闭注入（ES、Jetty 等个别版本），最大程度的减少对业务的影响。

组件的兼容性

JDK 兼容性：美团 RASP 除了使用 ASM 包之外基本上不使用第三方组件，降低供应链攻击，同时减少对不同版本 JDK 的专有特性依赖，对于 JDK 的代码也尽可能的本地化到 RASP 工程中，屏蔽 JDK 的版本差异性。

Java Agent 兼容性：公司有多种 Java Agent 包括性能诊断，安全扫描、动态调试、流量录制、热部署、链路追踪等约十多种，这些工具实现原理都是基于 Instrument。 冲突主要在还是在字节码修改上，例如 RASP 与 jdwp 的兼容上，最初版本的 RASP 在业务类中增加方法数量，当用户开启远程 debug 时，本地代码的方法数量与远程不一样，导致 JVM 崩溃。Java Agent 应该遵循的规范：

字节码的修改应该遵循下面的基本原则：不允许新增、修改和删除成员变量 ；不允许新增和删除方法 ；不允许修改方法签名（来源于：Java 字节码规范）;

Java Agent 的 jar 包应该采用自定义类加载加载，依赖包名称前缀替换等方式，避免与其他 Java Agent 和业务依赖的冲突；

与其他 Java Agent 约定，在类查找遍历修改时排除其他的 Java Agent 的包名称，避免相互引用；

对于热部署等 Java Agent，由于它不遵循字节码修改的基本规范，很遗憾，目前无法兼容，只能排除关闭注入；

RASP 的运行时注入与更新

运行时注入方式解决了 RASP 的首次注入不依赖业务重启服务的问题，但是随着部署场景的增加，不可避免的要对 RASP 进行更新迭代，如何升级成为一个让人头疼的问题。于是更新也不依赖业务重启，成为一个需要解决的最大问题。

插件热更新是一项具有挑战性的技术，也是 RASP 建设初期要求具备的核心特征之一。由于美团拥有上百万个 Java 服务节点，一般的 Java Agent 安装和升级都需要重启 Java 进程，对于如此庞大规模的服务来说，这并非易事。在超大规模下，如果依赖业务重新发布的方式来使 RASP 生效，需要等待所有的服务重启一遍。**RASP 项目没有权限重启业务**。因此，对于 RASP 来说，插件热更新是至关重要的。

在最初的版本中，当 RASP 注入到业务中后，如果需要更新功能（如修改策略或 hook 点），仍然需要重新启动 Java 进程。如果业务不重启，之前版本的 RASP 会残留在进程中无法卸载，而新版本需要兼容这些无法卸载的部分。这导致线上存在多个不同版本的 RASP，不同版本之间的兼容性几乎无法实现，这种方式是行不通的。

因此，RASP 借鉴了 Tomcat 的类加载器架构，将功能分为两类：第一类是需要频繁迭代的功能，如 hook 点、资源监控、检测引擎、通信等；第二类是几乎不需要改动的部分，如插件加载和初始化部分。将第一类功能抽取出来，形成一个单独的插件包（RASP Plugin），插件包由自定义类加载器加载，使得这部分具备运行时更新的能力。而 RASP Agent 引导包仅保留几个类，负责初始化插件 jar 包。下图 8 展示了拆分前后的对比：

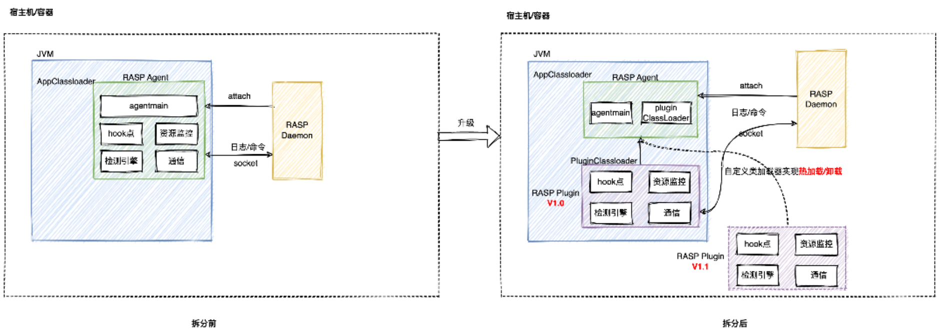


图 8　mt-rasp jar 包拆分前后对比

对于拆分后的架构，首次注入 RASP Agent 加载 V1.0 的插件，在需要对插件进行更新时，清除 RASP PluginV1.0 对象的引用和 `PluginClassLoader` 对象，然后创建新的 `PluginClassLoader` 实例重新加载并初始化 V1.1 版本插件，从而实现插件的卸载与热更新。上面拆分方案实现依靠自定义 RASP 类加载器，RASP 的类加载器层次结构（agentmain）如下图 9 所示：

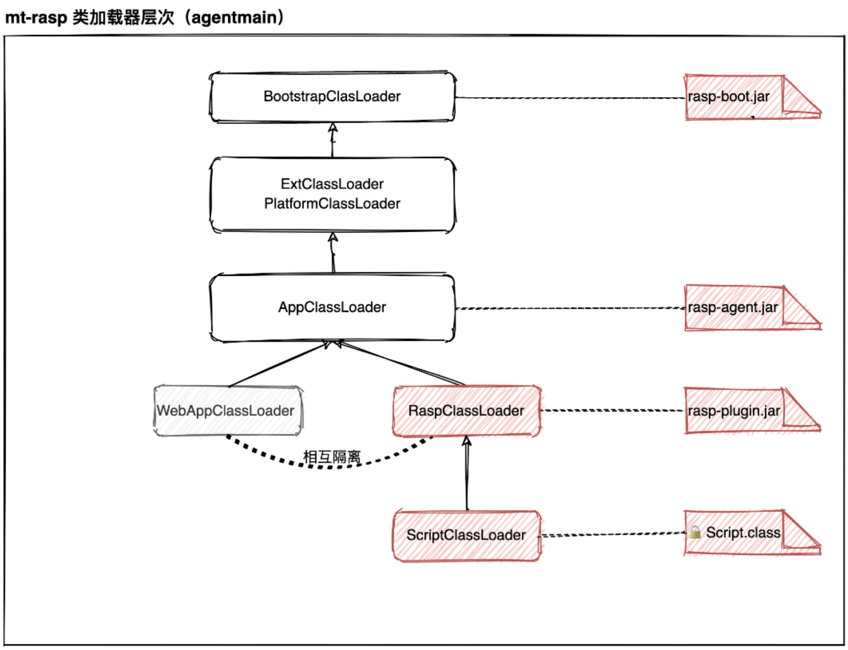


图 9　RASP 的类加载器层次结构

从顶层类加载器开始依次说明 RASP 包的功能和所属的类加载器。

- rasp-boot.jar：定义全局变量，能够被所有类访问到，使用 `Bootstrap-ClasLoader` 加载；
- rasp-agent.jar ：标准的 Java Agent 入口类，定义了 `agentmain/premain` 等 Agent 初始方法、加载 plugin 并初始化，使用 `AppClassLoader` 加载；
- rasp-plugin.jar ：RASP 核心实现，包括 hook 点、检测逻辑、资源监控等功能，使用自定义类加载器 `RaspClassLoader` 加载；
- Script.class ：定义检测逻辑，父加载器为 `RaspClassLoader`，使得脚本类能够访问 rasp-plugin.jar 中的类，使用自定义类加载器 `ScriptClass-Loader` 加载，并且脚本在磁盘加密在运行时解密。

premain & agentmain 两种方式兼顾

`agentmain` 和 `premain` 方是 Java Agent 的两种启动方式，`agentmain` 在 Java 进程启动后加载，而 `premain` 在 Java 进程启动前加载。由于启动时机不一样，带来的差异主要有 `agentmain` 更新加载更加灵活，但是字节码修改时存在性能问题，特别是对性能比较敏感的服务；而 `premain` 需要将 javaagent 参数加入到 JVM 启动命令行中，完全依赖业务启动，不太灵活，但是性能上比较稳定。美团 RASP 采用 `agentmain` 与 `premain` 结合方式，平衡灵活性与性能。原则上 `premain` 逻辑尽可能的简单，避免频繁的迭代与升级。

premain 一期方案

RASP 在加载时，Java 进程的 CPU 会短暂的升高甚至打满，并且 CPU 核数越少，升高越明显持续时间越长。根因是 **Java Agent 首次加载时会触发 JVM 中的 code cache 区域清零机制**（可以认为是 JDK 的 bug），大量热点代码的编译导致 JIT 编译线程将 CPU 打满，并且这种现象在 CPU 核数低于 4 核时表现尤为明显。

```
Manifest-Version: 1.0
Premain-Class: com.meituan.rasp.agent.RaspAgent
Agent-Class: com.meituan.rasp.agent.RaspAgent
```

```
Can-Redefine-Classes: true
Can-Retransform-Classes: true
Can-Set-Native-Method-Prefix: true
```

为了解决运行时 CPU 飙高问题，我们引入空的 `premain` 包（premain v1.0）（仅开启上面的字节码转换的开关 `Can-Redefine-Classes`，无任何逻辑，也不修改字节码），在应用启动前加载，该方案取得较大优化效果。因为无任何代码，代码兼容性风险极小（并不是没有），因此能快速上线解决 CPU 飙高问题。以某个业务的主机为例子，在优化前后的 `cpu.busy` 指标如下图 10 所示（注入前后 10 分钟）。

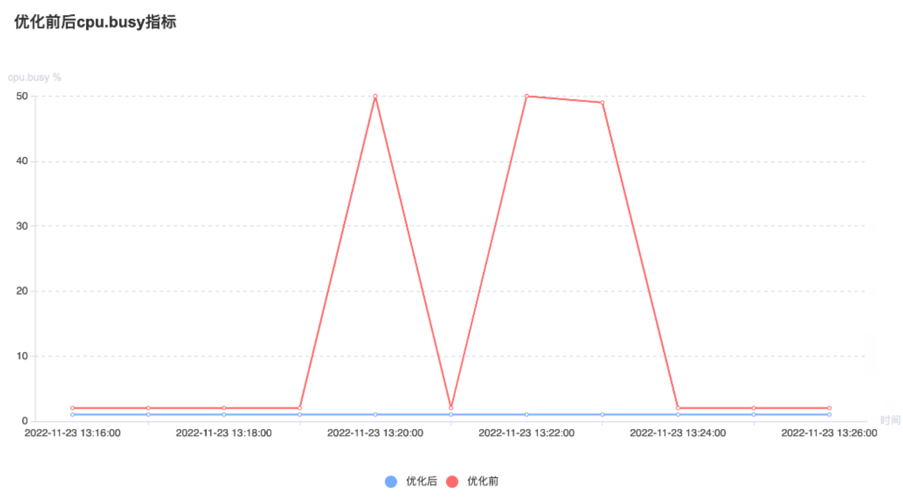


图 10　cpu.busy 指标优化前后对比

图 10 中红色为优化前的 `cpu.busy` 指标，优化前即使注入前系统负载很低（4 核 8G，cpu.busy <2%），注入瞬间 CPU 依然飙升很高（50%）；蓝色为优化后的 `cpu.busy` 指标，优化后 cpu.busy 曲线较平滑，无明显尖刺。

premain 二期方案

采用 `premain` 一期方案的原因是代码足够简单，几乎没有兼容性问题，因此能够快速大规模部署解决棘手的 cpu 抖动问题，上线效果较好。但是大部分服务虽然 CPU

不飙高了，但是还有少部分的服务 TP9999 指标依然影响较大。

修改字节码 TP9999 线升高分析

`premain` 一期方案主要用来快速解决 `cpu.busy` 抖动问题，但是对于性能比较敏感（如 tp9999 < 50ms）的业务，运行时字节码修改不可避免的造成 STW，从而导致 TP9999 线升高。因为字节码修改时需要进入 JVM 的 safepoint，执行字节码转换的方法 VM_RedefineClasses::doit 会导致应用短暂地暂停响应，这部分代码执行慢就会影响应用 `TP9999` 。大批量修改字节码测得火焰图如下图 11 所示：

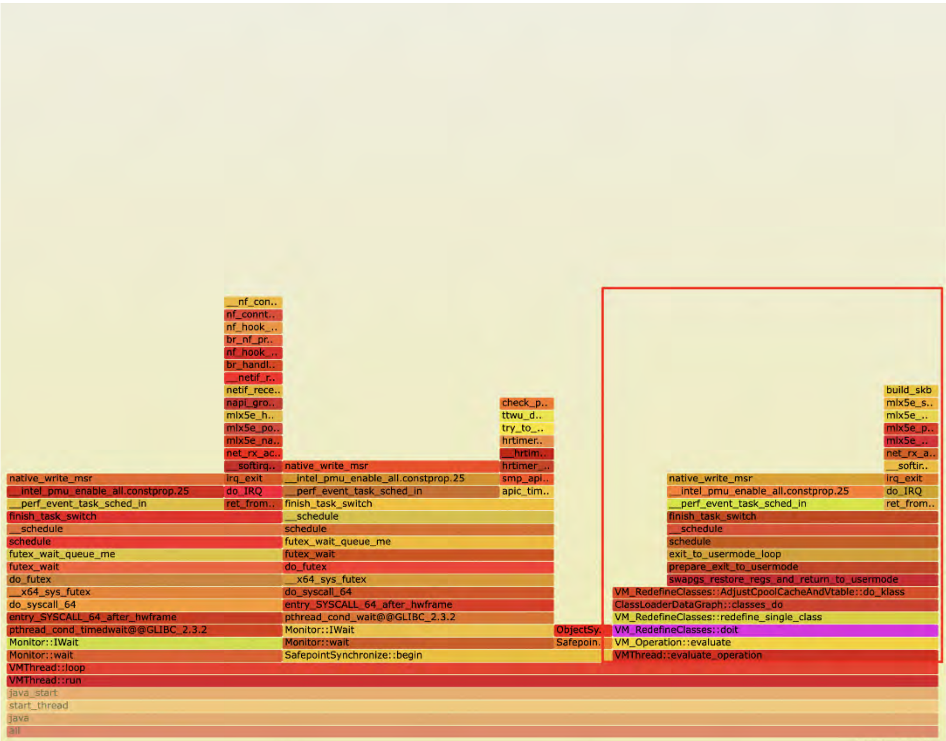


图 11　批量字节码转换时的 Java 进程火焰图

从火焰图 11 看出，`RedefineClasses` 耗时主要在 `VM_RedefineClasses::AdjustCpoolCacheAndVtable::do_klass` ，hotspot JDK 官方也有类似的 issue 。转换一个类的主要耗时在 `redefine_single_class` 方法，初步测试耗时占比在

40% ~ 80% 之间，并且一次转换类越多，占 STW 总耗时越大。

类转换 STW 时间与服务负载（QPS）关系

在同一服务（硬件配置 8 核 8G）测试了修改类的个数、服务负载和 STW 时间的关系如下图 12 所示：

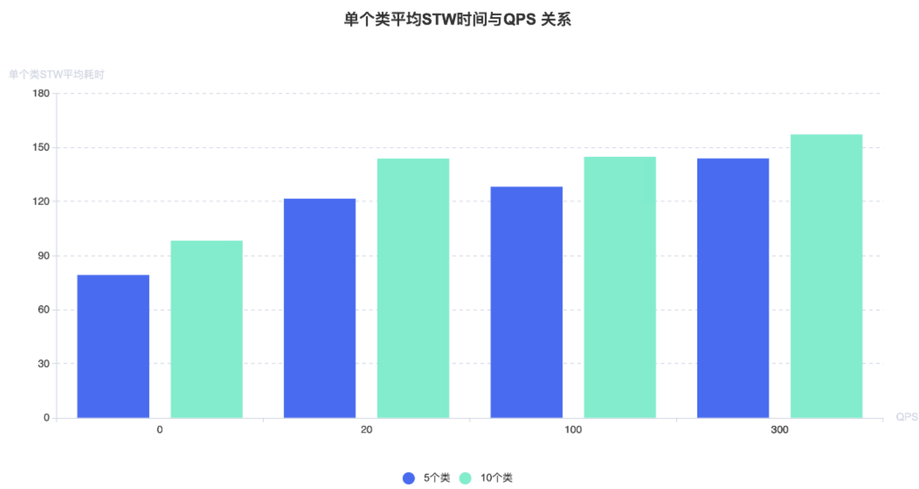


图 12　类转换 STW 时间与服务 QPS、转换类数量的关系

从上面数据可以对比看出：

1. 业务请求 QPS 为 0 时，STW 时间并不为 0；
2. QPS 为 20 时，一次转换的类越多，单个类的 STW 耗时越长；
3. 转换相同数量的类，随着 QPS 的增加，STW 时间有增加，但是不明显；

从上面的分析可以看出，修改字节码无法避免的产生 STW(当然，优化这部分 JDK 代码理论上是可以实现的，但是技术难度较高，短期内无法解决)，因此只能从**规避的角度**出发来解决。原则上只要保证字节码修改时没有请求即可。

一种可行的方案是将字节码转换的逻辑前移到 JVM 启动前（即业务没有流量或者主

动摘除流量），并且尽量避免有请求时大批量的回滚 / 修改字节码，能够在一定程度上避开或者缓解 STW 影响业务请求响应时间。

premain 修改字节码

对于高频率调用的方法如 http body 参数读取、`sql.execute` 等，使用 `premain` 修改字节码插入 RASP 检测逻辑，`premain` agent 做到轻量级。对 RASP 的架构做出相应修改，新增 `rasp-premain.jar`，让服务启动前进行加载并初始化，将字节码的转换逻辑前置到启动时，如下图 13 所示，蓝色 jar 包为新增的 `rasp-premain.jar`。

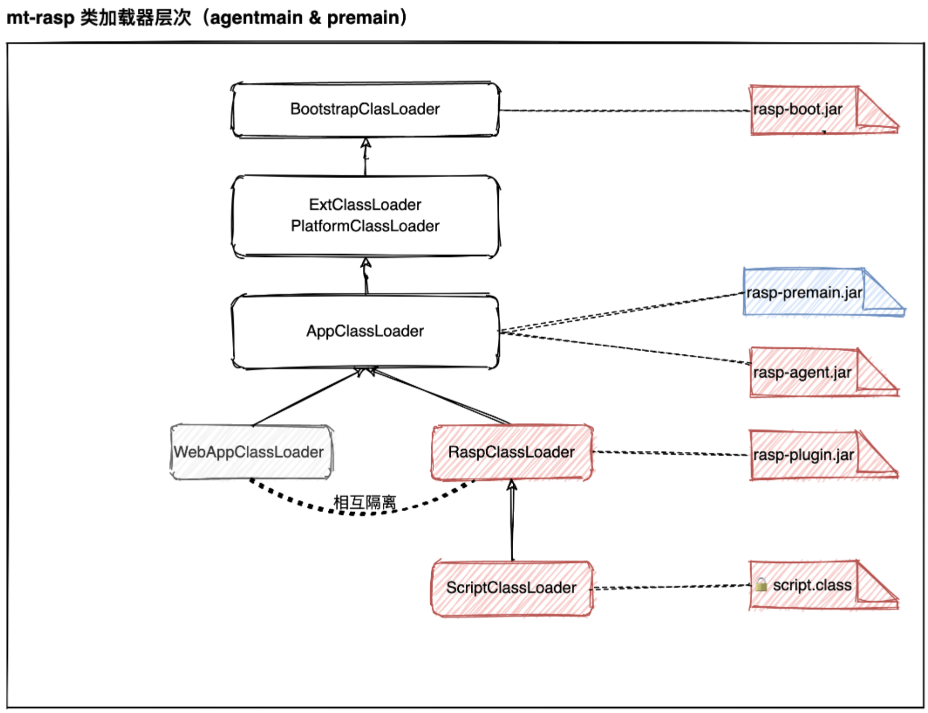


图 13　RASP 类加载器增加 premain agent

为了最大程度复用之前的系统架构，`premain` 加载后虽然字节码已经被转换，但 RASP 的功能在逻辑上是关闭的，需要等到 `agentmain` 注入之后打开检测开关。

`premain` 只做字节码转换，没有日志和通信等功能，不能单独工作。如图 14 所示，优化前后 TP9999 指标有较明显改善。

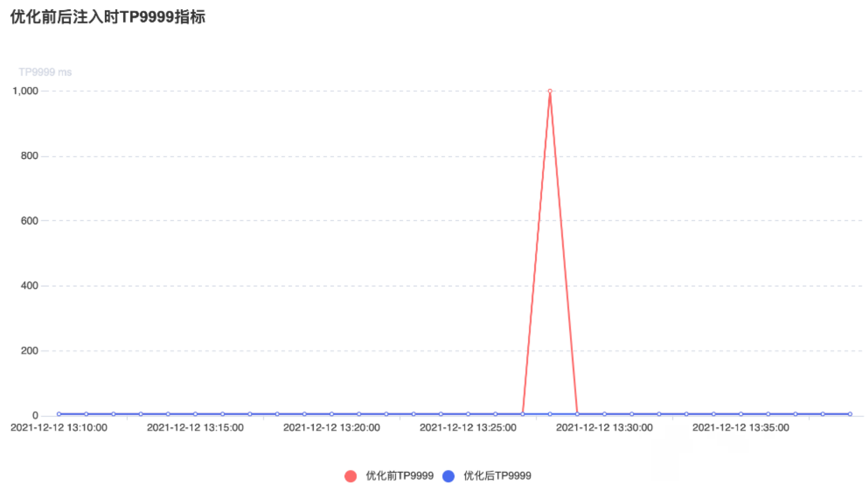


图 14　优化前后注入时 TP9999 指标

运行时性能优化与整体指标

- **rasp 加载之后的流量预热**

在 RASP 的流量控制层增加对流量的计数，RASP 初次接入流量时控制接入流量的比例（如 1%），使得业务业务流量能够预热 RASP 检测逻辑，预热时间或者次数达到设定的阈值后，再开启 100% 的流量检测。

- **软降级与逻辑开关**

业务负载较高的场景（CPU 飙高、hook 逻辑执行严重超时等），为了避免 RASP 检测逻辑加剧性能恶化，RASP 采样软降级措施，关闭对应 hook 类的逻辑开关，使得部分流量不执行检测逻辑。如果性能进一步恶化，RASP 运行模式降级为观察上报模式，待系统资源检恢复正常过后，资源监测通过后自动恢复到检测阻断模式。

- **延时回滚字节码**

RASP 更新插件代码时，需要将 plugin 的全部对象置空，否则会有内存泄漏问题，特别是元空间的内存泄漏，将导致业务将运行越来越慢，直到停止运行。从前面的 STW 时间结论来看，运行时的字节码回滚（和修改机制相同）也会产生 STW，因此 RASP 将 hook 代码的逻辑开关关闭后，字节码依然留在业务类中，在清理完各种对象引用关系后，依然能够卸载 plugin 插件。

监控体系建设

全局维度的监控指标：

- 主机注入覆盖率大盘；
- coredump 总数；
- 高峰期字节码的修改数量；
- 熔断超时数量和比例；

单机维度指标：从业务层面到系统层面如下（列举部分）

- 业务层面：检测引擎执行耗时、TP9999、请求出错率等
- JVM 层面：堆内存、元空间 / 永久代、线程死锁、插件加载次数限制、GC、STW 耗时、字节码转换等
- 进程层面：Java 进程 CPU、内存、coredump、守护进程状态等
- 系统维度：系统 CPU、系统内存、系统磁盘空间、网络等

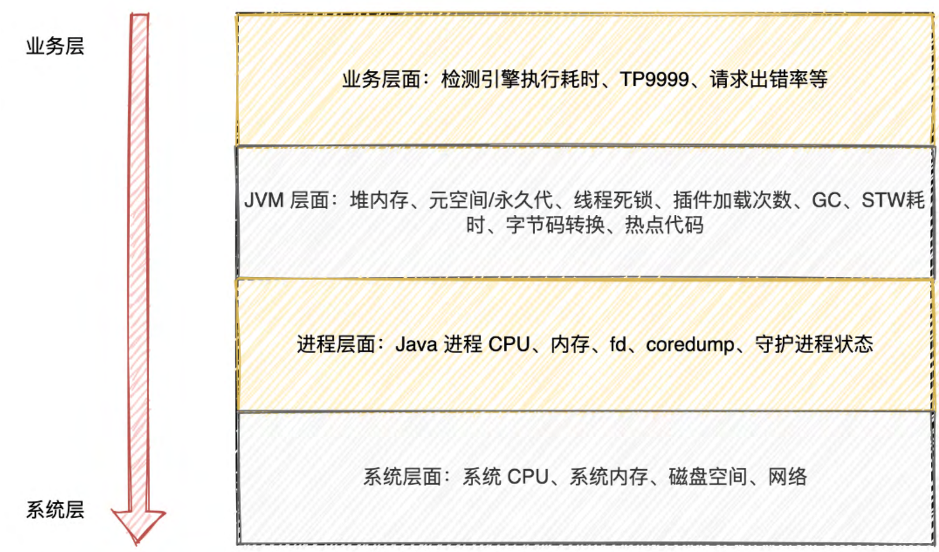


图 15 RASP 监控的指标分布

系统指标和进程指标对于 Golang 来说很容易获取，相关 api 较多。这里仅以 JVM 指标元空间使用率（MetaSpace）的检测为例子说明。RASP Daemon 执行 attach 获取目前 JVM 的最大元空间（`MaxMetaSpaceSize`）指标，然后读取 `/tmp/hsperfdata_${user}/pid` 文件解析元空间的占用（`usedMetaspaceSize` 参数在 jvm 里面是 `sun.gc.metaspace.used`），计算出元空间的占用比例和剩余空间，当剩余空间不足时，禁止 RASP Agent 注入，防止 RASP 成为压垮业务的最后一根稻草。

性能影响

测试配置 : 8 核 /8G/150G

压力：QPS 梯度 100，持续 120s，稳定施压 120s

- cpu.busy

表 1　注入前后的 `cpu.busy` 指标

	基准数据（不加载RASP）	注入数据（加载RASP）	CPU指标增量值
QPS=20	3.47%	4.18%	0.71%
QPS=100	11.70%	11.76%	0.06%
QPS=200	20.95%	21.05%	0.15%
QPS=300	32.12%	32.78%	0.66%
QPS=400	41.23%	44.2%	2.97%
QPS=500	52.78%	56.5%	3.73%
最大 QPS	620	587.8	–
拟合方程	拟合方程：y = 0.103x + 1.203 拟合度：0.999	拟合方程：y = 0.109x + 0.827 拟合度：0.998	

`cpu.busy` 绝对值增加：0.06% ~ 3.73%，整体性能与开源的 RASP 相当

- **堆内存增加**

QPS 超过 350 时系统 cpu 达到 35%，触发弹性扩容，QPS 压测到 350 可以测出最大内存损耗。

表 2　注入前后的内存增加值

	最小值	平均值	最大值
基准 (QPS=350)	422.38MB	638.34MB	3.10GB
注入 (QPS=350)	457.69MB	821.59MB	3.30GB
差值	32MB	183MB	0.2GB

注入前后对比，压测到系统弹性扩容的最大 QPS，最大堆内存增加约 200M，整体性能与开源的 RASP 相当

- **元空间增加**

元空间 / 永久代增加 2MB，优于开源 RASP 产品

- **请求耗时**

当前请求耗时控制在 5ms 内，优于开源 RASP 产品

漏洞检测

支持的漏洞类型

经过近多年的研发迭代，目前具备的漏洞检测类型如下，基本覆盖常见漏洞（部分）：命令执行（支持 Native 方法）、SQL 注入、文件访问、反序列化攻击、JNDI、表达式等等。

实时检测与阻断

不同语言实现的脚本性能比较

开源方案中采用了 JavaScript 引擎作为实现方式，JS 脚本可以被 Java、PHP 和 C++ 等各种语言兼容，具备较强的通用性。但是经过测试，与原生 Java 相比，这些方案在性能上存在较大的差距。尽管 JavaScript 引擎具有不同语言通用性的大优势，但在执行性能方面并不满足高性能场景下 RASP 的需求。在美团，相比于性能，检测引擎的语言通用性并不是最重要的考虑因素。下面简单对比一下 JavaScript 和 Java 实现的检测引擎的性能。因为检测脚本主要涉及字符串的各种操作，我们选择了字符串累加的 for 循环作为测试场景。

```
// java
c+='c'
// javascript
c=c+'c'
```

经过测试，我们发现 Java 在执行这种字符串操作的性能方面表现更好。Java 作为一种编译型语言，具有较高的执行效率和优化能力。它可以通过使用 `StringBuilder` 等高效的字符串操作类来提高性能。相比之下，JavaScript 作为一种解释型语言，执行效率相对较低。因此，在高性能场景下，使用 Java 实现的检测引擎往往能够更好地满足需求。尽管 JavaScript 引擎具有通用性，但在性能要求较高的场景下，**选**

择使用 Java 实现的原生检测引擎更为合适。

表 3　10 万量级的 for 循环中跑出结果如下（单位 ms）

	JavaScript	Java
平均执行耗时	585	6.5

可以看出 Java 语言实现的检测引擎，性能上具备优越性。美团 RASP 使用 Java 语言构建检测引擎，能够满足性能上的需求。

检测脚本的实现

在 RASP Plugin 中定义了检测脚本需要实现的接口，脚本的实现类由 RASP Daemon 下载到磁盘上；RASP Agent 定时检测脚本文件是否更新，如果脚本更新，使用新的类加载器加载磁盘上的 class 文件，并创建实例。

阻断与热修复

在 RASP 中，通常会在 hook 方法的执行之前（before）、返回（return）和抛出异常处（throw）增加检测逻辑。RASP 通过使用 ASM 字节码框架，在方法的 before、return 和 throw 处织入检测逻辑的字节码（下图 16 黄色框）。

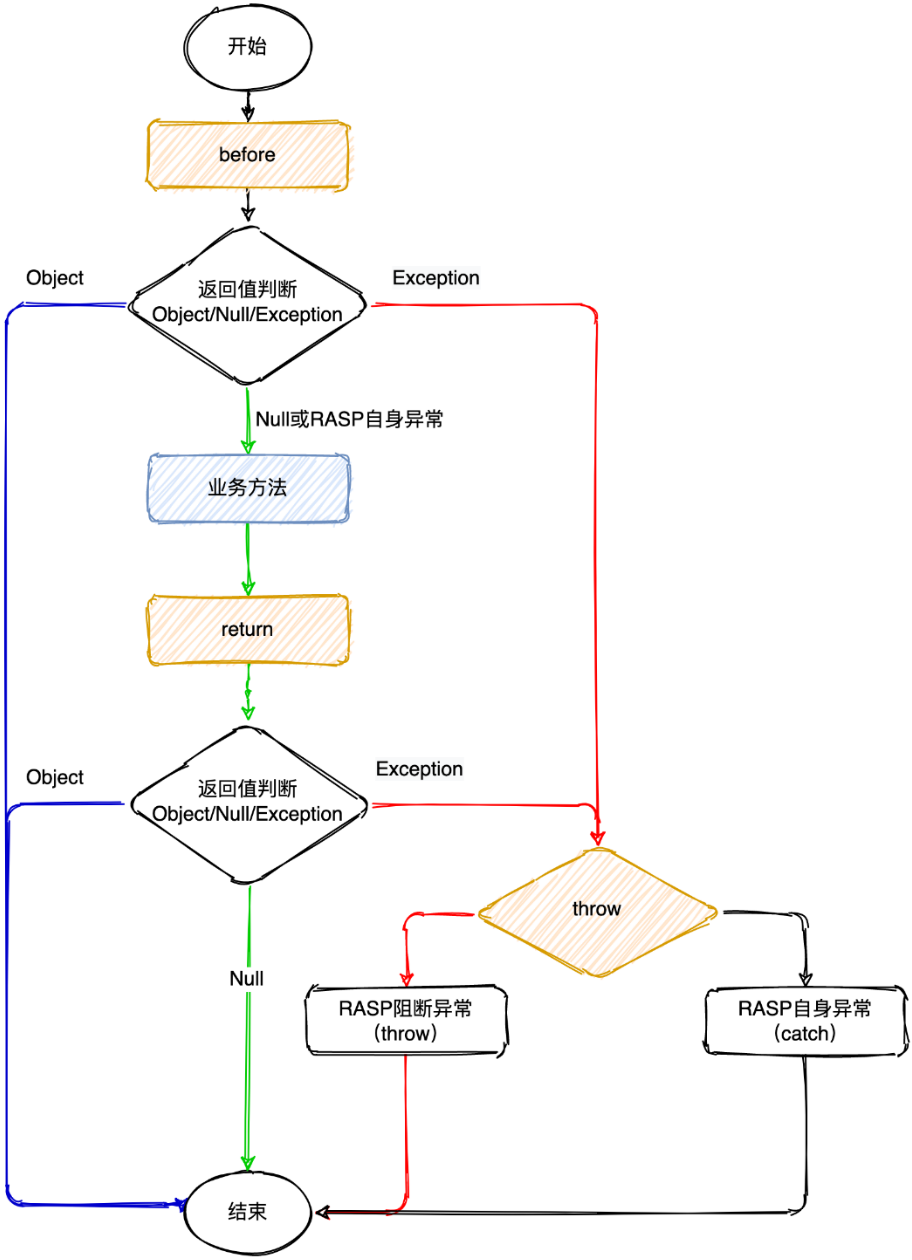


图 16 RASP 阻断热修复控制流程

这里以在方法返回之前增加 hook 逻辑为例子说明阻断 / 热修复的流程：

1. 字节码插桩：使用 ASM 工具识别方法中的返回指令如（return、areturn 等），在返回指令之前插入 RASP 的检测方法的字节码，使用 `instrument` 的 `restransform` api 将修改后的字节码替换原来的字节码。
2. 运行时检测：当检测引擎返回阻断异常对象时，方法的异常处理抛出阻断异常，终止方法的执行（上图 16 红色箭头的流程）；当检测引擎返回对象时，提前返回指定的对象，修改返回的返回值（上图 16 中蓝色箭头的流程）；返回 Null，表示既不阻断也不返回对象（上图 16 绿色箭头的流程），不改变当前方法的执行流程和返回对象。

热修复与阻断的区别在于热修复返回的是一个对象，这个对象是修复后的正确的对象。

成果

美团 RASP 经过多年的建设，在覆盖对象、部署方式、性能优化、兼容性和安全策略等多个方面逐步迭代，现在已覆盖绝大多数 Java 服务，支持众多 web 容器部署，基本覆盖常见的安全漏洞，整体覆盖率上达到了较高水位，并且多次检测出海量的漏洞攻击，成为美团 IDC 基础安全纵深防御体系中最重要的安全能力。

总结

本文主要介绍了美团 RASP 在研发过程中遇到的问题和解决方案。首先介绍了 RASP 的痛点问题，包括业务场景复杂、升级变更难、对业务性能影响大和缺少监控等。对于 RASP 的升级问题，引入了插件热更新的技术，可以在不重启 Java 进程的情况下，即时地更新 RASP 的功能。

为了降低对业务性能的影响，介绍了采取的优化措施，包括低峰期注入、启动时流量预热、软降级与逻辑开关以及插件卸载时不回滚字节码等关键技术。然后介绍了 RASP 的监控体系建设，包括监控指标的定义和收集。最后介绍了 RASP 的性能与灰度策略，通过对性能损耗的测试和分析，可以看出 RASP 对 CPU 和 QPS 的影响

较小。在灰度策略方面，RASP 结合了业务形态，特性影响等，选择合适的验证机制和测试方法。

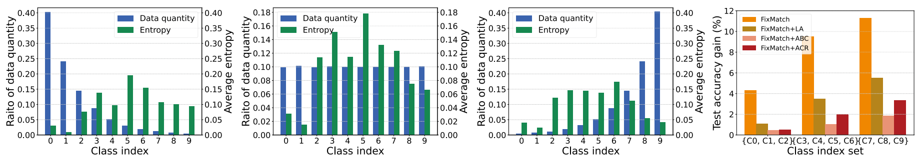
后续规划

- **新型容器形态支持**：美团 IDC 形态中，逐步从 VM、富容器过度到轻容器，未来轻容器会越来越多，RASP 的管控机制、容器隔离机制，都是未来 RASP 的挑战；
- **低打扰无感接入**：宿主业务的低打扰，注入性能影响，小众场景的覆盖，依旧是 RASP 的核心重点，让业务无感、自动、默认接入 RASP，提升整体 IDC 防御水位；
- **管控、监控自动化**：管控端的配置下发依赖链路较多、流程较长，配置变更成本风险高，优化为更高效、更实时、更准确的机制；

顶会论文

BEM: Balanced and Entropy-based Mix for Long-Tailed Semi-Supervised Learning

Hongwei Zheng[1,2], Linyuan Zhou[1], Han Li[2], Jinming Su[1], Xiaoming Wei[1], Xiaoming Xu[1] †
[1] Meituan [2] Shanghai Jiao Tong University
{zhenghongwei04, zhoulinyuan, sujinming, weixiaoming, xuxiaoming04}@meituan.com
{qingshi9974}@sjtu.edu.cn



(a) Consistent class distribution (b) Uniform class distribution (c) Reversed class distribution (d) Test accuracy gain

Figure 1. Experimental results on CIFAR10-LT [35]. **(a)-(c):** Class distribution of unlabeled data quantity and entropy for three typical settings, which have the same labeled data quantity distribution but differ in unlabeled ones. Both the data quantity and entropy are the statistical averages within one epoch after model convergence. Unexpected discrepancies are observed across all settings between the distribution of data quantity and entropy, particularly for head and tail classes. Notably, classes 3-6 exhibit the highest entropy, indicating greater uncertainty. **(d):** Test accuracy gain brought by BEM for various LTSSL frameworks in consistent setting.

Abstract

Data mixing methods play a crucial role in semi-supervised learning (SSL), but their application is unexplored in long-tailed semi-supervised learning (LTSSL). The primary reason is that the in-batch mixing manner fails to address class imbalance. Furthermore, existing LTSSL methods mainly focus on re-balancing data quantity but ignore class-wise uncertainty, which is also vital for class balance. For instance, some classes with sufficient samples might still exhibit high uncertainty due to indistinguishable features. To this end, this paper introduces the Balanced and Entropy-based Mix (BEM), a pioneering mixing approach to re-balance the class distribution of both data quantity and uncertainty. Specifically, we first propose a class balanced mix bank to store data of each class for mixing. This bank samples data based on the estimated quantity distribution, thus re-balancing data quantity. Then, we present an entropy-based learning approach to re-balance class-wise uncertainty, including entropy-based sampling strategy, entropy-based selection module, and entropy-based class balanced loss. Our BEM first leverages data mixing for improving LTSSL, and it can also serve as a complement to the existing re-balancing methods. Experimental results show that BEM significantly enhances various LTSSL frameworks and achieves state-of-the-art performances across multiple benchmarks.

1. Introduction

Semi-supervised learning (SSL) capitalizes on unlabeled data to reduce the cost of data labeling and boost the performance of models [22, 36, 47, 54, 61]. The general paradigm of most approaches is to randomly generate two views of an image with various augmentation methods and then use the output of one as the pseudo label to supervise the other [12, 55, 72]. As a simple and effective augmentation technique introduced in supervised learning [7, 26, 30, 62], data mixing is widely used in SSL algorithms [8, 9, 63], further enhancing model generalization and performance.

However, most existing SSL algorithms assume a balanced dataset, ignoring the real-world prevalence of long-tailed class distributions [6, 16, 27, 28, 74, 75]. To deal with this class imbalance scenario, various long-tailed semi-supervised learning (LTSSL) methods have been proposed, such as re-sampling [65], logit alignment [45, 66], and pseudo label alignment [39, 46]. Nevertheless, data mixing is rarely explored in LTSSL. The primary reason is that

† Corresponding Author

existing data mixing methods (*e.g.* MixUp [73], CutMix [69], and SaliencyMix [58]) often perform random mixing within a batch. Consequently, the infrequent tail classes may not be adequately sampled when the batch size is small. This hinders a balanced class distribution, which is crucial for LTSSL.

To make data mixing suitable for LTSSL, let us consider two key questions: **i) How can we apply data mixing to effectively re-balance the data quantity for each class? ii) Is it sufficient to solely focus on re-balancing data quantity to achieve class balance?** For the second question, we notice that previous LTSSL methods mainly focus on addressing the issue of long-tailed class distribution in terms of data quantity. These methods ignore the fact that class performance also depends on class-wise uncertainty [40, 42, 43, 52], which is associated with the training difficulty for each class. For instance, some classes with sufficient samples may still encounter high training difficulty due to the high uncertainty induced by indistinguishable features. As shown in Fig. 1 (a)-(c), we quantify the uncertainty by entropy [43] and compare this class-wise entropy with data quantity under three typical settings [66]. The results reveal a **significant disparity** between the class distribution of entropy and data quantity across all settings. This finding emphasizes the limitation of solely re-balancing data quantity, as it does not consider classes with high uncertainty, ultimately limiting performance improvement. Thus, it is crucial to also address the re-balancing of class-wise uncertainty, *i.e.* entropy.

To tackle the above problems, this paper presents a novel data mixing paradigm, called Balanced and Entropy-based Mix (BEM), for LTSSL. Specifically, we first introduce a simple mixing strategy, named as CamMix, which has a strong localization capability to avoid redundant areas for mixing. Then, we establish a class balanced mix bank (CBMB) to store and sample class-wise data for mixing. The sampling function follows the estimated class distribution of data quantity and we adopt the effective number [16] to represent the realistic data quantity of each class. Our CamMix incorporated CBMB can effectively re-balance the class-wise data quantity in an end-to-end optimized manner, which can not be achieved by the re-sampling methods [10, 32, 65, 80] with the complex training procedures.

Further, we present a novel entropy-based learning approach to re-balance class-wise uncertainty. Entropy-based sampling strategy (ESS) integrates class-wise entropy into the quantity-based sampling function. In addition, entropy-based selection module (ESM) adaptively determines the sampled data ratio between labeled and unlabeled data during mixing to manage the trade-off between guiding high-uncertainty unlabeled data [5, 64] with confident labeled data and maximizing the utilization of unlabeled data. Finally, we incorporate the class balanced loss [16] with class-wise entropy to form entropy-based class balanced (ECB) loss.

We highlight that our BEM is the first method that leverages data mixing to enhance LTSSL. Our results demonstrate that BEM can effectively complement existing re-balancing methods by boosting their performance across several benchmarks. As shown in Fig. 1 (d), our method enhances FixMatch [55], FixMatch+LA [45], FixMatch+ABC [39], FixMatch+ACR [66], achieving to 11.8%, 4.4%, 1.4% and 2.5% average gains on test accuracy, respectively. Additionally, BEM proves to be a versatile framework, performing well across different data distributions, diverse datasets, and various SSL learners.

2. Related Work

Data mixing. MixUp [73] and CutMix [69] are typical data mixing methods used in various computer vision tasks. While performing mixing at element-wise and region-wise levels respectively, they share a common limitation of neglecting class content, thus introducing substantial redundant context irrelevant to class content [19, 48]. To achieve class balance in LTSSL, it is essential to ensure that the selected region for mixing contains related class content and avoids redundancy. SaliencyMix [58] alleviates this issue by using a saliency map to ensure that selected regions contain class content, but the resulting region is still too coarse to avoid numerous redundant areas. In our paper, CamMix achieves tighter localization of class regions to minimize redundant areas, which is particularly well-suited for LTSSL.

Data mixing in semi-supervised learning. Data mixing is crucial in SSL, enhancing model performance by creating diverse training samples. For instance, MixMatch [9] utilizes MixUp [73] as the data mixing technique to learn a robust model. ReMixMatch [8] adds distribution alignment and augmentation anchoring to the MixMatch framework. ICT [63] employs the mean teacher model and implements MixUp on unsupervised samples. Despite widely used in SSL algorithms, almost no methods in LTSSL apply data mixing. This is mainly due to the limitation of employing in-batch mixing, which fails to address the class imbalance problem. Our method stands out as the first to incorporate data mixing in LTSSL.

Long-tailed semi-supervised learning. LTSSL is gaining attention due to its real-world applicability. For example, CReST [65] refines the model by iteratively enriching the labeled set with high-quality pseudo labels in multiple rounds. ABC [39] uses an auxiliary balanced classifier, trained by down-sampling majority classes. DASO [46] mitigates class bias by adaptively blending linear and semantic pseudo labels. ACR [66], the current state-of-the-art method, proposes a dual-branch network and dynamic logit adjustment. However, none of these methods utilizes data mixing to further enhance their performance as in SSL. CoSSL [20] uses MixUp at the feature level for minority classes and decouples representation learning and classifier learning. However,

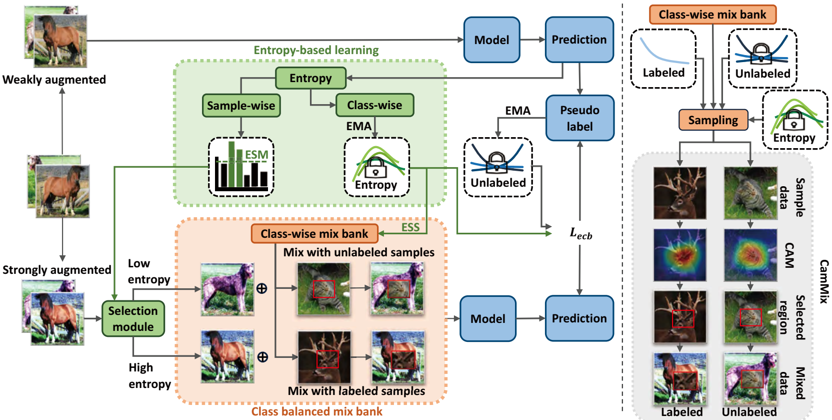


Figure 2. **Left**: The overview of Balanced and Entropy-based Mixing (BEM), incorporating with FixMatch [55] as an example in this figure. BEM consists of two sub-modules: class balanced mix bank (CBMB) and entropy-based learning (EL). CBMB re-balances data quantity through the proposed CamMix, guided by a class-balanced sampling function. EL further re-balances class-wise uncertainty using three techniques: entropy-based sampling strategy (ESS), entropy-based selection module (ESM) and entropy-based class balanced loss (L_{ecb}). **Right**: The sampling and CamMix process of BEM. The sampling process considers both the class distribution of data quantity and uncertainty, which are estimated on the fly. CamMix extracts the bounding box from the high response area of the CAM to form mixed data. (The lock icon denotes the unknown distribution that needs estimation, and the ⊕ icon denotes the process of CamMix.).

this feature mixing method assumes identical class distributions for labeled and unlabeled data and requires a complex training approach. Our study considers non-ideal class distributions and designs a simple image-level mixing method in an end-to-end training framework.

3. Preliminaries

Semi-supervised learning. In SSL, the training data consists of labeled data $X = \{(x_n, y_n)\}_{n=1}^N$ and unlabeled data $U = \{u_m\}_{m=1}^M$. Here, x_n and u_m are training samples, y_n is the ground truth, N and M denote the quantity of labeled data and unlabeled data, respectively. A representative framework of SSL is FixMatch [55], which utilizes unlabeled data with the *Weak and Strong Augmentation*. For an unlabeled sample u_m, it first takes a *weakly*-augmented version of u_m as the input of the model $f(\cdot)$ to compute the prediction. Then, it uses $q_m = \text{argmax}(f(A_w(u_m)))$ as one-hot pseudo label, while applying the prediction from a *strongly*-augmented of u_m to calculate the cross entropy loss L_u:

$$L_u = \sum_{m=1}^{B} \mathbb{I}(\max(f(A_w(u_m))) > \tau) \underbrace{\mathcal{H}(f(A_s(u_m)), q_m)}_{L_{cls}}, \tag{1}$$

where B denotes the batch size, $\mathcal{H}(\cdot)$ is cross entropy and L_{cls} denotes original classification loss term. $\mathbb{I}(\max(q_m) > \tau)$ is the mask to filter low-confidence pseudo label with a threshold of τ, abbreviated as $M_u(\cdot)$ in the following part. A_w and A_s denotes the *weak augmentation* (*e.g.*, random crop and flip) and *strong augmentation* (*e.g.*, RandAugment [15] and Cutout [18]), respectively.

Long-tailed semi-supervised learning. In LTSSL, a dataset with a long-tailed distribution is characterized by the minority of classes possessing a large number of samples, while the majority of classes contain only a few samples. Given C classes across the dataset, N_c represents the quantity of labeled data for class c. Without loss of generality, we assume that $N_1 \geq N_2 \geq \cdots \geq N_C$ and the imbalanced ratio is denoted by $\gamma_l = N_1/N_C$. Similarly, we can denote the quantity of unlabeled data as M_c for class c and the imbalanced ratio as $\gamma_u = \max_c M_c / \min_c M_c$.

4. Balanced and Entropy-based Mix (BEM)

The Balanced and Entropy-based Mix (BEM) is a plug-and-play method based on the existing SSL framework. Fig. 2 shows the overview of BEM, incorporating FixMatch [55] as an example. Specifically, our entropy-based learning (EL) takes the prediction of the weakly augmented samples as input to perform entropy-based sampling and selection. Then, strongly augmented samples are mixed with data from class balanced mix bank (CBMB) using our CamMix. Based on the estimated distribution of data quantity and uncertainty, we employ the entropy-based class balance (ECB) loss L_{ecb} to train the overall framework. Please refer to Appendix C for the pseudo-code of BEM.

4.1. CamMix

Most data mixing methods, such as MixUp [73] and CutMix [69], lack the localization ability for class re-balancing. Although SaliencyMix [58] has initial localization ability, it still tends to extract excessive redundant context. To this end, we propose CamMix to replace the saliency map of SaliencyMix with Class Activation Map (CAM) [79] to achieve more accurate localization. Specifically, we feed images into the prediction model (*i.e.* ResNet50 [23]) to generate the CAM, where the last layer of the third block of ResNet50 is used as the CAM layer. The resulting CAM is used to extract the largest connected region using a threshold of τ_c. Finally, we obtain the bounding box of this region and paste the corresponding patch onto the original image. The pseudo-code of CamMix can be found in the Appendix C.

4.2. Class Balanced Mix Bank (CBMB)

Previous in-batch data mixing methods used in SSL are limited to increasing the data quantity of tail classes, thus failing to re-balance the class distribution. To address this issue, we further propose a class balanced mix bank (CBMB) that stores samples for each class and adequately selects samples to be mixed based on a prior-based class-balancing rule. In essence, the more frequent a class, the more samples are used in the data mixing process. As noted in [16], there is overlap in the data, necessitating the use of the effective number E_c to measure the realistic class distribution of data quantity:

$$E_c = \frac{1-\beta^{N_c}}{1-\beta}, \tag{2}$$

where N_c represents the data quantity of class c, while the hyper-parameter β is set to 0.999 in our experiments.

The effective number of labeled data, denoted as E_c^x, can be obtained directly using Eq. 2. As the class distribution of unlabeled data is unknown, we estimate it using a simple yet effective approach. Specifically, at each iteration t, we obtain the class distribution of the pseudo label d_c^{ut} and update the class distribution of the entire unlabeled dataset d_c^u, using an Exponential Moving Average (EMA) approach once the training status stabilizes.

$$d_c^u \leftarrow \lambda_d d_c^u + (1-\lambda_d) d_c^{ut}, \tag{3}$$

where λ_d denotes the EMA weight. To obtain the effective number of unlabeled data for each class E_c^u, we substitute the class-wise data quantity $N_c^u = M d_c^u$ into Eq. 2, where M is the quantity of entire unlabeled dataset. Then, we obtain the effective number of total data for each class by $E_c = E_c^x + E_c^u$ and perform our CamMix using the initial sampling function as follows:

$$s_c = \frac{F_c}{\sum_{c=1}^{C} F_c}, \tag{4}$$

where $F_c = 1/E_c$ and s_c denotes the sampling probability for class c. By accurately estimating the class distribution of the dataset, we can enhance the precision of mixed data sampling. Our data mixing achieves class balance among training samples, equivalent to re-sampling during training.

4.3. Entropy-based Learning (EL)

In the previous section, we re-balance training samples to initially alleviate the long-tail distribution problem. However, class balance does not only depend on data quantity. Class-wise uncertainty, which can be quantified by entropy, is also vital for class performance as it reflects training difficulty. Thus, we propose an entropy-based learning approach to re-balance class-wise entropy, including entropy-based sampling strategy (ESS), entropy-based selection module (ESM) and entropy-based class balanced (ECB) loss.

Entropy-based Sampling Strategy. To consider class-wise uncertainty in the sampling process, we define the class-wise entropy e_c^x and e_c^u for the entire labeled and unlabeled dataset, and update them in EMA manner by using the average entropy e_c^{xt} and e_c^{ut} at each training iteration t as follows:

$$\begin{aligned} e_c^{xt} &= \frac{1}{N_c^t}\sum_{n=1}^{N_c^t}\sum_{c=1}^{C} -f_c(A_w(x_n))\log(f_c(A_w(x_n))) \\ e_c^{ut} &= \frac{1}{M_c^t}\sum_{m=1}^{M_c^t}\sum_{c=1}^{C} -f_c(A_w(u_m))\log(f_c(A_w(u_m))), \end{aligned} \tag{5}$$

$$\begin{aligned} e_c^x &\leftarrow \lambda_e e_c^x + (1-\lambda_e) e_c^{xt} \\ e_c^u &\leftarrow \lambda_e e_c^u + (1-\lambda_e) e_c^{ut}, \end{aligned} \tag{6}$$

where N_c^t and M_c^t represent the data quantity within one batch belonging to class c according to the ground truth and pseudo label respectively, and λ_e denotes the EMA weight. It's worth noting that we start estimating the entropy of data once the training status stabilizes. Then, we obtain the total class-wise entropy, *i.e.* $e_c = e_c^u + e_c^x$, and subsequently compute the final sampling probability $\hat{s}_c$:

$$\hat{s}_c = \delta(\alpha s_c + (1-\alpha) s_c'), \tag{7}$$

where s_c' is the normalization of e_c, denoted as $s_c' = e_c / \sum_{c=1}^{C} e_c$, the hyper-parameter α is used to balance between the effective number and entropy. The convex function $\delta(\cdot)$ is utilized to map the sampling function better according to FlexMatch [72]. Finally, we can obtain a more comprehensive sampling function $\hat{s}_c$ for CamMix.

Entropy-based Selection Module. Previous work [8, 9, 63] in SSL primarily uses unlabeled samples for data mixing.

Yet, some pseudo labels possess high uncertainty [5, 64], especially for challenging samples or in early training stages, causing confirmation bias [5]. Our data mixing approach allows the selection of both labeled and unlabeled data. We suggest augmenting high-uncertainty unlabeled data with confident labeled data. However, this beneficial mixing may under-utilize unlabeled data when the regions of labeled data cover unlabeled ones, leaving them unexploited in training. This trade-off is a crucial consideration in data mixing. Thus, we utilize sample-wise entropy e_m as the selection indicator between labeled and unlabeled samples in data mixing as:

$$e_m = \sum_{c=1}^{C} -f_c(A_w(u_m)) \log(f_c(A_w(u_m))). \tag{8}$$

We then define $M_h(\cdot)$ and $M_l(\cdot)$ as the masks of high and low entropy for selecting labeled and unlabeled samples respectively. They are also used to mask the unsupervised loss as in the Appendix A. These masks can be expressed as:

$$\begin{aligned} M_h(u_m) &= \mathbb{I}(e_m > \tau_e) \\ M_l(u_m) &= \mathbb{I}(e_m < \tau_e), \end{aligned} \tag{9}$$

where τ_e is the selection threshold of the entropy mask, updated in EMA manner:

$$\tau_e \leftarrow \lambda_\tau \tau_e + (1 - \lambda_\tau) e^t, \tag{10}$$

where λ_τ denotes the EMA weight, e^t is the average entropy of unlabeled data at each training iteration t, *i.e.* $e^t = \frac{1}{B}\sum_{m=1}^{B} e_m$. In the early training stages, we select more labeled samples for mixing due to the uncertainty of model prediction on some unlabeled data. As training progresses and predictions become more reliable, the utilization of unlabeled data increases.

Entropy-based class balanced loss. We further apply the class balanced loss, which is first introduced in [16] to re-balance the class distribution by utilizing the weighted loss based on the class-wise effective number as $L_{cb} = L_{cls}/E_c$. By normalizing $1/E_c$ as Eq. 4, we can obtain $L_{cb} = s_c L_{cls}$.

Moreover, to tackle the class-wise uncertainty problem in LTSSL, we propose entropy-based class balanced loss L_{ecb} on the unlabeled data. L_{ecb} uses $\hat{s}_c$ to measure both the effective number and uncertainty as:

$$L_{ecb} = \hat{s}_c^u L_{cls}, \tag{11}$$

where $\hat{s}_c^u$ is calculated by Eq. 7, but only based on unlabeled data. Finally, L_{ecb} can be weighted towards both tail classes and high uncertainty classes, further re-balancing the training process. Unlike previous entropy-based losses [22, 51], our loss focuses on class-wise uncertainty instead of sample-wise, making it ideally suited for the LTSSL problem characterized by large category gaps. A detailed description of the loss functions can be found in the Appendix A.

5. Experiments

5.1. Experimental setup

Datasets. We perform evaluation experiments of our proposed method on widely used long-tailed datasets, including CIFAR10-LT [35], CIFAR100-LT [35], STL10-LT [14] and ImageNet-127 [20]. To create imbalanced versions of the datasets, we randomly discard training samples to maintain the pre-defined imbalance ratio. With the imbalance ratio γ_l and maximum number N_1 of labeled samples, we can calculate the number of labeled samples for class c as $N_c = N_1 \times \gamma_l^{-\frac{c-1}{C-1}}$. Similarly, using the parameters γ_u and M_1, we can determine the class distribution of unlabeled data quantity as in the labeled samples. For a detailed introduction to the datasets, please refer to the Appendix B.

Implementation Details. Following DASO [46], we apply our method to various baseline frameworks, including FixMatch [55], FixMatch + LA [45], FixMatch + ABC [39] and FixMatch + ACR [66]. We compare our method with recent re-balancing methods like DARP [33], CReST/CReST+ [65] and DASO [46]. For a fair comparison, our code is developed based on DASO and ACR, implemented with Pytorch [49]. We conduct our experiments on CIFAR10-LT, CIFAR100-LT and STL-10 using Wide ResNet-28-2 [70], and on ImageNet-127 using ResNet-50 [23]. The top-1 accuracy on the test set is used as the evaluation metric. The mean and standard deviation of three independent runs are reported. Due to the page limitation, detailed training settings are provided in the Appendix B.

5.2. Results on CIFAR10/100-LT and STL10-LT.

We first consider the $\gamma_l = \gamma_u$ situation which is the most common scenario in SSL. Then, we investigate the performance of the methods by setting $\gamma_l \neq \gamma_u$, including uniform ($\gamma_u = 1$) and reversed ($\gamma_u = 1/100$) scenarios.

In case of $\gamma_l = \gamma_u$. As shown in Tab. 1, we compare our method with existing re-balancing methods under various baseline settings. When setting FixMatch as the baseline, our BEM shows superior performance improvement in most scenarios. When further adding LA to FixMatch for label re-balancing, our BEM outperforms all other configurations. When integrating ABC into FixMatch for pseudo label re-balancing, our BEM can benefit the baseline more than the DASO. Finally, we also demonstrate that our methods can complement ACR, achieving the SOTA performance with an average gain of 18.35% over FixMatch for CIFAR10-LT. In summary, our BEM achieves consistent and significant gain under all baseline settings, showing its great adaptability. The main reason is that, unlike most previous methods with pseudo label or logit adjustment, we directly re-balance the class distribution through data mixing, a vital technique missing in them, thus complementing these methods.

In case of $\gamma_l \neq \gamma_u$. In real-world datasets, the class dis-

Table 1. Comparison of test accuracy with combinations of different baseline frameworks under $\gamma_l = \gamma_u$ setup on CIFAR10-LT and CIFAR100-LT. The best results for each diversion are in **bold**.

	CIFAR10-LT				CIFAR100-LT			
	$\gamma=\gamma_l=\gamma_u=100$		$\gamma=\gamma_l=\gamma_u=150$		$\gamma=\gamma_l=\gamma_u=10$		$\gamma=\gamma_l=\gamma_u=20$	
Algorithm	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=50$ $M_1=400$	$N_1=150$ $M_1=300$	$N_1=50$ $M_1=400$	$N_1=150$ $M_1=300$
Supervised	47.3±0.95	61.9±0.41	44.2±0.33	58.2±0.29	29.6±0.57	46.9±0.22	25.1±1.14	41.2±0.15
w/LA [45]	53.3±0.44	70.6±0.21	49.5±0.40	67.1±0.78	30.2±0.44	48.7±0.89	26.5±1.31	44.1±0.42
FixMatch [55]	67.8±1.13	77.5±1.32	62.9±0.36	72.4±1.03	45.2±0.55	56.5±0.06	40.0±0.96	50.7±0.25
w/DARP [33]	74.5±0.78	77.8±0.63	67.2±0.32	73.6±0.73	49.4±0.20	58.1±0.44	43.4±0.87	52.2±0.66
w/CReST+ [65]	**76.3**±0.86	78.1±0.42	67.5±0.45	73.7±0.34	44.5±0.94	57.4±0.18	40.1±1.28	52.1±0.21
w/DASO [46]	76.0±0.37	79.1±0.75	**70.1**±1.81	75.1±0.77	49.8±0.24	**59.2**±0.35	43.6±0.09	52.9±0.42
w/BEM (ours)	75.8±1.13	**80.3**±0.62	69.7±0.91	**75.7**±0.22	**50.4**±0.34	59.0±0.23	**44.1**±0.18	**54.3**±0.36
FixMatch+LA [45]	75.3±2.45	82.0±0.36	67.0±2.49	78.0±0.91	47.3±0.42	58.6±0.36	41.4±0.93	53.4±0.32
w/DARP [33]	76.6±0.92	80.8±0.62	68.3±0.94	76.7±1.13	50.5±0.78	59.9±0.32	44.4±0.65	53.8±0.43
w/CReST [65]	76.7±1.13	81.1±0.57	70.9±1.18	77.9±0.71	44.0±0.21	57.1±0.55	40.6±0.55	52.3±0.20
w/DASO [46]	77.9±0.88	82.5±0.08	70.1±1.68	79.0±2.23	50.7±0.51	60.6±0.71	44.1±0.61	55.1±0.72
w/BEM (ours)	**78.6**±0.97	**83.1**±0.13	**72.5**±1.13	**79.9**±1.02	**51.3**±0.26	**61.9**±0.57	**44.8**±0.21	**56.1**±0.54
FixMatch+ABC [39]	78.9±0.82	83.8±0.36	66.5±0.78	80.1±0.45	47.5±0.18	59.1±0.21	41.6±0.83	53.7±0.55
w/DASO [46]	**80.1**±1.16	83.4±0.31	70.6±0.80	80.4±0.56	**50.2**±0.62	60.0±0.32	**44.5**±0.25	55.3±0.53
w/BEM (ours)	79.8±0.82	**83.9**±0.34	**70.7**±0.78	**80.8**±0.67	50.0±0.15	**60.9**±0.42	44.4±0.18	**55.5**±0.84
FixMatch+ACR [66]	81.6±0.19	84.1±0.39	77.0±1.19	80.9±0.22	55.7±0.12	65.6±0.16	48.0±0.75	58.9±0.36
w/BEM (ours)	**83.5**±0.33	**85.5**±0.28	**78.1**±0.99	**83.8**±1.12	**55.8**±0.32	**66.3**±0.24	**48.6**±0.45	**59.8**±0.37

Table 2. Comparison of test accuracy with combinations of different baseline frameworks under $\gamma_l \neq \gamma_u$ setup on CIFAR10-LT and STL10-LT. The γ_l is fixed to 100 for CIFAR10-LT, and the γ_l is set to 10 and 20 for STL10-LT. The N/A denotes the class distribution of data quantity is unknown. The best results for each diversion are in **bold**.

	CIFAR10-LT($\gamma_l \neq \gamma_u$)				STL10-LT($\gamma_u = N/A$)			
	$\gamma_u=1$(uniform)		$\gamma_u=1/100$(reversed)		$\gamma_l=10$		$\gamma_l=20$	
Algorithm	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=500$ $M_1=4000$	$N_1=1500$ $M_1=3000$	$N_1=150$ $M=100k$	$N_1=450$ $M=100k$	$N_1=150$ $M=100k$	$N_1=450$ $M=100k$
FixMatch [55]	73.0±3.81	81.5±1.15	62.5±0.94	71.8±1.70	56.1±2.32	72.4±0.71	47.6±4.87	64.0±2.27
w/DARP [33]	82.5±0.75	84.6±0.34	70.1±0.22	80.0±0.93	66.9±1.66	75.6±0.45	59.9±2.17	72.3±0.60
w/CReST [65]	83.2±1.67	87.1±0.28	70.7±2.02	**80.8**±0.39	61.7±2.51	71.6±1.17	57.1±3.67	68.6±0.88
w/CReST+ [65]	82.2±1.53	86.4±0.42	62.9±1.39	72.9±2.00	61.2±1.27	71.5±0.96	56.0±3.19	68.5±1.88
w/DASO [46]	86.6±0.84	88.8±0.59	**71.0**±0.95	80.3±0.65	**70.0**±1.19	78.4±0.80	**65.7**±1.78	75.3±0.44
w/BEM(ours)	**86.8**±0.47	**89.1**±0.75	70.0±1.72	79.1±0.77	68.3±1.15	**81.2**±1.42	61.6±0.98	**76.0**±1.51
FixMatch+ACR [66]	92.1±0.18	93.5±0.11	85.0±0.09	89.5±0.17	77.1±0.24	83.0±0.32	75.1±0.70	81.5±0.25
w/BEM(ours)	**94.3**±0.14	**95.1**±0.56	**85.5**±0.21	**89.8**±0.12	**79.3**±0.34	**84.2**±0.56	**75.9**±0.15	**82.3**±0.23

tribution of unlabeled data remains unknown or inconsistent with labeled data. For CIFAR10-LT, we consider two extreme scenarios: uniform and reversed. For STL10-LT, where the class distribution of unlabeled data is unknown, we set $\gamma_l \in \{10, 20\}$ and $N_1 \in \{150, 450\}$.

As shown in Tab. 2, our methods yield an average improvement of 14.1% and 11.1% over FixMatch in two scenarios for CIFAR10-LT. However, our method is less effective than DASO under the reversed setting. We speculate that data mixing methods cannot achieve thorough re-balancing in challenging scenarios, unlike approaches from the prediction perspective. However, integrating ACR results in the best performance on CIFAR10-LT, even under the reversed setting, with an average gain of 22.9% and 30.9% over FixMatch. Similarly, for STL-10, our method enhances the performance of FixMatch and achieves the best performance when combined with ACR. This highlights the value of our BEM for re-balancing methods. Further comparisons of our method with more re-balancing methods can be found in the Appendix D.

Table 3. Comparison of test accuracy with combinations of different SSL learners, including MeanTeacher, FlexMatch and SoftMatch.

	C10-LT		C100-LT	STL10-LT
	$N_1 = 1500$ $M_1 = 3000$		$N_1 = 150$ $M_1 = 300$	$N_1 = 450$ $M = 100k$
Algorithm	$\gamma_u = 100$	$\gamma_u = 1$	$\gamma_u = 10$	$\gamma_u = N/A$
MeanTeacher[57]	68.6±0.88	46.4±0.98	52.1±0.09	54.6±0.17
w/BEM(Ours)	**73.5**±0.56	**81.3**±1.67	**60.1**±0.43	**75.3**±0.59
FlexMatch [72]	79.2±0.92	82.2±0.23	62.1±0.86	74.9±0.42
w/BEM(Ours)	**81.2**±0.50	**88.0**±0.17	**68.4**±0.79	**81.2**±0.92
SoftMatch [12]	79.6±0.46	78.3±0.86	62.8±0.33	75.5±0.74
w/BEM(Ours)	**82.0**±0.38	**84.5**±0.25	**68.9**±1.08	**82.8**±0.49

BEM on the SSL learner. We further validate the adaptability of BEM with various SSL learners, including MeanTeacher [57], FlexMatch [72] and SoftMatch [12]. Notably, FlexMatch and SoftMatch outperform FixMatch on balanced datasets. For SoftMatch, we only apply L_{cb} considering its training process already re-weights the loss based on class-wise confidence. Following DASO, we set $\gamma_l = 100$ for CIFAR10-LT and $\gamma_l = 10$ for CIFAR100-LT and STL10-LT. As depicted in Tab. 3, our method enhances the performance of all SSL learners under each setting. Specially, MeanTeacher initially underperforms on the Long-Tailed dataset but achieves gains of 41.1%, 15.4%, and 37.9% on three datasets by applying BEM. SoftMatch, the state-of-the-art SSL method, also gains an additional 5.5%, 9.7% and 9.7% improvement with our BEM.

5.3. Results on ImageNet-127.

ImageNet127, initially introduced in [31] and later employed by CReST [65] for imbalanced SSL, is a naturally imbalanced dataset with an imbalance ratio $\gamma \approx 286$. It groups the 1000 classes of ImageNet [17] into 127 classes, based on the WordNet hierarchy. Due to resource constraints, we down-sample the origin ImageNet127 images to 32×32 or 64×64 pixel images [20] and randomly select 10% of training samples as the labeled set. Given the long-tailed test set, we set $\alpha = 0.2$ to reduce sampling and loss weight bias towards tail classes, favoring high uncertainty classes instead. Tab. 4 demonstrates the superiority of our method over FixMatch, even without other re-balancing techniques. When combined with ACR, our method achieves the best results for both image sizes (95.3% and 51.1% absolute gains over FixMatch). This shows the applicability of our BEM to long-tailed test datasets and its ability to enhance previous re-balancing methods.

5.4. Comprehensive analysis of the method.

We perform comprehensive ablation studies to further understand how our method enhances baseline frameworks. Following DASO, we use CIFAR10-LT (C10) with $N_1 = 500$, $\gamma = 100$ and STL10-LT (STL10) with $N_1 = 150$, $\gamma_l = 10$ to cover both $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$ cases. Our baseline framework is FixMatch. More results are provided in the Appendix D.

Table 4. Comparison of test accuracy with combinations of different baseline frameworks on ImageNet-127.

Algorithm	32×32	64×64
FixMatch [55]	29.7	42.3
w/DARP [33]	30.5	42.5
w/DARP+cRT [33]	39.7	51.0
w/CReST+ [65]	32.5	44.7
w/CReST++LA [45]	40.9	55.9
w/CoSSL [20]	43.7	53.9
w/TRAS [67]	46.2	54.1
w/BEM(Ours)	53.3	58.2
w/ACR [66]	57.2	63.6
w/ACR+BEM(Ours)	**58.0**	**63.9**

Table 5. Ablation study on different mixing strategies. Apart from BEM, all other methods perform mixing within the same batch.

Algorithm	C10	STL10
FixMatch [55]	67.8	56.1
w/MixUp [73]	69.9	63.2
w/CutMix [69]	70.2	62.5
w/SaliencyMix [58]	70.8	64.0
w/CamMix(Ours)	71.9	64.8
w/BEM(Ours)	**75.7**	**68.3**

Ablation study on different mixing strategies. We compare our mixing method with existing techniques including MixUp [73], CutMix [69] and SaliencyMix [58] to demonstrate its effectiveness in Tab. 5. First, we mix data within the same batch. SaliencyMix outperforms CutMix and MixUp on both datasets, and our CamMix surpasses SaliencyMix, indicating better localization ability. By further optimizing the in-batch mixing method, BEM achieves the best results.

Ablation study on each component of BEM. We verify each component in BEM by either removal or standard component replacement in Tab. 6. The accuracy on both datasets reduces sharply when replacing CamMix with CutMix. It highlights the importance of semantic region selection. We then remove CBMB and implement random sampling, resulting in a maximum performance decrease of 5.1% and 4.9%, respectively. This suggests our CBMB effectively tackles the long-tail problem. Removing ESS, denoted as setting $\alpha = 1$, also leads to a decline in the model's performance. When we remove ESM and merely use unlabeled data mixing, it results in a 4.4% performance decrease on STL10. This implies initial training phase guidance from confident labeled data resolves the problem of pseudo label errors especially when $\gamma_l \neq \gamma_u$. Finally, the removal of the ECB loss also

Table 6. Ablation study on each component of BEM.

	C10	STL10
BEM(Ours)	**75.7**	**68.3**
w/o CamMix	74.0	66.6
w/o CBMB	72.1	65.0
w/o ESS	74.7	67.0
w/o ESM	75.3	65.3
w/o ECB Loss	74.9	67.2

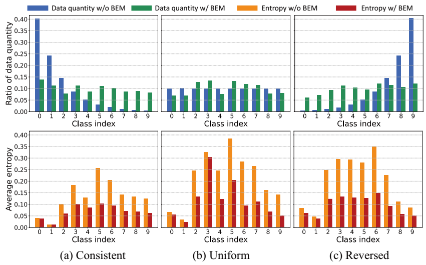


Figure 3. Class distribution of data quantity and entropy in three settings. Each mixed data is calculated as containing two classes.

causes a performance drop on both datasets.

Furthermore, we conduct a qualitative analysis of the performance enhancement achieved by BEM on CIFAR10-LT, setting $\gamma_l = \gamma_u = 100$, $N_1 = 500$ and $M_1 = 4000$. More visualization analysis can be seen in the Appendix E.

Visualization of the class distribution of unlabeled data quantity and entropy. To verify the effect of BEM on the re-balancing training process, we visualize the class distribution of data quantity and entropy. Fig. 3 reveals our method's effect on re-balancing data quantity across all settings. Moreover, our approach notably diminishes uncertainty via entropy and re-balances class-wise entropy, particularly in uniform settings where higher entropy classes engage more training samples, thus lowering uncertainty.

Visualization of T-SNE. Additionally, we visualize the learning representation on the balanced test set using t-distributed stochastic neighbor embedding (t-SNE) [59]. We apply our method to FixMatch and ACR respectively. The results in Fig. 4 suggest that our method generates clearer classification boundaries for representations.

Visualization of data mixing. As shown in Fig. 5, we compare intermediate images from various data mixing methods on STL10 due to the high-resolution input. Five images with different target sizes are selected to visualize. CutMix shows strong randomness and tends to miss the class content, especially when the target is small (see in (e)). Although SaliencyMix has initial target localization ability, it often fails to accurately locate key areas and tends to include numerous redundant contexts (see in (c) and (e)). CamMix shows the best localization ability, accurately locating the class content based on CAM. As τ_c increases, localization accuracy improves and inclusion of redundant context decreases.

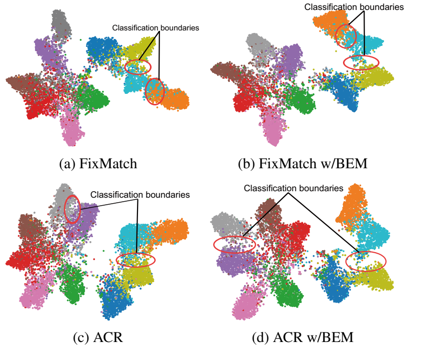


Figure 4. Comparison of t-SNE visualization with combinations of FixMatch and ACR.

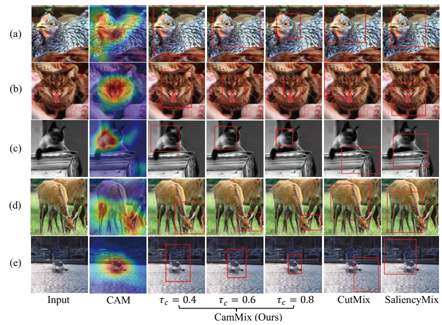


Figure 5. The visualization of data mixing process for CutMix, SaliencyMix, and CamMix on STL10-LT. The red box indicates the image area selected by data mixing.

6. Conclusion

In this work, we introduce a novel approach, Balanced and Entropy-based Mix (BEM), to enhance long-tailed semi-supervised learning by re-balancing the training process. Specially, we re-balance data quantity using the class balanced mix bank and re-balance class-wise uncertainty through the entropy-based learning approach. As the first method to leverage data mixing in LTSSL, BEM significantly boosts the accuracy of various LTSSL frameworks across multiple benchmarks, offering a complementary technique for other re-balancing methods.

References

[1] FirstName Alpher. Frobnication. *IEEE TPAMI*, 12(1):234–778, 2002.

[2] FirstName Alpher and FirstName Fotheringham-Smythe. Frobnication revisited. *Journal of Foo*, 13(1):234–778, 2003.

[3] FirstName Alpher and FirstName Gamow. Can a computer frobnicate? In *CVPR*, pages 234–778, 2005.

[4] FirstName Alpher, FirstName Fotheringham-Smythe, and FirstName Gamow. Can a machine frobnicate? *Journal of Foo*, 14(1):234–778, 2004.

[5] Eric Arazo, Diego Ortego, Paul Albert, Noel E O'Connor, and Kevin McGuinness. Pseudo-labeling and confirmation bias in deep semi-supervised learning. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 2, 5

[6] Samy Bengio. Sharing representations for long tail computer vision problems. In *Proceedings of the 2015 ACM on International Conference on Multimodal Interaction*, pages 1–1, 2015. 1

[7] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. *arXiv preprint arXiv:1807.07543*, 2018. 1

[8] David Berthelot, Nicholas Carlini, Ekin D Cubuk, Alex Kurakin, Kihyuk Sohn, Han Zhang, and Colin Raffel. Remixmatch: Semi-supervised learning with distribution alignment and augmentation anchoring. *arXiv preprint arXiv:1911.09785*, 2019. 1, 2, 4

[9] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin A Raffel. Mixmatch: A holistic approach to semi-supervised learning. *Advances in neural information processing systems*, 32, 2019. 1, 2, 4

[10] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. *Advances in neural information processing systems*, 32, 2019. 2

[11] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European conference on computer vision*, pages 213–229. Springer, 2020. 17

[12] Hao Chen, Ran Tao, Yue Fan, Yidong Wang, Jindong Wang, Bernt Schiele, Xing Xie, Bhiksha Raj, and Marios Savvides. Softmatch: Addressing the quantity-quality trade-off in semi-supervised learning. *arXiv preprint arXiv:2301.10921*, 2023. 1, 7, 15

[13] Xiaohua Chen, Yucan Zhou, Dayan Wu, Chule Yang, Bo Li, Qinghua Hu, and Weiping Wang. Area: adaptive reweighting via effective area for long-tailed classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19277–19287, 2023. 15

[14] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 5, 12

[15] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 702–703, 2020. 3

[16] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9268–9277, 2019. 1, 2, 4, 5

[17] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 7

[18] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[19] Fei Du, Peng Yang, Qi Jia, Fengtao Nan, Xiaoting Chen, and Yun Yang. Global and local mixture consistency cumulative learning for long-tailed visual recognitions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15814–15823, 2023. 2

[20] Yue Fan, Dengxin Dai, Anna Kukleva, and Bernt Schiele. Cossl: Co-learning of representation and classifier for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14574–14584, 2022. 2, 5, 7, 13

[21] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015. 17

[22] Yves Grandvalet and Yoshua Bengio. Semi-supervised learning by entropy minimization. *Advances in neural information processing systems*, 17, 2004. 1, 5

[23] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 5, 13

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 17

[25] Ruifei He, Jihan Yang, and Xiaojuan Qi. Re-distributing biased pseudo labels for semi-supervised semantic segmentation: A baseline investigation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6930–6940, 2021. 17

[26] Tong He, Zhi Zhang, Hang Zhang, Zhongyue Zhang, Junyuan Xie, and Mu Li. Bag of tricks for image classification with convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 558–567, 2019. 1

[27] Feng Hong, Jiangchao Yao, Zhihan Zhou, Ya Zhang, and Yanfeng Wang. Long-tailed partial label learning via dynamic rebalancing. In *The Eleventh International Conference on Learning Representations*, 2023. 1

[28] Feng Hong, Jiangchao Yao, Yueming Lyu, Zhihan Zhou, Ivor Tsang, Ya Zhang, and Yanfeng Wang. On harmonizing implicit subpopulations. In *The Twelfth International Conference on Learning Representations*, 2024. 1

[29] Hanzhe Hu, Fangyun Wei, Han Hu, Qiwei Ye, Jinshi Cui, and Liwei Wang. Semi-supervised semantic segmentation via adaptive equalization learning. *Advances in Neural Information Processing Systems*, 34:22106–22118, 2021. 17

[30] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. *arXiv preprint arXiv:1704.00109*, 2017. 1

[31] Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016. 7

[32] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. *arXiv preprint arXiv:1910.09217*, 2019. 2

[33] Jaehyung Kim, Youngbum Hur, Sejun Park, Eunho Yang, Sung Ju Hwang, and Jinwoo Shin. Distribution aligning refinery of pseudo-label for imbalanced semi-supervised learning. *Advances in neural information processing systems*, 33: 14567–14579, 2020. 5, 6, 7

[34] Jang-Hyun Kim, Wonho Choo, and Hyun Oh Song. Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *International Conference on Machine Learning*, pages 5275–5285. PMLR, 2020.

[35] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1, 5, 12

[36] Samuli Laine and Timo Aila. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016. 1

[37] FirstName LastName. The frobnicatable foo filter, 2014. Face and Gesture submission ID 324. Supplied as supplemental material `fg324.pdf`.

[38] FirstName LastName. Frobnication tutorial, 2014. Supplied as supplemental material `tr.pdf`.

[39] Hyuck Lee, Seungjae Shin, and Heeyoung Kim. Abc: Auxiliary balanced classifier for class-imbalanced semi-supervised learning. *Advances in Neural Information Processing Systems*, 34:7082–7094, 2021. 1, 2, 5, 6, 14

[40] Bolian Li, Zongbo Han, Haining Li, Huazhu Fu, and Changqing Zhang. Trustworthy long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6970–6979, 2022. 2

[41] Han Li, Bowen Shi, Wenrui Dai, Yabo Chen, Botao Wang, Yu Sun, Min Guo, Chenlin Li, Junni Zou, and Hongkai Xiong. Hierarchical graph networks for 3d human pose estimation. In *BMVC*, 2021. 17

[42] Han Li, Bowen Shi, Wenrui Dai, Hongwei Zheng, Botao Wang, Yu Sun, Min Guo, Chenglin Li, Junni Zou, and Hongkai Xiong. Pose-oriented transformer with uncertainty-guided refinement for 2d-to-3d human pose estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1296–1304, 2023. 2, 17

[43] Jun Li, Zichang Tan, Jun Wan, Zhen Lei, and Guodong Guo. Nested collaborative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6949–6958, 2022. 2

[44] Yen-Cheng Liu, Chih-Yao Ma, Zijian He, Chia-Wen Kuo, Kan Chen, Peizhao Zhang, Bichen Wu, Zsolt Kira, and Peter Vajda. Unbiased teacher for semi-supervised object detection. In *International Conference on Learning Representations*, 2021. 17

[45] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*, 2020. 1, 2, 5, 6, 7, 14

[46] Youngtaek Oh, Dong-Jin Kim, and In So Kweon. Daso: Distribution-aware semantics-oriented pseudo-label for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9786–9796, 2022. 1, 2, 5, 6, 12, 13, 14

[47] Yassine Ouali, Céline Hudelot, and Myriam Tami. An overview of deep semi-supervised learning. *arXiv preprint arXiv:2006.05278*, 2020. 1

[48] Seulki Park, Youngkyu Hong, Byeongho Heo, Sangdoo Yun, and Jin Young Choi. The majority can help the minority: Context-rich minority oversampling for long-tailed classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6887–6896, 2022. 2

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 5

[50] Jie Qin, Jiemin Fang, Qian Zhang, Wenyu Liu, Xingang Wang, and Xinggang Wang. Resizemix: Mixing data with preserved object information and true labels. *arXiv preprint arXiv:2012.11101*, 2020.

[51] Kuniaki Saito, Donghyun Kim, Stan Sclaroff, Trevor Darrell, and Kate Saenko. Semi-supervised domain adaptation via minimax entropy. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8050–8058, 2019. 5

[52] Dvir Samuel and Gal Chechik. Distributional robustness loss for long-tail learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9495–9504, 2021. 2

[53] Bowen Shi, Dongsheng Jiang, Xiaopeng Zhang, Han Li, Wenrui Dai, Junni Zou, Hongkai Xiong, and Qi Tian. A transformer-based decoder for semantic segmentation with multi-level context mining. In *European Conference on Computer Vision*, pages 624–639. Springer, 2022. 17

[54] Heereen Shim, Stijn Luca, Dietwig Lowet, and Bart Vanrumste. Data augmentation and semi-supervised learning for deep neural networks-based text classifier. In *Proceedings of the 35th annual ACM symposium on applied computing*, pages 1119–1126, 2020. 1

[55] Kihyuk Sohn, David Berthelot, Nicholas Carlini, Zizhao Zhang, Han Zhang, Colin A Raffel, Ekin Dogus Cubuk, Alexey Kurakin, and Chun-Liang Li. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *Advances in neural information processing systems*, 33:596–608, 2020. 1, 2, 3, 5, 6, 7, 12, 14, 15

[56] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 7262–7272, 2021. 17

[57] Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. *Advances in neural information processing systems*, 30, 2017. 7, 15

[58] AFM Uddin, Mst Monira, Wheemyung Shin, TaeChoong Chung, Sung-Ho Bae, et al. Saliencymix: A saliency guided data augmentation strategy for better regularization. *arXiv preprint arXiv:2006.01791*, 2020. 2, 4, 7

[59] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008. 8

[60] Stefan Van der Walt, Johannes L Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D Warner, Neil Yager, Emmanuelle Gouillart, and Tony Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014.

[61] Jesper E Van Engelen and Holger H Hoos. A survey on semi-supervised learning. *Machine learning*, 109(2):373–440, 2020. 1

[62] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International conference on machine learning*, pages 6438–6447. PMLR, 2019. 1

[63] Vikas Verma, Kenji Kawaguchi, Alex Lamb, Juho Kannala, Arno Solin, Yoshua Bengio, and David Lopez-Paz. Interpolation consistency training for semi-supervised learning. *Neural Networks*, 145:90–106, 2022. 1, 2, 4

[64] Yidong Wang, Hao Chen, Qiang Heng, Wenxin Hou, Yue Fan, Zhen Wu, Jindong Wang, Marios Savvides, Takahiro Shinozaki, Bhiksha Raj, et al. Freematch: Self-adaptive thresholding for semi-supervised learning. *arXiv preprint arXiv:2205.07246*, 2022. 2, 5

[65] Chen Wei, Kihyuk Sohn, Clayton Mellina, Alan Yuille, and Fan Yang. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10857–10866, 2021. 1, 2, 5, 6, 7

[66] Tong Wei and Kai Gan. Towards realistic long-tailed semi-supervised learning: Consistency is all you need. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3469–3478, 2023. 1, 2, 5, 6, 7, 14

[67] Tong Wei, Qian-Yu Liu, Jiang-Xin Shi, Wei-Wei Tu, and Lan-Zhe Guo. Transfer and share: semi-supervised learning from long-tailed data. *Machine Learning*, pages 1–18, 2022. 7

[68] Zhengzhuo Xu, Zenghao Chai, and Chun Yuan. Towards calibrated model for long-tailed visual recognition from prior perspective. *Advances in Neural Information Processing Systems*, 34:7139–7152, 2021. 15

[69] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6023–6032, 2019. 2, 4, 7, 12

[70] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016. 5, 13

[71] Yuhang Zang, Kaiyang Zhou, Chen Huang, and Chen Change Loy. Semi-supervised and long-tailed object detection with cascadematch. *International Journal of Computer Vision*, 131 (4):987–1001, 2023. 17

[72] Bowen Zhang, Yidong Wang, Wenxin Hou, Hao Wu, Jindong Wang, Manabu Okumura, and Takahiro Shinozaki. Flexmatch: Boosting semi-supervised learning with curriculum pseudo labeling. *Advances in Neural Information Processing Systems*, 34:18408–18419, 2021. 1, 4, 7, 15

[73] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 2, 4, 7

[74] Songyang Zhang, Zeming Li, Shipeng Yan, Xuming He, and Jian Sun. Distribution alignment: A unified framework for long-tail visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2361–2370, 2021. 1

[75] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023. 1

[76] Hongwei Zheng, Han Li, Bowen Shi, Wenrui Dai, Botao Wang, Yu Sun, Min Guo, and Hongkai Xiong. Actionprompt: Action-guided 3d human pose estimation with text and pose prompting. In *2023 IEEE International Conference on Multimedia and Expo (ICME)*, pages 2657–2662. IEEE, 2023. 17

[77] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6881–6890, 2021. 17

[78] Zhisheng Zhong, Jiequan Cui, Shu Liu, and Jiaya Jia. Improving calibration for long-tailed recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 16489–16498, 2021. 15

[79] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 4

[80] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. Bbn: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9719–9728, 2020. 2

[81] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 17

BEM: Balanced and Entropy-based Mix for Long-Tailed Semi-Supervised Learning

Supplementary Material

A. Detailed Loss Functions

We detail the loss functions for the training in this section. For the labeled data, we directly adopt the cross entropy $\mathcal{H}(\cdot)$ to calculate the supervised loss L_s. For the unlabeled data, we first follow FixMatch [55] to filter samples with low-confidence pseudo label by a mask $M_u(u_m) = \mathbb{I}(\max(f(A_w(u_m))) > \tau)$. Then, we can obtain M_h and M_l, the masks of high and low entropy for selecting labeled samples (x^s_m, y^s_m) and unlabeled samples u^s_m in data mixing. Given the mixed samples u'_m from CAMmix, we can obtain four types of unsupervised loss (*i.e.* L^h_u, L^l_u, $L^h_{u^s}$, and $L^l_{u^s}$), in which L^h_u, L^l_u and $L^l_{u^s}$ are weighted by the entropy-based class balanced weight $\hat{s}^u$ to form the L_{ecb}. Specifically, L^h_u and L^l_u are supervised by the pseudo label of the original unlabeled data q_m, while $L^h_{u^s}$ and $L^l_{u^s}$ are supervised by the ground truth of the sampled labeled data y^s_m and the pseudo label of the sampled unlabeled data q^s_m, respectively. The final loss function L is weighted by λ reflecting the proportion of area occupied by original and sampled data as in CutMix [69]. Detailed loss functions are as follows:

$$
\begin{aligned}
L_s &= \sum_{n=1}^{B} \mathcal{H}(f(A_w(x_n)), y_n) \\
L^h_u &= \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_h(u_m) \mathcal{H}(f(u'_m), q_m) \\
L^l_u &= \hat{s}^u \sum_{m=1}^{B} M_u(u_m) M_l(u_m) \mathcal{H}(f(u'_m), q_m) \\
L^h_{u^s} &= \sum_{m=1}^{B} M_h(u_m) \mathcal{H}(f(u'_m), y^s_m) \\
L^l_{u^s} &= \hat{s}^u \sum_{m=1}^{B} M_u(u^s_m) M_l(u_m) \mathcal{H}(f(u'_m), q^s_m) \\
L &= L_s + \lambda(L^h_u + L^l_u) + (1-\lambda)(L^h_{u^s} + L^l_{u^s})
\end{aligned}
\tag{12}
$$

Algorithm 1 Balanced and Entropy-based Mix (BEM)

Input: Labeled dataset X, unlabeled dataset U, model f, effective number of labeled data E^x_c, CAM threshold τ_c, area threshold τ_a, balanced parameter α, number of iterations T.

Require: Weak augmentation A_w, strong augmentation A_s.

for $t = 1$ to T **do**
- $\{(x_n, y_n)\}^B_{n=1} \leftarrow X$, $\{u_m\}^B_{m=1} \leftarrow U$
- Pseudo label $q_m \leftarrow \texttt{argmax}\ f(A_w(u_m))$
- **{Update training status}**
- Update CBMB according to (x_n, y_n), (u_m, q_m)
- Update class-wise data quantity E_c via Eq. 2, 3
- Update class-wise entropy e_c via Eq. 5, 6
- Update sampling probability $\hat{s}$, $\hat{s}^u$ via Eq. 7
- Update sample-wise entropy e_m, entropy masks M_h, M_l and entropy selection threshold τ_e via Eq. 8, 9, 10
- **{Sampling}**
- $\{(x^s_m, y^s_m)\}^B_{m=1}, \{u^s_m\}^B_{m=1} \leftarrow$ Sample labeled and unlabeled data from CBMB following $\hat{s}$
- **{Selection and CamMix}**
- $\{u'_m\}^B_{m=1}$, $\lambda \leftarrow \texttt{CamMix}(A_s(u_m),\ A_w(x^s_m),\ y^s_m,\ A_w(u^s_m), f)$ get mixed data and loss weight following M_h, M_l, τ_c and τ_a
- **{Compute losses}**
- Generate the mask of pseudo label M_u
- $L_s \leftarrow \sum^B_{n=1} \mathcal{H}(f(A_w(x_n)), y_n)$
- $L^h_u \leftarrow \hat{s}^u \sum^B_{m=1} M_u(u_m) M_h(u_m) \mathcal{H}(f(u'_m), q_m)$
- $L^l_u \leftarrow \hat{s}^u \sum^B_{m=1} M_u(u_m) M_l(u_m) \mathcal{H}(f(u'_m), q_m)$
- $L^h_{u^s} \leftarrow \sum^B_{m=1} M_h(u_m) \mathcal{H}(f(u'_m), y^s_m)$
- $L^l_{u^s} \leftarrow \hat{s}^u \sum^B_{m=1} M_u(u^s_m) M_l(u_m) \mathcal{H}(f(u'_m), q^s_m)$
- $L = L_s + \lambda(L^h_u + L^l_u) + (1-\lambda)(L^h_{u^s} + L^l_{u^s})$
- Update f based on ∇L using SGD

end for

return

B. Detailed Experimental Setup

In this section, we provide additional information about the datasets and implementation details.

Datasets. We evaluate our method in three scenarios, i.e., *1)* the class distribution of labeled data is consistent with the unlabeled data ($\gamma_l = \gamma_u$). *2)* the labeled and unlabeled data fail to share the same distribution ($\gamma_l \neq \gamma_u$). *3)* The test data possesses an imbalanced class distribution.

- **CIFAR10/100-LT** CIFAR-10/100 [35] are originally class-balanced datasets, each containing 500/5000 samples across 10 and 100 classes respectively. All images are 32×32 in size. Following previous work [46], we sample the training data to create imbalanced versions of the datasets. We employ different sampling ratios for labeled and unlabeled data to achieve various data distributions, including $\gamma_l = \gamma_u$ and $\gamma_l \neq \gamma_u$ scenarios. The test set contains $10k$ samples with a balanced class distribution. The CIFAR dataset can be downloaded from https://www.cs.toronto.edu/ kriz/cifar.html.
- **STL10-LT** The STL-10 [14] dataset consists of 5000

Algorithm 2 CamMix

Input: Strong augmentation of unlabeled data $A_s(u_m)$, weak augmentation of sampled labeled data $A_w(x^s_m)$, the label of sampled labeled data y^s_m, weak augmentation of sampled unlabeled data $A_w(u^s_m)$, model f, high entropy mask M_h, low entropy mask M_l, CAM threshold τ_c, area threshold τ_a, functions in skimage `label`$(\cdot)$ and `regionprops`$(\cdot)$, the function of CutMix `Mix`$(\cdot)$.

Output: Mixed data $\{u'_m\}^B_{m=1}$, loss weight λ.

1: **for** $m = 1$ to B **do**
2: $\quad q^s_m \leftarrow \texttt{argmax}\ f(A_w(u^s_m))$
3: $\quad CAM^u_m \leftarrow \texttt{GradCAM}(A_w(u^s_m), q^s_m)$
4: $\quad S^u_m \leftarrow \texttt{int}(CAM^u_m > \tau_c)$
5: $\quad P^u_m \leftarrow \texttt{max(regionprops(label(}S^u_m\texttt{)))}$ get largest connected region
6: $\quad$ **if** the area ratio of $P^u_m < \tau_a$ **then**
7: $\quad\quad bbox^u_m \leftarrow$ Random crop of $A_w(u^s_m)$
8: $\quad$ **else**
9: $\quad\quad bbox^u_m \leftarrow$ The bounding box of P^u_m
10: $\quad$ **end if**
11: $\quad bbox^x_m \leftarrow$ Calculate the bounding box for $(A_w(x^s_m), y^s_m)$ using a similar method in steps 3-10.
12: $\quad u'_m \leftarrow \texttt{Mix}(A_s(u_m), A_w(x^s_m)$ or $A_w(u^s_m))$ following $bbox^x_m$, $M_h(u_m)$, $bbox^u_m$, $M_l(u_m)$
13: $\quad (1 - \lambda_m) \leftarrow$ The area ratio of $bbox^x_m$ or $bbox^u_m$
14: **end for**
15: $\lambda \leftarrow$ The average of λ_m
16: **return** Mixed data $\{u'_m\}^B_{m=1}$, loss weight λ

class-balanced labeled data and 1000k unlabeled data with an unknown distribution. To make an imbalanced version of the dataset, we only sample the labeled data, while the distribution of unlabeled data naturally differs from that of labeled data, i.e.,$\gamma_l \neq \gamma_u$. All images are 96×96 in size and the dataset can be downloaded from https://cs.stanford.edu/ acoates/stl10/.

- **ImageNet-127** ImageNet-127 [20] is naturally an imbalanced dataset, thus it doesn't require any further processing. Moreover, it has an imbalanced test set, which can validate scenario *3)*. To conserve computation resources, all images are down-sampled to 32×32 or 64×64 in size and the dataset can be downloaded from https://image-net.org/download-images.

Implementation details. Following previous training protocol [46], we conduct our experiments on CIFAR10-LT, CIFAR100-LT and STL10-LT using Wide ResNet-28-2 [70], and on ImageNet-127 using ResNet-50 [23]. We train the model with a batch size of 64 for $250k$ iterations, with an evaluation every 500 iterations. We use SGD with momentum as our optimizer and adopt a cosine learning rate decay strategy by setting the learning rate to $\eta cos(\frac{7\pi t}{16T})$, where η is the initial learning rate, t is the current iteration number and T is the total number of iterations. We set the balance parameter $\alpha = 0.5$ on CIFAR10-LT, CIFAR100-LT and STL10-LT, and set it to 0.2 on ImageNet-127. We set all EMA update weights as $\lambda = \lambda_d = \lambda_e = \lambda_\tau = 0.999$. The CAM threshold τ_c and area threshold τ_a are set to 0.8 and 0.1, respectively. The epoch number for starting to estimate the data quantity and entropy of unlabeled data is set to 5. We designate the final block as the CAM layer. We adopt Softmax$(\cdot)$ as the mapping function $\delta(\cdot)$. Our experiments are conducted on one NVIDIA Tesla V100 with the CentOS 7 system, using PyTorch 1.11.0 and Torchvision 0.12.0.

C. Pseudo-code for Our BEM Algorithm

We define the pseudo-code for our BEM and CamMix algorithm in Alg. 1 and 2, respectively.

D. Additional Experiment Results

In this section, we conduct a series of additional experiments to further demonstrate the effectiveness of our BEM.

More results with re-balancing methods when $\gamma_l \neq \gamma_u$. We present the results of combining with FixMatch and ACR under $\gamma_l \neq \gamma_u$ setup in Tab. 2. As shown in Tab. 7, we further combine our BEM with more re-balancing methods, including LA and ABC. Without incorporating any re-balancing method, BEM's performance is weaker than DASO in some settings, particularly in the reversed setting. After combining two re-balancing methods, BEM outperforms DASO in almost all settings. Further integration with ACR achieves the state-of-the-art results in all scenarios with an average 31.5% performance gain. In summary, our method needs to combine with re-balancing methods to enhance the re-balancing ability in challenging scenarios, and it in turn complements these methods.

More results on CIFAR100-LT. We also conduct experiments on CIFAR100-LT under $\gamma_l \neq \gamma_u$ setup in Tab. 8. Results show that our BEM outperforms DASO in almost all settings. By integrating with ACR, we can achieve the best results in all scenarios (32.7% accuracy gain). It further demonstrates that the complementation of BEM can boost the performance of most re-balancing methods.

Fine-grained results. In this experiment, we present the fine-grained results in Tab. 9. We compare our BEM with DASO and ACR in three settings. Our method surpasses DASO in all scenarios and further enhances the state-of-the-art method (ACR). In particular, our method significantly improves the performance of few-shot classes at the cost of negligible drop on head classes in the consistent setting. Moreover, in all settings, our method shows a large improvement in medium classes, which is brought by entropy-based learning.

BEM on balanced datasets. To verify the effect of our BEM on balanced datasets, we conduct experiments on balanced datasets with combinations of different SSL methods, in-

Table 7. Comparison of test accuracy with combinations of different baseline models under $\gamma_l \neq \gamma_u$ setup on CIFAR10-LT and STL10-LT. The γ_l is fixed to 100 for CIFAR10-LT, and the γ_l is set to 10 and 20 for STL10-LT. The best results for each diversion are in **bold**.

	CIFAR10-LT($\gamma_l \neq \gamma_u$)				STL10-LT($\gamma_u = N/A$)			
	$\gamma_u = 1$(uniform)		$\gamma_u = 1/100$(reversed)		$\gamma_l = 10$		$\gamma_l = 20$	
Algorithm	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 500$ $M_1 = 4000$	$N_1 = 1500$ $M_1 = 3000$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$	$N_1 = 150$ $M = 100k$	$N_1 = 450$ $M = 100k$
FixMatch [55]	73.0±3.81	81.5±1.15	62.5±0.94	71.8±1.70	56.1±2.32	72.4±0.71	47.6±4.87	64.0±2.27
w/DASO [46]	86.6±0.84	88.8±0.59	**71.0** ±0.95	**80.3**±0.65	**70.0**±1.19	78.4±0.80	**65.7**±1.78	75.3±0.44
w/BEM(Ours)	**86.8**±0.47	**89.1**±0.75	70.0±1.72	79.1±0.77	68.3±1.15	**81.2**±1.42	61.6±0.98	**76.0**±1.51
w/LA [45]+DASO [46]	84.6±2.04	86.8±0.76	**72.6** ±0.38	78.5±1.31	72.7±1.45	79.7±0.44	**66.8**±0.62	75.7±0.50
w/LA [45]+BEM(Ours)	**85.3**±0.31	**88.5**±0.65	70.9±1.69	**79.8**±1.37	**72.9**±0.38	**81.8**±0.76	65.7±0.25	**76.8**±1.87
w/ABC [39]+DASO [46]	85.2±1.56	88.4±0.82	70.1±1.25	79.8±0.21	71.8±1.17	78.4±0.58	**67.3**±2.06	75.9±0.43
w/ABC [39]+BEM(Ours)	**85.9**±0.33	**89.0**±0.67	**71.2**±0.58	**80.1**±0.96	**73.1**±1.68	**81.4**±1.29	66.4±1.93	**76.7**±1.12
w/ACR [66]	92.1±0.18	93.5±0.11	85.0±0.09	89.5±0.17	77.1±0.24	83.0±0.32	75.1±0.70	81.5±0.25
w/ACR [66]+w/BEM(Ours)	**94.3**±0.14	**95.1**±0.56	**85.5**±0.21	**89.8**±0.12	**79.3**±0.34	**84.2**±0.56	**75.9**±0.15	**82.3**±0.23

Table 8. Comparison of test accuracy with combinations of different baseline models under $\gamma_l \neq \gamma_u$ setup on CIFAR100-LT. The γ_l is fixed to 10. The best results for each diversion are in **bold**.

	CIFAR100-LT($\gamma_l \neq \gamma_u$)			
	$\gamma_u = 1$(uniform)		$\gamma_u = 1/10$(reversed)	
Algorithm	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$	$N_1 = 50$ $M_1 = 400$	$N_1 = 150$ $M_1 = 300$
FixMatch [55]	45.5±0.71	58.1±0.72	44.2±0.43	57.3±0.19
w/DASO [46]	53.9±0.66	61.8±0.98	**51.0**±0.19	60.0±0.31
w/BEM(Ours)	**54.8**±0.55	**63.6**±0.91	50.8±0.25	**60.7**±0.12
w/LA [45]+DASO [46]	54.7±0.40	62.4±1.06	51.1±0.12	60.5±0.23
w/LA [45]+BEM(Ours)	**56.5**±0.43	**64.1**±0.87	**51.7**±0.20	**61.3**±0.17
w/ABC [39]+DASO [46]	53.4±0.53	62.4±0.61	**51.2**±0.19	60.8±0.39
w/ABC [39]+BEM(Ours)	**55.2**±0.35	**64.7**±0.87	51.1±0.10	**61.4**±0.29
w/ACR [66]	66.0±0.25	73.4±0.22	57.0±0.46	67.6±0.12
w/ACR [66]+BEM(Ours)	**68.1**±0.34	**75.9**±0.49	**58.0±0.28**	**68.4±0.13**

Table 9. Fine-grained results on CIFAR10-LT with $N_1 = 1500, M_1 = 3000, \gamma_l = 100$.

	Consistent($\gamma_u = 100$)				Uniform($\gamma_u = 1$)				Reversed($\gamma_u = 1/100$)			
Algorithm	Many	Medium	Few	All	Many	Medium	Few	All	Many	Medium	Few	All
DASO	**95.1**	78.6	60.4	78.1	89.6	84.4	85.7	86.3	84.0	71.6	68.2	74.3
BEM	94.7	78.0	67.0	79.8	91.7	88.1	90.7	89.4	82.3	80.2	73.3	78.7
ACR	93.9	81.6	75.3	83.4	92.8	90.6	97.9	93.5	90.7	83.8	**96.4**	89.7
ACR+BEM	92.3	**83.3**	**81.9**	**85.4**	**95.4**	**93.1**	**98.0**	**95.3**	**90.9**	**84.9**	95.8	**89.9**

cluding MeanTeacher, FixMatch, FlexMatch and SoftMatch. Specifically, we set $\alpha = 0$, meaning that we only consider the differences in class-wise uncertainty distribution. As shown in Tab. 12, our BEM enhances the performance of all baseline models, particularly the MeanTeacher, where our method gains an average of 21.4%, 26.9% and 25.0% improvement for three datasets. This demonstrates the potential of class-wise uncertainty re-balancing in enhancing model performance for balanced datasets.

Table 10. Ablation study on different sampling strategies. EFF. denotes the effective number.

	CBMB	ESS	EFF.	C10	STL10
Random				72.1	65.0
Quantity-based	✓		✓	74.9	66.5
Entropy-based		✓		74.4	65.9
w/o effective number	✓	✓		75.2	67.3
Ours	✓	✓	✓	**75.7**	**68.3**

Table 11. Ablation study on updating strategies of entropy selection threshold τ_e.

	C10	STL10
Baseline	67.8	56.1
$\tau_e = 0.1$	74.7	66.6
$\tau_e = 0.2$	75.2	67.2
$\tau_e = 0.4$	75.1	66.9
$\tau_e = 0.6$	74.4	66.4
w/ ours	**75.7**	**68.3**

Ablation study on sampling strategies. To evaluate the effect of our sampling strategy, we conduct a series of experiments by replacing the sampling function. Results are summarized in Tab. 10. Random sampling only improves performance slightly. Then, we split the class-balanced entropy-based sampling function and find that the results drop on both datasets. Further, we replace the effective number with the common number. Results indicate the effective number more accurately measures the class distribution of datasets.

Ablation on the updating strategy of entropy threshold τ_e. As shown in Tab. 11, we perform experiments to validate the effect of the entropy threshold τ_e updating strategy. When we filter the entropy mask with fixed thresholds, the

Table 12. Comparison of test accuracy on balanced datasets with combinations of different SSL methods, including MeanTeacher, FixMatch, FlexMatch and SoftMatch.

	CIFAR-10			CIFAR-100			STL-10	
Algorithm	40	250	4000	400	2500	10000	40	1000
MeanTeacher[57]	29.81±1.60	62.54±3.30	91.90±0.21	18.89±1.44	54.83±1.06	68.25±0.23	28.28±1.45	66.10±1.37
w/BEM(Ours)	**43.13**±2.55	**74.31**±1.79	**92.65**±0.23	**30.92**±3.69	**60.73**±2.14	**72.54**±0.19	**37.31**±2.59	**78.74**±1.38
FixMatch [55]	92.53±0.28	95.14±0.05	95.79±0.08	53.58±0.82	72.97±0.16	77.80±0.12	64.03±4.14	93.75±0.33
w/BEM(Ours)	**93.96**±0.37	**95.37**±0.03	**95.93**±0.11	**55.24**±0.93	**73.12**±0.14	**77.95**±0.11	**66.45**±3.29	**93.98**±0.65
FlexMatch [72]	95.03±0.06	95.03±0.09	95.81±0.01	60.06±1.62	73.51±0.20	78.10±0.09	70.85±0.01	94.23±1.62
w/BEM(Ours)	**95.08**±0.09	**95.21**±0.04	**95.98**±0.01	**60.83**±0.98	**73.94**±0.18	**78.72**±0.11	**72.11**±0.03	**94.39**±1.54
SoftMatch [12]	95.09±0.12	95.18±0.09	95.96±0.02	62.90±0.77	73.34±0.25	77.97±0.03	78.58±3.48	94.27±0.24
w/BEM(Ours)	**95.11**±0.08	**95.37**±0.06	**96.12**±0.07	**63.13**±0.92	**73.56**±0.08	**78.14**±0.08	**79.09**±3.87	**94.43**±0.38

Table 13. Ablation study on α.

	C10	STL10
1.0	74.7	67.0
0.7	75.5	67.3
0.5	**75.7**	68.3
0.3	74.4	**68.5**
0	73.8	67.5

Table 14. Ablation study on τ_c.

	C10	STL10
0.9	73.0	65.8
0.8	**75.7**	**68.3**
0.6	74.4	67.3
0.4	71.5	64.6
0.2	69.3	61.3

performance decreases and becomes unstable. Our EMA updating strategy achieves the best result, indicating that it adaptively adjusts the threshold following the training status of the model.

Ablation study on parameter α. As shown in Tab. 13, we verify the effect of α to balance the effective number and entropy in Eq. 7. Results show the best α on CIFAR10-LT and STL10-LT are 0.5 and 0.3, respectively. The visualization of sampling rate and class accuracy can be seen in Appendix E.

Ablation study on CAM threshold τ_c. In Tab. 14, we study the effect of CAM threshold τ_c on selected region. Results show that 0.8 is the best threshold on both datasets. It indicates that the precise selection of relevant regions is more advantageous for re-balancing long-tailed datasets.

Ablation study on the adding weight β. We conduct experiments to test the impact of the adding weight β in the equation $e_c = \beta e_c^u + (1-\beta)e_c^x$. The results in Tab. 15 indicate that weight addition has minimal impact. So we remove this parameter to simplify the number of hyperparameters.

Table 15. Ablation study on the adding weight β.

β	C10	STL10
0.7	75.1	67.7
0.5	75.7	**68.3**
0.3	**75.8**	68.1
0.1	74.9	67.9

Table 16. More comparison with class-wise data mixing methods.

	C10	STL10
FixMatch	67.8	56.1
w/UniMix [68]	72.9	66.0
w/MiSLAS [78]	73.4	66.2
w/Ours	**75.7**	**68.3**

Table 17. Comparison with AREA on supervised learning.

	C10-LT		C100-LT	
γ	200	50	200	50
CE	65.7	74.8	34.8	43.9
AREA [13]	**75.0**	82.7	**43.9**	**51.8**
Ours	74.7	**83.0**	40.3	49.7

More comparison with class-wise data mixing methods. We conduct additional experiments to compare our BEM with other class-wise data mixing methods [68, 78]. The results in Tab. 16 show that BEM outperforms them. We infer that these class-wise mixup methods are limited in not considering the uncertainty issue in LTSSL.

Comparison with AREA. We compare our BEM with AREA [13], which is a fully supervised learning method in long-tailed learning. Our BEM is different from AREA in three aspects: **1) Motivation:** AREA does not consider class-wise uncertainty. It optimizes the re-weighting strategy, which only focuses on data quantity, by exploring the spanned space of each class and relations between samples. While we propose to re-balance the class distribution of both data quantity and uncertainty, which is more suitable for LTSSL. **2) Task:** AREA focuses only on the class imbalance issue in the supervised learning diagram. While our method is specifically designed for LTSSL to further address the issue of uncertainty in unlabeled sample predictions, which can not be achieved by AREA. We also apply our BEM to supervised learning. Tab. 17 shows that BEM is competitive with AREA, demonstrating its flexibility and superiority. **3) Design:** AREA is based on the re-weighting strategy, using the effective area as class-wise weights in cross-entropy loss. While BEM is primarily based on re-sampling, where we use class-wise data quantity and uncertainty as sampling criteria for CamMix.

E. Additional Visualization Analysis

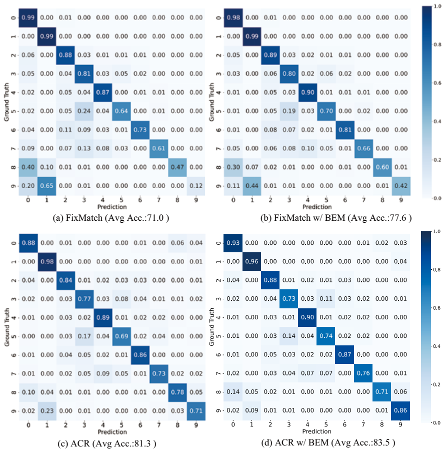
In this section, we provide additional visualization analysis to better understand our approach.

Visualization of confusion matrices on test set. We compare the confusion matrices of the prediction from the test set. We conduct experiments on CIFAR10-LT in the consistent scenario and apply our BEM to FixMatch and ACR, respectively. As shown in Fig. 6, the prediction of FixMatch is significantly biased towards the head classes, resulting in poor performance of the tail classes. Our method greatly alleviates this bias, improving both the tail performance and overall performance. ACR achieves good results in various classes, and our method further improves the performance of the tail classes, demonstrating the superiority and versatility of our method.

Visualization of precision and recall on the test set. We analyze the precision and recall on the test set to further verify the effect of our BEM. As shown in Fig. 7, we apply our method to FixMatch and ACR. The results show that the recall of tail classes achieves significant gains by combining our BEM with both models.

Visualization of train curves and test accuracy class distribution. We further assess the effect of BEM on FixMatch and ACR by plotting training curves and class-wise test accuracy. As shown in Fig. 8(a), the low entropy ratio increases, suggesting a large fraction of unlabeled data is used in the mixing as the training state becomes stable. As shown in Fig. 8(b), our method greatly improves the tail class performance of FixMatch and ACR.

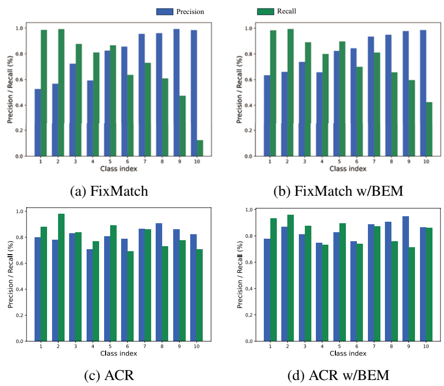
Visualization of the class distribution of sampling rate and accuracy under different α. We present the ablation study on α in Tab. 13. In addition, we further visualize the class distribution of sampling rate and accuracy under various α. Fig. 9 (a) shows that as α increases, the sampling rate of tail classes improves. When α is small, the sampling function pays attention not only to tail classes but also to middle



(a) FixMatch (Avg Acc.:71.0) (b) FixMatch w/ BEM (Avg Acc.:77.6)

(c) ACR (Avg Acc.:81.3) (d) ACR w/ BEM (Avg Acc.:83.5)

Figure 6. The confusion matrices of the test set on CIFAR10-LT under $\gamma_l = \gamma_u$ setup.



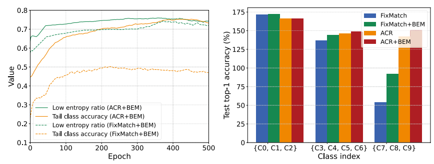
(a) FixMatch (b) FixMatch w/BEM

(c) ACR (d) ACR w/BEM

Figure 7. The precision and recall of the test set on CIFAR10-LT under $\gamma_l = \gamma_u$ setup.

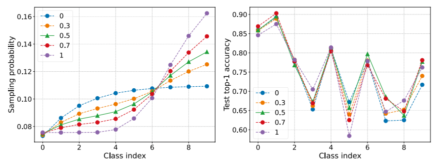
classes with high uncertainty. In Fig. 9 (b), we can see that when $\alpha = 0.5$, both the tail class and the middle class with high uncertainty have relatively high accuracy, indicating it achieves the balance of data quantity and uncertainty.

More visualization of data mixing. We provide the intermediate images of the data mixing on STL10 in Fig. 5. To further illustrate the effectiveness of our CamMix, we also present additional visualization results on CIFAR10 in



(a) Train curves (b) Class-wise test accuracy

Figure 8. **(a):** Train curves for tail low entropy ratio and tail class accuracy. **(b):** Class distribution of test accuracy over different methods. C0 and C9 are the head and tail classes, respectively.



(a) Class-wise sampling rate (b) Class-wise test accuracy

Figure 9. Class distribution of sampling rate and test accuracy under various α on CIFAR10-LT ($\gamma_l = \gamma_u = 100$) using FixMatch.

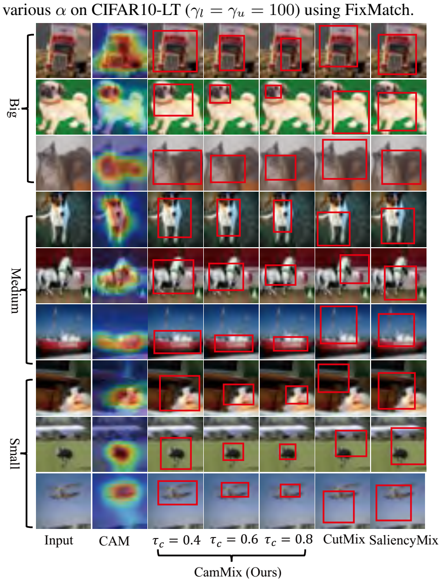
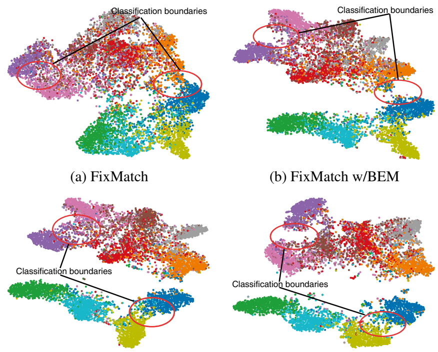


Figure 10. The visualization of data mixing process for CutMix, SaliencyMix, and CamMix on CIFAR10-LT. The red box indicates the image area selected by data mixing.



(a) FixMatch (b) FixMatch w/BEM

(c) ACR (d) ACR w/BEM

Figure 11. Comparison of t-SNE visualization with combinations of FixMatch and ACR on the test set of STL10-LT when $\gamma_l \neq \gamma_u$.

Fig. 10. We select three images for each target size. Based on the results from the two datasets, we can draw the following conclusions: 1) CutMix has a high degree of randomness and often selects the context region. 2) The localization ability of SaliencyMix needs to be optimized. The selection region is not precise and tends to choose numerous redundant areas. 3) CamMix greatly improves the localization ability due to the accuracy of CAM and excludes irrelevant redundant areas as τ_c value decreases.

More visualization of t-SNE As displayed in Fig. 4, we show the t-SNE of learning representations from the test data on CIFAR10-LT. We further conduct experiments on STL10-LT to visualize the learning representations when $\gamma_l \neq \gamma_u$. Results in Fig. 11 show that our method generates clearer classification boundaries for representations when $\gamma_l \neq \gamma_u$. Specially, the classification ability of FixMatch is relatively poor, with most clusters gathered together. Our method greatly enhances its classification ability.

F. Limitation and Future Work

A potential limitation is that the proposed BEM is restricted by only exploring the data mixing for the LTSSL classification task, while ignoring its further application for other vision tasks, such as object detection [11, 21, 81], semantic segmentation [24, 53, 56, 77] and others [41, 42, 76]. It is worth noting that the application of semi-supervised learning for long-tailed objection detection [44, 71] and semantic segmentation [25, 29] is not trivial but much harder than the pure classification task, as it requires further predict object location or semantic mask. In the future, we will extend our BEM to more complex vision tasks to further demonstrate its effectiveness and adaptability.

Cooperation Does Matter: Exploring Multi-Order Bilateral Relations for Audio-Visual Segmentation

Qi Yang[1,2] Xing Nie[1,2] Tong Li[3] Pengfei Gao[3] Ying Guo[3]
Cheng Zhen[3] Pengfei Yan[3] Shiming Xiang[1,2]
[1] School of Artificial Intelligence, University of Chinese Academy of Sciences (UCAS)
[2] Institute of Automation, Chinese Academy of Sciences (CASIA) [3] Meituan

Abstract

Recently, an audio-visual segmentation (AVS) task has been introduced, aiming to group pixels with sounding objects within a given video. This task necessitates a first-ever audio-driven pixel-level understanding of the scene, posing significant challenges. In this paper, we propose an innovative audio-visual transformer framework, termed COMBO, an acronym for COoperation of Multi-order Bilateral relatiOns. For the first time, our framework explores three types of bilateral entanglements within AVS: pixel entanglement, modality entanglement, and temporal entanglement. Regarding pixel entanglement, we employ a Siam-Encoder Module (SEM) that leverages prior knowledge to generate more precise visual features from the foundational model. For modality entanglement, we design a Bilateral-Fusion Module (BFM), enabling COMBO to align corresponding visual and auditory signals bi-directionally. As for temporal entanglement, we introduce an innovative adaptive inter-frame consistency loss according to the inherent rules of temporal. Comprehensive experiments and ablation studies on AVSBench-object (84.7 mIoU on S4, 59.2 mIou on MS3) and AVSBench-semantic (42.1 mIoU on AVSS) datasets demonstrate that COMBO surpasses previous state-of-the-art methods. Project page is available at https://yannqi.github.io/AVS-COMBO.

1. Introduction

Human visual attention is often *hear-guided*, *i.e.*, we tend to focus on the object with sounds [4]. For example, when we hear a cat meow, we pay more attention to the cat than other objects due to the strong association between the meow and the cat. Inspired by this potential interaction of auditory and visual signals, the cross-modal studies of vision and hearing have attracted numerous researchers, such as the audio-visual correspondence [1, 2], which only aims to match visual images and audio signals to the same

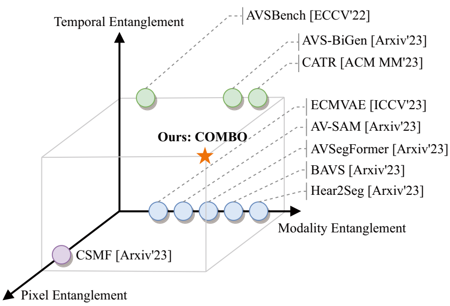


Figure 1. Comparison between the proposed COMBO and existing state-of-the-art methods. Our COMBO is the first work to simultaneously explore multi-order bilateral relations in modality, temporal and pixel levels.

scene, and sound source localization [5, 18, 34] which further seeks to locate the vocal visible regions. However, they have only focused on audio-visual tasks at the image or region levels, lacking pixel-level annotations. Recently, AVSBench [45, 46] integrates audio signals into video segmentation, called Audio-Visual Segmentation (AVS), which comprises two benchmarks: 1) *AVSBench-object*, which includes single source sound segmentation (S4) and multiple sound sources segmentation (MS3); 2) *AVSBench-semantic*, which further extends audio-visual semantic segmentation (AVSS) based on AVSBench-object.

Given that audio-visual segmentation is a burgeoning field that spans both audio and visual modalities, it presents a non-trivial task. Generally, when performing AVS, a cross-modal segmentation task involving video, there are mainly three challenges: (1) AVS contains audio and visual modalities, thus demanding explicit alignment of sequential audio features to spatial pixel-level activations; (2) AVS involves temporal information where the state of the current frame is dynamically affected by historical frames, therefore, exploring the correlation between adjacent frames is essential; (3) AVS includes an image segmentation task; compared to 1D audio signals, 2D image signals have more redundant information, which is prone to be affected by background noise, thus requiring precise extraction of features from the image. To resolve the issue (1), the prevailing methods [13, 25, 26, 31, 33] employ matrix multiplication and modified cross-attention module to encode pixel-wise audio-visual interaction. Though impressive, these designs ignore the temporal dependence of adjacent frames that have been proven to be important for AVS. To solve this problem, some methods [15, 23, 45] partition temporal relations into consideration to explore both issues (1) and (2) simultaneously since AVS is a cross-modal video task. Nevertheless, their approaches rely too much on implicit inter-frame relations, leading to inaccurate associations. Regarding issue (3), CSMF [3] leverages frozen large foundation models to extract pure visual features for AVS. However, it independently tackles the audio and visual signals by naively combining several existing foundation models, resulting in sub-optimal performance.

To this end, we present **COMBO**, a novel audio-visual transformer framework for AVS. According to the three issues mentioned above, COMBO simultaneously considers modality, temporal, and pixel levels by introducing their bilateral entanglements, as shown in Fig. 1. Specifically, nature itself has many bilateral relations. For example, in electricity and magnetism, due to their intrinsic correlation, the change of current causes the change of magnetic field, and vice versa. Motivated by this, we refer to this bilateral relationship of mutual influence as entanglement.

In this work, we explore three potential bilateral entanglements: pixel entanglement, modality entanglement, and temporal entanglement. Pixel entanglement refers to the interdependent relationship between an image and its corresponding mask. Since background noise in the image leads to inaccuracies in the image-to-mask prediction process, it is essential to utilize external masks from the foundation models to entangle the input image to assist the model. Therefore, we construct a Siam-Encoder Module (SEM) as a visual feature extractor to facilitate more precise visual features, which can liberate from the constraints of foundation models than other methods [26, 33, 37]. Besides, as for the alignments of the audio and visual signals, we explore the modality entanglement between audio and visual components to amplify the efficiency of cross-modal matching. Contrary to existing single-fusion methods [13, 45], we believe that the cooperation between the two modalities can produce a positive effect. Inspired by [24], we initially propose a potent and memory-efficient bidirectional audio-visual fusion module called Bilateral-Fusion Module (BFM). Our BFM amplifies the spatial awareness of visual features relevant to sounding objects and strengthens the attention of audio signals embodying visual targets. Moreover, the audio-visual tasks contain a solid temporal entanglement. Thus, we design an adaptive inter-frame consistency loss to better harness this inherent characteristic.

Our main contributions can be summarized as follows:

- We propose a Siam-Encoder Module (SEM) that transfers the knowledge of the foundation model for mining the potential pixel entanglement.
- We propose a Bilateral-Fusion Module (BFM) to take full advantage of the potential of both audio and visual modalities by exploring the modality entanglement.
- We propose an adaptive inter-frame consistency loss based on the inherent coherence of audio-visual tasks for enhanced temporal entanglement.
- We show that COMBO significantly outperforms existing state-of-the-art approaches in the challenging AVSBench-object and AVSBench-semantic datasets.

2. Related Work

2.1. Sound Source Localization

Sound source localization aims to estimate the position of a sound source in a video sequence, which is the most related task to the audio-visual segmentation task. LVS [5] utilizes a hard-mining strategy and a contrastive learning mechanism to discriminate challenging image fragments. DSOL [18] executes class-aware sounding object localization from mixed sound, which initially focuses on learning robust object representations from single-source localization. MSSL [34] localizes multiple sound sources in unconstrained videos without pairwise sound-object annotations. This approach involves the development of a two-stage learning framework, followed by the execution of cross-modal feature alignment. The pioneering methods in these areas have significantly inspired our research on AVS.

2.2. Semantic Segmentation

Semantic segmentation is a fundamental task that requires pixel-level classification. Early researchers take the Fully Convolutional Networks (FCN) [27] as the dominant approach and focus on aggregating long-range context in the feature map. PSPNet [42] performs spatial pyramid pooling at several grid scales. DeepLab [7, 8] utilizes atrous convolutions with different atrous rates. Furthermore, some methods [35, 38, 43] replace traditional convolutional backbone with transformer-based architectures. MaskFormer [9] and Mask2Former [11] propose a mask classifier with learnable queries and specialized designs for mask prediction. OneFormer [19] presents a universal image segmentation framework that unifies segmentation with a multi-task train-once design. Recently, a series of SAM models [20–22, 40] propose to build a foundation model for promptable segmentation with strong generalization. Given that the AVS task entails segmentation, these studies have significantly

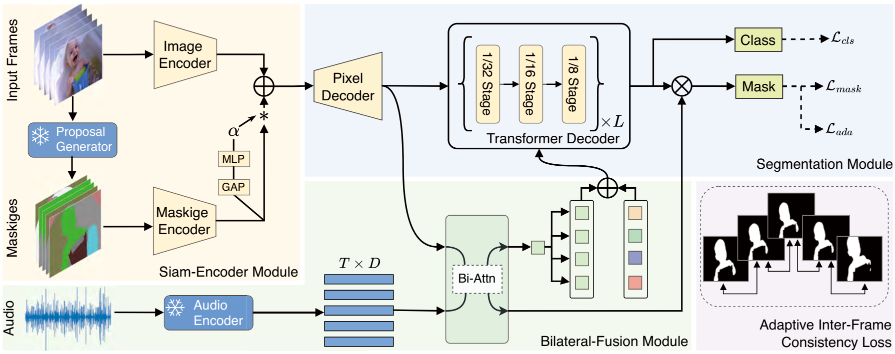


Figure 2. Overview of the proposed COMBO. COMBO adopts a novel audio-visual transformer framework specifically for audio-visual segmentation. Aiming at multi-oder bilateral entanglement, our method is composed of three independent modules. (1) We introduce the Siam-Encoder Module, which is designed for the exploration of pixel entanglement. (2) To integrate the entanglement of audio and visual signals, we propose a Bilateral-Fusion Module. (3) Given the inherent characteristics of temporal entanglement, we construct an adaptive inter-frame consistency loss in the segmentation module to enhance the consistency of the output.

contributed to our work.

2.3. Audio-Visual Segmentation

AVS is an emerging task that aims to locate sounding sources by predicting pixel-wise maps and attracts many researchers [13, 23, 26, 30, 31, 33, 37, 46]. AVSBench [46] first constructs the audio-visual segmentation benchmark and proposes a temporal pixel-wise audio-visual interaction module (TPAVI) to inject audio semantics as guidance for the visual segmentation process. AVSegformer [13] proposes a transformer architecture that introduces audio features into the transformer decoder, enabling the network to attend to interested visual features selectively. CATR [23] proposes a combinatorial dependence fusion approach that comprehensively accounts for the spatial-temporal dependencies of audio-visual combination. Some methods [15, 30, 31] take advantage of the generative manners with latent diffusion model or variational auto-encoder to address AVS task. In addition, AV-SAM [33], GAVS [37], and BAVS [26] utilize the large foundation model to bootstrap audio-visual segmentation. Different from the above methods [13, 26, 37, 45], our proposed COMBO rethinks AVS from bilateral relations of three entanglements, which enhances the model's representation ability by exploring the pixel, modality and temporal inherent relationships.

3. Method

3.1. Bilateral Visual Features Extraction

As illustrated in Fig. 2, our method initiates with the extraction of visual features primarily because the audio-visual segmentation (AVS), as a dense prediction task, exhibits extensive pixel entanglement in visual perception. Recent studies [21, 22, 40] have demonstrated that the Segment Anything Model [20] exhibits robust generalization performance in segmentation tasks. Consequently, transferring the impressive capabilities of the foundation model to more complex visual tasks, such as AVS, presents an intriguing and valuable research question. The extension, however, is not straightforward. Although some methods [3, 33] attempt to fine-tune or concatenate the pre-trained SAM model for AVS, the limited capacity of the frozen foundation model restricts its performance to address the AVS task. Additionally, the AVS task aims to predict all sound targets per pixel, whereas the SAM model is only capable of generating class-agnostic masks without any audio guidance, thus demonstrating a significant disparity. Therefore, transferring the foundation model's knowledge to the AVS task presents a tough challenge.

Maskige as Prior Knowledge. To solve the aforementioned issues, we believe that a feasible strategy is to incorporate the knowledge of the foundation model into visual features as a pixel entanglement, which is memory-efficient. Specifically, as depicted in the left area of Fig. 2, we introduce a Maskige generator. Given the input image $x \in \mathbb{R}^{3\times H\times W}$ and the frozen foundation model, one can derive the class-agnostic masks $c \in \mathbb{R}^{K\times H\times W}$, where K represents the number of potential targets. It is essential to highlight that K is dynamic and varies with input images. Thus, we first amplify the quantity of class-agnostic masks from K to N with zero masks and obtain a series of binary masks, where N is predetermined. However, given that the output of SAM is a series of binary masks, it is difficult to integrate them into visual features. Consequently,

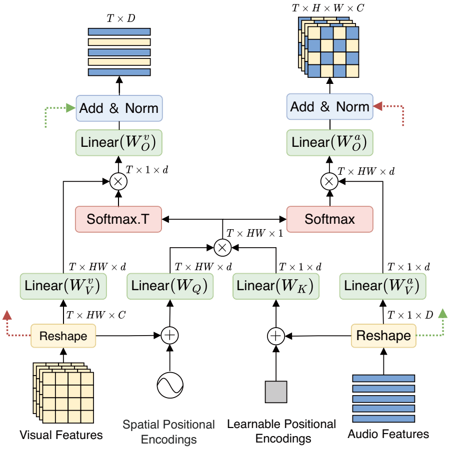


Figure 3. Illustration of Bilateral-Fusion Module (BFM). We input both visual and image signals, which are subsequently processed through bilateral attention to yield the fused visual and image features respectively. We omit the subscripts of H and W for better understanding. For enhanced visibility, the dashed line indicates a skip connection. Best viewed in color.

inspired by [6], we introduce Maskige $m \in \mathbb{R}^{3\times H\times W}$, which shares the exact dimensions as the input image, to integrate prior knowledge better. To efficiently generate Maskige m, we employ a random color encoding function $\mathcal{X}(\cdot) : \mathbb{R}^N \to \mathbb{R}^3$ that is capable of transforming the binary masks $c \in \mathbb{R}^{N\times H\times W}$ into Maskige $m \in \mathbb{R}^{3\times H\times W}$ without extra training. Specifically, $\mathcal{X}$ is designed to enhance the distinguishability of the Maskige and can be regarded as a linear layer $\mathcal{X}(c) = cA$, where $A \in \mathbb{R}^{N\times 3}$. To facilitate offline inference using the Maskige generator, the value of A is manually set appropriately without additional training. More details are in Appendix.

Siam-Encoder Module (SEM). To incorporate the image-like Maskiges as prior knowledge into input frames, we propose intertwining the features of Maskige and visual elements during the feature extraction stage. Accordingly, we design a Siam-Encoder Module, as depicted in Fig. 2. This module encompasses an Image encoder E_v and a Maskige encoder E_m, sharing a common framework. More precisely, for a short video clip with T frames $I \in \mathbb{R}^{T\times 3\times H\times W}$, the Maskiges can be generated using the Maskige generator, resulting in $M \in \mathbb{R}^{T\times 3\times H\times W}$. Subsequently, we extract multiple output features from both the image encoder and Maskige encoder, respectively. This process can be defined as follows:

$$F_{visual} = E_v(I; \theta_v),\ F_{visual} \in \{F_{v_i}\}_{i=1}^4, \tag{1}$$

$$F_{maskige} = E_m(M; \theta_m),\ F_{maskige} \in \{F_{m_i}\}_{i=1}^4, \tag{2}$$

in which F_{v_i} and $F_{m_i} \in \mathbb{R}^{T\times \frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C_i}$. C_i represents the dimension of the i-th stage output features. To integrate the Maskige features into COMBO, we introduce channel-weighted blocks that augment the original visual features. The formula can be written as follows:

$$F_{v_i} = F_{m_i}(\mathrm{GAP}(F_{m_i})W) + F_{v_i},\ i = \{1, 2, 3, 4\}, \tag{3}$$

where GAP stands for global average pooling, and $W \in \mathbb{R}^{C_i\times C_i}$ represents the linear weight. For simplicity, the bias is omitted in this context. After obtaining the Maskiges as prior information to boost pixel-level entanglement with visual features, the next critical aspect is exploring the modality entanglement between audio and visual signals.

3.2. Audio-Visual Bilateral Fusion

The relationship between any two modalities can be characterized as bilateral entanglement. For instance, an image can be described in text, and sound is inextricably linked with its visual counterpart. This entanglement among these distinct modalities provides an invaluable resource for researchers tackling multimodal tasks. Prior studies [38, 45] have overemphasized the influence of audio on visual features, thereby underestimating the significance of visual information to audio features. To address this imbalance, we propose a Bilateral-Fusion Module (BFM) in COMBO that surpasses a mere single fusion effect.

Audio Feature Extraction. For an audio clip corresponding to the input frames, we adopt VGGish [17] to extract audio features following [45]. Firstly, the audio clip is resampled to yield a 16kHz mono output $A_{mono} \in \mathbb{R}^{N_{samples}\times 96\times 64}$, where $N_{samples}$ is related to the duration of the audio. Then, a short-time Fourier transform is performed to yield a mel spectrum, denoted as $A_{mel} \in \mathbb{R}^{T\times 96\times 64}$. Finally, the mel spectrum is subsequently fed into the VGGish model, resulting in the extraction of audio features $F_a \in \mathbb{R}^{T\times D}$, where T denotes the number of frames and D represents dimension of the audio.

Bilateral-Fusion Module (BFM). We initially employ the pixel decoder [47] to gradually upsample visual features F_{v_i} derived from the SEM to further generate high-resolution per-pixel embeddings $P_i \in \mathbb{R}^{T\times\frac{H}{2^{i+1}}\times\frac{W}{2^{i+1}}\times C}$, $i \in \{1, 2, 3, 4\}$, where C denotes the output channel. Then, we design a Bilateral-Fusion Module (BFM) for constructing a bidirectional audio-visual mapping to assist with segmenting the sounding objects.

As shown in Fig. 3, our BFM utilizes audio features $F_a \in \mathbb{R}^{T\times D}$, in conjunction with the largest pixel-level embeddings $P_1 \in \mathbb{R}^{T\times H_1\times W_1\times C}$ as inputs which can propagate ample fine-grained semantic information to audio fea-

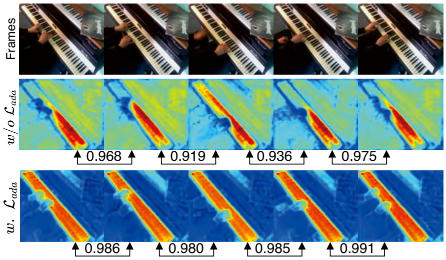


Figure 4. Illustration of the impact on Adaptive Inter-frame Consistency Loss. We visualize the heat map of the predicted masks without and with the consideration of $\mathcal{L}_{ada}$ based on the S4 subset. The results indicate that implementing $\mathcal{L}_{ada}$ promotes superior interframe consistency. Best viewed in color.

tures. Here, $H_1 = H/4, ; W_1 = W/4$. In order to incorporate both two signals, bilateral attention is designed within our BFM. Specifically, we initially add fixed sine spatial positional encodings and learnable positional encodings to P_1 and F_a, respectively. Next, in order to integrate audio-visual modalities in a memory-efficient way, our BFM comprises four point-wise linear layers that map P_1 and F_a to intermediate representations with dimension d. These representations share queries and keys with queries $Q = P_1 W_Q$, keys $K = F_a W_K$, visual values $V_v = P_1 W_V^v$, and audio values $V_a = P_1 W_V^a$. Following the mapping process, the bilateral attention is as follows:

$$P_1 = \text{Softmax}(QK^T/\sqrt{d})V_a + P_1, \tag{4}$$

$$F_a = \text{Softmax}((QK^T/\sqrt{d})^T)V_v + F_a, \tag{5}$$

where d is the embedding dimension. And $(QK^T/\sqrt{d})$ is calculated only once, which is more efficient.

After BFM, we proceed by expanding fused audio features F_a added with learnable embeddings within the transformer decoder as object queries. Additionally, following [10], we generate the output classes O^{cls} by incorporating the per-pixel embeddings P_4, P_3, P_2 into the transformer decoder. We acquire the predicted masks O^{mask} by multiplying the output embeddings from the transformer decoder with the fused embedding P_1.

3.3. Mining Temporal Relationships

Adaptive Inter-frame Consistency Loss. Temporal always implies a bilateral relationship in nature. For instance, in video clips, one can deduce the scenario of the current frame based on the past frame. Similarly, it is also feasible to predict the future frame based on the current frame. This interactive relations among frames can be construed as a type of temporal entanglement. To take advantage of this potential temporal entanglement, we introduce an adaptive inter-frame consistency loss for AVS. The similarity score for each successive frame can be calculated as follows:

$$\mathcal{S}_{t:t+1} = \cos(O_t^{mask}, O_{t+1}^{mask}), \tag{6}$$

where O_t^{mask} refers to the predicted masks at frame t, and $\cos(\cdot)$ symbolizes the cosine similarity function. The term $\mathcal{S}_{t:t+1}$ represents the similarity score between frames t and $t+1$. As illustrated in Fig. 4, it is evident that a significant similarity exists between distinct frames. Therefore, to leverage this prior information, we propose an adaptive inter-frame consistency loss, formulated as follows:

$$\mathcal{L}_{ada} = \sum_{t=1}^{T-1} \exp(\mathcal{S}_{t:t+1} - 1)(1 - \mathcal{S}_{t:t+1}), \tag{7}$$

where $\exp(\mathcal{S}_{t:t+1} - 1)$ represents adaptive weight. When the disparity between adjacent frames is substantial, the adaptive weight item is minimal, aligning with intuition.

3.4. Training and Inference

Overall Training Loss. The comprehensive training loss comprises three components: classification loss, mask loss, and adaptive inter-frame consistency loss, as previously discussed. The classification loss is formulated by a cross-entropy loss, denoted as $\mathcal{L}_{cls} = \mathcal{L}_{ce}$. The mask loss integrates the binary cross-entropy loss and the dice loss [32], and is depicted as $\mathcal{L}_{mask} = \mathcal{L}_{bce} + \mathcal{L}_{dice}$. Considering that in the AVS task, the ratio of segmented objects to the total image area is relatively small, employing dice loss allows the model to better concentrate on the foreground and minimizes distraction from the background. The overall training loss is expressed as follows:

$$\mathcal{L} = \lambda_{cls}\mathcal{L}_{cls} + \lambda_{mask}\mathcal{L}_{mask} + \lambda_{ada}\mathcal{L}_{ada}, \tag{8}$$

where λ_{cls}, λ_{mask}, and λ_{ada} are hyperparameters. More details about the λ parameters are available in Sec. 4.2.

Semantic Inference. After obtaining the predicted embeddings $O^{cls} \in \mathbb{R}^{T \times N_q \times (K_c+1)}$ and binary masks $O^{mask} \in \mathbb{R}^{T \times N_q \times H \times W}$, where K_c represents the total number of object classes and N_q is the number of object queries, we employ the identical post-processing as in [10] to yield the final semantic segmentation outputs. Specifically, we first calculate the output mask with classes $O = O^{cls} \times O^{mask} \in \mathbb{R}^{T \times (K_c+1) \times H \times W}$. Then, we execute $\arg\max$ and discard the *no object* class to obtain the ultimate results.

4. Experiments

4.1. AVSBench Datasets

We evaluate our proposed method on the AVSBench dataset [46], which consists of two scenarios: AVSBench-object and AVSBench-semantic.

Method	Backbone	S4		MS3	
		$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
LVS [5]	ResNet-18	37.9	51.0	29.5	33.0
MSSL [34]	ResNet-18	44.9	66.3	26.1	36.3
3DC [28]	ResNet-152	57.1	75.9	36.9	50.3
SST [12]	ResNet-101	66.3	80.1	42.6	57.2
iGAN [29]	ResNet-50	61.6	77.8	42.9	54.4
LGVT[41]	Swin-B	74.9	87.3	40.7	59.3
AVSBench [46]	ResNet-50	72.8	84.8	47.9	57.8
	PVT-v2	78.7	87.9	54.0	64.5
CSMF [3]	ViT-B	58.0	67.0	34.0	44.0
AVS-BiGen [15]	ResNet-50	74.1	85.4	45.0	56.8
	PVT-v2	81.7	90.4	55.1	66.8
CATR [23]	ResNet-50	74.8	86.6	52.8	65.3
	PVT-v2	81.4	89.6	59.0	70.0
DiffusionAVS [30]	ResNet-50	75.8	86.9	49.8	58.2
	PVT-v2	81.4	90.2	58.2	70.9
ECMVAE [31]	ResNet-50	76.3	86.5	48.7	60.7
	PVT-v2	81.7	90.1	57.8	70.8
BAVS [26]	ResNet-50	78.0	85.3	50.2	62.4
	PVT-v2	82.0	88.6	58.6	65.5
AVSegFormer [13]	ResNet-50	76.5	85.9	49.5	62.8
	PVT-v2	82.1	89.9	58.4	69.3
COMBO (ours)	**ResNet-50**	**81.7** (+3.7)	**90.1** (+4.8)	**54.5** (+2.7)	**66.6** (+1.3)
	PVT-v2	**84.7** (+2.6)	**91.9** (+2.0)	**59.2** (+0.2)	**71.2** (+1.2)

Table 1. Quantitative comparison results of different methods on AVSBench-object (Single-source, S4; Multi-source, MS3). We use the same backbones (ResNet-50 and PVT-v2) to demonstrate that our method outperforms other methods significantly.

Method	Backbone	AVSS	
		$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
3DC [28]	ResNet-18	17.3	21.6
AOT [39]	ResNet-50	25.4	31.0
AVSBench [46]	ResNet-50	20.2	25.2
	PVT-v2	29.8	35.2
BAVS [26]	ResNet-50	24.7	29.6
	PVT-v2	32.6	36.4
AVSegFormer [13]	ResNet-50	24.9	29.3
	PVT-v2	36.7	42.0
COMBO (ours)	**ResNet-50**	**33.3** (+8.4)	**37.3** (+8.0)
	PVT-v2	**42.1** (+5.4)	**46.1** (+4.1)

Table 2. Quantitative comparison results on AVSBench-semantic.

AVSBench-object. AVSBench-object [45] is an audio-visual dataset specifically designed for sound target segmentation. AVSBench-object includes two scenarios based on the number of audio sources in each frame: single sound source segmentation (S4) and multiple sound source segmentation (MS3). The S4 scenario incorporates 4,932 videos, with the ratio of train/validation/test split ratio configured at 70/15/15. This scenario is trained in a semi-supervised manner, wherein each video comprises five frames, but annotation during training is limited to the first frame only. Conversely, the MS3 scenario is characterized by multiple sound sources, including 424 videos, and maintains the same split ratios as in the S4 scenario. This scenario, unlike S4, employs a fully supervised training approach with all five frames being annotated.

Module	S4		AVSS	
	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
COMBO	**81.7**	**90.1**	**33.3**	**37.3**
w/o Siam-Encoder	80.6	88.7	31.9	35.7
w/o Bilateral-Fusion	81.1	89.9	33.1	36.7
w/o Inter-Frame Loss	81.0	89.8	33.0	37.1

Table 3. Ablation study of the various modules included in COMBO. We sequentially remove our proposed modules and compare their performance.

AVSBench-semantic. AVSBench-semantic [46] is an extension to AVSBench-object that incorporates additional semantic labels for the purpose of enhancing audio-visual semantic segmentation (AVSS). AVSBench-semantic includes a set of new multi-source videos as well as the original AVSBench-object videos, cumulatively accounting for a total of 11,356 videos spanning 70 categories. These videos are allocated as follows: 8,498 for training, 1,304 for validation, and 1,554 for testing. Additionally, the original videos have only been enhanced with semantic information, maintaining the same frames as prior. However, the new videos have been extended to 10 frames, increasing the difficulty due to the inclusion of extended audio-visual sequences.

4.2. Experimental Setup

Implementation Details. For fair comparison, we adopt the ImageNet pre-trained ResNet-50 [16] and Pyramid Vision Transformer (PVT-v2) [36] as the visual siam-encoders. All input frames are resized to 224×224. In terms of the proposal generator, we leverage the Semantic-SAM [22] to obtain class-agnostic masks. As for audio input, we take the Vggish encoder [17] pre-trained on AudioSet [14] to extract audio features. Following [10], the Multi-Scale Deformable Attention Transformer (MSDeformAttn) is our default pixel decoder. Besides, we adopt the standard transformer decoder, with $L = 3$ (i.e., a total of 9 layers) and $N_q = 100$ as the default. The hyperparameters are set as $\lambda_{cls} = 2$ and $\lambda_{mask} = 5$. For the inter-frame consistency loss, we set $\lambda_{ada} = 10$ for the AVSBench-object dataset, $\lambda_{ada} = 5$ for the AVSBench-semantic dataset due to longer frames per video. We calculate the inter-frame consistency loss only in the intermediate transformer decoder layer to prevent an over-dependence on temporal information. All models are trained using the Adam optimizer

SEM Module	S4		AVSS		#Params
	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	
$w.$ shared weights	81.5	90.2	31.1	34.4	135.5
w/o shared weights	**81.7**	**90.1**	**33.3**	**37.3**	158.9

Table 4. Ablation study of Siam-Encoder Module (SEM). We explore two configurations: using shared weights and using separate weights. #Params denotes the model parameters (M).

Fusion Mode	S4		AVSS	
	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
w/o fusion	81.1	89.9	33.1	36.7
visual fusion only	81.3	89.6	32.6	36.4
audio fusion only	81.3	**90.2**	33.2	36.7
bilateral fusion	**81.7**	90.1	**33.3**	**37.3**

Table 5. Ablation study of fusion mode. We conduct among four modes: no fusion, visual fusion only, audio fusion only, and our fusion mode.

with a learning rate of $1e-4$ and weight decay of 0.05. We train the S4 and AVSS subsets for 90k and MS3 for 20k iterations with a batch size of 8 on a single A100 40GB GPU.
Metrics. Folloing [45], we adopt two metrics to verify the effectiveness, namely, Jaccard index $\mathcal{J}$ and F-score $\mathcal{F}$. $\mathcal{J}$ computes the intersection over union (IoU) of the predicted segmentation and the ground truth mask. $\mathcal{F}$ considers both precision and recall, which is represented as $\mathcal{F}_\beta = \frac{(1+\beta^2)\times \text{presion}\times\text{recall}}{\beta^2\times\text{precision}+\text{recall}}$, where β^2 is set at 0.3. In our experiment, we use $\mathcal{M}_\mathcal{J}$ and $\mathcal{M}_\mathcal{F}$ to denote the mean metrics across the entire dataset.

4.3. Main Results

We conduct experiments on AVSBench-object and AVSBench-semantic datasets. As AVS is an emerging proposed problem recently introduced by [45], we compare our COMBO with some state-of-the-art methods from other related tasks, such as sound source localization (SSL) [5, 34], video object segmentation (VOS) [12, 28, 39], and salient object detection (SOD) [29, 41], all of which provide a comparative benchmark for our experiments. As evidenced in Tab. 1, COMBO demonstrates a substantial performance gap (+9.8 mIoU in S4; +16.3 mIoU in MS3) over other related methods, principally attributable to variances in setting specific task scenarios. We also compare our method against some recent state-of-the-art methods [13, 15, 23, 26, 30, 31, 45] that have been explicitly designed for audio-visual segmentation settings. On the AVSBench-object dataset, COMBO-R50 outperforms the current best performance by achieving 3.7 mIoU and 4.8 F-score improvements for S4 subset and 2.7 mIoU and 1.3 F-score improvements for MS3, while COMBO-PVT surpasses the top-performing method by 2.6 mIoU and 2.0 F-score for S4 and 0.2 mIoU and 1.2 F-score for MS3.

Besides, we compare the AVSBench-semantic dataset as displayed in Tab. 2, which presents a more challenging setting. Both COMBO-R50 and COMBO-PVT achieve significant results, with 8.4 and 5.4 mIoU improvements and significant F-score enhancements of 8.0 and 4.1, respectively. These experiments confirm that our COMBO model surpasses existing state-of-the-art methods across all subtasks, consequently setting a new benchmark for AVS.

Queries	S4		AVSS	
	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
all	80.9	89.9	30.7	34.2
add	**81.7**	**90.1**	**33.3**	**37.3**

Table 6. Ablation study of learnable queries.

λ_{ada}	S4		AVSS	
	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$	$\mathcal{M}_\mathcal{J}$	$\mathcal{M}_\mathcal{F}$
0	81.0	89.8	33.0	37.1
5	81.1	**90.1**	**33.3**	**37.3**
10	**81.7**	**90.1**	32.1	35.6
20	81.2	89.7	32.6	36.1

Table 7. Ablation study of the adaptive inter-frame consistency loss. λ_{ada} is the hyperparameter, while higher values constrain the output to be more similar.

4.4. Abaltion Study

In this section, we conduct ablation studies to verify the effectiveness of each essential design in the proposed COMBO. Specifically, we adopt ResNet-50 [16] as the backbone and carry out extensive experiments on the S4 and AVSS sub-tasks due to more videos in these tasks. Other training settings remain consistent with Sec. 4.2
Component analysis of COMBO. To validate the impact of our proposed method, we separately eliminate the Siam-Encoder Module (SEM), Bilateral-Fusion Module (BFM), and adaptive inter-frame consistency loss. As demonstrated in Tab. 3, the results indicate that COMBO has demonstrated superior influence on SEM with 1.4 mIoU improvement, particularly on the AVSS subset. Concurrently, our findings indicate that BFM is of substantial significance, demonstrating a performance enhancement of 0.6 mIoU over the S4 subset. More analysis are discussed later. In addition, we examine the effect of inter-frame consistency loss. The results reveal that our loss function contributes to performance improvements with 0.7 mIoU on the S4 subset. More details are provided subsequently.
Effects of Siam-Encoder Module (SEM). We first examine the significance of our SEM. Two designs are compared in Tab. 4: one with share weights and another with separate weights. When comparing shared and separate parameters, it is found that separable parameters can attain 0.2 mIoU on the S4 subset, but it is indeed more costly than the others.
Effects of Bilateral-Fusion Module (BFM). Moreover, we investigate the influence of our BFM. We compared our

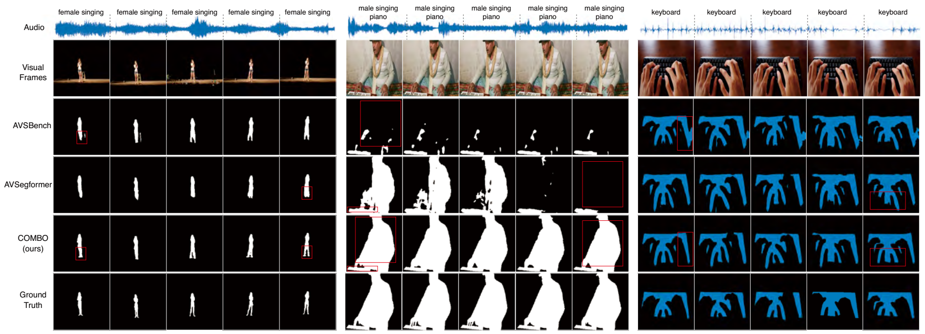


Figure 5. Comparison of Visual Examples on the AVSBench-object and AVSBench-semantic Datasets with AVSBench [46] and AVSegformer [13]. Wherein the leftmost example is derived from the S4 subset, the middle example is from the MS3 subset, and the rightmost example is from the AVSS subset. Red bounding boxes highlight the specific regions for comparison.

modules using different variants, which include without any fusion, video-only fusion (inject audio feature into visual), audio-only fusion (inject visual feature to audio), and our bilateral fusion. As demonstrated in Tab. 5, bilateral fusion achieves a performance improvement of 0.6 mIoU compared to the model without any fusion. It is worth noting that the transformer decoder inherently consists of a cross-attention function, potentially serving as a fusion process. Our model also realizes a performance improvement of 0.4 mIoU compared to only audio or visual fusion.

Effects of Audio Queries. As demonstrated in Tab. 6, *All* denotes the exclusive use of fused audio queries, and *Add* signifies the combination of fused audio queries and learnable queries. We first expand the audio features to the exact dimensions as learnable queries, then compare the experiments with the exclusive use of fused audio queries and the combination of fused audio queries and learnable queries. The results show that queries with *Add* have 0.8 and 2.6 mIoU improvements over *All* on S4 and AVSS subsets.

Effects of Adaptive Inter-frame Consistency Loss. To validate the effect of $\mathcal{L}_{ada}$, we adopt different values of λ_{ada}. As shown in Tab. 7, the experiments demonstrate that appropriate consistency constraints can enhance the model's performance. The appropriate value λ_{ada} can improve performance by 0.7 mIoU and 0.3 F-score for S4. In addition, the heat map of the predicted masks has been visualized in Fig. 4. It is observable that the use of $\mathcal{L}_{ada}$ facilitates the generation of a more distinct boundary output. Nevertheless, it is essential to note that exceedingly high values may lead to a decline in performance, given that the video does not exhibit complete consistency.

4.5. Qualitative Analysis

We provide a qualitative comparison between AVSBench [46], AVSegformer [13] and our proposed method on AVSBench-object and AVSBench-semantic datasets. As depicted in Fig. 5, our method, COMBO, exhibits superior audio-temporal and spatial localization quality, leading to better visualization and segmentation performance. For instance, in the case of the middle samples, our model accurately segments the singing man despite the presence of other sounds. Moreover, our method achieves more precise segmentation for background noise handling and provides richer details of the foreground in other examples.

5. Conclusion

We introduce a novel audio-visual transformer framework, termed COMBO, that archives state-of-the-art performance on AVSBench-object and AVSBench-semantic datasets. Contrary to previous methodologies that only factor in modality or temporal relations individually, our method explores multi-order bilateral relations for the first time, combining pixel entanglement, modality entanglement, and temporal entanglement. For these three kinds of entanglement, we propose Siam-Encoder Module (SEM), Bilateral-Fusion Module (BFM), and adaptive inter-frame consistency loss, respectively. Extensive experimental results verify the effectiveness of our proposed framework. We hope that our work will inspire further research in this significant and worthwhile field.

Acknowledgements

This research was supported by the Strategic Priority Research Program of Chinese Academy of Sciences (Grant No. XDB0500103), the National Natural Science Foundations of China (Grants No. 62076242, 62376267), the Pre-Research Project on Civil Aerospace Technologies (No. D030312), the National Defense Basic Scientific Research Program of China (No. JCKY2021203B063) and the innoHK project.

References

[1] Relja Arandjelovic and Andrew Zisserman. Look, listen and learn. In *Proceedings of the IEEE international conference on computer vision*, pages 609–617, 2017. 1

[2] Yusuf Aytar, Carl Vondrick, and Antonio Torralba. Soundnet: Learning sound representations from unlabeled video. *Advances in neural information processing systems*, 29, 2016. 1

[3] Swapnil Bhosale, Haosen Yang, Diptesh Kanojia, and Xiatian Zhu. Leveraging foundation models for unsupervised audio-visual segmentation. *arXiv preprint arXiv:2309.06728*, 2023. 2, 3, 6

[4] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020. 1

[5] Honglie Chen, Weidi Xie, Triantafyllos Afouras, Arsha Nagrani, Andrea Vedaldi, and Andrew Zisserman. Localizing visual sounds the hard way. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16867–16876, 2021. 1, 2, 6, 7

[6] Jiaqi Chen, Jiachen Lu, Xiatian Zhu, and Li Zhang. Generative semantic segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7111–7120, 2023. 4

[7] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L. Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(4):834–848, 2018. 2

[8] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. arxiv 2017. *arXiv preprint arXiv:1706.05587*, 2, 2019. 2

[9] Bowen Cheng, Alex Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. *Advances in Neural Information Processing Systems*, 34:17864–17875, 2021. 2

[10] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1280–1289, 2022. 5, 6, 1

[11] Bowen Cheng, Ishan Misra, Alexander G Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1290–1299, 2022. 2

[12] Brendan Duke, Abdalla Ahmed, Christian Wolf, Parham Aarabi, and Graham W. Taylor. Sstvos: Sparse spatiotemporal transformers for video object segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5912–5921, 2021. 6, 7

[13] Shengyi Gao, Zhe Chen, Guo Chen, Wenhai Wang, and Tong Lu. Avsegformer: Audio-visual segmentation with transformer. *arXiv preprint arXiv:2307.01146*, 2023. 2, 3, 6, 7, 8

[14] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 776–780, 2017. 6

[15] Dawei Hao, Yuxin Mao, Bowen He, Xiaodong Han, Yuchao Dai, and Yiran Zhong. Improving audio-visual segmentation with bidirectional generation. *arXiv preprint arXiv:2308.08288*, 2023. 2, 3, 6, 7

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 6, 7

[17] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017. 4, 6

[18] Di Hu, Rui Qian, Minyue Jiang, Xiao Tan, Shilei Wen, Errui Ding, Weiyao Lin, and Dejing Dou. Discriminative sounding objects localization via self-supervised audiovisual matching. *Advances in Neural Information Processing Systems*, 33:10077–10087, 2020. 1, 2

[19] Jitesh Jain, Jiachen Li, Mang Tik Chiu, Ali Hassani, Nikita Orlov, and Humphrey Shi. Oneformer: One transformer to rule universal image segmentation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2989–2998, 2023. 2

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollar, and Ross Girshick. Segment anything. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4015–4026, 2023. 2, 3, 1

[21] Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. *arXiv preprint arXiv:2308.00692*, 2023. 3

[22] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 2, 3, 6, 0, 1

[23] Kexin Li, Zongxin Yang, Lei Chen, Yi Yang, and Jun Xun. Catr: Combinatorial-dependence audio-queried transformer for audio-visual video segmentation. *arXiv preprint arXiv:2309.09709*, 2023. 2, 3, 6, 7

[24] Liunian Harold Li, Pengchuan Zhang, Haotian Zhang, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and

Jianfeng Gao. Grounded language-image pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10965–10975, 2022. 2

[25] Yuhang Ling, Yuxi Li, Zhenye Gan, Jiangning Zhang, Mingmin Chi, and Yabiao Wang. Hear to segment: Unmixing the audio to guide the semantic segmentation, 2023. 2

[26] Chen Liu, Peike Li, Hu Zhang, Lincheng Li, Zi Huang, Dadong Wang, and Xin Yu. Bavs: Bootstrapping audio-visual segmentation by integrating foundation knowledge. *arXiv preprint arXiv:2308.10175*, 2023. 2, 3, 6, 7

[27] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015. 2

[28] Sabarinath Mahadevan, Ali Athar, Aljosa Osep, Laura Leal-Taixé, Bastian Leibe, and Sebastian Hennen. Making a case for 3d convolutions for object segmentation in videos. In *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, 2020. 6, 7

[29] Yuxin Mao, Jing Zhang, Zhexiong Wan, Yuchao Dai, Aixuan Li, Yunqiu Lv, Xinyu Tian, Deng-Ping Fan, and Nick Barnes. Generative transformer for accurate and reliable salient object detection. *arXiv preprint arXiv:2104.10127*, 2021. 6, 7

[30] Yuxin Mao, Jing Zhang, Mochu Xiang, Yunqiu Lv, Yiran Zhong, and Yuchao Dai. Contrastive conditional latent diffusion for audio-visual segmentation. *arXiv preprint arXiv:2307.16579*, 2023. 3, 6, 7

[31] Yuxin Mao, Jing Zhang, Mochu Xiang, Yiran Zhong, and Yuchao Dai. Multimodal variational auto-encoder based audio-visual segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 954–965, 2023. 2, 3, 6, 7

[32] Fausto Milletari, Nassir Navab, and Seyed-Ahmad Ahmadi. V-net: Fully convolutional neural networks for volumetric medical image segmentation. In *2016 Fourth International Conference on 3D Vision (3DV)*, pages 565–571, 2016. 5

[33] Shentong Mo and Yapeng Tian. Av-sam: Segment anything model meets audio-visual localization and segmentation. *arXiv preprint arXiv:2305.01836*, 2023. 2, 3

[34] Rui Qian, Di Hu, Heinrich Dinkel, Mengyue Wu, Ning Xu, and Weiyao Lin. Multiple sound sources localization from coarse to fine. In *Computer Vision – ECCV 2020*, pages 292–308, Cham, 2020. Springer International Publishing. 1, 2, 6, 7

[35] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 7242–7252, 2021. 2

[36] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvt v2: Improved baselines with pyramid vision transformer. *Computational Visual Media*, 8(3):415–424, 2022. 6, 1

[37] Yaoting Wang, Weisong Liu, Guangyao Li, Jian Ding, Di Hu, and Xi Li. Prompting segmentation with sound is generalizable audio-visual source localizer. *arXiv preprint arXiv:2309.07929*, 2023. 2, 3

[38] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in Neural Information Processing Systems*, 34:12077–12090, 2021. 2, 4

[39] Zongxin Yang, Yunchao Wei, and Yi Yang. Associating objects with transformers for video object segmentation. *Advances in Neural Information Processing Systems*, 34:2491–2502, 2021. 6, 7

[40] Chaoning Zhang, Dongshen Han, Yu Qiao, Jung Uk Kim, Sung-Ho Bae, Seungkyu Lee, and Choong Seon Hong. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv preprint arXiv:2306.14289*, 2023. 2, 3, 1

[41] Jing Zhang, Jianwen Xie, Nick Barnes, and Ping Li. Learning generative vision transformer with energy-based latent space for saliency prediction. In *Advances in Neural Information Processing Systems*, pages 15448–15463. Curran Associates, Inc., 2021. 6, 7

[42] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6230–6239, 2017. 2

[43] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip H.S. Torr, and Li Zhang. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6877–6886, 2021. 2

[44] Bolei Zhou, Hang Zhao, Xavier Puig, Tete Xiao, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, 127:302–321, 2019. 1

[45] Jinxing Zhou, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, and Yiran Zhong. Audio–visual segmentation. In *European Conference on Computer Vision*, pages 386–403. Springer, 2022. 1, 2, 3, 4, 6, 7

[46] Jinxing Zhou, Xuyang Shen, Jianyuan Wang, Jiayi Zhang, Weixuan Sun, Jing Zhang, Stan Birchfield, Dan Guo, Lingpeng Kong, Meng Wang, et al. Audio-visual segmentation with semantics. *arXiv preprint arXiv:2301.13190*, 2023. 1, 3, 5, 6, 8

[47] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable {detr}: Deformable transformers for end-to-end object detection. In *International Conference on Learning Representations*, 2021. 4

Appendix

Supplementary Material

This appendix presents additional materials and results.

First, we describe the detailed training settings in Sec. A. Then, we give further descriptions of our proposed COMBO in Sec. B to enhance comprehension. Next, we provide more ablation studies for COMBO in Sec. C. Finally, a series of visual results are presented in Sec. D.

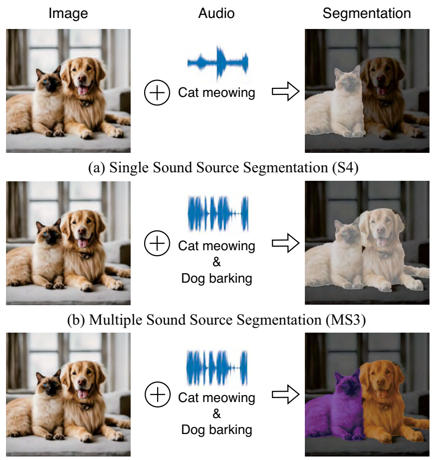
A. More Implementation Details

This section further explains the experimental details, which can be found in Tab. I. It should be noted that the batch size pertains to the number of videos entered, thereby implying $bs \times T$ frames per iteration, where bs denotes the batch size. Furthermore, pertaining to the AVSS task, given that the input video comprises varying numbers of frames, the number of frames within the batch size was dynamically altered without the need for padding zeros.

B. Further Descriptions

B.1. Task Description

We begin by providing an illustration description for the Audio-Visual Segmentation (AVS). As depicted in Fig. I, the purpose of AVS is to segment all sound objects pixel-by-pixel. There are two datasets included: (1) *AVSBench-object*. This dataset encompasses single source sound segmentation (S4) and multiple sound sources segmentation (MS3), as shown in Fig. I (a) and Fig. I (b), respectively. In other words, objects (such as a dog or cat) in an image can be categorized into class-agnostic masks based on their corresponding sounds. (2) *AVSBench-semantic*. In addition to the above, objects that emit sounds also carry class semantic information, a concept known as audio-visual semantic segmentation (AVSS), as shown in Fig. I (c). This represents a more challenging dataset due to its complexity.



(a) Single Sound Source Segmentation (S4)

(b) Multiple Sound Source Segmentation (MS3)

(c) Audio-Visual Semantic Segmentation (AVSS)

Figure I. Illustration of the three sub-tasks in AVSBench-object and AVSBench-semantic datasets. Best viewed in color.

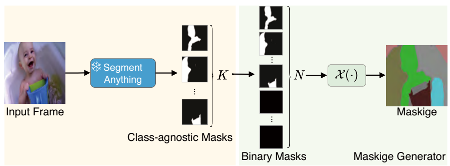


Figure II. Illustration of the Proposal Generator. The proposal generator consists of two parts: the yellow area on the left mainly contains a frozen foundation model for generating class-agnostic masks, and the green area on the right is used to convert the masks into Maskige, also called Maskige generator.

B.2. Proposal Generator

As shown in Fig. II, we provide a more detailed explanation of the proposal generator proposed in COMBO. Initially, we obtain class-agnostic masks denoted as $c \in \mathbb{R}^{K \times H \times W}$ from the input frame, using a pre-existing foundation model [22], where K denotes the number of potential targets. Subsequently, a Maskige generator, which is part of the proposal generator, is introduced to convert the class-agnostic masks $c \in \mathbb{R}^{K \times H \times W}$ into Maskige, denoted as $m \in \mathbb{R}^{3 \times H \times W}$ without the need for additional training. Particularly, as K is dynamic and fluctuates according to input frames, we amplify the quantity of class-agnostic masks to N using zero masks, thereby deriving a series of binary masks $c \in \mathbb{R}^{N \times H \times W}$, where N is a predetermined number and $N \geq K$. Next, considering that c denotes a series of binary masks that are challenging to incorporate into visual features, we utilize a random color encoding function $\mathcal{X}(\cdot) : \mathbb{R}^{N \times H \times W} \rightarrow \mathbb{R}^{3 \times H \times W}$ to convert the binary masks $c \in \mathbb{R}^{N \times H \times W}$ into Maskige $m \in \mathbb{R}^{3 \times H \times W}$. Formally, we define $\mathcal{X}(c) = cA$, where $A \in \mathbb{R}^{N \times 3}$. To facilitate the proposal generator offline, the value of A is manually set appropriately. Specifically, we set $N = 100$, a value

Settings	S4	MS3	AVSS
Resolution $H \times W$	224×224	224×224	224×224
Number of frames T	5	5	5 & 10
Data augmentation	horizontal flip & color aug	horizontal flip & color aug	horizontal flip & color aug
Audio dimension D	128	128	128
Embedding dimension d	256	256	256
Number of queries N_q	100	100	100
Number of transformer decoders L	3	3	3
Loss coefficient λ_{cls}	2.0	2.0	2.0
Loss coefficient λ_{mask}	5.0	5.0	5.0
Loss coefficient λ_{ada}	10.0	10.0	5.0
Batch size	8	8	8
Optimizer	AdamW	AdamW	AdamW
Learning rate	0.0001	0.0001	0.0001
Weight decay	0.05	0.05	0.05
Iterations	90k	20k	90k

Table I. Detailed settings. This table provides a detailed overview of the specific settings used for each sub-task.

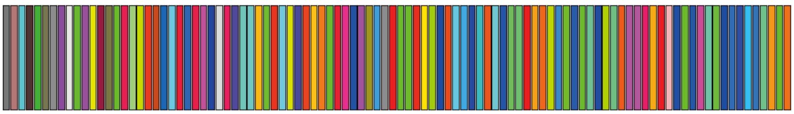


Figure III. Visualization of matrix $A \in \mathbb{R}^{100\times3}$. Each color bar has an RGB value of dimension 3, and there are 100 color bars.

considerably larger than K. To enhance the distinctiveness between various targets, as illustrated in Fig. III, we use the first 100 color mappings of the color mapping relationship in ADE20K dataset [44] as the parameters of matrix A. Further visualizations on Maskiges are available in Sec. D.

C. More Results

C.1. Effects of The Foundation Model

We continue our exploration by investigating the impact of the various foundation models in the proposal generator on performance. For comparison, we select the original Segment Anything Model (SAM) [20], the superior-performing Semantic-SAM [22], and the lighter MobileSAM [40] as the foundation models of the proposal generator to evaluate the performance alongside the backbone of PVT-v2 [36] on the S4 subset. As illustrated in Tab. II, the results depict a minimal performance discrepancy among the different foundational models. Nevertheless, it is evident that the performance of our model improves with the enhancement of the foundational model's ability. Accordingly, we choose the Semantic-SAM [22] as the foundation model of the proposal generator. In addition, we also provide a comparison of the visualizations of the Maskiges generated by different foundational models in Sec. D.

S4	SAM [20]	Semantic-SAM [22]	MobileSAM [40]
$\mathcal{M}_{\mathcal{J}}$	84.4	**84.7**	84.1
$\mathcal{M}_{\mathcal{F}}$	91.8	**91.9**	91.6

Table II. Impact of the different foundation models on COMBO.

N_q	S4		AVSS	
	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$	$\mathcal{M}_{\mathcal{J}}$	$\mathcal{M}_{\mathcal{F}}$
100	**81.7**	**90.1**	**33.3**	**37.3**
200	81.5	**90.1**	32.0	35.6
300	81.3	89.9	31.4	34.7

Table III. Impact of the number of queries on COMBO.

C.2. Effects of The Number of Queries

We present additional ablation studies concerning the number of queries, denoted as N_q, in our approach, as illustrated in Tab. III. In order to examine the influence of the query count on the model's performance, we conducted a series of experiments using varying quantities of queries within the transformer decoder, specifically 100, 200, and 300. Our findings suggest that 100 queries are sufficient, given the infrequency of maximum concurrent classes in an AVS task. Therefore, we established the default number of queries as 100 following [10].

D. More Qualitative Results

In this section, we introduce additional qualitative results of our proposed COMBO, along with its intermediate visualizations, to illustrate the effectiveness of our module. The quality of the generation of the Maskige is crucial to the assistance of our model. Therefore, we first show some exam-

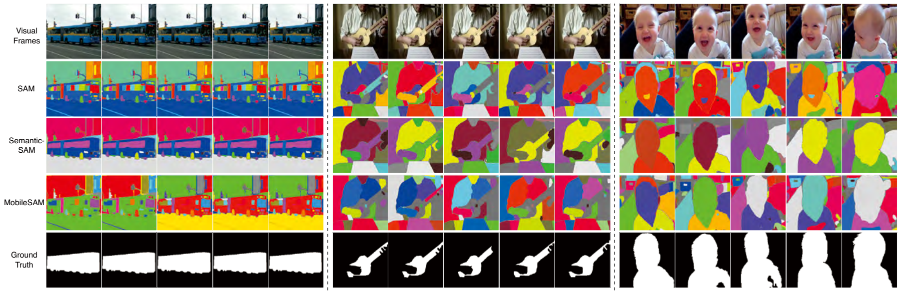


Figure IV. Visualization of Maskiges. The Maskiges are generated by proposal generator with various foundation models [20, 22, 40] and the color encoding function $\mathcal{X}(\cdot)$.

ples sampled from various sub-tasks with foundation models [20, 22, 40] in Fig. IV. It is evident that all foundational models exhibit exceptional proficiency in segmenting class-agnostic targets. However, given that Semantic-SAM [22] can produce a more complete target mask, we select it as the foundation model of our proposed proposal generator. Besides, we also provide additional heat maps of the predicted masks to illustrate the effectiveness of the adaptive inter-frame consistency loss, $\mathcal{L}_{ada}$. As depicted in Fig. V, when adjacent frames are similar, our loss module enables predicted masks to produce more accurate results. Conversely, when adjacent frames are dissimilar, our module can avoid mutual interference between adjacent frames due to the existence of adaptive. Finally, given that the audio-visual segmentation task is a video task with audio input, we present a visual comparison between our method and baseline in a video format, which can be reviewed on our project page.

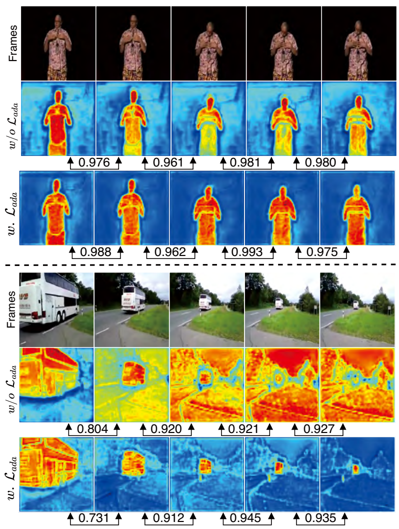


Figure V. Visualization of the heat map of the predicted masks without and with the consideration of $\mathcal{L}_{ada}$ based on the S4 subset.

Decision Focused Causal Learning for Direct Counterfactual Marketing Optimization

Hao Zhou*
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
Meituan
Beijing, China
zhouhao29@meituan.com

Rongxiao Huang*
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
rxhuang@smail.nju.edu.cn

Shaoming Li
Meituan
Beijing, China
shaoming.li@outlook.com

Guibin Jiang
Meituan
Beijing, China
jiangguibin@meituan.com

Jiaqi Zheng†
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
jzheng@nju.edu.cn

Bing Cheng
Meituan
Beijing, China
bing.cheng@meituan.com

Wei Lin
Meituan
Beijing, China
lwsaviola@163.com

Abstract

Marketing optimization plays an important role to enhance user engagement in online Internet platforms. Existing studies usually formulate this problem as a budget allocation problem and solve it by utilizing two fully decoupled stages, i.e., machine learning (ML) and operation research (OR). However, the learning objective in ML does not take account of the downstream optimization task in OR, which causes that the prediction accuracy in ML may be not positively related to the decision quality.

Decision Focused Learning (DFL) integrates ML and OR into an end-to-end framework, which takes the objective of the downstream task as the decision loss function and guarantees the consistency of the optimization direction between ML and OR. However, deploying DFL in marketing is non-trivial due to multiple technological challenges. Firstly, the budget allocation problem in marketing is a 0-1 integer stochastic programming problem and the budget is uncertain and fluctuates a lot in real-world settings, which is beyond the general problem background in DFL. Secondly, the counterfactual in marketing causes that the decision loss cannot be directly computed and the optimal solution can never be obtained, both of which disable the common gradient-estimation approaches in DFL. Thirdly, the OR solver is called frequently to compute the decision loss during model training in DFL, which produces huge computational cost and cannot support large-scale training data. In this paper, we propose a decision focused causal learning framework (DFCL) for direct counterfactual marketing optimization, which overcomes the above technological challenges. Both offline experiments and online A/B testing demonstrate the effectiveness of DFCL over the state-of-the-art methods. Currently, DFCL has been deployed in several marketing scenarios in Meituan, one of the largest online food delivery platform in the world.

CCS Concepts

• **Computing methodologies** → **Machine learning approaches**; • **Applied computing** → **Electronic commerce**.

Keywords

Causal Inference, Decision Focused Learning, Marketing Optimization

ACM Reference Format:
Hao Zhou, Rongxiao Huang, Shaoming Li, Guibin Jiang, Jiaqi Zheng, Bing Cheng, and Wei Lin. 2024. Decision Focused Causal Learning for Direct Counterfactual Marketing Optimization. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3637528.3672353

*Both authors contributed equally to this research.
†Corresponding author.

KDD '24, August 25–29, 2024, Barcelona, Spain.

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3672353

1 Introduction

Conducting marketing campaigns is a popular and effective way used by online Internet platforms to boost user engagement and revenue. For example, coupons in Taobao[37] can stimulate user activity, dynamic pricing in Airbnb[36] and discounts in Uber[11] encourage users to use the products.

Despite the incremental revenues, marketing campaigns could incur significant costs. In order to be sustainable, a marketing campaign is usually conducted under a limited budget. In other words, only a portion of individuals (e.g., shops or goods) may receive marketing treatments due to a limited budget. Hence, assigning the appropriate marketing treatments to different individuals is essential for the effectiveness of a marketing campaign since users would respond differently to various promotional offers. Such decision problems can be formalized as resource allocation problems and have been investigated for decades.

The mainstream solution for these problems is a two-stage method [2, 3, 11, 32, 38]. In the first stage, the individual-level (incremental) response under different treatments is predicted using ML models. The second stage is OR, and the predictions are fed into the combination optimization algorithms to achieve optimal overall revenue. However, the objectives of the two stages are not aligned: the former focuses on the predictive precision of the ML models, while the latter focuses on the quality of decisions. The method has some defects due to the isolation of ML and OR. First, the prediction precision of ML models has no strict positive correlation with the quality of the final decision. This is because standard loss functions (e.g., mean square error, cross-entropy error) do not take the interplay between the predictions into account, which can affect decision quality. Second, ML models often fall short of perfect precision, and the complex operations performed on the predictions in OR lead to the amplification or accumulation of prediction errors. Thus, the two-stage method usually obtains suboptimal decisions and is even inferior to heuristic strategies in some scenarios.

Recently, Decision-Focused Learning (DFL) [4, 12, 20, 26] has received increasing attention as an appropriate alternative to the two-stage method. The paradigm integrates prediction and optimization into an end-to-end system, which effectively aligns the objectives of both stages and achieves better performance on many challenging tasks. The key idea is to train ML models using a loss function that directly measures the quality of the decisions obtained from the predictions. Specifically, the ML models are trained under the predict-then-optimize framework [12], which (1) makes predictions based on historical data, (2) solves the optimization problem based on the predictions, and (3) computes the decision loss to update the ML model parameters using stochastic gradient descent (SGD).

Nevertheless, deploying DFL in marketing is non-trivial due to the following challenges.

Uncertainty of constraints. Most prior works of DFL have investigated the optimization problem where the unknown parameters appear in the objective function. The reason behind this is that the unknown parameters in the constraints lead to uncertainty in the solution space, and the optimal solution derived from the predictions may not be feasible under the real parameters. Within the constraints of our optimization problem, there are two distinct forms of uncertainty: intrinsic and extrinsic. The inherent uncertainty in the constraints refers to the costs consumed by the individuals under different treatments, which can be predicted based on historical data. Extrinsic uncertainty is the frequently changing marketing budget, determined by the external environment. An ML model is required to guarantee superior performance under different marketing budgets. Thus, our optimization objective is the effectiveness of the decision under any budget, and the optimization problem is a 0-1 integer stochastic programming.

Counterfactuals in marketing. Computing decision loss in marketing is challenging due to the presence of counterfactuals. Specifically, observing the values and costs of an individual under different treatments is impossible because the individual can only receive one treatment, which is also called the fundamental problem of causal inference [27]. In addition, the optimal solution of the optimization problem cannot be obtained based on offline data due to the counterfactuals, which disables the common gradient-estimation methods (e.g., SPO [12], LODL [28], LTR [18]) in DFL.

Computational cost of large-scale dataset. Computational cost is one of the major roadblocks for DFL involving large-scale optimization. As mentioned above, DFL integrates prediction and optimization into an end-to-end system, where the solver will be called frequently during training to solve the optimization problem. Therefore, the computational cost of DFL is high, leading prior works to investigate toy-level problems with few decision variables. In real-world applications, we need to train models for tens of millions of data, which is unsupportable by traditional DFL.

In this paper, we propose Decision-Focused Causal-Learning (DFCL) to address the above challenges. The main contributions of this work can be summarized as follows.

Generalization. In order to address both endogenous uncertainty (cost of individual consumption) and exogenous uncertainty (marketing budget) in the constraints, the uncertainty constraints are transformed into the objective function of the dual problem using Lagrangian duality theory. The optimization objective of the dual problem is then used as the decision loss. Moreover, we prove that the budget of the primal problem corresponds to the Lagrange multipliers of the dual problem, and thus optimizing the dual solution under different Lagrange multipliers is equivalent to optimizing the quality of decisions under different budgets.

Counterfactual Decision Loss. Optimal solution, decision loss, and gradient cannot be computed directly due to the existence of counterfactuals in marketing, thus we propose two solutions: (1) surrogate loss function and (2) black-box optimization based on the Expected Outcome Metric (EOM) [2, 39, 40]. Inspired by Policy Gradient in Reinforcement Learning, we transform the decision problem of discrete actions into the problem which maximizes expected revenue under the probability distribution of the actions, and combine the Maximum Entropy Regularizer as well as the Lagrangian duality theory to give two kinds of surrogate loss functions: Policy Learning Loss and Maximun Entropy Regularized Loss. We theoretically guarantee continuity, convexity and equivalence of the surrogate loss functions. For black-box optimization, we employ the EOM to give an unbiased estimation of the decision loss and improve the finite difference strategy to develop an efficient estimator of the gradient, which enables us to update the model parameters using gradient descent.

Scalability. In real-world applications, we need to train models for tens-of-millions of data. The surrogate functions proposed in this paper are smooth convex loss functions with almost the same computational efficiency as the two-stage method. For black-box optimization, frequently solving the optimization problem after perturbation incurs huge computational overhead. We accelerate the problem solving and modify the gradient estimator using the Lagrangian duality theory, which significantly improves the training efficiency and reduces the training time from hour-level to second-level per epoch compared to the black-box method based on the primal problem.

We conduct extensive experiments to evaluate the performance of DFCL. Both offline experiments and online A/B testing show the superior performance of our method over state-of-the-art baselines. DFCL is deployed to several scenarios in Meituan, an online food delivery platform, and achieves significant revenue.

2 Related Works

Two-stage Method. The mainstream solution to the resource allocation problem in marketing usually follows the two-stage paradigm [2, 3, 32, 38], which handles the two stages—machine learning (ML) and operation research (OR)—independently. In the first stage, the uplift models are deployed to predict the treatment effects of individuals. Some prior works have focused on the design of uplift models, including Meta-Learners [16, 23], Causal Forests [2, 5, 31, 39], representation learning [13, 29, 35] and rank model [8, 17]. However, standard loss functions (such as mean square error and cross-entropy error) for training uplift models do not take the downstream OR into account. In the second stage, the resource allocation problem is represented as a multi-choice knapsack problem (MCKP), which is NP-Hard and efficiently solved based on Lagrangian duality theory [2, 3, 32, 40].

Decision-Focused Learning(DFL). DFL is considered an appropriate alternative to the two-stage method, which integrates prediction and optimization into an end-to-end system. Since computing decision loss requires solving optimization problems, which usually involve non-differentiable operations, automatic differentiation in machine learning frameworks (such as Pytorch [25] and Tensorflow [1]) cannot give the correct gradient. Three categories of approaches to gradient computation are proposed by prior DFL works: analytical smoothing of optimization mappings, smoothing by random perturbations, and differentiation of surrogate loss function. The first method derives the analytic gradient of decision loss by using the KKT condition or the homogenous self-dual formulation, including Optnet [4], DQP [10], QPTL [33], and IntOpt [19]. However, when the optimization problem is discrete, the method requires a continuous relaxation of the primal problem, which results in suboptimality. A potential resolution is to consider every optimization problem as a black-box optimization and utilize random perturbations, such as DBB [26], DPO [7], and I-MLE [24], to generate approximate gradient. Furthermore, the decision loss is typically discontinuous and nonconvex, so some of these works suggest convex surrogate functions, including SPO [12], LTR [18], NCE [22], and LODL [28], for the decision loss.

The most related works to ours are DRP [11] and DPM [40]. DRP proposes to directly learn ROI (ratio between incremental values and incremental costs) to rank and choose individuals in the binary treatment setting. It has been shown by [40] that the loss function in DRP is unable to converging to a stable extreme point. DPM extend the idea to the multiple treatments setting by directly learning the unbiased estimation of the decision factor in OR. However, the construction of the decision factor in multi-treatment setting relies on the law of diminishing marginal utility, which does not hold strictly in some scenarios of marketing.

3 Problem Formulation

In this section, we formalize the resource allocation problem and introduce the overall optimization objective in marketing.

We start with a common marketing scenario that has M types of treatments. Let r_{ij} and c_{ij} be the revenue and cost of individual i under treatment j, respectively. The objective is to find an optimal allocation strategy for a group of individuals to maximize the revenue of the platform, given a limited budget B. Therefore, the budget allocation problem with multiple treatments (MTBAP) can be formulated as an integer programming problem (1):

$$\begin{aligned}\max_{z} \quad F(z,B) \;&=\; \sum_i \sum_j z_{ij} r_{ij},\\ \text{s.t.} \quad &\sum_i \sum_j z_{ij} c_{ij} \le B,\\ &\sum_j z_{ij} = 1, \forall i,\\ &z_{ij} \in \{0,1\}, \forall i, j,\end{aligned} \tag{1}$$

where $z_{ij} \in \{0, 1\}$ is the decision variable to denote whether to assign treatment j to individual i. The first constraint is the limitation of the budget and the second one requires that only one treatment is assigned to each individual. Since the budget B fluctuates a lot in real-world settings, the objective is regared as a function of the budget and the overall marketing goal is to maximize revenue $F(z, B)$ within arbitrary given budget.

Combinatorial Optimization Algorithm. When the value of r_{ij} and c_{ij} are known in advance, MTBAP is a classical multiple choice knapsack problem (MCKP) [30], which remains NP-Hard. Existing studies usually solve this problem by using greedy algorithms or Lagrangian duality theory, both of which can provide a approximation ratio of

$$\rho = 1 - \frac{\max_{ij} r_{ij}}{\text{OPT}},$$

where OPT is the optimal solution. In the above equation, $\max_{ij} r_{ij}$ refers to the revenue of one individual (e.g., one user or one shop), which is negligible compared with OPT that is the sum of the revenue of all the individuals in marketing. Therefore, it indicates that both greedy algorithms and Lagrangian duality theory can achieve near optimal performance, which are also the most common algorithms to solve MTBAP in marketing. The details can be found in existing works, which will not be discussed in this paper.

Model Prediction. However, the value of r_{ij} and c_{ij} are unknown during decision making in real-world applications, which are usually replaced with the prediction value. Therefore, how to make the prediction of r_{ij} and c_{ij} plays important roles in marketing effectiveness, which will be addressed in this paper. In the traditional two-stage approaches, the machine learning (ML) model is trained with the direction of optimizing prediction accuracy, which may be not consistent with the direction of optimizing decision quality. In the following sections, we mainly focus on the design of the loss function, to make a tradeoff between the prediction accuracy and the decision quality.

4 Learning Framework of DFCL

In the learning framework, the loss function includes two parts, the prediction loss and the decision loss, i.e.,

$$\mathcal{L}_{DFCL} = \alpha \mathcal{L}_{PL} + \mathcal{L}_{DL}.$$

The former $\mathcal{L}_{PL}$ aims to decrease the prediction error, which contributes to improving the generalization ability of a ML model. The latter $\mathcal{L}_{DL}$ measures the decision quality of the downstream task, which is exactly the objective of marketing optimization.

4.1 Prediction Loss

In the traditional two-stage method, the ML model is trained by minimizing the difference between the predictions $\hat{r}, \hat{c}$ and the ground-truth values r, c. For instance, in a regression problem, the mean squared error (MSE) is usually used to train the ML model:

$$\mathcal{L}_{MSE}(r, c, \hat{r}, \hat{c}) = \frac{1}{NM} \sum_i \sum_j (r_{ij} - \hat{r}_{ij})^2 + (c_{ij} - \hat{c}_{ij})^2. \quad (2)$$

Due to the counterfactuals in marketing, observing the revenue or cost of an individual under different treatments is impossible because each individual can only receive one treatment, which is also called the fundamental problem of causal inference.

Definition 1 (The fundamental problem of causal inference). *For all individuals, only one of all the potential outcomes under different treatments can be observed in real-world data.*

Therefore, $\mathcal{L}_{MSE}$ cannot be directly computed according to Eq. 2 since r_{ij_1} and r_{ij_2} (or equivalently, c_{ij_1} and c_{ij_2}) cannot be simultaneously observed for any $j_1 \neq j_2$. To solve this problem, we first formulate the training data set and then develop a equivalent prediction loss in marketing.

Data Set. Suppose that there is a data set of size N collected from random control trials (RCT). The i-th sample is denoted by $(x_i, t_i, r_{it_i}, c_{it_i})$, where x_i is the features of individual i, t_i is the assigned treatment, and r_{it_i}, c_{it_i} are the revenue and the cost of individual i under treatment t_i. Denote the count of the samples (individuals) receiving treatment j by N_j.

Prediction Loss. Given the above data set, we present the prediction loss in marketing in Eq. (3). Theorem 1 presents the equivalency and the detailed proof can be found in Appendix A.

$$\mathcal{L}_{PL}(r, c, \hat{r}, \hat{c}) = \frac{1}{M} \sum_i \frac{1}{N_{t_i}} [(r_{it_i} - \hat{r}_{it_i})^2 + (c_{it_i} - \hat{c}_{it_i})^2]. \quad (3)$$

Theorem 1. *The prediction loss $\mathcal{L}_{PL}$ is equivalent to $\mathcal{L}_{MSE}$, i.e.,*

$$\mathcal{L}_{PL} = \mathcal{L}_{MSE}.$$

4.2 Decision Loss

As is stated in Sec. 3, the ground-truth value of r and c are usually unknown in advance, which are replaced with the prediction $\hat{r}$ and $\hat{c}$ during decision making. Therefore, denote the original optimization problem $F(z, B)$ by $F(z, B, \hat{r}, \hat{c})$, and the solution $z^*(B, \hat{r}, \hat{c})$ is obtained by solve MTBAP $F(z, B, \hat{r}, \hat{c})$, i.e.,

$$z^*(B, \hat{r}, \hat{c}) = \arg\max_z F(z, B, \hat{r}, \hat{c}).$$

The objective value achieved by the current solution $z^*(B, \hat{r}, \hat{c})$ can be expressed with the ground-truth value of r as

$$\sum_i \sum_j r_{ij} z^*_{ij}(B, \hat{r}, \hat{c}).$$

The decision loss under budget B is defined as the negative of the objective value with ground-truth r and predicted decision $z^*(B, \hat{r}, \hat{c})$, i.e.,

$$\mathcal{L}_{DL}(B, r, c, \hat{r}, \hat{c}) = -\sum_i \sum_j r_{ij} z^*_{ij}(B, \hat{r}, \hat{c}).$$

As is descripted in Sec. 3, the budget B fluctuates a lot in real-world settings and the overall marketing objective is to maximize the revenue under arbitrary budget. Therefore, the decision loss in marketing is defined as

$$\begin{aligned}\mathcal{L}_{DL}(r, c, \hat{r}, \hat{c}) &= \int_0^\infty \mathcal{L}_{DL}(B, r, c, \hat{r}, \hat{c}) dB \\ &= \int_0^\infty -\sum_i \sum_j r_{ij} z^*_{ij}(B, \hat{r}, \hat{c}) dB.\end{aligned}$$

For ease of calculation, we can also discretize the budget and compute the decision loss by

$$\mathcal{L}_{DL}(r, c, \hat{r}, \hat{c}) = \sum_B \mathcal{L}_{DL}(B, r, c, \hat{r}, \hat{c}).$$

4.3 Learning Framework

Algorithm 1 presents the framework of decision focused causal learning (DFCL). The most crucial step in this framework is the gradient estimation of $\mathcal{L}_{DFCL}$ in line 10 of Algorithm 1. However, it is non-trival in marketing due to the following technological challenges, i.e., uncertainty of constraints, counterfactual and computation cost. In this paper, we will show how to address these challenges and how to deploy DFCL in marketing optimization.

Algorithm 1 Decision Focused Causal Learning (DFCL)

Input: training data D $\equiv \{(x_i, t_i, r_{it_i}, c_{it_i})\}_{i=1}^N$

1: Initialize ω
2: **for** each epoch **do**
3: $\quad \hat{r}, \hat{c} = m_\omega(x)$.
4: $\quad \mathcal{L}_{PL} = \sum_i \frac{1}{N_{t_i}} [(r_{it_i} - \hat{r}_{it_i})^2 + (c_{it_i} - \hat{c}_{it_i})^2]$.
5: $\quad$ **for** each budget B **do**
6: $\quad\quad z^*(B, \hat{r}, \hat{c}) = \arg\max_z F(z, B, \hat{r}, \hat{c})$.
7: $\quad\quad \mathcal{L}_{DL}(B) = -\sum_i \sum_j r_{ij} z^*_{ij}(B, \hat{r}, \hat{c})$.
8: $\quad \mathcal{L}_{DL} = \sum_B \mathcal{L}_{DL}(B)$
9: $\quad \mathcal{L}_{DFCL} = \alpha \mathcal{L}_{PL} + \mathcal{L}_{DL}$.
10: $\quad \omega = \omega - \eta \frac{\partial \mathcal{L}_{DFCL}}{\partial \omega}$

5 Gradient Estimation of DFCL

The loss of DFCL consists of the prediction loss and the decision loss. The former is a continuously differentiable function whose gradient can be directly computed. Hence, the gradient estimation of the decision loss is the key focus of this section. Firstly, we introduce the equivalent dual decision loss to remove the uncertain constraints and reduce the computation cost of combinatorial optimization algorithms. Secondly, we develop two surrogate loss functions and improve the black-box optimization algorithm to provide a gradient estimation of the dual decision loss.

5.1 Dual Decision Loss

Based on the Lagrangian duality theory, the upper bound of the original problem $F(z, B, r, c)$ can be obtained by solving the following dual problem (4).

$$\begin{aligned}
&\min_{\lambda\geq 0}\begin{pmatrix} \max\limits_{z} \lambda B + \sum_i \sum_j (r_{ij} - \lambda c_{ij}) z_{ij} \\ s.t.\ \sum_j z_{ij} = 1, \forall j \\ z_{ij} \in \{0,1\}, \forall i, j \end{pmatrix} \\
&= \min_{\lambda\geq 0} \max_{z} H(z, \lambda, B, r, c) \\
&= \min_{\lambda\geq 0} G(\lambda, B, r, c). \qquad (4)
\end{aligned}$$

The optimal Lagrange multiplier λ^* for the dual problem (4) can be obtained by using a gradient descent algorithm or a binary search method with the terminal condition of $B - \sum_i \sum_j c_{ij} z_{ij} \leq \epsilon$ or $\lambda \leq \epsilon$. In addition, an approximately optimal solution for the original problem can be derived by maximizing $H(z, \lambda^*, B, r, c)$. Theorem 2 presents the relationship between the original problem $F(z, B, r, c)$ and the dual problem $G(\lambda, B, r, c)$.

Theorem 2. *Denote by $F_c(z, B, r, c)$ the relaxation form of $F(z, B, r, c)$ where the decision variables z are relaxed to continuous variables (i.e., $z_{ij} \in [0,1]$ for $\forall i, j$). Denote the optimal solution by*

$$\begin{aligned}
z_c^*(B, r, c) &= \arg\max_{z} F_c(z, B, r, c), \\
z^*(B, r, c) &= \arg\max_{z} F(z, B, r, c), \\
\lambda^*(B, r, c) &= \arg\min_{\lambda\geq 0} G(\lambda, B, r, c).
\end{aligned}$$

Given the optimal Lagrange multiplier λ^*, an approximation solution for the original problem can be derived by

$$z^d(\lambda^*, B, r, c) = \arg\max_{z} H(z, \lambda^*, B, r, c).$$

Based on these definitions, we claim that λ^* is monotonic decreasing with the increment of the budget B, and we have

$$\begin{aligned}
F(z^d, B, r, c) &\leq F(z^*, B, r, c) \\
&\leq F_c(z_c^*, B, r, c) \\
&= G(\lambda^*, B, r, c) \\
&\leq F(z^d, B, r, c) + \max_{ij} r_{ij}
\end{aligned}$$

The detailed proof can be found in [14]. Given the optimal λ^*, Theorem 2 indicates that the solution $z_d(\lambda^*, B, r, c)$ obtained by maximizing $H(z, \lambda^*, B, r, c)$ is approximately optimal with an apprimation ratio of

$$\begin{aligned}
\rho = \frac{F(z^d, B, r, c)}{F(z^*, B, r, c)} &\geq \frac{F(z^*, B, r, c) - \max_{ij} r_{ij}}{F(z^*, B, r, c)} \\
&= 1 - \frac{\max_{ij} r_{ij}}{F(z^*, B, r, c)} \\
&\approx 1
\end{aligned}$$

The last equality holds because $F(z^*, B, r, c)$ is the sum of the revenue of millions of individuals in marketing, which means that $F(z^*, B, r, c) \gg \max_{ij} r_{ij}$.

Therefore, instead of the original problem, the optimization of the dual problem $H(z, \lambda^*, B, r, c)$ is taken as the learning objective, which we call the dual decision loss. Given the optimal λ^* and the prediction value $\hat{r}, \hat{c}$, the solution $z^d(\lambda^*, B, \hat{r}, \hat{c})$ is obtained by maximizing $H(z, \lambda^*, B, \hat{r}, \hat{c})$, i.e.,

$$z^d(\lambda^*, B, \hat{r}, \hat{c}) = \arg\max_{z} H(z, \lambda^*, B, \hat{r}, \hat{c}).$$

Notice that $\lambda^* B$ can be taken as a constant, and removing it from $H(z, \lambda^*, B, \hat{r}, \hat{c})$ does not influence z^d. Therefore, the solution z^d can be rewritten as

$$z^d(\lambda^*, \hat{r}, \hat{c}) = \arg\max_{z} H(z, \lambda^*, \hat{r}, \hat{c}),$$

where $H(z, \lambda^*, \hat{r}, \hat{c})$ is the form of $H(z, \lambda^*, B, \hat{r}, \hat{c})$ after removing $\lambda^* B$. The dual decision loss achieved by the current solution z_d is

$$\mathcal{L}_{DDL}(\lambda^*, B, r, c, \hat{r}, \hat{c}) = -(\lambda^* B + \sum_i \sum_j (r_{ij} - \lambda^* c_{ij}) z_{ij}^d(\lambda^*, \hat{r}, \hat{c})).$$

Similarly, since λ^* and B is irrelevant to the prediction value $\hat{r}, \hat{c}$, $\lambda^* B$ can be regarded as a constant and removed from the dual decision loss. According to Theorem 2, λ^* is monotonic decreasing with the increment of the budget B and there is an unique λ^* for the dual problem when given the budget B. Therefore, the decision loss $\mathcal{L}_{DL}(r, c, \hat{r}, \hat{c})$ in the original problem under arbitrary budget B can be transformed to the dual decision loss $\mathcal{L}_{DDL}(r, c, \hat{r}, \hat{c})$ under arbitrary Lagrange multiplier λ^*, i.e.,

$$\begin{aligned}
\mathcal{L}_{DDL}(r, c, \hat{r}, \hat{c}) &= \int_0^\infty \mathcal{L}_{DDL}(\lambda^*, r, c, \hat{r}, \hat{c}) d\lambda^* \\
&= \int_0^\infty \mathcal{L}_{DDL}(\lambda, r, c, \hat{r}, \hat{c}) d\lambda \\
&= -\int_0^\infty \sum_i \sum_j (r_{ij} - \lambda c_{ij}) z_{ij}^d(\lambda, \hat{r}, \hat{c}) d\lambda.
\end{aligned}$$

By discretizing the Lagrange multiplier λ, the dual decision loss can be computed by

$$\mathcal{L}_{DDL}(r, c, \hat{r}, \hat{c}) = \sum_{\lambda} \mathcal{L}_{DDL}(\lambda, r, c, \hat{r}, \hat{c}).$$

5.2 Policy Learning Loss

Notice that the dual problem $H(z, \lambda, \hat{r}, \hat{c})$ can be solved by

$$\begin{aligned}
\max_{z} H(z, \lambda, \hat{r}, \hat{c}) &= \begin{pmatrix} \max\limits_{z} \sum_i \sum_j (\hat{r}_{ij} - \lambda \hat{c}_{ij}) z_{ij} \\ s.t.\ \sum_j z_{ij} = 1, \forall j \\ z_{ij} \in \{0,1\}, \forall i, j \end{pmatrix} \\
&= \sum_i \max_j (\hat{r}_{ij} - \lambda \hat{c}_{ij})
\end{aligned}$$

Therefore, the solution $z^d(\lambda,\hat{r},\hat{c}) = \arg\max_z H(z,\lambda,\hat{r},\hat{c})$ can be expressed by

$$z^d_{ij}(\lambda,\hat{r},\hat{c}) = \mathbb{I}_{j=\arg\max_j \hat{r}_{ij}-\lambda\hat{c}_{ij}}.$$

Hence, the dual decision loss is rewritten as

$$\mathcal{L}_{DDL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda\sum_i\sum_j (r_{ij}-\lambda c_{ij})\mathbb{I}_{j=\arg\max_j \hat{r}_{ij}-\lambda\hat{c}_{ij}}$$

However, $\mathcal{L}_{DDL}(r,c,\hat{r},\hat{c})$ is not differentiable with respect to $\hat{r}$ and $\hat{c}$ due to the indicator function. Instead, we utilize a softmax function to smooth the dual decision loss, i.e.,

$$\mathcal{L}'_{DDL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda\sum_i\sum_j (r_{ij}-\lambda c_{ij})\frac{\exp(\hat{r}_{ij}-\lambda\hat{c}_{ij})}{\sum_k \exp(\hat{r}_{ik}-\lambda\hat{c}_{ik})} \quad (5)$$

Let $p_{ij}(\lambda,\hat{r},\hat{c}) = \exp(\hat{r}_{ij}-\lambda\hat{c}_{ij})/\sum_k \exp(\hat{r}_{ik}-\lambda\hat{c}_{ik})$ be the probability of assigning treatment j to individual i. Take $r_{ij}-\lambda c_{ij}$ as the reward of assigning treatment j to individual i. Hence, minimizing $\mathcal{L}'_{DDL}(r,c,\hat{r},\hat{c})$ is equivalent to maximizing the expected reward of policy $\pi = p_{ij}(\lambda,\hat{r},\hat{c})$ under different Lagrange multipliers. Therefore, $\mathcal{L}'_{DDL}$ is also called the policy learning loss.

Due to the counterfactual in marketing, $\mathcal{L}'_{DDL}(r,c,\hat{r},\hat{c})$ cannot be directly computed by Eq. (5) in training data sets. Instead, we propose a surrogate loss, i.e.,

$$\mathcal{L}_{PLL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda\sum_i \frac{N}{N_{t_i}}(r_{it_i}-\lambda c_{it_i})\frac{\exp(\hat{r}_{it_i}-\lambda\hat{c}_{it_i})}{\sum_j \exp(\hat{r}_{ij}-\lambda\hat{c}_{ij})}.$$

Theorem 3 presents the equivalence between the original dual decision loss $\mathcal{L}_{DDL}$ and the surrogate policy learning loss L_{PLL}. The detailed proof can be found in Appendix B.

Theorem 3. *$\mathcal{L}_{DDL}, \mathcal{L}'_{DDL}$ and $\mathcal{L}_{PPL}$ are equivalent, i.e.,*

$$\mathcal{L}_{PLL}(\lambda,\hat{r},\hat{c}) = \mathcal{L}'_{DDL}(\lambda,\hat{r},\hat{c})$$

and

$$\min_{\hat{r},\hat{c}} \mathcal{L}_{PLL}(\lambda,\hat{r},\hat{c}) = \min_{\hat{r},\hat{c}} \mathcal{L}_{DDL}(\lambda,\hat{r},\hat{c}).$$

5.3 Maximum Entropy Regularized Loss

In order to obtain a differentiable closed form of $z^d(\lambda,\hat{r},\hat{c})$ with respect to $\hat{r}$ and $\hat{c}$, we relax the discrete constraint $z \in \{0,1\}$ to a continuous one $x \in [0,1]$ and add a maximum entropy regularizer to the objective function in $H(z,\lambda,\hat{r},\hat{c})$. Hence, $H(z,\lambda,\hat{r},\hat{c})$ is transformed to a nonlinear convex function, i.e.,

$$\begin{aligned}\max_z \ & \sum_i\sum_j(\hat{r}_{ij}-\lambda\hat{c}_{ij})z_{ij} - \tau\sum_i\sum_j z_{ij}\ln z_{ij},\\ s.t. \ & \sum_j z_{ij} = 1, \forall i,\\ & z_{ij} \in [0,1],\end{aligned}$$

where τ denotes the penalty weight. The Lagrange relaxation function can be further rewritten as

$$L(z,\beta) = \sum_{i=1}^N\sum_{j=1}^M (r_{ij}-\lambda c_{ij})z_{ij} - \tau\sum_{i=1}^N\sum_{j=1}^M z_{ij}\ln z_{ij} - \sum_i \beta_i(1-\sum_j z_{ij}),$$

where β is the dual variables on the equality constraint. When $\frac{\partial L(z,\beta)}{\partial z} = 0$ and $\frac{\partial L(z,\beta)}{\partial \beta} = 0$, the optimal solution is obtained by

$$z^d_{ij} = \frac{\exp[(\hat{r}_{ij}-\lambda\hat{c}_{ij})/\tau]}{\sum_k \exp[(\hat{r}_{ik}-\lambda\hat{c}_{ik})/\tau]},$$

which is continuously differentiable with respect to $\hat{r}$ and $\hat{c}$. Hence, the dual decision loss can be rewritten as

$$\mathcal{L}''_{DDL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda\sum_i\sum_j (r_{ij}-\lambda c_{ij})\frac{\exp[(\hat{r}_{ij}-\lambda\hat{c}_{ij})/\tau]}{\sum_k \exp[(\hat{r}_{ik}-\lambda\hat{c}_{ik})/\tau]}$$

Similarly, $\mathcal{L}''_{DDL}$ cannot be directly computed due to the counterfactual in marketing. We propose a surrogate loss $\mathcal{L}_{MERL}(r,c,\hat{r},\hat{c})$ as follows, which we call the maximum entropy regularized loss,

$$\mathcal{L}_{MERL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda\sum_i \frac{N}{N_{t_i}}(r_{it_i}-\lambda c_{it_i})\frac{\exp[(\hat{r}_{it_i}-\lambda\hat{c}_{it_i})/\tau]}{\sum_j \exp[(\hat{r}_{ij}-\lambda\hat{c}_{ij})/\tau]}.$$

Notice that $\mathcal{L}_{PLL}$ is a special case of $\mathcal{L}_{MERL}$, where the solution z^d can be regarded as a temperature softmax function in $\mathcal{L}_{MERL}$.

5.4 Improved Finite-Difference Strategy

In addition to constructing surrogate loss functions, we can also use the Expected Outcome Metric (EOM) [2, 39, 40] to give an unbiased estimate of the decision loss and leverage black-box optimization for decision-focused learning.

EOM is a commonly used method for offline strategy evaluation based on randomized dataset. Given a batch of N random samples and model predictions $\hat{r}$ and $\hat{c}$, an arbitrary allocation strategy $z(\hat{r},\hat{c})$ can be evaluated: (1) find the set of individuals whose received treatment is equal to the treatment in the allocation strategy $z(\hat{r},\hat{c})$, (2) then empirically estimate their per capita revenue and per capita cost:

$$\bar{r}(r,c,\hat{r},\hat{c}) = \frac{1}{N}\sum_i \frac{1}{p_{t_i}} r_{t_i}\mathbb{I}_{t_i=\arg\max_j\ z_{ij}},$$

$$\bar{c}(r,c,\hat{r},\hat{c}) = \frac{1}{N}\sum_i \frac{1}{p_{t_i}} c_{t_i}\mathbb{I}_{t_i=\arg\max_j\ z_{ij}},$$

where p_{t_i} denotes the probability that a treatment is equal to t_i in the randomized dataset. For the primal MCKP with budget B, we can use binary search to empirically estimate the per capita revenue under a per capita budget $\frac{B}{N}$ as is shown in Appendix C.

Therefore, we can redefine the decision loss as follows:

$$\mathcal{L}_{DL}(r,c,\hat{r},\hat{c}) = -\sum_B \bar{r}(B,r,c,\hat{r},\hat{c}).$$

Since the computation of $\bar{r}(B,r,c,\hat{r},\hat{c})$ involves many nondifferentiable operations, we consider them as black-box functions and estimate the gradient by perturbation. Using the finite difference strategy, the gradient of the decision quality with respect to $\hat{r}_{ij}$ is estimated as:

$$\frac{\partial\mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial\hat{r}_{ij}} = \frac{\mathcal{L}_{DL}(r,c,\hat{r}+e_{ij}h,\hat{c}) - \mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{h},$$

where h is a small constant, and $e^{ij} \in \{0,1\}^{N\times M}$ is a matrix where only the element in the i-th row and j-th column is 1, and all other elements are 0. The gradient term $\frac{\partial\mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial\hat{c}_{ij}}$ can be computed similarly. We estimate the gradient by perturbing the predictions one by one and obtain the gradient matrix $\frac{\partial\mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial\hat{r}}$

and $\frac{\partial \mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial \hat{c}} \in R^{N \times M}$. Finally, we derive the following loss function, which is able to train the ML model via gradient descent:

$$\mathcal{L}_{FDL}(r,c,\hat{r},\hat{c}) = \sum_i \sum_j \frac{\partial \mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial \hat{r}_{ij}} \hat{r}_{ij} + \frac{\partial \mathcal{L}_{DL}(r,c,\hat{r},\hat{c})}{\partial \hat{c}_{ij}} \hat{c}_{ij}.$$

Since the perturbations are performed one by one, $\bar{r}(B,r,c,\hat{r},\hat{c})$ requires frequent evaluation, leading to a considerable time complexity of black-box optimization based on the primal MCKP [34]. In practice, we find that the number of samples N tends to be in the millions or even tens of millions, so the time consumption for a training epoch reaches the level of hours. A possible approach is to only perturb some of the samples by sampling, but this may incur the loss of much of the gradient information.

Instead, we accelerate the problem solving and modify the gradient estimator by using Lagrangian duality theory. Since the budget B in the primal MCKP corresponds one-to-one to the λ in the duality problem, the dual decision loss can be redefined as:

$$\mathcal{L}_{DDL}(r,c,\hat{r},\hat{c}) = -\sum_\lambda \bar{r}(\lambda,r,c,\hat{r},\hat{c}) - \lambda \bar{c}(\lambda,r,c,\hat{r},\hat{c}).$$

Although we avoid solving the primal MCKP, it is still necessary to frequently evaluate the per capita revenue and per capita cost after perturbation under multiple Lagrangian multipliers. We observe that the decision making is independent for each individual thanks to the decomposition of the Lagrangian duality theory. Thus, for each sample, the smallest perturbation that causes a change in the dual decison loss is first calculated, and the loss after the perturbation is obtained by correcting only the original result. Appendix D provides details of the modified gradient estimator, which greatly reduces the computational overhead. Finally, the black-box optimization loss function can be rewritten as

$$\mathcal{L}_{IFDL}(r,c,\hat{r},\hat{c}) = \sum_i \sum_j \frac{\partial \mathcal{L}_{DDL}(r,c,\hat{r},\hat{c})}{\partial \hat{r}_{ij}} \hat{r}_{ij} + \frac{\partial \mathcal{L}_{DDL}(r,c,\hat{r},\hat{c})}{\partial \hat{c}_{ij}} \hat{c}_{ij}.$$

It is sufficient to support model training on tens of million of data since the computational cost of incremental updating $L_{DDL}(r,c,\hat{r},\hat{c})$ after perturbation is much less than that of re-evaluating it. To improve numerical stability in training, we truncate the perturbation matrix $P \in \mathbb{R}^{N \times M}$. Further, the loss function can be smoothed using Softmax to reduce the difficulty of training.

$$\mathcal{L}_{IFDL-Softmax}(r,c,\hat{r},\hat{c}) = \sum_i \sum_j \frac{\partial \mathcal{L}_{DDL}(r,c,\hat{r},\hat{c})}{\partial a_{ij}} a_{ij},$$

where $a_{ij} = Softmax(\hat{r}_{ij} - \lambda \hat{c}_{ij})$, and only a is perturbed for gradient estimation and no longer for $\hat{r}, \hat{c}$.

6 Evaluation

In this section, we will conduct large-scale offline and online experiments to compare our methods with other benchmarks to validate their performance.

6.1 Offline Experiment

6.1.1 Dataset. Two types of datasets are provided in this paper.

- **CRITEO-UPLIFT v2.** This public dataset is provided by the AdTech company Criteo in the AdKDD'18 workshop[9]. The dataset contains 13.9 million samples collected from a random control trial (RCT) that prevents a random part of users from being targeted by advertising. Each sample has 12 features, 1 binary treatment indicator and 2 response labels(visit/conversion). In order to study resource allocation problem under limited budget using the dataset, we follow[40] and take the visit/conversion label as the cost/value respectively. We randomly sample 70% samples for training and the remaining samples for test.
- **Marketing data.** Discounting is a common marketing campaign in Meituan, an online food delivery platform. We conduct a two-week RCT to collect data in this platform. The online shops on the platform offer daily discounts to users. Note that to avoid price discrimination, the discount of a shop is the same for all individuals, but it changes randomly each day and varies from shop to shop. The data in the first week is used for training and the other for test. The discount $T \in \{0, 5, 10, 15, 20\}$ is taken as the treatment, where $T = t$ means t% off for each order whose price meets a given threshold. The dataset contains 2.8 million samples, and each sample has 107 features, 1 treatment label and 2 response labels (daily cost/orders).

6.1.2 Evaluation Metrics. Multiple evaluation metrics are provided for offline evaluation in this experiment. In addition to adopting the evaluation metrics commonly used in two-stage models, such as Logloss and MSE, we also use the following metrics for policy evaluation with counterfactuals, which are more significant.

- **AUCC (Area under Cost Curve).** A common metric used in existing works [2, 11, 40], which is designed for evaluating the performance to rank ROI of individuals in the binary treatment setting. We use the metric to compare the performance of different methods in CRITEO-UPLIFT v2.
- **EOM (Expected Outcome Metric).** EOM is also commonly used in [2, 39, 40]. Based on RCT data, an unbiased estimation of the expected outcome (per-capita revenue/per-capita cost) for arbitrary budget allocation policy can be obtained. The details of EOM are shown in Sec. 5.4. We use the metric to compare the performance of different methods in Marketing data.

6.1.3 Benchmarks. For each dataset in this paper, multiple models and algorithms are implemented and taken as benchmarks.

- **TSM-SL.** The two-stage method is mentioned in many exsting works[2, 3, 32, 38]. In the first stage, a well-trained S-Learner model is used to predict the response (revenue/cost) of individuals under different treatments. In the second stage, we find the optimal budget allocation solution for an MCKP formulation based on the predictions.
- **TSM-CF.** Also a two-stage method, the difference with TSM-SL is that instead of S-learner, we use a Causal Forests [5] to predict the incremental response in the first stage. It is implemented here base on EconML packages [6], which can support binary treatment and multiple treatments.
- **DPM.** This method[40] designs the decision factor for the MCKP, and proposes a surrogate loss to directly learn the decision factor.
- **CN.** This method[32] imposes a monotonic constraint between outcome predictions and treatments, which is particularly useful for ITE estimation under multiple treatments. The method is trained with MSE loss and evaluated only on marketing data, which is a multi-treatment experiment.

- **CN+DFCL-PL.** The constraint network is trained with Decision-Focused Causal Learning (DFCL) loss, which comprises MSE loss ($\mathcal{L}_{PL}$) and policy learning loss ($\mathcal{L}_{PLL}$).
- **DFCL-PL.** The DFCL method based on policy learning loss proposed in this paper.
- **DFCL-MER.** The DFCL method proposed in this paper utilizes the surrogate loss derived by Maximun Entropy Regularizer
- **DFCL-IFD.** The DFCL method proposed in this paper for gradient estimation using the improved finite difference strategy.

6.1.4 Implementation Details.

- **CRITEO-UPLIFT v2.** For the baseline methods (TSM-SL, TSM-CF and DPM), we cite the results directly from[40]. The DFCL model uses the same DNN architecture with a shared layer that is a single-layer MLP of dimension 128 and four head networks that are two-layer MLPs of dimension [64, 1]. Except for the final output layer, the remaining layers use ReLU activations. For DFCL-MER, we set the temperature $\tau = 3$. Our models are trained for 40 epochs with the Adam optimizer [15]. In order to accelerate the training, the first twenty epochs are warmstarting [21] using the cross-entropy loss, and then the models are trained using the DFCL loss.
- **Marketing data.** In the multi-treatment experiment, the models need to predict the revenue and cost under five treatments. TSM-SL, CN, CN+DFCL-PL, DFCL-PL, DFCL-MER and DFCL-IFD use the same DNN architecture: a 4 layers MLP (64-32-32-10). The first five outputs of the models are the predicted revenue, and the remaining outputs are the predicted cost. For DFCL-MER, we set the temperature $\tau = 0.01$. For DPM, a S-Learner model is trained using the customized loss proposed in [40] to directly predict marginal utility under different treatments. The DPM model has 4 layers of MLP (64-32-32-1), with the last layer using a sigmoid activation and each of the remaining layers with ReLU activations. All neural network-based models are trained for 500 epochs using the Adam optimizer. For TSM-CF, we set $n\_estimators = 256$, $min\_sample\_leaf = 300$ and $depth = 24$.

All experiments are run on AMD EPYC 7502P Rome 32x@ 2.50GHz processor with 64GB memory.

Table 1: Comparison of common metrics, noting that DPM and TSM-CF predict the decision factor and the incremental intervention effect, respectively, and thus do not apply to these two metrics.

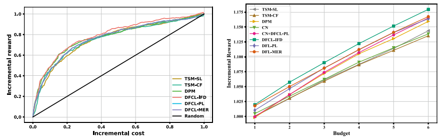
Model	CRITEO-UPLIFT v2	Marketing data
	Logloss	MSE
TSM-SL	0.2165 ± 0.0001	0.2625 ± 0.0009
CN	/	0.2639 ± 0.0012
CN+DFCL-PL	/	0.2703 ± 0.0015
DFCL-PL	0.2186 ± 0.0008	0.2678 ± 0.0010
DFCL-MER	0.2178 ± 0.0012	0.2650 ± 0.0005
DFCL-IFD	0.2170 ± 0.0003	0.2642 ± 0.0009

6.2 Experimental Results

6.2.1 Overall performance. In Table 1, we present the prediction loss of different models on the two datasets. Clearly, the two-stage method performs best on common metrics, which minimizes MSE or Logloss on the training set. However, what we really focus on is the decision quality of predictions. Fig. 1a and Table 2 present the comparison between our proposed methods and other benchmarks in CRITEO-UPLIFT v2 on AUCC [11], which represents the decision quality under binary treatments. We can see that DFCL-IFD achieves the best performance, DFCL-PL, and DFCL-MER perform similarly to DPM, and the two-stage methods perform the worst.

Table 2: AUCC(CRITEO-UPLIFT v2)

Model	AUCC	Improvement
TSM-SL	0.7561 ± 0.0113	/
TSM-CF	0.7558 ± 0.0012	-0.03%
DPM	0.7739 ± 0.0002	+2.35%
DFCL-PL	0.7713 ± 0.0025	+2.01%
DFCL-MER	0.7727 ± 0.0015	+2.20%
DFCL-IFD	0.7859 ± 0.0021	+3.94%



(a) AUCC(CRITEO-UPLIFT v2) (b) EOM (Marketing data)

Figure 1: Offline experiment results

In marketing data, we use EOM method to calculate per-capita orders and per-capita budgets based on predictions. The results are shown in Table 3 and Fig. 1b. Our models significantly outperform the baseline models in terms of per-capita orders at all per-capita budgets. DPM is on par with the two stage methods in the low per-capita budgets and outperforms them in the high per-capita budgets. CN has a marginal improvement of 0.16% compared to the two-stage methods. Further evaluation is carried out on the model trained with DFCL loss, which comprises MSE loss ($\mathcal{L}_{PL}$) and policy learning loss ($\mathcal{L}_{PLL}$). The integration of policy learning loss yielded a notable enhancement in performance, with the constrained network showing a significant increase of 1.26%. These findings suggest that our proposed DFCL approach is versatile and can be integrated into existing methodologies. Interestingly, the constraint network combined with policy learning loss (CN + DFCL-PL) did not outperform DFCL-PL alone. We hypothesize that this may be due to the predefined constraints within the network, which potentially restrict the expansiveness of the decision space.

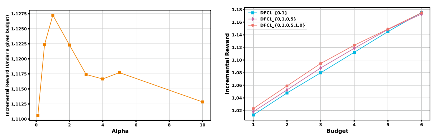
6.2.2 Prediction Loss vs Decision Loss tradeoff. As mentioned above, we integrate the prediction loss as a regularizer into the training objective. In this experiment, we will consider how the weight of the prediction loss affects the performance of DFCL. We set $\alpha \in \{0.1, 0.5, 1, 2, 3, 4, 5, 10\}$ and measure the per-capita orders under a fixed per-capita budget. As shown in Fig. 2a, increasing α in a certain range does not lead to a decrease in model performance.

Table 3: EOM (Marketing data)

Model	Budget 1	2	3	4	5	6	Improvement
TSM-SL	1.0000 ± 0.0023	1.0300 ± 0.0022	1.0611 ± 0.0023	1.0873 ± 0.0022	1.1140 ± 0.0020	1.1437 ± 0.0021	/
TSM-CF	1.0006 ± 0.0007	1.0306 ± 0.0006	1.0592 ± 0.0004	1.0866 ± 0.0006	1.1109 ± 0.0003	1.1353 ± 0.0008	-0.19%
DPM	0.9983 ± 0.0011	1.0366 ± 0.0010	1.0720 ± 0.0006	1.1050 ± 0.0003	1.1305 ± 0.0010	1.1594 ± 0.0007	1.00%
CN	1.0047 ± 0.0013	1.0339 ± 0.0010	1.0622 ± 0.0007	1.0910 ± 0.0009	1.1151 ± 0.0011	1.1384 ± 0.0015	0.16%
CN+DFCL-PL	0.9995 ± 0.0003	1.0366 ± 0.0008	1.0739 ± 0.0009	1.1071 ± 0.0005	1.1367 ± 0.0006	1.1650 ± 0.0009	1.26%
DFCL-PL	1.0104 ± 0.0005	1.0465 ± 0.0006	1.0812 ± 0.0004	1.1118 ± 0.0007	1.1407 ± 0.0011	1.1638 ± 0.0018	1.98%
DFCL-MER	1.0178 ± 0.0008	1.0501 ± 0.0005	1.0810 ± 0.0002	1.1121 ± 0.0010	1.1410 ± 0.0013	1.1674 ± 0.0009	2.06%
DFCL-IFD	1.0197 ± 0.0012	1.0574 ± 0.0022	1.0902 ± 0.0024	1.1221 ± 0.0026	1.1516 ± 0.0028	1.1796 ± 0.0030	2.85%

However, if α is too large, the prediction loss dominates the training objective and the model will be reduced to the two-stage method. The experiment suggests it is possible to choose a value of α so that we can achieve better performance and more accurate predictions.

6.2.3 Impact of Lagrange multiplier. Next, we would like to discuss the impact of the Lagrange multiplier λ on the performance of DFCL model. Since a given Lagrange multiplier λ corresponds to the MCKP for a certain budget constraint, DFCL model can learn allocation policies for different budgets simultaneously by changing or adding λ to the DFCL loss. We set up different combinations of Lagrange multipliers ($\lambda \in \{\{0.1\}, \{0.1, 0.5\}, \{0.1, 0.5, 1.0\}\}$) and use to train DFCL models. Fig. 2b shows the results using DFCL-IFD models trained by combinations of Lagrange multipliers. We can observe that λ is a hyperparameter that can have a significant impact on model performance. A small λ enables the model to learn the allocation policy efficiently under high budget and vice versa. Moreover, models trained with multiple Lagrange multipliers can balance performance with different budgets.

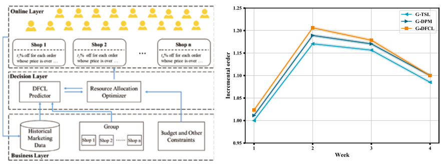


(a) Impact of the prediction loss weight α **(b) Impact of Lagrange multiplier**

Figure 2: Offline experiment results

6.3 Online A/B testing

6.3.1 Setups. We deploy DFCL, DPM and TSM-SL to support a discount campaign in Meituan (a food delivery platform), and conduct an online A/B testing for four weeks. The experiment contains 310K online shops and they are randomly divided every day into three groups called G-DFCL, G-DPM and G-TSL respectively. Each shop will be assigned a discount $t \in \{0, 5, 10, 15, 20\}$ as the treatmemt, which means t% off for each order whose price meets a given threshold. The marketing goal is to maximize the orders by assigning an appropriate discount to each store every day for a limited budget. The online deployment of DFCL is shown in Fig. 3a: (1) Before the campaign starts each day, we use the DFCL model to make predictions and allocate the appropriate discounts to each store based on budget and other constraints in an offline environment. (2) The users visit the online shop and get discounts which will stimulate them to make purchases. (3) During model training, we use historical random data and resource allocation optimizer to update the model parameters.



(a) Online deployment of DFCL **(b) Orders**

Figure 3: Online A/B testing

6.3.2 Results. Fig. 3b illustrates the improvement in weekly orders for G-DFCL and G-DPM relative to G-TSM. In order to preserve data privacy, all data points in Fig. 3b have been normalized that are divided by the orders of TSM-SL in the first week. We can see that DFCL achieves a significant average improvement of 2.17% relative to TSM-SL and also outperforms DPM with a relative improvement of 0.85%. The detailed results can be found in Appendix E.

7 Conclusion

In this paper, we propose a decision focused causal learning framework (DFCL) for direct counterfactual marketing optimization, which overcomes the technological challenges of DFL deployment in marketing. By designing surrogate losses and constructing black-box optimisation, we efficiently align the objectives of ML and OR. Both offline experiments and online A/B testing demonstrate the effectiveness of DFCL over the state-of-the-art methods.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2023YFB4502400), the NSF of China (62172206), and the Xiaomi Foundation.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467* (2016).

[2] Meng Ai, Biao Li, Heyang Gong, Qingwei Yu, Shengjie Xue, Yuan Zhang, Yunzhou Zhang, and Peng Jiang. 2022. LBCF: A Large-Scale Budget-Constrained Causal Forest Algorithm. In *Proceedings of the ACM Web Conference 2022*. 2310–2319.

[3] Javier Albert and Dmitri Goldenberg. 2022. E-commerce promotions personalization via online multiple-choice knapsack with uplift modeling. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*. 2863–2872.

[4] Brandon Amos and J Zico Kolter. 2017. Optnet: Differentiable optimization as a layer in neural networks. In *International Conference on Machine Learning*. PMLR, 136–145.

[5] Susan Athey, Julie Tibshirani, and Stefan Wager. 2019. Generalized random forests. (2019).

[6] Keith Battocchi, Eleanor Dillon, Maggie Hei, Greg Lewis, Paul Oka, Miruna Oprescu, and Vasilis Syrgkanis. 2019. EconML: A Python package for ML-Based heterogeneous treatment effects estimation. *Version 0. x* (2019).

[7] Quentin Berthet, Mathieu Blondel, Olivier Teboul, Marco Cuturi, Jean-Philippe Vert, and Francis Bach. 2020. Learning with differentiable pertubed optimizers. *Advances in neural information processing systems* 33 (2020), 9508–9519.

[8] Artem Betlei, Eustache Diemert, and Massih-Reza Amini. 2021. Uplift modeling with generalization guarantees. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 55–65.

[9] Eustache Diemert, Artem Betlei, Christophe Renaudin, and Massih-Reza Amini. 2018. A large scale benchmark for uplift modeling. In *KDD*.

[10] Priya Donti, Brandon Amos, and J Zico Kolter. 2017. Task-based end-to-end model learning in stochastic optimization. *Advances in neural information processing systems* 30 (2017).

[11] Shuyang Du, James Lee, and Farzin Ghaffarizadeh. 2019. Improve User Retention with Causal Learning. In *The 2019 ACM SIGKDD Workshop on Causal Discovery*. PMLR, 34–49.

[12] Adam N Elmachtoub and Paul Grigas. 2022. Smart "predict, then optimize". *Management Science* 68, 1 (2022), 9–26.

[13] Fredrik Johansson, Uri Shalit, and David Sontag. 2016. Learning representations for counterfactual inference. In *International conference on machine learning*. PMLR, 3020–3029.

[14] Hans Kellerer, Ulrich Pferschy, David Pisinger, Hans Kellerer, Ulrich Pferschy, and David Pisinger. 2004. The multiple-choice knapsack problem. *Knapsack Problems* (2004), 317–347.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Sören R Künzel, Jasjeet S Sekhon, Peter J Bickel, and Bin Yu. 2019. Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the national academy of sciences* 116, 10 (2019), 4156–4165.

[17] Finn Kuusisto, Vitor Santos Costa, Houssam Nassif, Elizabeth Burnside, David Page, and Jude Shavlik. 2014. Support vector machines for differential prediction. In *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2014, Nancy, France, September 15-19, 2014. Proceedings, Part II 14*. Springer, 50–65.

[18] Jayanta Mandi, Vıctor Bucarey, Maxime Mulamba Ke Tchomba, and Tias Guns. 2022. Decision-focused learning: through the lens of learning to rank. In *International Conference on Machine Learning*. PMLR, 14935–14947.

[19] Jayanta Mandi and Tias Guns. 2020. Interior point solving for lp-based prediction+ optimisation. *Advances in Neural Information Processing Systems* 33 (2020), 7272–7282.

[20] Jayanta Mandi, James Kotary, Senne Berden, Maxime Mulamba, Victor Bucarey, Tias Guns, and Ferdinando Fioretto. 2023. Decision-focused learning: Foundations, state of the art, benchmark and future opportunities. *arXiv preprint arXiv:2307.13565* (2023).

[21] Jayanta Mandi, Peter J Stuckey, Tias Guns, et al. 2020. Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 34. 1603–1610.

[22] Maxime Mulamba, Jayanta Mandi, Michelangelo Diligenti, Michele Lombardi, Victor Bucarey, and Tias Guns. 2020. Contrastive losses and solution caching for predict-and-optimize. *arXiv preprint arXiv:2011.05354* (2020).

[23] Xinkun Nie and Stefan Wager. 2021. Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika* 108, 2 (2021), 299–319.

[24] Mathias Niepert, Pasquale Minervini, and Luca Franceschi. 2021. Implicit MLE: backpropagating through discrete exponential family distributions. *Advances in Neural Information Processing Systems* 34 (2021), 14567–14579.

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[26] Marin Vlastelica Pogančić, Anselm Paulus, Vit Musil, Georg Martius, and Michal Rolinek. 2019. Differentiation of blackbox combinatorial solvers. In *International Conference on Learning Representations*.

[27] Jasjeet S Sekhon. 2008. The Neyman-Rubin Model of Causal Inference and Estimation via Matching Methods. *The Oxford Handbook of Political Methodology* 2 (2008), 1–32.

[28] Sanket Shah, Kai Wang, Bryan Wilder, Andrew Perrault, and Milind Tambe. 2022. Decision-focused learning without decision-making: Learning locally optimized decision losses. *Advances in Neural Information Processing Systems* 35 (2022), 1320–1332.

[29] Claudia Shi, David Blei, and Victor Veitch. 2019. Adapting Neural Networks for the Estimation of Treatment Effects. *Advances in Neural Information Processing Systems (NIPS)* 32 (2019).

[30] Prabhakant Sinha and Andris A Zoltners. 1979. The multiple-choice knapsack problem. *Operations Research* 27, 3 (1979), 503–515.

[31] Stefan Wager and Susan Athey. 2018. Estimation and inference of heterogeneous treatment effects using random forests. *J. Amer. Statist. Assoc.* 113, 523 (2018), 1228–1242.

[32] Chao Wang, Xiaowei Shi, Shuai Xu, Zhe Wang, Zhiqiang Fan, Yan Feng, An You, and Yu Chen. 2023. A Multi-stage Framework for Online Bonus Allocation Based on Constrained User Intent Detection. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5028–5038.

[33] Bryan Wilder, Bistra Dilkina, and Milind Tambe. 2019. Melding the data-decisions pipeline: Decision-focused learning for combinatorial optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1658–1665.

[34] Ziang Yan, Shusen Wang, Guorui Zhou, Jingjian Lin, and Peng Jiang. 2023. An End-to-End Framework for Marketing Effectiveness Optimization under Budget Constraint. *arXiv preprint arXiv:2302.04477* (2023).

[35] Liuyi Yao, Sheng Li, Yaliang Li, Mengdi Huai, Jing Gao, and Aidong Zhang. 2018. Representation learning for treatment effect estimation from observational data. *Advances in neural information processing systems* 31 (2018).

[36] Peng Ye, Julian Qian, Jieying Chen, Chen-hung Wu, Yitong Zhou, Spencer De Mars, Frank Yang, and Li Zhang. 2018. Customized regression model for airbnb dynamic pricing. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 932–940.

[37] Yang Zhang, Bo Tang, Qingyu Yang, Dou An, Hongyin Tang, Chenyang Xi, Xueying Li, and Feiyu Xiong. 2021. BCORLE (λ): An Offline Reinforcement Learning and Evaluation Framework for Coupons Allocation in E-commerce Market. *Advances in Neural Information Processing Systems* 34 (2021), 20410–20422.

[38] Kui Zhao, Junhao Hua, Ling Yan, Qi Zhang, Huan Xu, and Cheng Yang. 2019. A unified framework for marketing budget allocation. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1820–1830.

[39] Yan Zhao, Xiao Fang, and David Simchi-Levi. 2017. Uplift modeling with multiple treatments and general response types. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 588–596.

[40] Hao Zhou, Shaoming Li, Guibin Jiang, Jiaqi Zheng, and Dong Wang. 2023. Direct heterogeneous causal learning for resource allocation problems in marketing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37. 5446–5454.

A The Proof of Theorem 1

Proof. First of all, we introduce some notations. Following the potential outcome framework [27], let $X \in \mathbb{R}^d$ denote the feature vector and $T \in \{1, 2, \ldots, M\}$ be the treatment. Let $Y^r(T)$ and $Y^c(T)$ be the potential outcome of the revenue and the cost respectively when the individual receives treatment T. let $\widehat{Y}^r(T)$ and $\widehat{Y}^c(T)$ be the predicted outcome of the revenue and the cost respectively when the individual receives treatment T. For $\mathcal{L}_{MSE}$, we have

$$\begin{aligned}\mathcal{L}_{MSE}(r, c, \hat{r}, \hat{c}) &= \frac{1}{NM} \sum_i \sum_j (r_{ij} - \hat{r}_{ij})^2 + (c_{ij} - \hat{c}_{ij})^2 \\ &= \mathbb{E}_{X,T}[(Y^r(T) - \widehat{Y}^r(T))^2 + (Y^c(T) - \widehat{Y}^c(T))^2].\end{aligned}$$

For $\mathcal{L}_{PL}$, we have

$$\begin{aligned}
&\mathcal{L}_{PL}(r,c,\hat{r},\hat{c})\\
=&\frac{1}{M}\sum_i \frac{1}{N_{t_i}}[(r_{it_i}-\hat{r}_{it_i})^2+(c_{it_i}-\hat{c}_{it_i})^2]\\
=&\frac{1}{M}\sum_j \sum_{i:t_i=j}\frac{1}{N_{t_i}}[(r_{it_i}-\hat{r}_{it_i})^2+(c_{it_i}-\hat{c}_{it_i})^2]\\
=&\frac{1}{M}\sum_j \frac{1}{N_{t_i}}\sum_{i:t_i=j}[(r_{it_i}-\hat{r}_{it_i})^2+(c_{it_i}-\hat{c}_{it_i})^2]\\
=&\frac{1}{M}\sum_j \mathbb{E}_X[(Y^r(j)-\widehat{Y}^r(j))^2+(Y^c(j)-\widehat{Y}^c(j))^2|T_i=j]\\
=&\frac{1}{M}\sum_j \mathbb{E}_X[(Y^r(j)-\widehat{Y}^r(j))^2+(Y^c(j)-\widehat{Y}^c(j))^2] \qquad (T\perp X)\\
=&\mathbb{E}_{X,T}[(Y^r(T)-\widehat{Y}^r(T))^2+(Y^c(T)-\widehat{Y}^c(T))^2],
\end{aligned}$$

where $T\perp X$ holds because the data set is from random control trials (RCT). (RCT). Therefore, we finish the proof. □

B The Proof of Theorem 3

Proof. Follow the notations in Appendix A and let

$$\text{softmax}(\hat{r}_{ij}-\lambda\hat{c}_{ij})=\frac{\exp(\hat{r}_{ij}-\lambda\hat{c}_{ij})}{\sum_k \exp(\hat{r}_{ik}-\lambda\hat{c}_{ik})}$$

be the softmax function. Hence, $\mathcal{L}'_{DDL}$ can be rewritten as

$$\begin{aligned}
&\mathcal{L}'_{DDL}(r,c,\hat{r},\hat{c})\\
&=-\sum_\lambda\sum_i\sum_j(r_{ij}-\lambda c_{ij})\text{softmax}(\hat{r}_{ij}-\lambda\hat{c}_{ij})\\
&=-NM\sum_\lambda\frac{1}{NM}\sum_i\sum_j(r_{ij}-\lambda c_{ij})\text{softmax}(\hat{r}_{ij}-\lambda\hat{c}_{ij})\\
&=-NM\sum_\lambda\mathbb{E}_{X,T}[(Y^r(T)-\lambda Y^c(T))\text{softmax}(\widehat{Y}^r(T)-\lambda\widehat{Y}^c(T))]
\end{aligned}$$

In addition, we have

$$\begin{aligned}
&\mathcal{L}_{PLL}(r,c,\hat{r},\hat{c})\\
&=-\sum_\lambda\sum_i\frac{N}{N_{t_i}}(r_{it_i}-\lambda c_{it_i})\text{softmax}(\hat{r}_{it_i}-\lambda\hat{c}_{it_i})\\
&=-\sum_\lambda\sum_i\frac{N}{N_{t_i}}(r_{it_i}-\lambda c_{it_i})\text{softmax}(\hat{r}_{it_i}-\lambda\hat{c}_{it_i})\\
&=-N\sum_\lambda\sum_j\sum_{i:t_i=j}\frac{1}{N_{t_i}}(r_{it_i}-\lambda c_{it_i})\text{softmax}(\hat{r}_{it_i}-\lambda\hat{c}_{it_i})\\
&=-N\sum_\lambda\sum_j\mathbb{E}_X[(Y^r(j)-\lambda Y^c(j))\text{softmax}(\widehat{Y}^r(j)-\lambda\widehat{Y}^c(j))|T_i=j]\\
&=-N\sum_\lambda\sum_j\mathbb{E}_X[(Y^r(j)-\lambda Y^c(j))\text{softmax}(\widehat{Y}^r(j)-\lambda\widehat{Y}^c(j))]\\
&=-NM\sum_\lambda\frac{1}{M}\sum_j\mathbb{E}_X[(Y^r(j)-\lambda Y^c(j))\text{softmax}(\widehat{Y}^r(j)-\lambda\widehat{Y}^c(j))]\\
&=-NM\sum_\lambda\mathbb{E}_{X,T}[(Y^r(T)-\lambda Y^c(T))\text{softmax}(\widehat{Y}^r(T)-\lambda\widehat{Y}^c(T))].
\end{aligned}$$

Therefore, $\mathcal{L}'_{DDL}(r,c,\hat{r},\hat{c})=\mathcal{L}_{PLL}(r,c,\hat{r},\hat{c})$ holds.

Algorithm 2 Lagrangian duality gradient estimator

Input: training data $\mathrm{D}\equiv\{(x_i,t_i,r_{it_i},c_{it_i})\}_{i=1}^N$, Lagrange multiplier λ, the predicted revenue/cost $\hat{r}/\hat{c}$

Output: $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}}$, $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}}$

- Initialize $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}}=0$, $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}}=0$, $z_{ij}=0\ \forall i,j$
- $a=\hat{r}-\lambda\hat{c}$
- $\forall i,j,\ z_{ij}=\mathbb{I}_{j=\arg\max_j(a_{ij})}$
- $\bar{r}(\hat{r},\hat{c},\lambda)=\frac{1}{N}\sum_i\frac{1}{p_{t_i}}r_{t_i}\mathbb{I}_{t_i=\arg\max_j z_{ij}}$
- $\bar{c}(\hat{r},\hat{c},\lambda)=\frac{1}{N}\sum_i\frac{1}{p_{t_i}}c_{t_i}\mathbb{I}_{t_i=\arg\max_j z_{ij}}$
- $\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})=\bar{r}(\lambda,r,c,\hat{r},\hat{c})-\lambda\bar{c}(\lambda,r,c,\hat{r},\hat{c})$
- matching_indices= $\{i|t_i=\arg\max_j\ z_{ij},\ \forall i\}$
- mismatching_indices= $\{i|t_i\neq\arg\max_j\ z_{ij},\ \forall i\}$
- **for all** $i\in$ matching_indices **do**
  - $h^r_{it_i}=max_{j\neq t_i}a_{ij}-a_{it_i}$, $h^c_{it_i}=\frac{(a_{it_i}-\max_{j\neq t_i}a_{ij})}{\lambda}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}_{it_i}}=\frac{-\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^r_{it_i}}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}_{it_i}}=\frac{-\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^c_{it_i}}$
  - **for all** $j\in\{1,2,...,M\}$ *and* $j\neq t_i$ **do**
    - $h^r_{ij}=a_{it_i}-a_{ij}$, $h^c_{ij}=\frac{(a_{ij}-a_{it_i})}{\lambda}$
    - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}_{ij}}=\frac{-\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^r_{ij}}$
    - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}_{ij}}=\frac{-\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^c_{ij}}$
- **for all** $i\in$ mismatching_indices **do**
  - $j=\arg\max_j\ a_{ij}$
  - $h^r_{it_i}=a_{ij}-a_{it_i}$, $h^r_{ij}=-h^r_{it_i}$
  - $h^c_{it_i}=\frac{(max_j a_{ij}-a_{it_i})}{\lambda}$, $h^c_{ij}=-h^c_{it_i}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}_{it_i}}=\frac{\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^r_{it_i}}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}_{it_i}}=\frac{\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^c_{it_i}}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}_{ij}}=\frac{\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^r_{ij}}$
  - $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}_{ij}}=\frac{\frac{1}{Np_{t_i}}(r_{it_i}-\lambda c_{it_i})}{h^c_{ij}}$
- **return** $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{r}}$, $\frac{\partial\mathcal{L}_{DDL}(\lambda,r,c,\hat{r},\hat{c})}{\partial\hat{c}}$

For $\forall i,j=\arg\max_k r_{ik}-\lambda c_{ik}$, let $\hat{r}_{ij}-\lambda\hat{c}_{ij}\to+\infty$; for $\forall i,j\neq\arg\max_k r_{ik}-\lambda c_{ik}$, let $\hat{r}_{ij}-\lambda\hat{c}_{ij}\to-\infty$. Hence, we have

$$\text{softmax}(\hat{r}_{ij}-\lambda\hat{c}_{ij})\to\mathbb{I}_{j=\arg\max_k\hat{r}_{ik}-\lambda\hat{c}_{ik}}.$$

Therefore, we further get

$$\min_{\hat{r},\hat{c}}\mathcal{L}_{PLL}(\lambda,\hat{r},\hat{c})=\min_{\hat{r},\hat{c}}\mathcal{L}'_{DDL}(\lambda,\hat{r},\hat{c})=\min_{\hat{r},\hat{c}}\mathcal{L}_{DDL}(\lambda,\hat{r},\hat{c}).$$

□

C Policy Evaluation Based on EOM

Given a batch of N random samples and model predictions $\hat{r}$ and $\hat{c}$, we can use binary search to empirically estimate the per capita

Table 4: Online A/B testing results with the confidence interval

Group	Week				Improvement
	1st	2nd	3rd	4th	
G-TSL	1.0000 ± 0.00285	1.1706 ± 0.00298	1.1565 ± 0.00293	1.0851 ± 0.00289	/
G-DPM	1.0113 ± 0.00284	1.1891 ± 0.00298	1.1704 ± 0.00293	1.1000 ± 0.00288	1.32%
G-DFCL	1.0235 ± 0.00285	1.2062 ± 0.00297	1.1786 ± 0.00293	1.1000 ± 0.00288	2.17%

Algorithm 3 An implementation of per capita revenue estimation for primal MCKP with budget B

Input: training data $D \equiv \{(x_i, t_i, r_{it_i}, c_{it_i})\}_{i=1}^{N}$, the budget B, the predicted revenue/cost $\hat{r}/\hat{c}$, a small constant ϵ

Output: the expected per capita revenue $\bar{r}(B, r, c, \hat{r}, \hat{c})$

Initialize $\lambda_{\min} = 0$, $\lambda_{\max} = \max_{i,j}(\frac{\hat{r_{ij}}}{\hat{c_{ij}}})$, $z_{ij} = 0\ \forall i, j$
while $\frac{B}{N} - \bar{c}(\lambda, r, c, \hat{r}, \hat{c}) < \epsilon$ **do**
  $\lambda = \frac{\lambda_{\max} + \lambda_{\min}}{2}$
  $\forall i, j,\ z_{ij} = \mathbb{I}_{j=\arg\max_j(\hat{r_{ij}} - \lambda \hat{c_{ij}})}$
  $\bar{r}(\lambda, r, c, \hat{r}, \hat{c}) = \frac{1}{N}\sum_i \frac{1}{p_{t_i}} r_{t_i} \mathbb{I}_{t_i=\arg\max_j\ z_{ij}}$
  $\bar{c}(\lambda, r, c, \hat{r}, \hat{c}) = \frac{1}{N}\sum_i \frac{1}{p_{t_i}} c_{t_i} \mathbb{I}_{t_i=\arg\max_j\ z_{ij}}$
  if $\frac{B}{N} - \bar{c}(\lambda, r, c, \hat{r}, \hat{c}) > 0$ **then**
    $\lambda_{\max} = \lambda$
  else
    $\lambda_{\min} = \lambda$
$\bar{r}(B, r, c, \hat{r}, \hat{c}) = \bar{r}(\lambda, r, c, \hat{r}, \hat{c})$
return the expected per capita revenue $\bar{r}(B, r, c, \hat{r}, \hat{c})$

revenue under a per capita budget $\frac{B}{N}$, Algorithm 3 summarizes this approach.

D Lagrangian Duality Gradient Estimator

The decision making is independent for each individual thanks to the decomposition of the Lagrangian duality theory. Thus, for each sample, the smallest perturbation that causes a change in the dual decison loss is first calculated, and the loss after the perturbation is obtained by correcting only the original result. Algorithm 2 provides details of the modified gradient estimator, which greatly reduces the computational overhead. Note that for comprehensibility, Algorithm 2 is described with for loops, while in practice we work with matrix operations.

E Supplementary Experimental Results

Tabel 4 presents the detailed online A/B testing results. In order to preserve data privacy, all data points have been normalized by dividing by the orders of TSM-SL in the first week. The confidence interval ($\alpha = 0.05$) is computed by a t-test.

DolphCoder: Echo-Locating Code Large Language Models with Diverse and Multi-Objective Instruction Tuning

Yejie Wang[1*], Keqing He[2*], Guanting Dong[1], Pei Wang[1], Weihao Zeng[1], Muxi Diao[1]
Yutao Mou[1], Mengdi Zhang[2], Jingang Wang[2], Xunliang Cai[2], Weiran Xu[1†]
[1]Beijing University of Posts and Telecommunications, Beijing, China
[2]Meituan, Beijing, China
{wangyejie,dongguanting,wangpei,zengwh,dmx,myt,xuweiran}@bupt.edu.cn
{hekeqing,zhangmengdi02,wangjingang02,caixunliang}@meituan.com

Abstract

Code Large Language Models (Code LLMs) have demonstrated outstanding performance in code-related tasks. Several instruction tuning approaches have been proposed to boost the code generation performance of pre-trained Code LLMs. In this paper, we introduce a diverse instruction model (**DolphCoder**) with self-evaluating for code generation. It learns diverse instruction targets and combines a code evaluation objective to enhance its code generation ability. Our model achieves superior performance on the HumanEval and MBPP benchmarks, demonstrating new insights for future code instruction tuning work. Our key findings are: (1) Augmenting more diverse responses with distinct reasoning paths increases the code capability of LLMs. (2) Improving one's ability to evaluate the correctness of code solutions also enhances their ability to create it.

1 Introduction

Code pre-trained models have achieved remarkable progress in the era of large language models (LLMs), such as Codex (Chen et al., 2021), AlphaCode (Li et al., 2022), and PaLM-Coder (Chowdhery et al., 2022b). Code-related tasks are also the key factors in evaluating the capability of LLMs. Numerous code LLMs have been proposed, including closed-source models (Chen et al., 2021; Li et al., 2022; OpenAI, 2023) and open-source models (Li et al., 2023; Rozière et al., 2023). They perform expensive pre-training using substantial amounts of code data and display impressive performance.

In contrast to these pre-trained code LLMs, another lightweight paradigm of enhancing code capability is instruction tuning using relatively small high-quality code-related data. For example, Code Alpaca (Chaudhary, 2023) employs a similar self-instruct method as Alpaca (Taori et al., 2023) to generate code instructions via OpenAI's ChatGPT[1]. Further, WizardCoder (Luo et al., 2023) introduces a more complicated Evol-Instruct method (Xu et al., 2023a) which evolves existing instruction data to generate more complex and diverse datasets. Instead, OctoPack (Muennighoff et al., 2023) and Magicoder (Wei et al., 2023) construct code instructions by mining existing code corpus. All of these methods enhance the performance of the open-source Code LLMs.

However, these methods have two weaknesses: (1) They take the only golden answer but ignore the diversity of answers in code generation. We find that augmenting more diverse responses using different system prompts increases the code capability of LLMs. (2) Current models generate plausible code snippets in terms of grammar and logic but are unable to identify subtle errors, such as corner cases and wrong input/output formats. It has no guarantee that temperature sampling will consistently produce accurate answers over time. We suppose that LLMs are capable of generating correct solutions while struggling to discriminate correct from incorrect ones. Improving one's ability to evaluate the correctness of code also enhances their ability to create it.

Inspired by the two insights, we introduce a diverse instruction model (**DolphCoder**) with self-evaluating for code generation. Specifically, we use Code Llama-python as our base model and obtain evolved instruction data following WizardCoder. Then motivated by rejection sampling (Touvron et al., 2023b) and ORCA (Mukherjee et al., 2023), we use different system prompts to generate diverse answers via ChatGPT. After removing low-quality and similar data using heuristic rules (Luo et al., 2023; Di et al., 2023a), we perform supervised

* Equal contribution.
† Corresponding author.

[1]https://openai.com/blog/ChatGPT

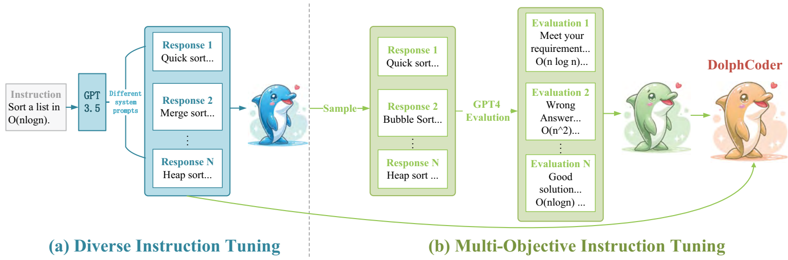


Figure 1: The overall architecture of our proposed diverse instruction tuning with self-evaluating for code generation, DolphCoder. Stage (a) denotes Diverse Instruction Tuning (DIT) and Stage (b) denotes Multi-Objective Instruction Tuning (MOT) for self-evaluating.

fine-tuning on the remaining instruction data. Further, we explore whether improving one's ability to evaluate code helps generate it. We propose a self-evaluate multi-task learning framework by adding a code evaluation objective to the traditional instruction fine-tuning task. We find training the model for both code generation and code evaluation benefits the code capability.

Our key contributions are summarized as follows:

1. We introduce a diverse instruction model (**DolphCoder**) with self-evaluating for code generation. It learns diverse instruction targets and combines a code evaluation objective to enhance its code generation ability.

2. DolphCoder outperforms strong open-source code LLMs by a large margin, including CODELLAMA-INSTRUCT, OctoCoder, and WizardCoder.

2 Method

In this section, we elaborate on the methodological details of DolphCoder. As shown in Figure 1, DolphCoder has two training stages: (1) The first is **D**iverse **I**nstruction **T**uning (DIT) with multiple chain-of-thought answers to the same instruction. (2) The second is **M**ulti-**O**bjective **T**uning (MOT) of combining the code generation task and code evaluation task both in the form of a natural language generation task.

2.1 Diverse Instruction Tuning

We follow the Evol-Instruct technique (Xu et al., 2023a; Luo et al., 2023) to construct our training corpus. Based on the Code Alpaca dataset [2], we iteratively evolve the programming problems in this dataset via in-depth evolving to obtain new instructions.[3] For each instruction, then we use different system prompts to query ChatGPT and obtain diverse targets. These system prompts aim at augmenting user instructions and task descriptions to give more code solutions with diverse reasoning paths. We display our system prompts in Figure 2. We find more diverse answers can increase the code capability of LLMs (see Section 4.2) and argue that different styles of code solutions provide more supervised signals for the model. Similar to Luo et al. (2023); Di et al. (2023a), we also use several heuristic rules to remove low-quality and similar data. Finally, we get a diverse code instruction dataset with a size of 510k and utilize Code LLama as the foundation LLM to finetune.

2.2 Multi-Objective Instruction Tuning

We discover that the current instruction models produce both correct and incorrect code solutions when randomly sampled. We argue that LLMs can generate correct solutions while struggling to discriminate correct from incorrect ones. Therefore, we explore whether improving one's ability to evaluate code helps generate it.

We sample 5k instructions from the above dataset and use our model in the first stage to generate 100 answers using temperature sampling. Next,

[2] https://huggingface.co/datasets/sahil2801/CodeAlpaca-20k

[3] Since the original datasets or code are not released, we reproduce the evolve procedure following WizardLM (Xu et al., 2023a) and modify the evolve prompts according to the original WizardCoder paper.

System Prompts

1. [EMPTY]

2. You are a code assistant who knows multiple programming languages and how to translate between them. Given a task, you explain in simple steps what the task is asking, any guidelines it provides, and how to use those guidelines to find the solution.

3. You are an AI assistant. You will be given a task. You must generate a detailed and long answer.

4. You are an AI assistant, who knows every language and how to translate one language to another. Given a task, you explain in simple steps what the task is asking, any guidelines that it provides.

5. You solve the task and show how you used the guidelines to solve the task.

6. You are a teacher. Given a task, you explain in simple steps what the task is asking, any guidelines it provides and how to use those guidelines to find the answer.

Figure 2: We use these system prompts to generate more diverse responses where [EMPTY] means no system prompt.

we use GPT-4[4] to verify the correctness of the deduplicated generated answers in terms of grammar, logic and efficiency. In our preliminary experiments, We encounter difficulties in assessing the correctness of code using ChatGPT or other existing LLMs. We also consider the use of compiled signals from the code executor but find most generated code is grammatically correct. Previous work (Liu et al., 2023a; Shen et al., 2023a) use existing unit tests in the training set, which is not applicable to our situation. We leave out more evaluation methods to future work. Finally, we obtain the code evaluation dataset with a size of 370k.[5] We formulate the code evaluation task in the form of a natural language generation task as

[4] https://openai.com/GPT-4

[5] We remove failed GPT-4 requests and responses without <passed> or <not passed> results.

Evaluation Prompt

You are a code evaluation model. Your task is to analyze and inspect the input code snippet. This involves evaluating whether it can run smoothly, whether it will produce the correct results, and whether there are issues like timeouts. You should output the result of the code with <passed> or <not passed>. Additionally, you should also provide an explanation for the evaluation result.

Figure 3: We use the evaluation prompt to query GPT-4 to access the correctness of the generated code solutions of our model.

shown in Figure 1. We hope its similar training form can provide multiple meaningful supervision signals to the original code generation model. In our experiments, we find it difficult to balance the two code generation and code evaluator tasks because the model always tends to overfit the code generation objective. Therefore, we use a multi-step training paradigm where we first finetune the above model as an evaluator and then as a generator. Specifically, we finetune the DIT model using the code evaluation dataset for 1 epoch and then continue training on the diverse instruction data for 100 steps. We find training more steps will not get further improvements. Experiment results demonstrate that the multi-step training stably enhances its code generation ability (shown in Figure 6).

3 Experiments

3.1 Benchmarks

In this paper, we focus on two of the most widely used benchmarks in the field of code generation.

- **HumanEval (base)**[6] **and HumanEval+ (plus)**. HumanEval (base) is a widely used benchmark proposed by OpenAI for code synthesis. It consists of 164 handcrafted programming problems, with an average of 9.6 test cases allocated to each one for correctness checking. Further, Liu et al. (2023b) find that test cases in the benchmark may be insufficient and propose HumanEval+ (plus) powered by the EvalPlus framework to obtain 80× test cases.

[6] https://github.com/openai/human-eval

- **MBPP (base) (Austin et al., 2021) and MBPP+ (plus)**. MBPP (base) is also a code synthesis benchmark offering a set of 500 crowd-sourced Python programming problems covering programming fundamentals, standard library functionality, and so on. Each problem consists of a task description, code solution and 3 automated test cases. EvalPlus also provides the MBPP+ (plus) benchmark which expands by 35x test cases.

We perform n-gram matching de-duplication between our training datasets and the benchmarks to prevent data leakage. For all the experiments, we use greedy decoding and report the pass@1 metric. The inference prompt template is shown in Figure 4 following WizardCoder. To keep a fair comparison, we use the same EvalPlus [7] framework to compute metrics.

3.2 Baselines

In this paper, we categorize the baseline models into the following two types. **(1) Closed-source models:** We have specifically incorporated OpenAI's GPT-3.5 and GPT-4, which are developed privately by leading technology companies, demonstrating the current state-of-the-art in LLM proficiency. **(2) Open-source models:** Several open-source LLMs have been made available to the AI community, although their performance generally lags behind the closed-source models a lot. As part of our research, we incorporate a significant number of these open-source models as our baselines, which include CodeGen (Nijkamp et al., 2023), CodeT5+ (Wang et al., 2021), StarCoder, CODELLAMA, OctoCoder and WizardCoder series.

3.3 Implementation Details

Data generation For Diverse Instruction data, we devise six system prompts shown as Figure 2, inclusive of a blank one, and prompt GPT-3.5-turbo to generate more diverse responses. Inspired by CodeFuse (Di et al., 2023b), we use heuristic rules to filter out low-quality data and deduplicate data based on the test set. The specific rules are as follows:

- **Filtering low-quality data**
  1. Filter data with instruction length less than 10 words or greater than 1000 words;
  2. Filter data with output length less than 80 words;
  3. Filter out data with invalid markdown format, such as: code blocks not closed;
  4. Filter data with more than 2048 tokens;
- **Filtering data similar to test dataset**
  1. Filter data containing any function name from the test dataset.
  2. Using NLTK to remove stop words and punctuation from the docstring of HumanEval, obtain the core words such as "sort array prime", etc. Filter data containing more than 40% of the core words from the test dataset.

[7] https://evalplus.github.io/leaderboard.html

Prompt Template

Below is an instruction that describes a task, paired with an input that provides further context. Write a response that appropriately completes the request.

Instruction: Create a Python script for this problem: {Question}

Response:

Figure 4: Inference prompt when testing on HumanEval and MBPP.

For Multi-Objective Instruction data, we extract 5000 instructions from the Diverse Instruction data and subsequently generate 100 responses for each instruction in 0.5 temperature and 0.95 top-p with the model we get after diverse instruction tuning. To ascertain the correctness of each code solution, we prompt GPT-4 to evaluate whether this code solution can meet the instruction requirement. And then they will be classified to either 'passed' or 'not passed'. The prompt we used to evaluate is as Figure 3.

Training In our study, we utilize the CODELLAMA-PYTHON-7B and CODELLAMA-PYTHON-13B as our foundational models. We train DIT for 3 epochs and MOT for 1 epoch of code evaluation data and 100 steps of code generation data. During the training phase, we establish a global batch size of 512 and a sequence length of 2048, with a learning rate initialized at 5e-6 and a warmup fraction of

Model	Size	HumanEval		MBPP	
		Base	Plus	Base	Plus
GPT-3.5 (Nov 2023)	-	72.6	65.9	81.7	69.4
GPT-4 (Nov 2023)	-	85.4	81.7	83.0	70.7
CODELLAMA-PYTHON	34B	51.8	42.7	67.2	52.9
WizardCoder-CL	34B	73.2	64.6	73.2	59.9
CodeT5+	16B	31.7	26.2	54.6	44.4
CodeGen-Mono	16B	32.9	27.4	52.6	43.6
StarCoder	15B	34.1	29.3	55.1	46.1
CODELLAMA-PYTHON	13B	42.7	36.6	61.2	50.9
CODELLAMA-INSTRUCT	13B	42.7	-	49.4	-
OctoCoder	15B	46.2	-	-	-
WizardCoder	13B	60.4*	54.3*	65.2*	53.1*
DolphCoder(ours)	13B	**67.7**	**57.9**	**67.2**	**54.1**
StarCoder	7B	24.4	20.7	33.1	28.8
CodeT5+	6B	29.3	23.8	51.9	40.9
CodeGen-Mono	6B	29.3	25.6	49.9	42.1
CODELLAMA-PYTHON	7B	37.8	34.1	57.6	45.4
CODELLAMA-INSTRUCT	7B	34.8	-	44.4	-
WizardCoder	7B	48.2	40.9	56.6	47.1
DolphCoder(ours)	7B	**62.8**	**54.9**	**64.9**	**52.6**

Table 1: Pass@1 results of different code LLMs for HumanEval and MBPP. Base means the original benchmark and Plus denotes the extended benchmark. * denotes the reproduced results via the EvalPlus scripts and other baseline results are cited from the official leaderboard. The best results in each column are in bold.

15%. After the warmup, the learning rate decays following a cosine schedule. We employ the Adam optimizer, with β_1 and β_2 parameters set at 0.9 and 0.95 respectively. And to ensure training stability, we incorporate gradient clipping with a value set at 1.0, a technique designed to prevent excessive gradient escalation that could lead to numerical instability or model divergence.

3.4 Main Results

The primary outcomes of our proposed method in comparison with the baselines are illustrated in Table 1. We conduct the following comparisons: (1) Generally, Our proposed method significantly outperforms all the previous methods in different sizes of model parameters. (2) Compared to our base model (CODELLAMA), DolphCoder shows significant improvements in both HumanEval(+) and MBPP(+). In detail, DolphCoder-7b has a 25 percentage point increase on HumanEval and a 7.3 percentage point increase on MBPP. (3) As we compare our model DolphCoder with the recent baseline model WizardCoder-7b, which is built on the same base model (CODELLAMA), DolphCoder still consistently outperforms it across all benchmarks and model sizes. (4) DolphCoder-7b even outperforms CODELLAMA-PYTHON 34b, demonstrating the efficiency of using small-size LLMs. These results indicate the generalizability and robustness of our method.

4 Analysis

4.1 Ablation Study

To investigate the characteristics of the main components in DolphCoder, we conduct ablation experiments in Table 2. From the results, we have the following observations: (1) Both DIT and MOT contribute to the performance improvement. Specifically, on the CODELLAMA-PYTHON-7b

Model	Size	HumanEval		MBPP	
		Base	Plus	Base	Plus
Evol Instruct	7B	50.0	44.5	59.4	50.1
+DIT	7B	57.9	51.2	64.2	52.1
+MOT	7B	55.5	46.3	64.2	52.6
+ALL(DolphCoder)	7B	**62.8**	**54.9**	**64.9**	**52.6**
Evol Instruct	13B	64.0	55.5	65.7	51.6
+DIT	13B	66.5	57.3	65.2	52.1
+MOT	13B	65.2	56.7	66.7	53.9
+ALL(DolphCoder)	13B	**67.7**	**57.9**	**67.2**	**54.1**

Table 2: Ablation study of DolphCoder.

Ratios	Size	Diversity	HumanEval	
			Base	Plus
1	7B	1.0	52.4	45.1
3	7B	2.1	**57.9**	50.6
6	7B	2.7	**57.9**	**51.2**
1	13B	1.0	63.4	53.7
3	13B	2.1	65.2	**58.5**
6	13B	2.7	**66.5**	57.3

Table 3: The effect of different sampling ratios of DIT. Code diversity shows the average number of different code solutions for a code instruction, and we use syntax analysis to parse different code solutions. All indicators refer to pass@1.

base, DIT yields an improvement of 7.9 pp on HumanEval and 4.8 pp on MBPP, compared to the baseline evolving instruction approach. MOT results in an improvement of 5.5 pp and 4.8 pp respectively on HumanEval and MBPP. Moreover, we observe that the MOT training data constructed by GPT-4 contains approximately 20% error noise, which severely limits the upper-bound performance of the MOT method in this work. A more detailed discussion can be found in Appendix B. (2) The combination of DIT and MOT yields further benefits. DolphCoder-7b exhibits an enhancement of 12.8 pp on HumanEval and 5.5 pp on MBPP, demonstrating the relatively orthogonal relationship between these two methods. These results demonstrate the effectiveness of our proposed methods.

4.2 Effect of Diverse Instruction Tuning

To further explore the source of the model's improvement brought by DIT, we test the model's performance under different sampling ratios which is shown in Table 3. The sampling ratio represents the number of system prompts. As the sampling ratio increases, the model's performance on all indicators gradually increases, which proves that using more diverse responses can enhance the model's performance.

In addition, we extract code snippets from the responses and check whether the implementation of these codes is different. For ease of statistics, we only focus on Python-related code. Specifically, we use AST to parse each piece of code and calculate the similarity between codes, then remove duplicates. We count the average number of unique code solutions left for each code task as an indicator of code diversity. Table 3 shows that as the sampling ratio increases, the number of different code solutions corresponding to the same code instruction

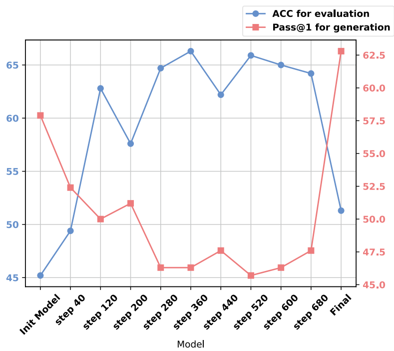


Figure 5: The trend of code evaluation capability and code generation capability during the MOT stage where step 200 refers to the training step in the first step of MOT. Init model means DIT model and Final means DolphCoder. Pass@1 refer to pass@1 on HumanEval.

also gradually increases and the model's performance on HumanEval improves too. Moreover, we observe that the increase in code diversity is not linear. Specifically, when we increase the sampling ratio from 3 to 6, we only observe marginal performance gains which we argue that the DIT data may contain unnecessary redundancy. Concurrent work (Lu et al., 2023; Liu et al., 2023c) explore more complicated diversity-based compression methods for general instruction fine-tuning. We leave it to future work.

4.3 Effect of Multi-Objective Instruction Tuning

To explore how the code evaluation capability and code generation capability benefit each other, we evaluate them in the MOT training stage. Specif-

Model	Greedy	Pass@1	Pass@10
DolphCoder	62.8	59.9	71.3
-w/o MOT	57.9	56.6	69.5
Improvement	4.9	3.3	1.8

Table 4: Pass@k with different decoding methods and different k values, where Greedy employs greedy sampling, pass@1 and pass@10 samples at temperature=0.2. All indicators are tested on HumanEval based on DolphCoder-7b.

ically, we evaluate the model's code evaluation capability by sampling 40 answers for each of the 164 test questions in HumanEval and classifying their correctness using the given golden unit tests. The experimental results are shown in Figure 5.

From the results, we observe that there exists a strong association between the evaluation and generation capabilities of the model. As the model continues to train on the evaluation task, its evaluation capability keeps improving and gradually stabilizes. However, this process impairs the model's code generation capability, potentially due to catastrophic forgetting caused by the multi-step training. After the model undergoes the generator training process, the model's code generation capability is restored and the pass@1 metric surpasses the limit of DIT model performance. Meanwhile, its evaluation capability significantly decreases. We suppose that the improvement in code generation capability is achieved through the transformation of the evaluation capability, but these two capabilities are challenging to coexist.

To further scrutinize the impact of the MOT, we compare the metrics of pass@1 and pass@10 to ascertain whether the improvements stem from the model's heightened preference for the correct response. From Table 4, we find that the DolphCoder outperforms the DIT Model across all metrics. The improvement of MOT on the pass@1 greedy decoding indicator is the largest, with an increase of 4.9%. Compared with pass@1, the improvement on pass@10 is significantly reduced to only 1.8%. This implies that the impact of the MOT does not focus on enhancing the model's capability to generate robust solutions. Instead, it primarily augments the model's ability to distinguish responses, leading to a higher preference for the correct answer.

Training	Inference	HumanEval	
		Base	Plus
✗	✗	57.9	50.6
✗	✓	55.5	48.2
✓	✗	**66.5**	**57.3**
✓	✓	61.0	53.7

Table 5: The effect of system prompt during training and inference process. Considering efficiency, we only conduct experiments on the DIT model based on CODELLAMA-13B-PYTHON and use pass@1 metric on HumanEval with greedy decoding. For inference with system prompts, we randomly choose a system prompt in the training for each test query.

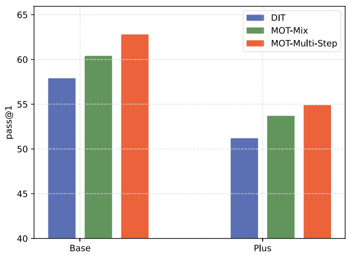


Figure 6: Effect of different settings of multi-task learning based on DolphCoder-7B. Mix denotes directly adding the generation and evaluation loss together and Multi-Step means sequential training.

4.4 Analysis of Key Designs

System Prompt Ablation We obtain more diverse training data by transforming the system prompt. In order to explore whether the system prompt should still be used during training and inference, we conduct experimental verification. Table 5 shows the effect of the system prompts during the training and inference process. From the results, we observe that, compared to training with system prompts, training without the system prompt results in a significant decline. This implies that assigning completely identical code instructions to different answers is not a good training method. When comparing whether to use the system prompt during inference, we find that erasing the system prompt results in a significant improvement compared to using the system prompt. This improvement may stem from the model's freedom to determine the most suitable inference path.

Multi-Task Training Method Ablation We also explore different multi-task training methods as shown in Figure 6. The results show that both MOT variants significantly outperform the DIT baseline and the multi-step training setting gets superior performance compared to the mix training one. We also find the multi-step training gets a more stable training process to balance the two code generation and code evaluator tasks.

5 Related Work

5.1 Instruction Fine-Tuning

Large language models (LLMs) experience the Instruction fine-tuning (IFT) stage, which enhances their capability to accomplish tasks and adhere to human instructions. The term IFT is broadly used here to encompass a range of sequence-to-sequence fine-tuning applications. T5 (Raffel et al., 2023) was one of the first models to explore this approach, training on multiple supervised text-to-text tasks. Recent studies have delved into the multi-task instruction-based fine-tuning of pre-trained LLMs, aiming to improve their innate ability to perform various downstream NLP tasks effectively, such as FLAN (Wei et al., 2022), T0 (Sanh et al., 2022), and UnifiedQA (Khashabi et al., 2020), further expanded the task scope to improve the overall generalization ability of LMs.

Following the notable achievements of proprietary LLMs, particularly ChatGPT, there has been a growing focus on utilizing Instruction Fine-Tuning (IFT) to better align LLMs with human intentions, as highlighted in the research by (Brown et al., 2020; Ouyang et al., 2022). A common finding in these studies is that fine-tuning LMs on more diverse data can significantly improve model performance. Taori et al. (2023) chooses another approach, adopting a self-guided method, using ChatGPT to generate broader training data. Chiang et al. (2023) trains their model using user-shared conversations collected from ShareGPT.com. Xu et al. (2023b) introduced the Evol-Instruct method, which involves evolving existing instruction data to generate more complex and diversified datasets.

5.2 Large Language Models for Code

Large language models are often pre-trained on trillions of tokens following the scaling laws (Hoffmann et al., 2022; Kaplan et al., 2020), which demonstrates remarkable achievements across a broad spectrum of tasks and such an amount of text data is often a diverse composite with a non-negligible part of code (Zhang et al., 2023). Pioneered by Codex, researchers have also found continual pretraining on code to significantly benefit language models' performance on code. For example, Chowdhery et al. (2022a) post-trains PaLM on 7.8B additional code tokens to get PaLM-Coder and Rozière et al. (2023) train LLaMA 2 (Touvron et al., 2023a) on more than 500B code tokens to acquire Code LLaMA, which leads open-source models to a new height. For supervised fine-tuning, many works utilize larger, more capable teacher models to synthesize instruction data to finetune small language models (Mitra et al., 2023; Luo et al., 2023; Di et al., 2023b). Instead, Muennighoff et al. (2023); Wei et al. (2023) construct code instructions by mining existing code corpus, which are orthogonal to our method. Since code-supervised signals are easily collected by compiling and running them, reinforcement learning becomes another important branch. Numerous works have attempted to utilize reinforcement learning with feedback information offered by compilation and other sources. PanGu-Coder2 (Shen et al., 2023b) introduces a ranking loss based on unit tests, which helps to align with the capable code model deeply. However, these methods rely on human-annotated unit tests, which limits the application in the practical scenario.

6 Conclusion and Future Work

In this paper, we investigate two fine-tuning methods to improve the LLMs' performance on code generation. We first introduce a response augmentation strategy by using different ChatGPT system prompts to increase the diversity of code solutions. We find different chain-of-thought reasoning paths improve performance. Then, We adopt a multi-step training approach that combines traditional code generation and code evaluation objectives. We find improving one's ability to evaluate the correctness of code also enhances their ability to create it. For future work, we aim to explore the effect of our methods on the larger foundation models and parameter-efficient fine-tuning mechanisms. We also plan to increase the accuracy of evaluation signals via other ways like automatic unit tests and reinforcement learning methods.

7 Limitations

Our limitations are there-fold: (1) We only explore our method on the 7B/13B base models due to the computation cost. More experiments on the larger models and other code models should be conducted to confirm our conclusion. (2) We only use GPT-4 to evaluate the quality of generated code solutions. The performance of GPT-4 is still poor and limits the performance of our proposed method. More precise and open-source evaluation models should be explored in future work. (3) There is still room for optimization in our training data. For example, we find continually increasing the number of system prompts only gets marginal performance gains. Diversity-based compression methods (Lu et al., 2023; Liu et al., 2023c) may be valuable while the number of system prompts is large.

8 Broader Impacts

Similar to the other LLMs, our DolphCoder could also generate unethical, harmful, or misleading information, which is not considered in our work. Future research to address the ethical and societal implications is needed. DolphCoder is also susceptible to hallucination in ungrounded generation use cases due to its smaller size. This model is solely designed for research settings, and its testing has only been carried out in such environments. It should not be used in downstream applications, as additional analysis is needed to assess potential harm or bias in the proposed application.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie J. Cai, Michael Terry, Quoc V. Le, and Charles Sutton. 2021. Program synthesis with large language models. *ArXiv*, abs/2108.07732.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Sahil Chaudhary. 2023. Code alpaca: An instruction-following llama model for code generation. `https://github.com/sahil280114/codealpaca`.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde, Jared Kaplan, Harrison Edwards, Yura Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, David W. Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William H. Guss, Alex Nichol, Igor Babuschkin, S. Arun Balaji, Shantanu Jain, Andrew Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew M. Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code. *ArXiv*, abs/2107.03374.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022a. Palm: Scaling language modeling with pathways.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022b. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Peng Di, Jianguo Li, Hang Yu, Wei Jiang, Wenting Cai, Yang Cao, Chaoyu Chen, Dajun Chen, Hongwei Chen, Liang Chen, Gang Fan, Jie Gong, Zi Gong, Wen Hu, Tingting Guo, Zhichao Lei, Ting Li, Zheng Li, Ming Liang, Cong Liao, Bingchang Liu, Jiachen Liu, Zhiwei Liu, Shaojun Lu, Mingquan Shen, Guangpei Wang, Huan Wang, Zhi Yu Wang, Zhaogui Xu, Jiawei Yang, Qing Ye, Gehao Zhang, Yu Zhang, Zelin Zhao, Xunjin Zheng, Hailian Zhou, Lifu Zhu, and Xianying Zhu. 2023a. Codefuse-13b: A pretrained multi-lingual code large language model. *ArXiv*, abs/2310.06266.

Peng Di, Jianguo Li, Hang Yu, Wei Jiang, Wenting Cai, Yang Cao, Chaoyu Chen, Dajun Chen, Hongwei Chen, Liang Chen, et al. 2023b. Codefuse-13b: A pretrained multi-lingual code large language model. *arXiv preprint arXiv:2310.06266*.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. 2022. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. UNIFIEDQA: Crossing format boundaries with a single QA system. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1896–1907, Online. Association for Computational Linguistics.

Raymond Li, Loubna Ben Allal, Yangtian Zi, Niklas Muennighoff, Denis Kocetkov, Chenghao Mou, Marc Marone, Christopher Akiki, Jia Li, Jenny Chim, Qian Liu, Evgenii Zheltonozhskii, Terry Yue Zhuo, Thomas Wang, Olivier Dehaene, Mishig Davaadorj, Joel Lamy-Poirier, João Monteiro, Oleh Shliazhko, Nicolas Gontier, Nicholas Meade, Armel Zebaze, Ming-Ho Yee, Logesh Kumar Umapathi, Jian Zhu, Benjamin Lipkin, Muhtasham Oblokulov, Zhiruo Wang, Rudra Murthy, Jason Stillerman, Siva Sankalp Patel, Dmitry Abulkhanov, Marco Zocca, Manan Dey, Zhihan Zhang, Nourhan Fahmy, Urvashi Bhattacharyya, W. Yu, Swayam Singh, Sasha Luccioni, Paulo Villegas, Maxim Kunakov, Fedor Zhdanov, Manuel Romero, Tony Lee, Nadav Timor, Jennifer Ding, Claire Schlesinger, Hailey Schoelkopf, Jana Ebert, Tri Dao, Mayank Mishra, Alexander Gu, Jennifer Robinson, Carolyn Jane Anderson, Brendan Dolan-Gavitt, Danish Contractor, Siva Reddy, Daniel Fried, Dzmitry Bahdanau, Yacine Jernite, Carlos Muñoz Ferrandis, Sean M. Hughes, Thomas Wolf, Arjun Guha, Leandro von Werra, and Harm de Vries. 2023. Starcoder: may the source be with you! *ArXiv*, abs/2305.06161.

Yujia Li, David H. Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom, Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, Thomas Hubert, Peter Choy, Cyprien de, Masson d'Autume, Igor Babuschkin, Xinyun Chen, Po-Sen Huang, Johannes Welbl, Sven Gowal, Alexey, Cherepanov, James Molloy, Daniel Jaymin Mankowitz, Esme Sutherland Robson, Pushmeet Kohli, Nando de, Freitas, Koray Kavukcuoglu, and Oriol Vinyals. 2022. Competition-level code generation with alphacode. *Science*, 378:1092 – 1097.

Jiate Liu, Yiqin Zhu, Kaiwen Xiao, Qiang Fu, Xiao Han, Wei Yang, and Deheng Ye. 2023a. Rltf: Reinforcement learning from unit test feedback. *ArXiv*, abs/2307.04349.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. 2023b. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *arXiv preprint arXiv:2305.01210*.

Wei Liu, Weihao Zeng, Keqing He, Yong Jiang, and Junxian He. 2023c. What makes good data for alignment? a comprehensive study of automatic data selection in instruction tuning.

Keming Lu, Hongyi Yuan, Zheng Yuan, Runji Lin, Junyang Lin, Chuanqi Tan, Chang Zhou, and Jingren Zhou. 2023. #instag: Instruction tagging for analyzing supervised fine-tuning of large language models.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Wizardcoder: Empowering code large language models with evol-instruct. *ArXiv*, abs/2306.08568.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason.

Niklas Muennighoff, Qian Liu, Qi Liu, Armel Zebaze, Qinkai Zheng, Binyuan Hui, Terry Yue Zhuo, Swayam Singh, Xiangru Tang, Leandro von Werra, and S. Longpre. 2023. Octopack: Instruction tuning code large language models. *ArXiv*, abs/2308.07124.

Subhabrata Mukherjee, Arindam Mitra, Ganesh Jawahar, Sahaj Agarwal, Hamid Palangi, and Ahmed Hassan Awadallah. 2023. Orca: Progressive learning from complex explanation traces of gpt-4. *ArXiv*, abs/2306.02707.

Erik Nijkamp, Bo Pang, Hiroaki Hayashi, Lifu Tu, Huan Wang, Yingbo Zhou, Silvio Savarese, and Caiming Xiong. 2023. Codegen: An open large language model for code with multi-turn program synthesis.

OpenAI. 2023. Gpt-4 technical report. *ArXiv*, abs/2303.08774.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, I. Evtimov, Joanna Bitton, Manish P Bhatt, Cristian Cantón Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre D'efossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code. *ArXiv*, abs/2308.12950.

Baptiste Rozière, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Tal Remez, Jérémy Rapin, Artyom Kozhevnikov, Ivan Evtimov, Joanna Bitton, Manish Bhatt, Cristian Canton Ferrer, Aaron Grattafiori, Wenhan Xiong, Alexandre Défossez, Jade Copet, Faisal Azhar, Hugo Touvron, Louis Martin, Nicolas Usunier, Thomas Scialom, and Gabriel Synnaeve. 2023. Code llama: Open foundation models for code.

Victor Sanh, Albert Webson, Colin Raffel, Stephen H. Bach, Lintang Sutawika, Zaid Alyafeai, Antoine Chaffin, Arnaud Stiegler, Teven Le Scao, Arun Raja, Manan Dey, M Saiful Bari, Canwen Xu, Urmish Thakker, Shanya Sharma Sharma, Eliza Szczechla, Taewoon Kim, Gunjan Chhablani, Nihal Nayak, Debajyoti Datta, Jonathan Chang, Mike Tian-Jian Jiang, Han Wang, Matteo Manica, Sheng Shen, Zheng Xin Yong, Harshit Pandey, Rachel Bawden, Thomas Wang, Trishala Neeraj, Jos Rozen, Abheesht Sharma, Andrea Santilli, Thibault Fevry, Jason Alan Fries, Ryan Teehan, Tali Bers, Stella Biderman, Leo Gao, Thomas Wolf, and Alexander M. Rush. 2022. Multitask prompted training enables zero-shot task generalization.

Bo Shen, Jiaxin Zhang, Taihong Chen, Daoguang Zan, Bing Geng, An Fu, Muhan Zeng, Ailun Yu, Jichuan Ji, Jingyang Zhao, Yuenan Guo, and Qianxiang Wang. 2023a. Pangu-coder2: Boosting large language models for code with ranking feedback. *ArXiv*, abs/2307.14936.

Bo Shen, Jiaxin Zhang, Taihong Chen, Daoguang Zan, Bing Geng, An Fu, Muhan Zeng, Ailun Yu, Jichuan Ji, Jingyang Zhao, Yuenan Guo, and Qianxiang Wang. 2023b. Pangu-coder2: Boosting large language models for code with ranking feedback.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models.

Hugo Touvron, Louis Martin, Kevin R. Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Daniel M. Bikel, Lukas Blecher, Cristian Cantón Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony S. Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel M. Kloumann, A. V. Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, R. Subramanian, Xia Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zhengxu Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and

Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *ArXiv*, abs/2307.09288.

Yue Wang, Weishi Wang, Shafiq Joty, and Steven C. H. Hoi. 2021. Codet5: Identifier-aware unified pre-trained encoder-decoder models for code understanding and generation.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022. Finetuned language models are zero-shot learners.

Yuxiang Wei, Zhe Wang, Jiawei Liu, Yifeng Ding, and Lingming Zhang. 2023. Magicoder: Source code is all you need. *ArXiv*, abs/2312.02120.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023a. Wizardlm: Empowering large language models to follow complex instructions. *ArXiv*, abs/2304.12244.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023b. Wizardlm: Empowering large language models to follow complex instructions.

Ziyin Zhang, Chaoyu Chen, Bingchang Liu, Cong Liao, Zi Gong, Hang Yu, Jianguo Li, and Rui Wang. 2023. Unifying the perspectives of nlp and software engineering: A survey on language models for code.

A Case Study

Table 6 presents comparative examples of DolphCoder and WizardCoder. In this case, the models are required to determine whether an array can become a non-decreasing sequence after applying a circular right shift operation. From the case, we can observe that WizardCoder-13b generates an obvious logical error, where it only checks if the sequence is non-decreasing without performing any right shift operation. In contrast, DolphCoder accurately simulates the circular right shift operation and correctly identifies the termination condition. Furthermore, it is worth noting that DolphCoder considers more robust boundary input cases, which can be attributed to its training on a code evaluation task and a more diverse training dataset.

B Effect of Code Evaluation Capability of GPT-4

In our study, we utilize GPT-4 to create a training dataset for code evaluation. However, there is a concern regarding whether GPT-4 has the ability to generate a perfect golden label. To obtain the answer we test it on HumanEval. Specifically, we use the DIT model to generate multiple candidate answers for each question. Subsequently, we leverage GPT-4 to evaluate each of these candidate answers. And then we select the code solutions that GPT-4 identified as correct. Since the test cases are available in HumanEval, we can get the binary <passed> or <not passed> accuracy of GPT-4 which is just 79.4%, which indicates that the train data we constructed through GPT-4 may contain about 20% noise. The noise limits the upper-bound performance of the MOT method in this work. Additionally, we report the average pass@1 results before and after filtering by GPT-4, as shown in Table 7. We can observe that the pass@1 before filtering is 56.6% which is generated by the DIT model directly at 0.2 temperature. Then we calculate the pass@1 after golden filtering is 72.6% where we use golden unit tests to filter wrong solutions. We consider this percentage of 72.6% is a theoretical upper bound of GPT-4. We also report pass@1 of MOT and GPT-4 Filtering. We summarize key insights as follows: (1) GPT-4 can not perfectly perform code evaluation (62.7% vs 72.6%). (2) Our MOT can achieve a slightly worse performance than the GPT-4 Filtering (59.9% vs 62.7%). (3) GPT-4 evaluation capability limits the upper bound of MOT. How to improve automatic code evaluation is essential to future work.

<table>
<tr><td>Instruction</td><td>

```python
def move_one_ball(arr):
"""
    We have an array 'arr' of N integers arr[1],
    arr[2], ..., arr[N]. The numbers in the array will be randomly ordered.
      Your task is to determine if it is possible to get an array sorted in
      non-decreasing order by performing the following operation on the
      given array: You are allowed to perform right shift operation any
      number of times. One right shift operation means shifting all elements
       of the array by one position in the right direction. The last element
       of the array will be moved to the starting position in the array i.e.
       0th index. If it is possible to obtain the sorted array by performing
       the above operation then return True else return False. If the given
      array is empty then return True. Note: The given list is guaranteed to
       have unique elements ...
"""
```

</td></tr>
<tr><td>WizardCoder-13b</td><td>

Here's the Python script for the given problem:

```python
def move_one_ball(arr):
    if not arr:
        return True
    for i in range(len(arr)):
        if arr[i] > arr[i+1]:
            return False
    return True
```

The function takes an array 'arr' as input and returns 'True' if it is possible to obtain a sorted array in non-decreasing order by performing right shift operations on the given array. Otherwise, it returns 'False'.The function first checks if the given array is empty. If it is empty, it returns 'True' as it is possible to obtain a sorted array in non-decreasing order by performing no right shift operations.**Next, the function iterates through the array and checks if the current element is greater than the next element. If it is, then it is not possible to obtain a sorted array in non-decreasing order by performing any number of right shift operations.** Therefore, the function returns 'False'.If the function has not returned 'False' by this point, it means that the array is already sorted in non-decreasing order. Therefore, it returns 'True'.

</td></tr>
<tr><td>DolphCoder-13b</td><td>

Here is the Python script for the given problem:

```python
def move_one_ball(arr):
    if len(arr) == 0:
        return True
    if len(arr) == 1:
        return True
    if arr == sorted(arr):
        return True
    for i in range(len(arr)):
        arr.append(arr.pop(0))
        if arr == sorted(arr):
            return True
    return False
```

Explanation:
1. First, **we check if the given array is empty or has only one element.** If so, we return True as it is already sorted.
2. Next, we check if the given array is already sorted. If it is, we return True.
3. If the array is not sorted, **we perform a right shift operation on the array by appending the first element to the end of the array and popping the first element from the beginning.**
4. **We repeat step 3 until the array is sorted or we have performed all possible right shift operations.**
5. If the array is sorted after performing all possible right shift operations, we return True. Otherwise, we return False.

</td></tr>
</table>

Table 6: Examples of DolphCoder compared to WizardCoder.

Method	**Pass@1**
DIT	56.6
+MOT	59.9
+GPT-4 Filtering	62.7
+Golden Filtering	72.6

Table 7: Effect of code evaluation capability of GPT-4. We sample 10 answers with a temperature=0.2 and report the average pass@1 metric. +MOT denotes the MOT model based on the DIT model. GPT-4 or golden filtering means that we use GPT-4 or golden unit test cases to filter out these wrong answers from the DIT model, and then we report the average pass@1 among the remaining answers.

Graph-Structured Speculative Decoding

Zhuocheng Gong[1]*, Jiahao Liu[2], Ziyue Wang[3], Pengfei Wu[1]
Jingang Wang[2], Xunliang Cai[2], Dongyan Zhao[1,4]†, Rui Yan[5]†
[1]Wangxuan Institute of Computer Technology, Peking University
[2]Meituan; [3]Tianjin University; [4]National Key Laboratory of General Artificial Intelligence
[5]Gaoling School of Artificial Intelligence, Renmin University of China
{gzhch,zhaody}@pku.edu.cn, pengfeiwu1999@stu.pku.edu.cn
ruiyan@ruc.edu.cn, wangziyue@tju.edu.cn
{liujiahao12,wangjingang02,caixunliang}@meituan.com

Abstract

Speculative decoding has emerged as a promising technique to accelerate the inference of Large Language Models (LLMs) by employing a small language model to draft a hypothesis sequence, which is then validated by the LLM. The effectiveness of this approach heavily relies on the balance between performance and efficiency of the draft model. In our research, we focus on enhancing the proportion of draft tokens that are accepted to the final output by generating multiple hypotheses instead of just one. This allows the LLM more options to choose from and select the longest sequence that meets its standards. Our analysis reveals that hypotheses produced by the draft model share many common token sequences, suggesting a potential for optimizing computation. Leveraging this observation, we introduce an innovative approach utilizing a directed acyclic graph (DAG) to manage the drafted hypotheses. This structure enables us to efficiently predict and merge recurring token sequences, vastly reducing the computational demands of the draft model. We term this approach Graph-structured Speculative Decoding (GSD). We apply GSD across a range of LLMs, including a 70-billion parameter LLaMA-2 model, and observe a remarkable speedup of 1.73× to 1.96×, significantly surpassing standard speculative decoding[1].

1 Introduction

The impressive performance of Large Language Models (LLMs) comes with an efficiency bottleneck that hinders their broader adoption (Vaswani et al., 2017; Touvron et al., 2023a; OpenAI, 2022; Touvron et al., 2023b). In this context, speculative decoding (SD) emerges as a promising direction to accelerate the decoding process by reducing the number of forward passes of LLMs (Chen et al., 2023; Leviathan et al., 2023; Zhou et al., 2023; Spector and Ré, 2023; Miao et al., 2023a). The underlying idea of SD is "draft then verify": rather than generating one token at a time using the LLM, SD employs a smaller model to draft a hypothesis sequence of tokens covering several decoding steps and then uses the LLM to verify the hypothesis. Consequently, the decoding process includes a *draft stage* and a *verification stage*. In this scheme, the number of forward calls of LLMs can be significantly reduced.

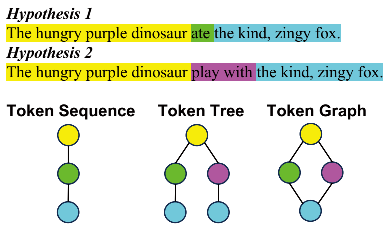


Figure 1: An illustrative comparison between the tree- and graph-structured draft token management.

However, SD faces its own set of challenges: the trade-off between performance and efficiency of the draft model limits the potential for acceleration. Ideally, the draft model should generate high-quality hypotheses while maintaining computational efficiency — a balance that is notoriously difficult to strike, echoing the adage that "there's no such thing as a free lunch." In this study, we address

[1]Code available at https://github.com/gzhch/gsd
*Work done during an internship at Meituan.
†Corresponding authors: Dongyan Zhao (zhaody@pku.edu.cn) and Rui Yan (ruiyan@ruc.edu.cn).

the challenge of enhancing the acceptance rate of the draft model's hypotheses without increasing the computational burden. Inspired by beam search (Graves, 2012) and tree attention (Spector and Ré, 2023; Miao et al., 2023a), our approach involves producing a bunch of hypotheses instead of a solitary one. Then, the LLM verifies these multiple hypotheses in a singlar forward pass and accepts the longest one. While tree decoding, which adopts a tree structure to organize the drafted tokens, presents an efficient implementation for simultaneously drafting all hypotheses, it also leads to exponential growth in the number of tokens at deeper levels of the tree, resulting in a prohibitive computational overhead. Consequently, the length of the hypotheses must be kept relatively short, which in turn leads to suboptimal use of the draft model's capabilities.

Our objective is to extend the length of drafted hypotheses without a corresponding rise in computational cost. To this end, we meticulously examined the hypotheses to find opportunities for improvement. We observe that hypotheses based on the same context are often semantically similar or related, and the variations among differing hypotheses typically boil down to only a handful of tokens. Notably, more than 70% of the drafted tokens tend to recur across various hypotheses. If we could discern when the draft model is likely to predict these re-occurring tokens, we could simply reuse them from previous drafts, thereby reducing the overall number of tokens that need to be generated. Capitalizing on this revelation, we propose Graph-structured Speculative Decoding (GSD), which uses a directed acyclic graph to organize the drafted tokens (Figure 1). In this graph, each path that stems from the root node corresponds to a unique hypothesis. This approach allows different hypotheses to share a substantial number of common nodes.

The pipeline of GSD follows that of standard SD (also the Sequence-structured SD, SSD), which encompasses a draft stage and a verification stage. In the draft stage, the draft model constructs a token graph containing multiple hypotheses. In the verification stage, the token graph is flattened into a sequence, enabling the LLM to validate all hypotheses concurrently. The longest one is then adopted as part of the final output. We conduct extensive experiments using LLaMA-70b, one of the largest open-source LLMs, showing that GSD drafts tokens not exceeding 2$\times$ the amount drafted by SSD on average, while tree-structured SD (TSD) drafted a token count that is more than 15 times greater. In terms of speedup, GSD outperforms all other methods, marking a significant advancement in speculative decoding techniques

2 Related Works

2.1 LLM Compression

Improving the efficiency of LLM inference has emerged as a pivotal research focus in recent years. The primary objective of model compression is to decrease computational demands and speed up the inference process. Research into the compression of large language models branches out into several directions, including knowledge distillation (Jiao et al., 2020; Sanh et al., 2019; Wang et al., 2021; Passban et al., 2021), quantization (Tao et al., 2022; Liu et al., 2023a,b; Dettmers et al., 2023; Xiao et al., 2023), network pruning (Liang et al., 2021; Frantar and Alistarh, 2023). Despite their innovations, these methods can be classified as lossy compression. This means that their efficiency improvements are intrinsically linked to a trade-off in performance, leading to the likelihood that a compressed LLM might produce compromised results.

2.2 LLM Decoding Acceleration

Alongside conventional model compression techniques, there is another branch of research that focuses on accelerating LLM inference without incurring information loss. Among these studies, speculative decoding (SD) (Chen et al., 2023; Leviathan et al., 2023; Zhou et al., 2023; Spector and Ré, 2023; Miao et al., 2023a) emerges as a promising technique. SD does not modify the model architecture, nor does it require supplemental data or retraining. SD typically employs a smaller model to draft initial predictions for “easy" tokens, while the LLM itself verifies these drafted tokens and generates “hard" tokens. Some researchers suggest that the smaller model is not essential for SD. For instance, the smaller model can be substituted with the LLM itself (Zhang et al., 2023) or a large text database (He et al., 2023). In addition to SD, other efforts are being made to enhance the decoding efficiency of LLMs. Blockwise parallel decoding (Stern et al., 2018), for example, is introduced to make predictions for multiple time steps in parallel. More recently, Medusa (Cai et al., 2023) has trained multiple prediction heads to predict the next set of tokens simultaneously.

3 Preliminaries: Sequence-structured Speculative Decoding

In this section, we establish the notation and provide a foundational overview of sequence-structured speculative decoding (SSD).

Consider an input sequence at time step t, denoted by $x_{\leq t} = \{x_1, x_2, ..., x_t\}$, where each x_i symbolizes the i-th token from the sequence. Let M_p be the target LLM we want to accelerate, and let M_q denote the draft model. The probabilities $p(x_{t+1}|x_{\leq t})$ and $q(x_{t+1}|x_{\leq t})$ represent the predictive distributions for the next token as given by M_p (the LLM) and M_q (the draft LM), respectively.

SSD leverages the draft model, M_q, to propose a hypothesis comprising γ tokens, which we denote as $h = \{\tilde{x}_{t+1}, \tilde{x}_{t+2}, ..., \tilde{x}_{t+\gamma}\}$. The drafting of each token, $\tilde{x}_t + i$, is modeled as follows:

$$\tilde{x}_{t+i} \sim q(x|x_{\leq t}, \tilde{x}_{t+1}, ..., \tilde{x}_{t+i-1}) \quad (1)$$

Upon completion of the draft stage, the LLM verifies the γ drafted tokens in a singular forward pass. The verification process, which compares predictions made by M_p and M_q to determine which tokens shall be accepted, can be conducted in both deterministic and non-deterministic ways. Deterministic verification accepts drafted tokens only if the LLM would generate the same. The non-deterministic way employs the sampling method used in previous studies (Chen et al., 2023). For the i-th token in the hypothesis, the acceptance probability is calculated as $\min(1, p(\tilde{x}_{t+i})/q(\tilde{x}_{t+i}))$. Should the token $\tilde{x}_{t+i}$ face rejection, all subsequent tokens in the hypothesis are also discarded, the verification process comes to a halt, and M_p regenerates the discarded token. This method ensures that the tokens that are ultimately accepted are representative of the output distribution characterized by M_p.

4 A Step Forward: Tree-structured Speculative Decoding

An intuitive idea for improving SSD is to draft multiple hypotheses instead of merely one. This is where Tree-structured SD (TSD) comes into play.

In each drafting step of SSD, the draft model predicts a single next token as described in Equation 1. After γ steps, the drafted tokens compose a sequence $\{\tilde{x}_{t+1}, \tilde{x}_{t+2}, ..., \tilde{x}_{t+\gamma}\}$. In contrast, TSD allows the draft model to consider k different alternatives for the next token at each drafting step. The resulting drafted tokens thus create a tree structure, with the root representing the context at the commencement of drafting, and each branch from the root depicting a different hypothesis.

After γ drafting steps, the resulting token tree has a depth of γ and a maximum out-degree of k and can contain up to $\frac{k^{\gamma+1}-1}{k-1}$ nodes, representing as many as k^γ unique hypotheses. Let's denote the collection of all hypotheses as $\{h_i\}_{i=1}^{k^\gamma}$. TSD holds a significant advantage over SSD; by enabling the generation of a larger pool of hypotheses in a single drafting stage, it raises the chances of having longer sequences of tokens accepted by the LLM. This boosts the acceptance rate of the SD process. Fundamentally, TSD operates in a manner analogous to beam search, maintaining multiple potential hypotheses within its tree structure during the draft stage and then selecting the most promising one during the verification stage.

4.1 Parallelized drafting and verifying via tree attention

The draft stage of TSD generates a multitude of hypotheses. A significant challenge within this framework is the efficient drafting of these multiple hypotheses. If one were to adhere to the traditional inference scheme that decodes one token at a time (akin to extending one branch of the token tree), the computational demands are apparently unacceptable given that the token tree contains $\frac{k^{\gamma+1}-1}{k-1}$ tokens to be decoded.

A promising resolution to this problem is by employing meticulous tree attention. Tree attention operates by flattening the token tree into a sequence and then simultaneously predicting the next node for all branches during a single forward draft, thus circumventing the necessity of performing a forward pass for each potential sequence. As illustrated in Figure 2, it accomplishes this by customizing the attention mask in such a way that each token is only allowed to attend to its ancestor nodes in the tree hierarchy, thus maintaining the correct dependencies amongst tokens.

The verification stage benefits from tree attention by validating all hypotheses within a single forward pass. After this process, the longest path that unfolds from the root node is chosen as the sequence to be accepted.

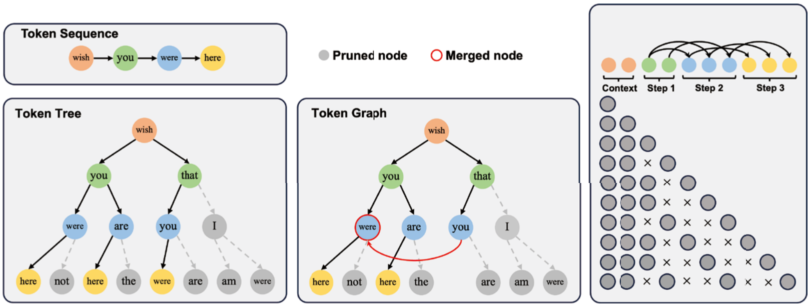


Figure 2: Overview of our method. (Left) GSD advances beyond TSD and SSD by implementing pruning strategies along with a re-occurring node merging technique. (Right) An illustration demonstrates the process by which the token tree (or graph) is flattened to a sequence. The sequence is then paired with a customized attention mask designed to uphold the proper dependencies between tokens to perform efficient drafting and verifying.

4.2 Pruning inferior branches

Despite the parallel drafting and verification with tree attention, TSD still consumes significantly more computation than SSD. The root cause lies in the exponentially increased length of input sequences processed in each forward pass. Transformer attention has a computational complexity that scales quadratically, $\mathcal{O}(l^2)$, with the sequence length l. While kv-caching does alleviate the computational load to some degree, the burden remains substantially heavier than that of SSD. Thus, to reduce the input sequence length, we need to perform pruning on the token tree.

We introduce two pruning strategies to moderate the size of the token tree. The first strategy is *probability pruning*. For a given node c within the token tree, where s_c denotes the path from the root to c, the logit probability is given by $q(c|x_{\leq t}, s_c)$. By setting a probability threshold θ_{prob}, we can filter out nodes: if $q(c|x_{\leq t}, s_c) < \theta_{prob}$, the node is deemed unlikely to be verified successfully and is marked as a leaf, halting further speculation.

The second strategy, *sibling pruning*, focuses on a node's child nodes $\{c_i\}_{i=1}^{k}$. Among these, we discern which nodes should remain as non-leaf nodes based on their logit probabilities relative to the highest probability among them. Specifically, let $m_q = \max_{i=1,\ldots,k} p(c_i|x_{\leq t}, s_{c_i})$. A child node c_i is then designated as a leaf if $p(c_i|x_{\leq t}, s_{c_i}) < \theta_{sib} \cdot m_q$. This approach ensures that the logit probabilities among sibling nodes do not deviate excessively from the maximum observed, m_q. The underlying idea is that, during probabilistic sampling, if the generation probabilities across a node's children vary greatly, the tokens associated with lower probabilities are less likely to be chosen. Therefore, it may not be necessary to keep these less probable nodes in the tree. Hence, when the output distribution for a current token is peaked—indicating high model confidence in its prediction—we need not preserve many child nodes. However, if the distribution is flatter, meaning multiple tokens have similar probabilities, it then becomes prudent to maintain a broader set of child nodes as candidates.

5 Graph-structured Speculative Decoding

Empirically, we observe that TSD often fails to surpass SSD, contrary to expectations. It appears that despite the utilization of pruning and tree attention, the cost of drafting multiple hypotheses still counterbalances the potential benefits that TSD offers. So we would like to ask: Can we further reduce the quantity of drafted tokens to enhance TSD's efficiency and effectiveness?

5.1 Same tokens re-occur among hypotheses

Before delving into GSD, we first conduct a pilot study to investigate the drafted hypotheses generated by TSD. We analyze the token trees from 100 distinct TSD runs, documenting the statistics of n-gram co-occurrences across various branches. The findings of this analysis are presented in Figure 3, and they give rise to several key insights:

- There is a high degree of commonality among the tokens in different hypotheses. As depicted in Figure 3, within a token tree of 10-

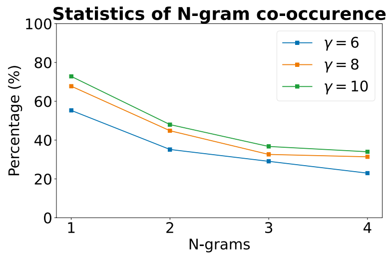


Figure 3: The proportion of tokens that are part of re-occurring n-grams within the token tree where the maximum out-degree k is 4. $\theta_{prob} = 0.2$ and $\theta_{sib} = 0.3$.

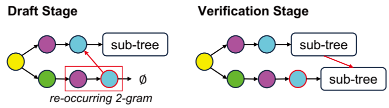


Figure 4: An illustration of how the token graph operates during the draft stage and the verification stage.

depth and 4-width, approximately 70% of tokens appear across multiple branches. This suggests that the generated hypotheses tend to form a cluster of semantically similar or related candidates, rather than branching off in completely disparate semantic directions.

- There is also a notable frequency of recurring n-grams within the token tree. This observation suggests that the similarities between different hypotheses extend beyond single tokens — entire segments of tokens (n-grams) are often duplicated among the various branches of the tree. This pattern points to redundancy in the token sequences being drafted, which may have implications for optimizing the efficiency of the speculative decoding process.

5.2 Identifying redundant nodes

We leverage the findings of identical tokens reappearing across different hypotheses to reduce computation.To this end, we introduce the concept of a τ-redundant node. A node is designated as τ-redundant when it corresponds to the last token of a re-occurring τ-gram. We assume that the presence of a τ-gram, defined as a sequence of τ consecutive identical tokens, signals a high degree of similarity between the current hypothesis and an alternate hypothesis already explored. This implies a strong likelihood that the sequence will continue to predict identical subsequent tokens.

5.3 Merging redundant nodes

Building on the concept of τ-redundant nodes, we implement a procedure to merge these nodes to enhance efficiency. The approach is straightforward: we mark τ-redundant nodes as leaf nodes, effectively ceasing their further expansion within the token tree. To merge the nodes, we first locate the first occurrence of the re-occurring τ-gram. We then draw a directed edge from the τ-redundant node to this first occurrence. By doing so, we establish that the nodes following the τ-redundant node will not need to be generated anew. Rather, we can directly reuse the results previously computed for the initial τ-gram occurrence. As a result of this merging process, the token tree is transformed into a directed acyclic graph (DAG), wherein no n-grams longer than τ will be repeated.

How does node merging hurt the performance? Merging nodes can result in a divergence from the nodes that would have otherwise been generated, potentially impacting the quality of the generated content. To quantify this effect, we calculate the KL divergence between the probability distributions of the next token across the vocabulary with or without node merging. Experimental results demonstrate that the KL divergence decreases rapidly with the increase of τ, suggesting that the impact of node merging diminishes significantly as the threshold τ is heightened. (Detailed results in Appendix C)

5.4 Token graph verification

There is still one step to go to fulfill GSD: the verification process. In the verification stage, we need to flatten the token graph to a sequence so that the LLM can verify all hypotheses simultaneously. To convert a DAG into a sequence while preserving the correct dependencies between tokens, we start by reverting the graph to its original tree structure. This is done by “unmerging" all previously merged nodes. During this process, the successor nodes of any redundant node are replicated from the relevant merged nodes (Figure 4). With the structure now back in the form of a tree, we can apply the same verification procedure as used in TSD.

Datasets	Mehtod	Model	Acceptance Rate	Drafted Token Num	Graph Success	Speedup
GSM8k	Self SSD	LLaMA-2-70b	-	-	-	1.37×
GSM8k	SSD	LLaMA-2-70b	0.795	629.9	-	1.85×
GSM8k	TSD	LLaMA-2-70b	0.894	8574.6	0%	1.81×
GSM8k	GSD	LLaMA-2-70b	0.917	793.1	27.7%	**1.96**×
XSUM	Self SSD	LLaMA-2-70b	-	-	-	1.28×
XSUM	SSD	LLaMA-2-70b	0.652	773.2	-	1.56×
XSUM	TSD	LLaMA-2-70b	0.784	22512.4	0%	1.42×
XSUM	GSD	LLaMA-2-70b	0.831	1544.8	32.8%	**1.73**×
XSUM	SSD	LLaMA-2-70b-chat	0.496	989.4	-	1.19×
XSUM	TSD	LLaMA-2-70b-chat	0.634	4601.2	0%	1.30×
XSUM	GSD	LLaMA-2-70b-chat	0.642	1545.7	30.4%	**1.32**×

Table 1: Evaluation results on 70b model. Self SSD is the method proposed by Zhang et al. (2023), which uses the LLM itself as the draft model. Speedup is the averaged result of greedy and top-p sampling. Here we only present the results of 70b models, full results can be found in Appendix D.

6 Experiments

6.1 Setup

There are two settings for verifying the drafted tokens: a deterministic setting where accepting the drafted tokens only if the LLM would generate tokens the same, and a non-deterministic setting where accepting the drafted tokens if they follow the same distribution with the LLM-itself generated tokens. In our main experiments, we adhere to the deterministic decoding setting if not specified. Under this condition, the generated output sequence is guaranteed to be identical to what would be produced via standard generation methods, so we can concentrate solely on efficiency metrics. Other details can be found in Appendix A.

Models We experiment on various backbone LLMs, including LLaMA (Touvron et al., 2023a), OPT (Zhang et al., 2022), and BLOOM (Workshop et al., 2022). For LLaMA, we use LLaMA-70b, LLaMA-70b-chat, and LLaMA-7b as large LLMs and LLaMA-7b and LLaMA-7b-chat, LLaMA-160m as draft models respectively. Note that LLaMA-160m is not an official checkpoint but a LLaMA-like model (Miao et al., 2023b). For OPT, we use OPT-13b as the LLM and OPT-350m as the draft model. For BLOOM, we use BLOOM-7b1 as the LLM and BLOOM-560m as the draft model.

Datasets We evaluate on Extreme Summarization (XSum) (Narayan et al., 2018), GSM8K (Cobbe et al., 2021), Alpaca (Taori et al., 2023), and WMT-14 (En-De) (Bojar et al., 2014). For GSM8K and WMT-14, we evaluate the full test set. For XSum and Alpaca, we randomly select 5000 instances for evaluation.

6.2 Main Results

Table 1 illustrates a comparison of our method against other speculative decoding approaches. Focusing on the speed-up ratio, we can see that GSD offers a significant advantage over the alternatives, achieving up to 1.94 and 1.70 times faster speeds. When examining the acceptance rate, we observe that both TSD and GSD have an acceptance rate that exceeds that of SSD by more than 10%. This indicates that tokens generated by the draft model are more likely to pass the verification process. Comparing the number of drafted tokens, we can see that TSD produces an order of magnitude more tokens than SSD. Hence, while TSD also has a high acceptance rate, this advantage is negated by the excessive number of tokens generated.

Additionally, we assess what proportion of tokens, which passed verification during the speculative decoding process, contained nodes from the merged subtrees, and find that approximately 30% of the drafting stages include such tokens. This indicates that, while the token graph is significantly smaller in node count compared to the token tree, we have successfully preserved the decoding information by recognizing and grafting nodes from different branches.

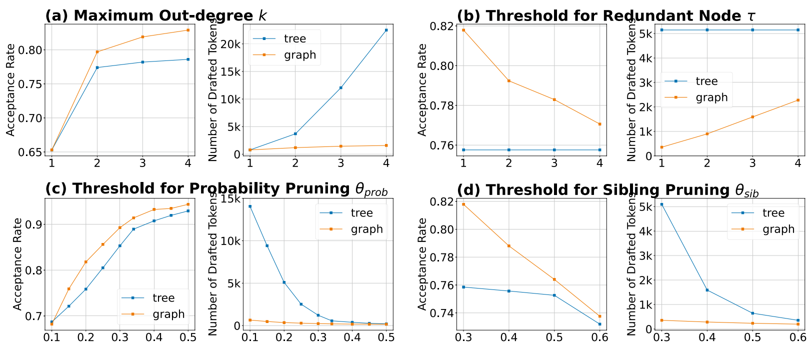


Figure 5: A series of ablation studies to investigate the hyperparameter configuration of maximum out-degree, redundant threshold, and two pruning techniques. All other hyperparameters adhere to the configuration described in section A.

Methods	SSD	TSD	GSD
GSM8k	1.80×	1.81×	2.14×
XSUM	1.58×	1.46×	1.89×

Table 2: Speedup results on non-deterministic speculative decoding on LLaMA-2-70b.

6.3 Ablation Study

Maximum Out-degree k Maximum out-degree k refers to the maximum number of child nodes that each node within the token tree (or graph) can possess. As depicted in Figure 5(a), as the k increases, the model is more likely to accept longer sequences in the verification stage due to the more diverse set of candidate hypotheses, thereby significantly enhancing the acceptance rate. However, the total number of nodes in the token tree increases exponentially as the increase of k as we have discussed in Section 4. When setting k to 4, the token tree contains more than 20000 tokens which leads to a heavy computation budget. In contrast, the token graph prevents the uncontrolled swell of node count that could impede computational efficiency by merging repeating sub-trees. This optimization allows the GSD to achieve a much higher acceptance rate while free from a rapid increase in nodes with the increase of k.

Threshold for Redundant Node τ As mentioned in Section 5.2, when two different hypotheses emanating from different branches share a common token sequence of length τ, they are identified as repetition and subsequently merged as a single branch. Thus, the larger the τ, the more radical the node merging becomes. As shown in Figure 5(b), as the increase of τ, the method becomes more conservative in fusing repeated branches, retaining more nodes in the token graph. Besides, the acceptance rate is inversely correlated with the redundant threshold. This implies that more aggressive node fusion leads to a more diverse set of candidate hypotheses. At first glance, this might seem paradoxical, since one would expect that aggressive node fusion, which reduces the number of nodes in the token graph, would decrease the diversity of hypotheses by merging similar sequences. However, when the merging happens, the two nodes that are merged as one then share a common child subtree in later drafting steps. By merging, the newly generated tokens within the subtree are simultaneously added to two different branches, while these tokens might not be generated by both independent branches if not merged. Thus, the node merging effectively introduces a greater variety of hypotheses by allowing for increased sharing of information between different parts of the token graph, which might otherwise remain isolated, leading to less efficient search space coverage.

Pruning Threshold $\theta_{prob}, \theta_{sib}$ The probability pruning technique prunes tokens of low logit probability and the sibling pruning technique involves pruning sibling nodes that had passed the probability-based pruning based on the maximum logit probability. As illustrated in the figure, both pruning strategies significantly reduce the number of generated tokens. However, these two pruning strategies have opposite effects on the acceptance rate. When the threshold is raised, probability pruning leads to an increase in the acceptance rate, while sibling pruning has a diminishing effect. This indicates that while probability pruning can help in focusing the speculative decoding process on more likely hypotheses, sibling pruning might lead to the removal of potential candidate hypotheses that could have been valid. The implications of these findings suggest that a delicate balance must be struck between pruning enough to maintain computational efficiency and avoiding overly aggressive pruning that could eliminate valid hypotheses.

6.4 Non-deterministic Setting

Table 2 represents performance under the non-deterministic decoding setting. This non-deterministic verification process determines whether a drafted token should be accepted by comparing the generating probability of the draft model and the LLM. Implementing GSD in this setting is a little tricky because GSD uses a shared logit distribution for redundant tokens, which could slightly deviate from the actual distribution. We have addressed the potential effects of this issue in Section 5.3 through experimental analysis. Furthermore, we conduct an explicit evaluation of the text quality, confirming that the performance disruption due to node merging is inconsequential. Detailed results can be found in Appendix E.

7 Analysis

7.1 Breakdown of Computation

Table 3 presents a computational analysis comparing different speculative decoding methods. Compared to TSD, the primary improvement offered by GSD lies in the reduction of time consumed during the draft stage, which can be attributed to the fewer number of nodes in the token graph, resulting in a reduced count of tokens that need to be processed during each drafting forward pass.

Besides, we find that, in addition to drafting and verifying, there is a significant portion of computation that should not be overlooked. We find that this computation is primarily associated with the update of the kv-cache of the draft model. Thus, improving the efficiency of the kv-caching represents a potential direction for further accelerating the speculative decoding.

Methods	Draft	Verification	Others
SSD	224.9 ms	133.5 ms	45.8 ms
TSD(k=2)	257.0 ms	172.4 ms	46.9 ms
GSD(k=2)	225.9 ms	170.0 ms	45.5 ms
TSD(k=4)	323.9 ms	184.4 ms	49.8 ms
GSD(k=4)	209.0 ms	178.3 ms	50.2 ms

Table 3: Breakdown of computation of a **single** draft-verification iteration.

Graph-like Speculative Decoding

[Input] Write a story about life on Mars:

[Output] The Martian Chronicles is a 1950 science fiction short story collection by Ray Brad bury that chronicles the exploration and settlement of Mars in the near future. The stories range from the first expeditions to the Red Planet to the eventual demise of the human race. The collection is notable for its use of science fiction to explore social and political themes, as well as its vivid and imaginative depiction of life on Mars. The stories are also notable for their use of allegory and symbolism, as well as their exploration of the human condition. The Martian Chronicles is considered one of the most influential works of science fiction ever written and has been adapted into a number of different media, including a television series, a radio drama, and a comic book. The Martian Chronicles is a collection of short stories by Ray Brad bury that chronicles the colonization of Mars by humans.

Figure 6: A visualization of the generation process of graph-structured speculative decoding. The black color represents the token generated by the verification model. Both red and blue are the accepted tokens. Red tokens are ordinarily drafted while blue tokens are from the merged nodes of the token graph.

7.2 Case Study

Figure 6 presents an illustrative example of GSD. This case demonstrates how the token graph assists in maintaining various hypotheses while simultaneously decreasing the total number of drafted tokens. Notably, approximately 30% of the accepted drafted tokens are derived from the subtrees associated with merged nodes, illustrating the efficiency gains achieved through GSD.

8 Conclusion

In this paper, we introduce graph-structured speculative decoding (GSD), a novel decoding strategy that utilizes a token graph to concurrently record a multitude of sequence hypotheses within a single draft stage. We propose a redundant node merging technique and two pruning strategies to constrain the size of the token graph without unduly compromising the diversity of hypotheses. Our extensive experiments demonstrate that GSD significantly increases the acceptance rate of drafted tokens while not introducing much computation, achieving a noticeable acceleration in speed compared to previous speculative decoding methods.

Limitations

We discuss the limitations of our work as follows: While our investigation has highlighted an interesting phenomenon of hypotheses generated from the same context contexts, we have not thoroughly examined the underlying mechanism that gives rise to this phenomenon. A deeper exploration into why these hypotheses exhibit such close semantic ties could unveil further insights that may benefit future research and applications.

Acknowledgement

References

Ondrej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Ale s Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, and Tri Dao. 2023. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. `https://github.com/FasterDecoding/Medusa`.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *arXiv preprint arXiv:2305.14314*.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Alex Graves. 2012. Sequence transduction with recurrent neural networks. *CoRR*, abs/1211.3711.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D Lee, and Di He. 2023. Rest: Retrieval-based speculative decoding. *arXiv preprint arXiv:2311.08252*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Chen Liang, Simiao Zuo, Minshuo Chen, Haoming Jiang, Xiaodong Liu, Pengcheng He, Tuo Zhao, and Weizhu Chen. 2021. Super tickets in pre-trained language models: From model compression to improving generalization. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6524–6538, Online. Association for Computational Linguistics.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023a. LLM-QAT: data-free quantization aware training for large language models. *CoRR*, abs/2305.17888.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023b. Llm-qat: Data-free quantization aware training for large language models. *arXiv preprint arXiv:2305.17888*.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023a. Specinfer: Accelerating generative LLM serving with speculative inference and token tree verification. *CoRR*, abs/2305.09781.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023b. Specinfer: Accelerating generative llm serving with speculative inference and token tree verification. *arXiv preprint arXiv:2305.09781*.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2022. Openai chatgpt.

Peyman Passban, Yimeng Wu, Mehdi Rezagholizadeh, and Qun Liu. 2021. Alp-kd: Attention-based layer projection for knowledge distillation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13657–13665.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Benjamin Spector and Christopher Ré. 2023. Accelerating LLM inference with staged speculative decoding. *CoRR*, abs/2308.04623.

Mitchell Stern, Noam Shazeer, and Jakob Uszkoreit. 2018. Blockwise parallel decoding for deep autoregressive models. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 10107–10116.

Chaofan Tao, Lu Hou, Wei Zhang, Lifeng Shang, Xin Jiang, Qun Liu, Ping Luo, and Ngai Wong. 2022. Compression of generative pre-trained language models via quantization. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4821–4836, Dublin, Ireland. Association for Computational Linguistics.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Wenhui Wang, Hangbo Bao, Shaohan Huang, Li Dong, and Furu Wei. 2021. MiniLMv2: Multi-head self-attention relation distillation for compressing pretrained transformers. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2140–2151, Online. Association for Computational Linguistics.

BigScience Workshop, Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Guangxuan Xiao, Ji Lin, Mickael Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning*, pages 38087–38099. PMLR.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2023. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *CoRR*, abs/2309.08168.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yongchao Zhou, Kaifeng Lyu, Ankit Singh Rawat, Aditya Krishna Menon, Afshin Rostamizadeh, Sanjiv Kumar, Jean-François Kagy, and Rishabh Agarwal. 2023. Distillspec: Improving speculative decoding via knowledge distillation. *CoRR*, abs/2310.08461.

A Additional Implementation Details

We establish both the maximum input sequence length and output sequence length at 512. Any input sequences exceeding 512 tokens are truncated. We set the maximum drafting step at 10 and adopt a draft-exiting mechanism to prematurely exit the drafting stage when the token probability drops below θ_{prob}. For the top-p sampling decoding, we set the top-p to 0.7 and temperature to 0.7. For graph decoding and tree decoding, we set the maximum out-degree k as 4. For the pruning configurations, we default to $\theta_{prob} = 0.2$ and $\theta_{sib} = 0.3$. We set $\tau = 2$.

B Comparison with Other Inference Acceleration Methods

Methods	GSM8K	XSUM
Medusa	2.01×	1.62×
GSD	1.96×	1.73×

Table 4: Speedup ratios on LLaMA-2-70b.

Except for speculative decoding, there have been other methods for accelerating the decoding of LLM. Among these studies, Medusa (Cai et al., 2023) is a simple yet effective method. We compare with Medusa on GSM8K and XSUM (Table 4). Besides, we want to mention that Medusa is dedicated to the same deterministic setting and employs a similar tree structure to manage the generated tokens, so it is possible to incorporate Medusa with our proposed graph structure to further optimize the token management. Hopefully, this would bring further acceleration.

C Impact of Node Merging on Logits Distribution

maximum out-degree	$\tau = 0$	$\tau = 1$	$\tau = 2$	$\tau = 3$	$\tau = 4$
k=3	1.19e-4	2.70e-6	5.77e-7	3.34e-7	5.13e-7
k=5	1.78e-4	4.70e-6	4-21e-7	7.62e-7	8.49e-7
k=∞	1.30e-4	3.11e-6	1.03e-6	9.27e-7	7.64e-7

Table 5: Averaged KL-divergence between the probability distributions across the vocabulary with or without node merging. Results are averaged over 1000 examples. We test on a series of k (maximum out-degree) and τ (the threshold for redundant node), showing that in most cases, merging redundant nodes brings minimal affection to the generation probability of subsequent tokens.

D Additional Results on Deterministic Setting

We present the evaluation results on BLOOM-7b1, OPT-13b, and LLaMA-7b in Table 6,7, and 8.

Methods	Alpaca	WMT-14 en-de	gsm8k
SSD	0.628 (1.12×)	0.705 (1.30×)	0.653 (1.18×)
TSD	0.783 (0.44×)	0.798 (0.59×)	0.741 (0.32×)
GSD	0.819 (1.48×)	0.812 (1.52×)	0.755 (1.26×)

Table 6: BLOOM-7b1 performance under $k = 4$, $\tau = 1$, $\theta_{prob} = 0.4$, $\theta_{sib} = 0.1$. BLOOM-560m serves as the draft model.

Methods	Alpaca	WMT-14 en-de	gsm8k
SSD	0.563 (1.12×)	0.621 (1.16×)	0.602 (1.08×)
TSD	0.672 (0.37×)	0.705 (0.38×)	0.770 (0.62×)
GSD	0.691 (1.15×)	0.733 (1.28×)	0.793 (1.22×)

Table 7: OPT-13b performance under $k = 4$, $\tau = 1$, $\theta_{prob} = 0.4$, $\theta_{sib} = 0.1$. OPT-350m serves as the draft model.

Methods	Alpaca	WMT-14 en-de	gsm8k
SSD	0.729 (1.22×)	0.783 (1.29×)	0.601 (1.04×)
TSD	0.846 (0.65×)	0.851 (0.56×)	0.775 (0.60×)
GSD	0.860 (1.31×)	0.863 (1.36×)	0.793 (1.16×)

Table 8: LLaMA-2-7b performance under $k = 4$, $\tau = 1$, $\theta_{prob} = 0.4$, $\theta_{sib} = 0.1$. LLaMA-160m serves as the draft model.

E Additional Results on Non-deterministic Setting

Table 9 shows results on LLaMA-2-7b under the non-deterministic setting. In this scenario, the text produced by the model is not necessarily identical to that which would be generated via a standard decoding process. Consequently, to ensure that GSD does not significantly impair output quality, we assess the quality of the generated text. The results are shown in Table 10.

F Further Analysis on GSD

We present some extra explorations in this section. GSD introduces a novel directed acyclic graph structure to manage the drafted tokens. Every branch starting from the root node forms a unique hypothesis. We analyze the positional structure of the accepted/rejected nodes within the graph.

Methods	Alpaca	WMT-14 en-de	gsm8k
SSD	0.695 (1.16×)	0.737 (1.21×)	0.540 (0.96×)
GSD	0.793 (1.34×)	0.848 (1.31×)	0.836 (1.18×)

Table 9: LLaMA-2-7b performance under $k = 4$, $\tau = 1$, $\theta_{prob} = 0.4$, $\theta_{sib} = 0.1$. The hyperparameter settings might not be optimal.

	Rouge-1	Rouge-2	Rouge-l
vanilla decoding	0.25	0.09	0.19
SSD	0.24	0.09	0.19
GSD ($\tau = 1$)	0.23	0.09	0.18
GSD ($\tau = 2$)	0.23	0.09	0.18

Table 10: Rouge-1/2/l scores on LLaMA-2-7b under non-deterministic setting.

Figure 7 shows the benefit of considering multiple hypotheses in enhancing the acceptance rate, with approximately half of the accepted tokens originating from Child-k nodes (where $k > 1$). These tokens are typically not taken into account in SSD. Comparing TSD and GSD, we can see that GSD slightly increases the acceptance rate for tokens positioned as Child-1. Figure 8 shows how varying the θ_{sib} threshold impacts the acceptance rate for tokens at each position. A higher θ_{sib} corresponds to more stringent pruning, resulting in fewer sibling nodes being retained. We can see a clear negative correlation between the increase of θ_{sib} and the acceptance rate for tokens at latter positions.

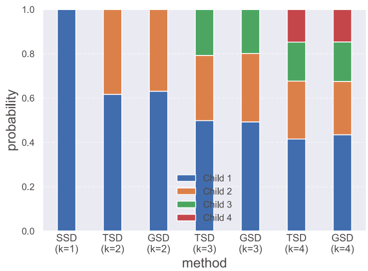


Figure 7: Percentage of i-th child being accepted. Results are averaged across all nodes within the token graph. We compare various speculative decoding configurations on LLaMA-7b. The child nodes within the decoding graph are ranked according to their probability, such that Child-1 corresponds to the token with the highest probability, while Child-k represents the token with the k-th highest logit probability.

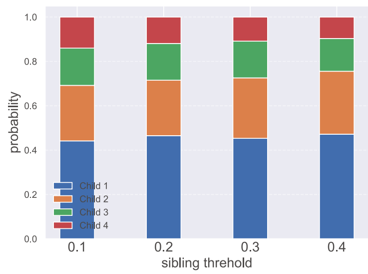


Figure 8: Percentage of i-th child being accepted. Results are obtained from LLaMA-7b with $k = 4$, $\tau = 1$, $\theta_{orob} = 0.4$

Harvesting Efficient On-Demand Order Pooling from Skilled Couriers: Enhancing Graph Representation Learning for Refining Real-time Many-to-One Assignments

Yile Liang
Meituan
Beijing, China
yileliang0412@163.com

Jiuxia Zhao
Meituan
Beijing, China
zhaojiuxia@meituan.com

Donghui Li
Meituan
Beijing, China
lidonghui03@meituan.com

Jie Feng*
Tsinghua University
Beijing, China
fengj12ee@hotmail.com

Chen Zhang†
Tsinghua University
Beijing, China
zhangchen0715@gmail.com

Xuetao Ding
Meituan
Beijing, China
dingxuetao@meituan.com

Jinghua Hao
Meituan
Beijing, China
haojinghua@meituan.com

Renqing He
Meituan
Beijing, China
herenqing@meituan.com

ABSTRACT

The recent past has witnessed a notable surge in on-demand food delivery (OFD) services, offering delivery fulfillment within dozens of minutes after an order is placed. In OFD, pooling multiple orders for simultaneous delivery in real-time order assignment is a pivotal efficiency source, which may in turn extend delivery time. Constructing high-quality order pooling to harmonize platform efficiency with the experiences of consumers and couriers, is crucial to OFD platforms. However, the complexity and real-time nature of order assignment, making extensive calculations impractical, significantly limit the potential for order consolidation. Moreover, offline environment is frequently riddled with unknown factors, posing challenges for the platform's perceptibility and pooling decisions.

Nevertheless, delivery behaviors of skilled couriers (SCs) who know the environment well, can improve system awareness and effectively inform decisions. Hence a SC delivery network (SCDN) is constructed, based on an enhanced attributed heterogeneous network embedding approach tailored for OFD. It aims to extract features from rich temporal and spatial information, and uncover the latent potential for order combinations embedded within SC trajectories. Accordingly, the vast search space of order assignment can be effectively pruned through scalable similarity calculations of low-dimensional vectors, making comprehensive and high-quality pooling outcomes more easily identified in real time. In addition, the acquired embedding outcomes highlight promising subspaces embedded within this space, i.e., scale-effect hotspot areas, which can offer significant potential for elevating courier efficiency.

SCDN has now been deployed in Meituan dispatch system. Online tests reveal that with SCDN, the pooling quality and extent have been greatly improved. And our system can boost couriers' efficiency by 45-55% during noon peak hours, while upholding the timely delivery commitment.

KEYWORDS

on-demand food delivery, order pooling, many-to-one assignment problem, graph representation learning

1 INTRODUCTION

1.1 Backgrounds

In recent years, there has been a remarkable upsurge in the widespread adoption of on-demand food delivery (OFD) services worldwide. With a mere few clicks, consumers can enjoy delicious meals without stepping out, all delivered right to their doorstep within just a few dozen minutes. This trend is attributable to the overarching shifts in technological innovation, including the popularity of apps and online platforms, and the growing dependence on third-party services for OFD. Global revenues for OFD sector were about \$90 billion in 2018, rose to \$294 billion in 2021, and are expected to exceed \$466 billion by 2026 [16]. Meituan Waimai, China's pioneering OFD platform has witnessed remarkable growth over the last decade. In 2023, the platform handles over 70 million orders daily, encompassing an extensive reach across almost 3,000 cities, counties and regions throughout China. 6.24 million couriers earned income via Meituan, with over 1 million actively engaged daily.

In OFD, orders are placed continuously by consumers from various locations. In response, the platform promptly gathers these newly initiated orders, channels them to merchants, and assigns dedicated couriers for pick-up and delivery within the promised delivery time. The platforms act as intermediaries, linking a multitude of consumers, merchants and couriers within the ecosystem, and strike a balance between gains and losses among these stakeholders to achieve sustained growth and prosperity [14]. Among these, consumers desire prompt services, merchants seek to maintain food freshness, couriers aim to fulfill enough orders to earn a

*Corresponding author.
†This work was fulfilled when Chen Zhang interned at Meituan.

Conference acronym 'XX, June 03–05, 2018, Woodstock, NY
2018.

decent income in a safe environment, while OFD platforms focus on boosting efficiency to reduce costs and increase profits.

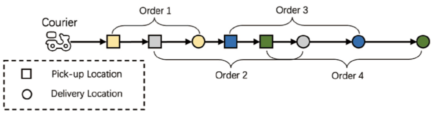


Figure 1: A courier's concurrent execution and route sequence of four orders.

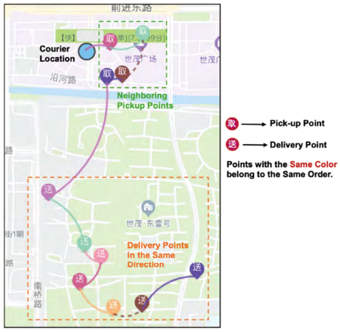
In this context, couriers often engage in concurrent execution of multiple delivery tasks, including order pick-up and delivery. **A pivotal efficiency source in OFD is to pool multiple orders for simultaneous delivery of a single courier in order assignment,** leveraging shared pick-up and delivery behaviours and travelling distances, enabling couriers serve more orders within committed delivery time limits. Facilitating comprehensive order pooling can effectively reduce delivery costs and enhance OFD sustainability[23, 24]. Figure 2(a) presents a high-quality order pooling example, where the courier's pickup points are highly concentrated, and the delivery destinations are aligned along a coherent route, enabling the courier to fulfill the deliveries with remarkable efficiency. However, unreasonable order pooling may result in detours and prolonged delivery times, severely undermining the stakeholders' experiences. Figure 2(b) illustrates a scenario in which unreasonable order pooling negatively impacts a courier's route, leading to an inefficient delivery trajectory.

In Meituan Waimai, the dispatch system conducts city-level batch order assignments every 30 seconds[14]. In each dispatch cycle, the system identifies available couriers for new orders, and assesses the matching degree (MD) between them, including convenience of route, over-time risk, and courier acceptance willingness. This evaluation process demands massive computations for pick-up and delivery route planning (PDRP) to simulate courier's behaviors after accepting orders [5]. Subsequently, through the resolution of a multi-objective many(order)-to-one(courier) assignment (MOA) problem, the system matches orders with the most suitable couriers to optimize the overall MD scores.

Constructing comprehensive and high-quality order pooling in order assignments stands as a key issue for OFD platforms to harmonize platform efficiency with stakeholder experience. Practically, there are two primary methods to facilitate comprehensive and high-quality order pooling in order assignments during each dispatch cycle. The first approach entails identifying suitable order combinations among all the pending orders, such as those with shared pick-up/delivery tasks or minimal detours, aiming to increase the ratio of MOA outcomes. The second approach focuses on matching orders with couriers whose existing assignments can share pick-up/delivery tasks or travel routes with the new orders, thereby optimizing the delivery process.

1.2 Challenges

However, OFD's distinct features present considerable challenges.



(a) High-quality order pooling.

(b) Unreasonable order pooling.

Figure 2: Order pooling examples.

(1) Computational complexity in real time. On one hand, the MD scores based on PDRP outcomes, are ***non-additive***. Specifically, the MD score of assigning multiple orders concurrently to a courier, is not equivalent to the sum of the scores of assigning each order individually to the same courier. Hence, to model the MOA problem and to obtain sufficient order combination results usually demands massive MD score calculations, which suffers from combinatorial explosion, as depicted in Figure 3. The MOA problem details can be found in Appendix A. For some big cities in China during noon peak, there amounts to over 3 thousand orders [1] to be assigned in each dispatch cycle, while each order can retrieve hundreds of couriers available for delivery on average. Assuming at most 5 orders assigned to a courier, and the average courier candidates for a order (combination) is 100, the calculation volume is

[1]It is the order volume in several geographically adjacent areas within a city, not the total order volume for the entire city.

$(C^1_{3000} + C^2_{3000} + C^3_{3000} + C^4_{3000} + C^5_{3000}) \times 100$. On the other hand, the MOA problem itself is categorized as an ***NP-hard*** integer programming problem, known for its extremely vast search space. Crafting online algorithms that perform effectively for the MOA is an exceptionally challenging task[3, 15, 35]. Moreover, the fast movement of couriers requires assignment decisions be made within a mere 10 seconds. This imperative time frame ensures the consistency of courier status between the information acquisition phase and the actual assignment moment.

Consequently, the platform tends to favor one(order)-to-one(courier) assignments during each dispatch cycle, a strategy that reduces computational volume and complexity, albeit at the expense of comprehensive order pooling.

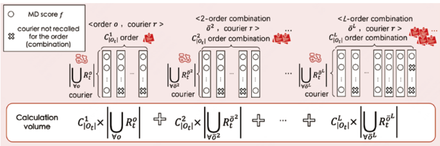


Figure 3: Calculation volume and search space for modeling and solving MOA problems in each dispatch cycle.

(2) Limited system awareness on the "last mile" offline environment. In OFD, the "last mile" offline environment is highly intricate and dynamic [34], encompassing unforeseen road closures, unknown natural obstacles, and pandemic-related lockdowns. OFD platforms are unable to fully access these extensive, finely-detailed spatiotemporal data during large-scale decision-making, due to insufficient map precision and digital capabilities, along with computational and storage constraints. Consequently, order pooling decisions based on coarse data and limited awareness, may not be reasonable, potentially harming courier experiences, causing delivery delays, and reducing delivery efficiency.

1.3 Related Work

Prior research on order pooling algorithms primarily focused on batching issues in traditional warehouse management [1, 19, 30]. However, the more relaxed time constraints of warehouse batching algorithms, typically in minutes, or even hours, are not well-suitable for the urgency required in OFD.

In recent years, research pertaining to OFD has gradually gained traction. The prevalent method for order pooling batches orders based on geographical proximity and closeness of their promised delivery time [22]. However, while these criteria-based batching rules are straightforward, they limit the scope for consolidation. An exact algorithm for order batching and assignment is proposed in [31], under the unrealistic assumption of perfect information about the arrival of orders. The study in [9] produces monthly OFD task groupings offline to facilitate order consolidation, However, their effectiveness is heavily reliant on order structure stability. Work in [10, 11] achieve order consolidation using iterative clustering on an order graph, but the batching algorithm's complexity and computational load hinder real-time processing. Similar work in [24] leverages additional decomposition mechanisms to reduce computational cost, yet it falls short of enabling real-time application despite notable performance gains. To satisfy the need for solutions within seconds, XGBoost models are built through supervised learning on historical order assignment results in [29, 32], to promote combined order assignments. However, the consolidation results struggle to break through the constraints of historical decisions, resulting in limited effectiveness.

1.4 Motivations

In light of the limitations present in existing work, it's worth noting that OFD platforms are equipped with a vast fleet of couriers, and extensive data on courier behaviors, especially from the skilled ones, which offer insights for high-efficiency and quality delivery services and enhance system intelligence. Skilled couriers (SCs) often possess a comprehensive grasp of the offline environment, including order distribution and road logistics, and continually improve their delivery skills to adapting to complex conditions. Moreover, our couriers can reject or transfer system-assigned orders, leveraging their expertise to optimize routes, minimizing detours and overtime. Additionally, the platform gathers courier preferences for pick-up and delivery locations via their apps, promoting efficient operations with fewer bottlenecks. **Thus SCs' behaviours of order selection, route sequence and feedback can provide the system superior courier-oriented pooling outcomes and help improve decision quality.**

In the past decade, the work on word representation learning has achieved cutting-edge results [7, 17, 20, 25]. Neural language models replace traditional high-dimensional and sparse word vectors with low-dimensional and dense embeddings, which assume that frequently co-occurring words share stronger statistical dependencies. Recently, graph representation learning (GRL) methods [4, 13] have increasingly been applied in various fields, including e-commerce [6, 8, 28], job search [12, 21], ride-sharing [26, 27], to discover diverse types of recommendations on the Web. These approaches have had a major impact in both academia and industry.

Drawing on prior achievements and the principle that **orders frequently combined together in SCs' routes tend to yield top-tier pooling results**, this paper aims to **using GRL methods to uncover the latent potential for order pooling embedded within the SCs' behaviour data**. Therefore, through scalable low-dimension vector calculations, instead of massive and time-consuming PDRP computations, we effectively prune the MOA problem's search space, shown in Figure 3, meanwhile extract small-scale and isolated subspaces promising for high-quality order consolidation results, facilitating real-time, effective order pooling.

1.5 Contributions

Accordingly, a systemic solution framework, named as SC delivery network (SCDN), is proposed. The novel contributions are:

(1) Graph Modelling: We construct a delivery network from SC route sequences, with *flow unit* (FU) as nodes linked by SC behavior sequences. An FU is a directed vector from pick-up areas of interest (AOI [2])[36] to delivery AOI. Orders of an FU share the

[2] AOIs are defined as non-overlapping irregular polygons that comprehensively divide and cover the space

same pick-up and delivery AOIs. The network is formulated as an ***attributed multiplex heterogeneous network*** (AMHEN), with FU nodes featuring multiple attributes for temporal and spatial information, and links representing two different types of courier behaviors, namely pick-up and delivery.

(2) Learning Algorithm: Based on GATNE [2], an effective GRL method for AMHEN, an enhanced attributed heterogeneous network embedding (EATNE) approach tailored for OFD is derived to obtain FU embeddings. First, given the fact that couriers move within a confined region[3] in a city, a **region-congregated negative sampling mechanism** is proposed as an enhancement over traditional randomized negative sampling to improve algorithm performance. Second, we employ a **customized margin ranking loss** instead of cross-entropy used by GATNE, aiming to refine embedding quality. Last, to address dispersed order distribution and limited FU coverage in SC behaviors, we build a **cold start mitigation mechanism**, using geographic information to generate embeddings of FUs previously unseen, thus broadening coverage.

(3) MOA Search Space Refinement and OFD Application: Utilizing FU embedding, we reconstruct the order combination and courier recall mechanisms within Meituan's dispatch system, facilitating superior real-time order pooling. Our use of SCDN refines order structure profiles and pinpoints scale-effect hotspots within MOA's vast search space, uncovering independent and small-scale subspaces for thorough and high-quality order pooling. Accordingly, an innovative delivery mode is developed to enhance courier efficiency without compromising service reliability.

To our knowledge, this is the first application of GRL methods in achieving real-time order pooling in OFD, now deployed in Meituan Waimai's dispatch system. Online tests shows significant improvement in order pooling. The total MD score of the MOA problem is improved by 5.3%, indicating more efficient order assignments with reduced detours and overtime risks. The newly-built mode cut the average incremental pick-up time for couriers [4] during noon peak by 51% and delivery time by 21%. These enhancements have led to a 45-55% boost in efficiency, maintaining consistent work hours and on-time delivery standards.

2 GRAPH REPRESENTATION LEARNING APPROACH

In this section, we will detail the step-by-step process by which the FU embeddings are acquired.

2.1 AMHEN Construction

The AMHEN is constructed based on SC route sequences as described below. The definition of SC and selection criteria of SC route sequences are introduced in Appendix B.

We first divide a SC's route sequence into distinct ***sessions***, using the rest or no action interval as a separator, presently set to 30 minutes. Then we transform the route sessions into ***FU sequences*** via replacing the orders in the sessions with their FUs. Since couriers participate in both pick-up and delivery actions during order fulfilment, there are two kinds of FU sequences: one based on pick-up behavior and the other on delivery, as shown in Figure 4. Diverse couriers' FU sequences may incorporate some common FUs.

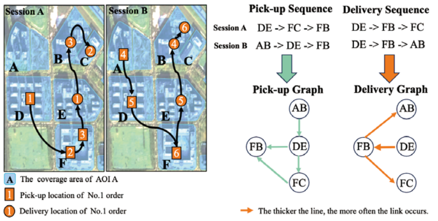


Figure 4: Illustration of AMHEN Construction, including 2 sessions. Session A contains 3 orders for FUs DE, FB and FC. The pick-up FU sequence is DE->FC->FB. And the delivery FU sequence is DE->FB->FC. Session B follows the same process.

To capture shared experiences of SCs, by treating FU as nodes and their connections in the FU sequence as links, we can integrate all the FU sequences into a unified yet ***heterogeneous graph***. Moreover, it is crucial to utilize the rich temporal and spatial information to enhance learning accuracy, e.g. average historical order amount and delivery distance of each FU, which makes the above graph an ***AMHEN***. More about node attributes is in Appendix C.

Denote AMHEN by $G = (V, E, A)$, where V is the FU node set, A is the attribute set for all nodes. FU node $v_i \in V$ owns fruitful attributes $\mathbf{x}_i \in A$ to describe its crucial characters. $E = (E^p, E^d)$ is the set of edges, which contains two types: pick-up and delivery. Specifically, there may be two types of edges between the FU nodes v_i and v_j, where $e^p_{ij} \in E^p$ indicates a pick-up edge and $e^d_{ij} \in E^d$ a delivery one. If two orders, belonging to FU nodes v_i and v_j, are successively picked up by the same SC, there exists a pick-up edge e^p_{ij} connecting v_i and v_j. Similarly, a delivery edge e^d_{ij} indicates there exist orders of FU nodes v_i and v_j that are consecutively delivered by the same SC. Hence, an AMHEN is constructed by merging massive records from tens of thousands of SCs.

2.2 Graph Representation Learning Model

Treating the AMHEN as input, we apply the model in GATNE [2] to produce node vector representation, i.e. FU embedding, which can be regarded as the aggregation of various node attributes and topology information in the graph, as depicted in Figure 5.

We divide the whole embedding of node v_i on each edge type τ into two parts, base embedding and edge embedding. The base embedding $\mathbf{b}_i$ is defined as a parameterized function of its attributes $\mathbf{x}_i$ as $\mathbf{b}_i = \mathbf{h}(\mathbf{x}_i)$, where $\mathbf{h}$ is a transformation function, while the k-th level edge embedding $\mathbf{u}^{(k)}_{i,\tau} \in \mathbb{R}^s, (1 \le k \le K)$ of node v_i on edge type τ is aggregated from the edge embeddings of neighbors:

$$\mathbf{u}^{(k)}_{i,\tau} = \text{aggregator}\left(\left\{\mathbf{u}^{(k-1)}_{j,\tau}, \forall v_j \in \mathcal{N}_{i,\tau}\right\}\right), \tag{1}$$

[3]A circular area with a diameter of 3-5 km, and the courier's designated residence as the center.

[4]defined as the interval between picking up the current order and the preceding one in the courier's route.

where $\tau \in \{p, d\}$ indicates the edge type, s is the dimension of edge embeddings, and $\mathcal{N}_{i,\tau}$ is the neighbors of node v_i on edge type τ. The initial edge embedding $\mathbf{u}_{i,\tau}^{(0)}$ is parameterized as the function of attributes $\mathbf{x}_i : \mathbf{u}_{i,\tau}^{(0)} = \mathbf{g}_\tau(\mathbf{x}_i)$, where $\mathbf{g}_\tau$ is a transformation function. The *aggregator* function is mean operation in practice.

We denote the K-th level edge embedding $\mathbf{u}_{i,\tau}^{(K)}$ by $\mathbf{u}_{i,\tau}$. Then the pick-up edge embedding $\mathbf{u}_{i,p}$ and the delivery edge embedding $\mathbf{u}_{i,d}$ of node v_i are combined as $\mathbf{U}_i = (\mathbf{u}_{i,p}, \mathbf{u}_{i,d})$. Given that the pick-up edge and delivery edge have different impacts, self attention mechanism is used to calculate the weights $\mathbf{a}_{i,\tau} \in \{\mathbf{a}_{i,p}, \mathbf{a}_{i,d}\}$.

$$\mathbf{a}_{i,\tau} = \operatorname{softmax}\left(\mathbf{w}_\tau^\top \tanh\left(\mathbf{W}_\tau \mathbf{U}_i\right)\right)^\top, \tag{2}$$

where $\mathbf{w}_\tau \in \mathbb{R}^{d_a}, \mathbf{W}_\tau \in \mathbb{R}^{d_a \times s}$ are trainable parameters for edge type τ. Thus, the overall embedding of node v_i for pick-up edge $\mathbf{v}_{i,p}$ and delivery edge $\mathbf{v}_{i,d}$ can be computed as:

$$\mathbf{v}_{i,p} = \mathbf{h}(\mathbf{x}_i) + \alpha_p \mathbf{a}_{i,p} \mathbf{M}_p^\top \mathbf{u}_{i,p} + \beta_p \mathbf{g}_p \mathbf{x}_i, \tag{3}$$

$$\mathbf{v}_{i,d} = \mathbf{h}(\mathbf{x}_i) + \alpha_d \mathbf{a}_{i,d} \mathbf{M}_d^\top \mathbf{u}_{i,d} + \beta_d \mathbf{g}_d \mathbf{x}_i, \tag{4}$$

where α_p and α_d indicate importance of pick-up and delivery edge embeddings, respectively, characterizing how pick-up and delivery behaviors affect courier efficiency. $\mathbf{M}_p, \mathbf{M}_d \in \mathbb{R}^{s \times d}$ are trainable parameters. β_p and β_d control the importance of node attributes.

The FU embedding $\mathbf{v}_i$ is the average of $\mathbf{v}_{i,p}$ and $\mathbf{v}_{i,d}$. The detailed implementation of EATNE can be found in Appendix D.

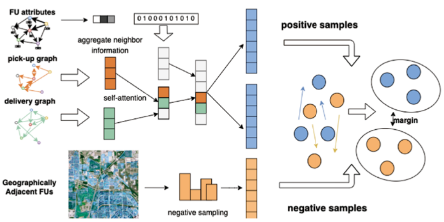


Figure 5: Illustration of the GRL Model.

2.3 Model Optimization

The positive data for training is generated by a meta-path-based random walk method and skip-gram model [17]. Given a set of pick-up FU sequences S, supposing that random walk with length l on S follows a path $S_p = (v_{s_1}, \ldots, v_{s_l})$, the pick-up context of v_{s_t} is denoted as $C_{v_{s_t}}^{S_p} = \{v_{s_k} \mid v_{s_k} \in S_p, |k - t| \le c, t \ne k\}$, where c is the size of the sampling window. Thus, given a node v_i and its all pick-up contexts, we can generate a positive pick-up data set $\mathcal{D}_p^P$ of positive pairs (v_i, v_j), which indicates SCs frequently pool the orders of these FU together. Similarly, we can generate a positive data set $\mathcal{D}_d^P$ from the delivery FU sessions.

Negative Sampling. Since couriers usually move within a confined region, negative samples from different regions are so easy for the model to distinguish in the whole training stage which makes the learning inefficient. Therefore, the negative data sets $\mathcal{D}_p^N$, $\mathcal{D}_d^N$ are constructed by random sampling from pick-up and delivery FU pairs in the same delivery region but excluding positive pairs, respectively. In other words, we select k-hop (k>2) neighbors of the FU node that share the same confined region as the challenging negative samples to enable the effective training of the proposed model. Traditional GATNE uses randomized negative sampling, yet ignores the regional effects in OFD. We find that the performance of GATNE decreases as the negative sampling scope expands and the effect becomes almost random as it reaches the city size.

Margin Ranking Loss. The learning task is to make the representation of positive FU pairs lying nearby in the embedding space, and the negative pairs different. However, achieving this with cross-entropy can be challenging. Therefore, a customized optimization objective based on margin ranking loss is proposed to maximize the distance between positive and negative samples in Equation 5, where $\gamma_p^P, \gamma_d^P, \gamma_p^N$ and γ_d^N are hyperparmeters representing the weights of various data sets, m_p and m_d are the minimum distance between negative pairs for pick-up and delivery, and cos represents the cosine similarity between FU embeddings.

$$\begin{aligned} L = & \frac{\gamma_p^P}{|D_p^P|} \sum_{(v_i,v_j) \in D_p^P} (1 - \cos(\mathbf{v}_{i,p}, \mathbf{v}_{j,p})) \\ & + \frac{\gamma_d^P}{|D_d^P|} \sum_{(v_i,v_j) \in D_d^P} (1 - \cos(\mathbf{v}_{i,d}, \mathbf{v}_{j,d})) \\ & + \frac{\gamma_p^N}{|D_p^N|} \sum_{(v_i,v_j) \in D_p^N} \max\left(0, \cos(\mathbf{v}_{i,p}, \mathbf{v}_{j,p}) - m_p\right) \\ & + \frac{\gamma_d^N}{|D_d^N|} \sum_{(v_i,v_j) \in D_d^N} \max\left(0, \cos(\mathbf{v}_{i,d}, \mathbf{v}_{j,d}) - m_d\right), \end{aligned} \tag{5}$$

2.4 Embedding Coverage Improvement

SC behaviors cover only 60% FUs. To compensate for the loss, we construct an extended delivery network based on geographical adjacency, shown in Figure 6. The criterion for judging spatial adjacency between FUs is the pick-up AOIs should be same [5] and the distance between delivery point is less than a threshold (currently 1km). If no adjacent FUs found, we will relax it to only consider the same pick-up AOI as a fallback. Then the embeddings of FUs previously unseen, can be estimated by aggregating the embeddings of their existing neighboring FUs in the network constructed above. This increases FU embedding coverage to over 80%.

3 APPLICATION AND DEPLOYMENT

3.1 Model Deployment

As introduced above, FU embeddings are learned from SC behavior data using EATNE. Different models are created for diverse scenarios, like weekday/weekend and peak/idle time, due to their significant differences in order structure. Moreover, to accelerate training in big cities, we use community detection algorithms to

[5] The emphasis on the same pick-up points is due to existing data analysis and courier feedback.

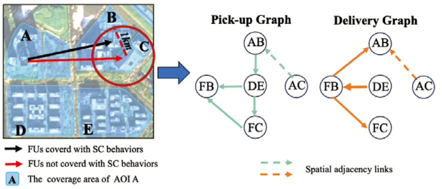


Figure 6: Illustration of spatial adjacency relationship in the extended delivery network.

partition the city network into separate regional groups for parallel training at regional group level.

The models are trained using 4 weeks of data across the country. They are trained for less than 2 weeks on 4 NVIDIA Tesla V100 GPUs with 32GB of memory each, and the models get updated every 2 weeks.

3.2 Information Mining

Leveraging the FU embeddings, we've created a set of indices.

(1) High-quality pooling probability (HPP) quantifies how well multiple orders can be consolidated together, sharing common pick-up and delivery times and travel distances. Since two FUs that consecutively appear in the SC behavior sequence often possess the above traits, this metric is calculated by the cosine similarity between the FU embeddings of these orders, reflecting the frequency of consecutive co-occurrence of the two FUs in SC behavior data.

$$p_{ij} = cos(\mathbf{v}_i, \mathbf{v}_j), \forall i, j \in V \tag{6}$$

Orders with high HPP values can be consolidated and assigned to the same courier to achieve efficient delivery.

(2) FU efficiency indicator (FEI) measures how much an order in this FU improves efficiency, based on how likely it is to be combined with orders from other FUs to form an efficient delivery sequence. It is calculated by the weighed aggregate of HPPs for the FU and its neighbouring FUs that share same or nearby pick-up or delivery AOIs. The weights are determined by the order volume of those neighboring FUs.

$$\eta_i = \sum_{j \in V_i} p_{ij} \times w_{ij}, \forall i \in V \tag{7}$$

The higher FEI values, the more likely for the order to be efficiently pooled with other orders, thus improving courier efficiency. FEI values are normalized at the city level for ease of comparisons.

(3) Scale-effect hotspot (SEH) for OFD refers to a local network of geographically proximate FUs, wherein the marginal cost and time of delivery for couriers fulfilling orders in this network progressively diminishes, allowing for comprehensive order consolidation within promised delivery time. In accordance, FUs in an SEH should have high FEI values, and any pair of FUs in the same SEH exhibit a relatively high HPP. And the total order volume for each SEH should exceed certain criteria.

$$S = \left\{ i \in V \middle| \begin{matrix} \eta_i > Thre_\eta; \\ p_{ij} > Thre_p, \forall i, j \in S \end{matrix} \right\} \tag{8}$$

3.3 Deployment in Dispatch System

The above information, including FU embeddings, FEI, SEH, are introduced in the system via offline features. As either low-dimensional vectors or scalars, they are performance-friendly to the real-time storage of the system. The main system framework is shown in Figure 7.

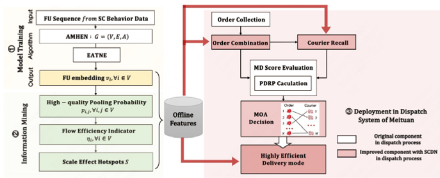


Figure 7: The main execution process of the dispatch system in each dispatch cycle.

3.3.1 Order Combination and Courier Recall. The MOA problem of our system is now solved by well-crafted constructive heuristics, i.e. imitation learning-enhanced iterated matching algorithm (ILIMA) [3], since metaheuristic algorithms with in-depth search fail to meet the real-time requirements [35]. Meanwhile, a few orders are combined in mutually exclusive groups based on the closeness of their origins and destinations, as well as promised delivery time, before MD score evaluation. However, the real-time performance severely restricts the search depth of the algorithm, resulting in insufficient and suboptimal order pooling.

With SCDN, we develop scalable mechanisms for courier recall and order combination, which can cut down the MOA search space, and let us focus our limited computation time on promising areas. Generally, orders with high HPP are formed as favorable combinations in advance, which can greatly expand the proportion of combined orders. Order combinations with low HPP and couriers whose on-hand orders mostly share low HPP with the new order are filtered out. Hence, we can facilitate high-quality order pooling in real time, without obvious increase in score calculation volume and computation time.

Order Combination. Based on HPP, high-quality order combinations can be identified and incorporated into ILIMA as expanding decision entities rather than single orders. As illustrated in Figure 8, on one hand, order combinations with very low HPP can be pruned to avoid unnecessary score calculation. On the other hand, since top-tier order combinations found by high HPP should be pooled to the same courier, other combinations containing partial orders, and conflicting orders themselves can be removed from the search space. It can guide ILIMA to search deeply and effectively without obviously increasing score calculation volume.

Courier Recall. When retrieving available couriers for an order (combination), we calculate the average value of HPP between it and the courier's on-hand orders, to quickly estimate MD between the order (combination) and the courier, instead of time-consuming score calculations. For the on-hand orders already picked up by the courier, its FU can be considered as the FU starting from the AOI

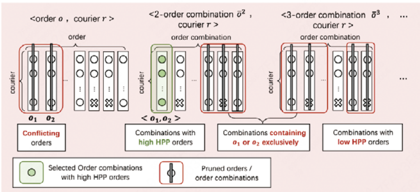


Figure 8: Order combination mechanism pruning MOA search space using HPP information. For example, for candidate orders A, B, C, and combinations AB, AC, BC, if AB and AC have higher HPP, then only AB, AC, B and C are preserved for MD evaluation, while A and BC can be eliminated.

where the courier is currently located and ending at its delivery AOI. This further helps to prune the MOA search space and reduce real-time computational pressure while maintaining solution quality, as shown in Figure 9.

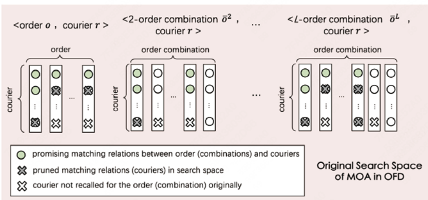


Figure 9: Courier recall mechanisms pruning MOA search space using HPP information.

The implementation details of order combination and courier recall mechanisms can be found in Appendix F.

3.3.2 Highly Efficient Delivery Mode. SEHs identified by SCDN, essentially represent small-scale subspaces deeply embedded within the MOA search space, where thorough and high-quality order pooling outcomes can be found, as shown in Figure 10. Then a new delivery mode can be built, wherein a dedicated group of couriers is assigned to each SEH, as opposed to receiving assignments in the entire region. Accordingly, the original large-scale MOA problem, initially solved within a vast search space shown in Figure 3, can be effectively decomposed into **a collection of small-scale MOA problems, defined within much smaller and independent subspaces, paving the way for comprehensive and in-depth real-time searching**. This approach serves to continually enhance the courier efficiency potential.

In the delivery mode, order assignments for each SEH are conducted as follows:

(1) Hourly SEH Identifications. SEHs for certain time periods in a city are found using binary programming (BP), which categorizes FUs with high FEI within a specified time period into a number of mutually exclusive sets. It aims to maximize the average HPP among FUs within each set, with FU quantities and total historical order volume in each set as constraints. Practically, in some mega cities like Beijing, SEHs in peak periods are determined every 30 minutes to capture the changes in order structure. The BP problems for SEH identification can be solved via genetic algorithm [18] within 10 minutes. More information is in Appendix G.

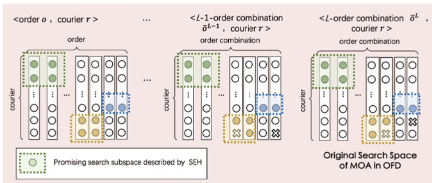


Figure 10: Promising MOA search subspace described by SEH.

(2) Real-time Parallel MOA Solutions. Order assignment for SEH is a scaled-down MOA problem. Given the limited area and stable order structure for SEH in a certain time period, the behavioral patterns of mode couriers are highly certain, thus simplifying the MD evaluation. In reality, we evaluate the MD via a weighted sum of average order increments for pick-up and delivery AOIs in a courier's route after the new order acceptance for SEH, instead of time-intensive PDRP calculations to simulate couriers' routes. Hence we can evaluate the MD between any promising order combinations and candidate couriers in real time, and solve the completely-modeled MOA problem for each SEH using a Hill-Climbing heuristic algorithm [33] in parallel, helping to pool orders effectively and thoroughly. Orders outside SEHs keep the existing assignment rules.

4 EXPERIMENTAL EVALUATION

4.1 Model Performance Evaluation

4.1.1 Model Learning Performance. Link prediction task is used to evaluate the performance of EATNE, with AUC, F1 score and PR as evaluation criteria. The experiments are conducted on a real-world dataset collected from Meituan delivery platform, using a single Linux server with NVIDIA Tesla V100 GPU with 32GB memory. The dataset contains 28,000 SC behavior records from 28 days in Beijing, China, forming a delivery network with about 70,000 FUs. For each edge type, the test set is generated with 10% randomly chosen positive edges and an equal number of negative edges, selected by regional negative sampling. Parameter details are in Appendix E.

First we examine the effectiveness of EATNE. Figure 11 shows that the original GATNE is hard to converge in this situation. While EATNE, armed with regional negative sampling and margin loss, produces superior outcomes in addition to converging much faster. Next the performance of EATNE in various graph configurations is

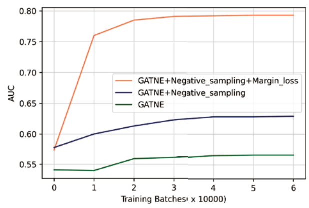


Figure 11: The convergence curve for different algorithms.

investigated. Table 1 shows that optimal performance is achieved by graphs with pick-up and delivery edges and node attributes, proving the validity of the proposed ANHEN. Notably, pick-up connections are more important than delivery ones, indicating pick-up behaviours have a greater effect on courier efficiency. Moreover, adding node attributes is highly impactful, highlighting order structure's key role in affecting courier efficiency.

Table 1: Model performance under different graph settings

Node Attr.	Pick-up Edge	Delivery Edge	AUC	F1	PR
✓	✓	✓	**0.79**	**0.72**	**0.75**
✗	✓	✓	0.64	0.60	0.59
✓	✗	✓	0.74	0.69	0.71
✓	✓	✗	0.76	0.71	0.73

4.1.2 FU Embedding Effectiveness. To evaluate the effectiveness of FU embeddings, we examine the training results via the data of the same district in Beijing. First, by performing DBSCAN clustering on learned embeddings, we evaluate if geographical similarity is encoded. Figure 12, which shows resulting 33 clusters, confirms the FUs from close locations are clustered together in the hidden space.

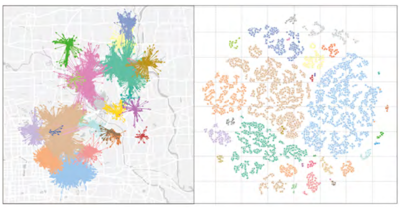
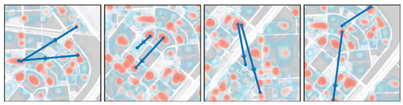
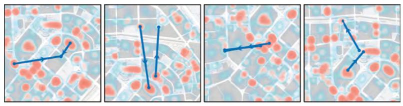


Figure 12: FU embedding clusters of a district in Beijing on map (left) and after T-SNE (right).

Next we demonstrate high-quality pooling potential can be captured by FU embedding similarity, i.e. HPP. Figure 13(a) shows four cases of FU pairs with high HPP, including (1) FU pair with pick-up and delivery AOIs located closely, (2) nearby parallel FU pair, (3) FU pair where one runs alongside the other, and (4) head-to-tail FU pair, with the tail one pointing high-order-density AOIs, leading to less courier empty run time [6] after completing deliveries. Orders in these FU pairs can be pooled for simultaneous delivery to improve courier efficiency. Meanwhile, we also identify FU pairs with low HPP. Figure 13(b) illustrates four cases of this situation, including (1) FU pair with the same delivery AOI but pick-up AOIs located far apart, (2) reverse parallel FU pair, (3) FU pair where one FU runs alongside the other but points a low-order-density AOI, leading to longer courier empty run time, and (4) head-to-tail FU pair that also leads to a low-order-density area. These FU pairs are unlikely to be efficiently pooled together and may undermine courier efficiency.



(a) FU pairs with high similarity.



(b) FU pairs with low similarity.

Figure 13: FU pair cases in different similarity levels.

4.2 Order Combination and Courier Recall

The proposed method, ILIMA + SCDN, is evaluated against the current online implementation, which utilizes ILIMA with ruled batching method, and MNDS, a metaheuristic algorithm used in [3]. Experiments are conducted in a mid-sized Chinese city, involving around 500 orders and 2,500 couriers in a dispatch cycle during noon peak.

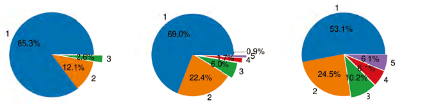
The comparison results on both computational cost and solution quality are presented in Table 2. The ILIMA+SCDN approach enhances the total MD score of MOA solutions by 5.3% compared to ILIMA+Rule method, without incurring a significant increase in time consumption. However, it lags by 1.2 *pp* behind MNDS. Despite this, MNDS requires exploration of a much larger search space and massive PDRP calculations, which takes over 20 seconds on average, making it unsuitable for online use. Hence, the proposed method excels at balancing computational time and solution quality, securing more optimal MOA solutions in real-time. Moreover, Figure 14 illustrates that the overall combination level grows as the percentage of couriers assigned only one order decreases by 16.3 *pp*. This shift results in increasing order consolidation. Online A/B test show that while maitaining delivery experience, couirer efficiency, i.e.orders completed per hour, is augmented by 3.7%.

Table 3 presents the results of offline experiments conducted with varying order volumes. In different order size scenarios, the

[6]Empty run time refers to the empty cruising time before carriers deliver next orders.

Table 2: Computation cost and score improvement of MOA.

Method	Online PDRP Calculations	Computation Time Online/s	MD Score Improvement
ILIMA+Ruled	44,541	5.6	0%
ILIMA+SCDN	48,998	6.9	5.3%
MNDS	/	/	6.5%



(a) ILIMA+Rule. (b) ILIMA+SCDN. (c) MNDS.

Figure 14: Combination level distribution.

proposed ILIMA+SCDN method significantly enhances the MD score over the existing ILIMA+Ruled method. Regarding PDRP Calculations, for orders fewer than 400, our proposed ILIMA+SCDN method demonstrates lower PDRP Calculations compared to the ILIMA+Ruled method. Nevertheless, as the order volume escalates, the computational burden of both methods exhibits nearly linear growth, aligning with the online time requirements.

Table 3: MOA results across various order sizes.

Method	(0, 200]	(200, 400]	(400, 600]	(600, 800]	(800, 1000]
MD Score Improvement					
ILIMA+Ruled	0%	0%	0%	0%	0%
ILIMA+SCDN	1.0%	4.0%	4.4%	5.5%	3.7%
MNDS	1.7%	5.3%	5.3%	6.9%	5.6%
PDRP Calculations					
ILIMA+Ruled	4,285	21,910	37,323	57,700	79,596
ILIMA+SCDN	4,250	20,589	38,358	65,011	94,292

4.3 Highly Efficient Delivery Mode

Figure 15 depicts 5 SEHs identified in a specific district of Beijing during weekday noon peak period (11:00-12:59). In response to fluctuations in order structures, the network configuration of each SEH is updated every half hour. On average, each SEH processes about 81 orders every half hour with an average HPP of 0.65, ensuring high order density and strong network connectivity. Moreover, the maximum number of orders pending assignment in each cycle is less than 10. By allocating 5 to 8 couriers per SEH, we significantly simplify the complexity of MOA solutions for each SEH.

Taking a SEH in Beijing as an example, online tests show a major boost in order pooling. During noon peak, a courier can accept over 7 orders at once. And the percentage of SEH couriers picking up over 5 orders simultaneously in the same AOI has risen by 23.5 *pp* compared to past performance. Likewise, the percentage of SEH couriers delivering over 5 orders at once in the same AOI has increased by 20 *pp*. The average courier incremental pick-up time has

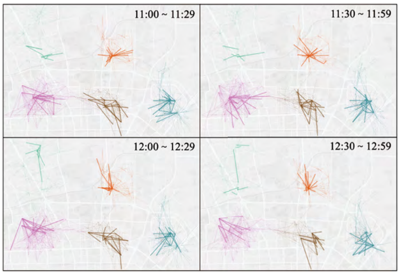


Figure 15: SEHs over time, with each image capturing half an hour. Colors denote different areas, bold lines for internal SEH FUs, and thin lines for external FUs.

been reduced by 51% and delivery time by 21%. These enhancements lead to a 45-55% boost in courier efficiency, i.e.orders completed per hour, while maintaining consistent work hours and on-time delivery standards. Figure 16 illustrates the superior performance of SEH mode against city average level in noon peak, where each bar corresponds to the trial performance of a specific courier.

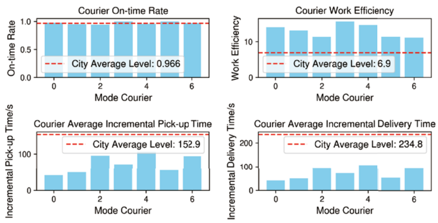


Figure 16: Courier performance in a SEH mode in noon peak.

5 CONCLUSION

This paper proposed a systemic solution framework, SCDN, based on an Enhanced GATNE method tailored for OFD, to resolve real-time OFD order pooling problem. It uncovers the latent potential for order pooling embedded within SC trajectories, which can strengthen system awareness and effectively inform decisions. Accordingly, the vast search space of NP-hard MOA problems in OFD is effectively pruned through scalable similarity calculations of simple vectors. Thus high-quality and comprehensive pooling outcomes are found in real time. Moreover, the outcomes highlight SEHs for OFD, where highly-efficient delivery modes are built for continuously improving efficiency. SCDN has now been deployed in Meituan. Online tests show it has achieved excellent performance and well-acknowledged by all the stakeholders.

REFERENCES

[1] Olivier Briant, Hadrien Cambazard, Diego Cattaruzza, Nicolas Catusse, Anne-Laure Ladier, and Maxime Ogier. 2020. An efficient and general approach for the joint order batching and picker routing problem. *European journal of operational research* 285, 2 (2020), 497–512.

[2] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 1358–1368.

[3] Jing-Fang Chen, Ling Wang, Hao Ren, Jize Pan, Shengyao Wang, Jie Zheng, and Xing Wang. 2022. An imitation learning-enhanced iterated matching algorithm for on-demand food delivery. *IEEE Transactions on Intelligent Transportation Systems* 23, 10 (2022), 18603–18619.

[4] Peng Cui, Xiao Wang, Jian Pei, and Wenwu Zhu. 2018. A survey on network embedding. *IEEE transactions on knowledge and data engineering* 31, 5 (2018), 833–852.

[5] Tao Feng, Huan Yan, Huandong Wang, Wenzhen Huang, Yuyang Han, Hongsen Liao, Jinghua Hao, and Yong Li. 2023. ILRoute: A Graph-based Imitation Learning Method to Unveil Riders' Routing Strategies in Food Delivery Service. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 4024–4034.

[6] Mihajlo Grbovic and Haibin Cheng. 2018. Real-time personalization using embeddings for search ranking at airbnb. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 311–320.

[7] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[8] Qingbo Hu, Sihong Xie, Jiawei Zhang, Qiang Zhu, Songtao Guo, and Philip S Yu. 2016. HeteroSales: Utilizing heterogeneous social networks to identify the next enterprise customer. In *Proceedings of the 25th International Conference on World Wide Web*. 41–50.

[9] Shenggong Ji, Yu Zheng, Zhaoyuan Wang, and Tianrui Li. 2019. Alleviating users' pain of waiting: Effective task grouping for online-to-offline food delivery services. In *The World Wide Web Conference*. 773–783.

[10] Manas Joshi, Arshdeep Singh, Sayan Ranu, Amitabha Bagchi, Priyank Karia, and Puneet Kala. 2021. Batching and matching for food delivery in dynamic road networks. In *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2099–2104.

[11] Manas Joshi, Arshdeep Singh, Sayan Ranu, Amitabha Bagchi, Priyank Karia, and Puneet Kala. 2022. FoodMatch: Batching and Matching for Food Delivery in Dynamic Road Networks. *ACM Transactions on Spatial Algorithms and Systems (TSAS)* 8, 1 (2022), 1–25.

[12] Krishnaram Kenthapadi, Benjamin Le, and Ganesh Venkataraman. 2017. Personalized job recommendation system at linkedin: Practical challenges and lessons learned. In *Proceedings of the eleventh ACM conference on recommender systems*. 346–347.

[13] Shima Khoshraftar and Aijun An. 2022. A survey on graph representation learning methods. *arXiv preprint arXiv:2204.01855* (2022).

[14] Yile Liang, Donghui Li, Jiuxia Zhao, Xuetao Ding, Huanjia Lian, Jinghua Hao, and Renqing He. 2023. Enhancing Dynamic On-demand Food Order Dispatching via Future-informed and Spatial-temporal Extended Decisions. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*. 4702–4708.

[15] Vittorio Maniezzo, Thomas Stützle, and Stefan Voß. 2021. *Matheuristics*. Springer.

[16] Eva-Marie Meemken, Marc F Bellemare, Thomas Reardon, and Carolina M Vargas. 2022. Research and policy for the food-delivery revolution. *Science* 377, 6608 (2022), 810–813.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* 26 (2013).

[18] Seyedali Mirjalili and Seyedali Mirjalili. 2019. Genetic algorithm. *Evolutionary Algorithms and Neural Networks: Theory and Applications* (2019), 43–55.

[19] Eduardo G Pardo, Sergio Gil-Borrás, Antonio Alonso-Ayuso, and Abraham Duarte. 2023. Order Batching Problems: taxonomy and literature review. *European Journal of Operational Research* (2023).

[20] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[21] Rohan Ramanath, Hakan Inan, Gungor Polatkan, Bo Hu, Qi Guo, Cagri Ozcaglar, Xianren Wu, Krishnaram Kenthapadi, and Sahin Cem Geyik. 2018. Towards deep and representation learning for talent search at linkedin. In *Proceedings of the 27th ACM international conference on information and knowledge management*. 2253–2261.

[22] Damian Reyes, Alan Erera, Martin Savelsbergh, Sagar Sahasrabudhe, and Ryan O'Neil. 2018. The meal delivery routing problem. *Optimization Online* 6571 (2018).

[23] Akhil Shetty, Junjie Qin, Kameshwar Poolla, and Pravin Varaiya. 2022. The Value of Pooling in Last-Mile Delivery. In *2022 IEEE 61st Conference on Decision and Control (CDC)*. IEEE, 531–538.

[24] Michele D Simoni and Matthias Winkenbach. 2023. Crowdsourced on-demand food delivery: An order batching and assignment algorithm. *Transportation Research Part C: Emerging Technologies* 149 (2023), 104055.

[25] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*. 1067–1077.

[26] Lei Tang, Zihang Liu, Rongguo Zhang, Zongtao Duan, and Yunji Liang. 2021. Who Will Travel With Me? Personalized Ranking Using Attributed Network Embedding for Pooling. *IEEE Transactions on Intelligent Transportation Systems* 23, 8 (2021), 12311–12327.

[27] Lei Tang, Zihang Liu, Yaling Zhao, Zongtao Duan, and Jingchi Jia. 2020. Efficient ridesharing framework for ride-matching via heterogeneous network embedding. *ACM Transactions on Knowledge Discovery from Data (TKDD)* 14, 3 (2020), 1–24.

[28] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 839–848.

[29] Xing Wang, Ling Wang, Shengyao Wang, Yang Yu, Jing-fang Chen, and Jie Zheng. 2021. Solving online food delivery problem via an effective hybrid algorithm with intelligent batching strategy. In *International Conference on Intelligent Computing*. Springer, 340–354.

[30] Jianglong Yang, Li Zhou, and Huwei Liu. 2021. Hybrid genetic algorithm-based optimisation of the batch order picking in a dense mobile rack warehouse. *Plos one* 16, 4 (2021), e0249543.

[31] Baris Yildiz and Martin Savelsbergh. 2019. Provably high-quality solutions for the meal delivery routing problem. *Transportation Science* 53, 5 (2019), 1372–1388.

[32] Yang Yu, Qingte Zhou, Shenglin Yi, Huanyu Zheng, Shengyao Wang, Jinghua Hao, Renqing He, and Zhizhao Sun. 2021. Delay to group in food delivery system: A prediction approach. In *International Conference on Intelligent Computing*. Springer, 540–551.

[33] Lingyu Zhang, Tao Hu, Yue Min, Guobin Wu, Junying Zhang, Pengcheng Feng, Pinghua Gong, and Jieping Ye. 2017. A taxi order dispatch model based on combinatorial optimization. In *Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining*. 2151–2159.

[34] Jie Zheng, Ling Wang, Li Wang, Shengyao Wang, Jing-Fang Chen, and Xing Wang. 2022. Solving stochastic online food delivery problem via iterated greedy algorithm with decomposition-based strategy. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 53, 2 (2022), 957–969.

[35] Qingte Zhou, Huanyu Zheng, Shengyao Wang, Jinghua Hao, Renqing He, Zhizhao Sun, Xing Wang, and Ling Wang. 2020. Two fast heuristics for online order dispatching. In *2020 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 1–8.

[36] Yida Zhu, Liying Chen, Daping Xiong, Shuiping Chen, Fangxiao Du, Jinghua Hao, Renqing He, and Zhizhao Sun. 2023. C-AOI: Contour-based Instance Segmentation for High-Quality Areas-of-Interest in Online Food Delivery Platform. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 5750–5759.

A MANY-TO-ONE ASSIGNMENT PROBLEM AT EACH DISPATCH CYCLE

As shown in Figure 3, the calculation volume increases very fast with the number of orders and couriers. Different order combinations of order set O_t are considered. For example, the number of $l-$order combinations is $C^l_{|O_t|}$. Since the MD score of assigning combinations of orders is not equivalent to the sum of scores of individual assignments. The calculation volume of MD score is $\sum_{l \in L} C^l_{|O_t|} \times |\cup_{\forall \bar{o}^l} R_t^{\bar{o}^l}|$, where $R_t^{\bar{o}^l}$ is the set of couriers for $l-$order combination $\bar{o}^l$ at dispatch time t.

$$
\begin{aligned}
&\min_{x_t \in \aleph_t} \sum_{\bar{o} \in comb(O_t)} \sum_{r \in R_t^{\bar{o}}} \left(\sum_{g \in G} \eta_t^g \times f_{t,r}^{g,\bar{o}} \right) \times x_r^{\bar{o}} \\
&s.t. \aleph_t = \left\{ \begin{array}{l} \sum_{\bar{o}(o)} \sum_{r \in R_t^{\bar{o}(o)}} x_r^{\bar{o}(o)} = 1, \forall o \in O_t \\ x_r^{\bar{o}} = \prod_{o \in \bar{o}} x_r^o, \forall \bar{o} \in comb(O_t) \end{array} \right\}
\end{aligned}
\tag{9}
$$

Algorithm 1 EATNE for OFD

Input: Network G; Embedding dimension d; Edge embedding dimension s; Window size c; Learning rate η; Marigin loss min distance m_p, m_d; coefficient $\alpha, \beta, \gamma_p^P, \gamma_d^P, \gamma_p^D, \gamma_d^D$.
Output: Embedding $\mathbf{v_i}$, and Embeddding $\mathbf{v}_{i,p}$ and $\mathbf{v}_{i,d}$ on the pick-up and delivery edge for all $v_i \in V$.

```
Initialize all the model parameters θ.
Generate positive data sets D_p^P and D_d^P by random walk on the pick-up and delivery edge, respectively.
Randomly sample FU pairs within the same delivery region, then add to negative data set D_p^N and D_d^N.
while not converged do
    for each FU pair in D_p^P, D_d^P do
        Calculate v_{i,p} and v_{i,d} using Equation (4) and (5) respectively;
        Sample m negative samples and calculate loss value using Equation (6).
        Update model parameters θ by ∂E/∂θ.
    end for
end while
Set v_i as the average of v_{i,p} and v_{i,d}.
```

After getting all these MD scores, the MOA problem can be formulated into an integer programming problem in Equation (9). The objective function is to minimize the total MD scores for different goals, and $f_{t,r}^{g,\bar{o}}$ is the MD score of assigning order combination $\bar{o}$ to courier r at time t for goal g, $comb(O_t)$ refers to all the possible combinations constructed by orders in O_t, η_t^g is the weight of goal g in the objective function at time t. The constraint is to make sure each combination $\bar{o}$ can only be assigned to one courier and only one combination of each order can be selected. $\bar{o}(o)$ represents the order combination containing order o.

B DEFINITION OF SKILLED COURIER AND SELECTION CRITERIA OF ROUTE SESSIONS

As mentioned above, SC refers to the couriers with relatively high efficiency, currently set top rank 5%-35% in a delivery region. It should be noted that in order to prevent extreme cases from affecting the validity of the learning outcomes, the top 5% of couriers have been excluded.

The SC route sessions of both pick-up and delivery type, for constructing the network are selected based on the following criteria:

(1) time interval between the execution of two consecutive orders less than 30 minutes;

(2) no overtime orders;

(3) no speeding behaviours;

(4) no orders with negative feedback reported.

Then based on the carefully selected sessions of SCs, we construct the corresponding AMHEN using the method outlined in Section 2.

C FU NODE ATTRIBUTES IN AMHEN

We incorporate rich spatial and temporal information as attributes of a FU node, for a specific scenario (i.e., weekday/weekend, peak/idle time), mainly including:

(1) average order volume of FU, and the corresponding pick-up and delivery AOIs in the scenario for last 30 days;

(2) average meal-waiting and pick-up time duration of the corresponding pick-up AOI in the scenario for last 30 days ;

(3) average delivery time duration of the corresponding delivery AOI in the scenario for last 30 days ;

(4) average delivery distance of the FU;

(5) average FU delivery period of time since consumers order in the scenario for last 30 days;

(6) type and number of natural barriers (e.g. bridge, river, highway) along the FU path;

(7) latitudes and longitudes of the center points of the corresponding pick-up and delivery AOIs;

(8) the proportion of SCs who chose the corresponding pick-up and delivery AOIs as their preferred locations for the scenario in the past 30 days.

D IMPLEMENTATION OF EATNE ALGORITHM

The proposed EATNE algorithm is summarized in Algorithm 1.

E EATNE MODEL PARAMETER CONFIGURATION

The detailed parameter setting is shown in Table 4. We employ the Adam optimizer with default settings for training. The model implements early stopping if there's no improvement in the ROC-AUC on the validation set within a single training epoch.

Table 4: Parameter configuration of EATNE model.

Notation	Description	Setting Value
d	base embedding dimension	200
s	edge embedding dimension	20
l	random walk length	10
c	sampling window size	3
m_p, m_d	margin loss min distance	0.3
η	learning rate	0.001
$\alpha_p, \alpha_d, \beta_p, \beta_q$	edge weights	1
$\gamma_p^P, \gamma_d^P, \gamma_p^D, \gamma_d^D$	weights in loss objective	1

F IMPLEMENTATION DETAILS OF ORDER COMBINATION AND COURIER RECALL.

The MOA problem in our system is now solved using a constructive heuristic framework. The process during each dispatch cycle may require multiple iterations. Let O^k denote the set of pending orders during iteration k, with $O^0 = O$ initially, where O represents all pending orders during this dispatch cycle. And at iteration k,

1. **Evaluation stage**: For the pending orders O^k and their associated recalled courier candidates $R_o^k, o \in O^k$, MD scores $\{\{f_o^r\}_{r \in R_o^k}\}_{o \in O^k}$ are calculated.
2. **Matching stage**: Based on current MD scores, a one(order)-to-one(courier) assignment decision is made following greedy policy (aiming to optimize the sum of MD scores for all matching relations at the current iteration). This may result in only a subset O^k being successfully assigned.

(3) **Termination condition**: Denote the remaining unassigned orders as $\overline{O}^k$. If $\overline{O}^k = \emptyset$, stop the iterations. Otherwise, update the state of couriers by including newly assigned orders, let $O^{k+1} = \overline{O}^k$, $k = k + 1$, proceed to Step (1).

F.1 Order Combination Mechanism.

Although the above algorithm has good performance in solving, it tends to promote one-to-one assignment results, which is not conducive to sufficient order pooling. To facilitate many-to-one assignments, high-quality and mutually exclusive order combinations are identified based on HPP, and incorporated into the algorithm as expanding entities rather than single orders. The evaluation stage at iteration k is executed as follows:

(1) For pending orders O^k, calculate the HPPs between the FUs of any two orders and denote the combination set as C^k. Set the order combination set preserved for MD evaluation at iteration k as $\widehat{C^k} = \emptyset$.
(2) Prune two-order combinations with low HPPs ($p_{o_1,o_2} < P_1$), i.e., let $C^k = C^k - C^k_{low}$.
(3) Repeat this step until $C^k = \emptyset$: pickup $c = \{o_1, o_2\} \in C^k$ with the highest HPP value, let $\widehat{C^k} = \widehat{C^k} + \{c\}$. Then remove its related entries in C^k and O^k, i.e, let $C^k = C^k - \{c|o_1 \in c \text{ or } o_2 \in c, c \in C^k\}$, $O^k = O^k - \{o_1\} - \{o_2\}$.
(4) Use $\widehat{C^k}$ and O^k as decision entities and calculate the MD scores with their associated couriers.

The above process is illustrated in Figure 8. And in practice, P_1 is set to 0.6.

F.2 Courier Recall Mechanism.

To reduce MD score calculation volume, we can further refine the courier candidates recalled for each order/order combination using HPP. For the evaluation stage at iteration k , the pending entity sets are $\widehat{C^k}$ and O^k, and the courier recall mechanism is executed as follows:

(1) For $o \in O^k$, denote the corresponding courier candidate set as R^k_o. For $r \in R^k_o$, if the on-hand order set $O^k_r \neq \emptyset$, calculate the average HPP of o and orders in O^k_r as an estimation of MD score, i.e. $\widetilde{f^r_o} = \frac{1}{|O^k_r|} \sum_{o' \in O^k_r} p_{o,o'}$.
For the on-hand order already picked up by courier r, its FU can be considered as the FU starting from the AOI where the courier is currently located and ending at its delivery AOI. For the on-hands whose FU embedding is absent, the associated HPP is set as 0.
If $\widetilde{f^r_o}$ is lower than threshold P_2, courier r will be removed from the candidate set, i.e. $R^k_o = R^k_o - \{r\}$.
(2) For $c = \{o_1, o_2\} \in \widehat{C^k}$, denote the corresponding courier candidate set as R^k_c, which is the intersection of courier candidate sets of o_1 and o_2. For $r \in R^k_c$, calculate the average HPP of o_1 and o_2 as Step (1), respectively.
If either $\widetilde{f^r_{o_1}}$ or $\widetilde{f^r_{o_2}}$ is lower than threshold P_2, courier r will be removed from the candidate set, i.e. $R^k_c = R^k_c - \{r\}$.
(3) For orders in O^k and combinations in $\widehat{C^k}$, calculate the MD scores with their refined couriers.

The above process is illustrated as in Figure 9. And in practice, P_2 is set to 0.5.

G SEH IDENTIFICATION APPROACH

We utilize BP to identify SEHs during each time interval from FUs with high FEI in a city or nearby areas. In this section, we introduce the variable definitions, objective function, and constraints of the model.

The decision variable x^g_f represents whether FU f belongs to SEH g. To calculate the average HPP in each SEH, we introduce a binary auxiliary variable $y^g_{f,f'}$, which indicates whether FU f and f' belong to SEH g simultaneously. The objective function in Equation (10) is to maximize the average HPP in each SEH, where $p_{f,f'}$ is the HPP between FU f and f'.

The constraint in Equation (11) limits each FU to appear in only one SEH. Equation (12) limits the minimum and maximum number of FUs in each SEH. Equation (13) limits the minimum number of orders in each SEH, where n_f is the number of orders of FU f. Equation (14) and Equation (15) ensure that $y^g_{f,f'} = 1$ if and only if $x^g_f = x^g_{f'} = 1$. Equation (16) constrains the minimum average HPP in each SEH g. Equation (17) and Equation (18) ensure that all the decision variables are binary.

$$\max \sum_{g \in G} \frac{\sum_{f \in F} \sum_{f' \in F, f' \neq f} p_{f,f'} \times y^g_{f,f'}}{\sum_{f \in F} \sum_{f' \in F, f' \neq f} y^g_{f,f'}} \tag{10}$$

$$s.t. \sum_{g \in G} x^g_f = 1, \forall f \in F \tag{11}$$

$$|g|^{\min} \leq \sum_{f \in F} x^g_f \leq |g|^{\max}, \forall g \in G \tag{12}$$

$$\sum_{f \in F} n_f \times x^g_f \geq N, \forall g \in G \tag{13}$$

$$y^g_{f,f'} \geq x^g_f + x^g_{f'} - 1 \tag{14}$$

$$y^g_{f,f'} \leq x^g_f, y^g_{f,f'} \leq x^g_{f'} \tag{15}$$

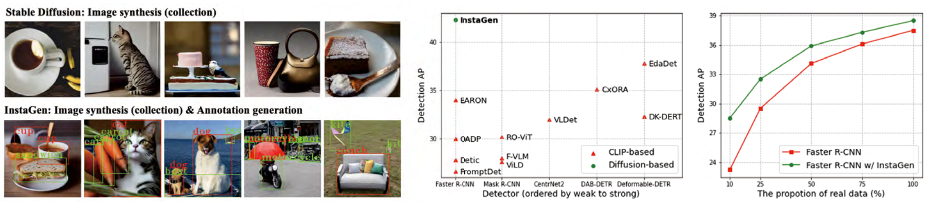
$$\sum_{f \in F} \sum_{f' \in F, f' \neq f} (p_{f,f'} - P) \times y^g_{f,f'} \geq 0, \forall g \in G \tag{16}$$

$$x^g_f \in \{0, 1\}, \forall f \in F, \forall g \in G \tag{17}$$

$$y^g_{f,f'} \in \{0, 1\}, \forall f, f' \in F, f \neq f', \forall g \in G \tag{18}$$

InstaGen: Enhancing Object Detection by Training on Synthetic Dataset

Chengjian Feng[1] Yujie Zhong[1] Zequn Jie[1,†] Weidi Xie[2,†] Lin Ma[1]
[1] Meituan Inc. [2] CMIC, Shanghai Jiao Tong University
fcjian@outlook.com jaszhong@hotmail.com zequn.nus@gmail.com
weidi@sjtu.edu.cn forest.linma@gmail.com
https://fcjian.github.io/InstaGen



(a) Demonstration of the synthetic images generated from Stable Diffusion and InstaGen with text prompt: 'a photograph of [base category] and [novel category]'.

(b) The detection AP of novel categories of different paradigms on COCO-OVD benchmarks.

(c) The detection AP with respect to the portion (%) of real data (COCO) usage.

Figure 1. (a) The synthetic images generated from **Stable Diffusion** and our proposed **InstaGen**, which can serve as a *dataset synthesizer* for sourcing photo-realistic images and instance bounding boxes at scale. (b) On open-vocabulary detection, training on synthetic images demonstrates significant improvement over CLIP-based methods on novel categories. (c) Training on the synthetic images generated from InstaGen also enhances the detection performance in close-set scenario, particularly in data-sparse circumstances.

Abstract

*In this paper, we present a novel paradigm to enhance the ability of object detector, e.g., expanding categories or improving detection performance, by training on **synthetic dataset** generated from diffusion models. Specifically, we integrate an instance-level grounding head into a pre-trained, generative diffusion model, to augment it with the ability of localising instances in the generated images. The grounding head is trained to align the text embedding of category names with the regional visual feature of the diffusion model, using supervision from an off-the-shelf object detector, and a novel self-training scheme on (novel) categories not covered by the detector. We conduct thorough experiments to show that, this enhanced version of diffusion model, termed as* **InstaGen**, *can serve as a data synthesizer, to enhance object detectors by training on its generated samples, demonstrating superior performance over existing state-of-the-art methods in open-vocabulary (+4.5 AP) and data-sparse (+1.2 ∼ 5.2 AP) scenarios.*

1. Introduction

Object detection has been extensively studied in the field of computer vision, focusing on the localization and categorization of objects within images [3, 5, 12, 26, 27]. The common practise is to train the detectors on large-scale image datasets, such as MS-COCO [20] and Object365 [30], where objects are exhaustively annotated with bounding boxes and corresponding category labels. However, the procedure for collecting images and annotations is often laborious and time-consuming, limiting the datasets' scalability.

In the recent literature, text-to-image diffusion models have demonstrated remarkable success in generating high-quality images [28, 29], that unlocks the possibility of training vision systems with synthetic images. In general, existing text-to-image diffusion models are capable of synthesizing images based on some free-form text prompt, as shown in the first row of Figure 1a. Despite being photo-realistic, such synthesized images *can not* support training sophisticated systems, that normally requires the inclusion of instance-level annotations, *e.g.*, bounding boxes for object detection in our case. In this paper, we investigate a novel paradigm of *dataset synthesis* for training object detector, *i.e.*, augmenting the text-to-image diffusion model to generate instance-level bounding boxes along with images.

To begin with, we build an image synthesizer by fine-tuning the diffusion model on existing detection dataset. This is driven by the observation that off-the-shelf diffusion models often generate images with only one or two objects on simplistic background, training detectors on such

†: corresponding author.

images may thus lead to reduced robustness in complex real-world scenarios. Specifically, we exploit the existing detection dataset, and subsequently fine-tune the diffusion model with the image-caption pairs, constructed by taking random image crops, and composing the category name of the objects in the crop. As illustrated in the second row of the Figure 1a, once finetuned, the image synthesizer now enables to produce images with multiple objects and intricate contexts, thereby providing a more accurate simulation of real-world detection scenarios.

To generate bounding boxes for objects within synthetic images, we propose an instance grounding module that establishes the correlation between the regional visual features from diffusion model and the text embedding of category names, and infers the coordinates for the objects' bounding boxes. Specifically, we adopt a two-step training strategies, *firstly*, we train the grounding module on synthetic images, with the supervision from an off-the-shelf object detector, which has been trained on a set of base categories; *secondly*, we utilize the trained grounding head to generate pseudo labels for a larger set of categories, including those not seen in existing detection dataset, and self-train the grounding module. Once finished training, the grounding module will be able to identify the objects of arbitrary category and their bounding boxes in the synthetic image, by simply providing the name in free-form language.

To summarize, we explore a novel approach to enhance object detection capabilities, such as expanding detectable categories and improving overall detection performance, by training on *synthetic dataset* generated from diffusion model. We make the following contribution: (i) We develop an image synthesizer by fine-tuning the diffusion model, with image-caption pairs derived from existing object detection datasets, our synthesizer can generate images with multiple objects and complex contexts, offering a more realistic simulation for real-world detection scenarios. (ii) We introduce a data synthesis framework for detection, termed as **InstaGen**. This is achieved through a novel grounding module that enables to generate labels and bounding boxes for objects in synthetic images. (iii) We train standard object detectors on the combination of *real and synthetic* dataset, and demonstrate superior performance over existing state-of-the-art detectors across various benchmarks, including open-vocabulary detection (increasing Average Precision [AP] by +4.5), data-sparse detection (enhancing AP by +1.2 to +5.2), and cross-dataset transfer (boosting AP by +0.5 to +1.1).

2. Related Work

Object Detection. Object detection aims to simultaneously predict the category and corresponding bounding box for the objects in the images. Generally, object detectors [3, 4, 6, 26, 27] are trained on a substantial amount of training data with bounding box annotations and can only recognize a predetermined set of categories present in the training data. In the recent literature, to further expand the ability of object detector, open-vocabulary object detection (OVD) has been widely researched, for example, OVR-CNN [37] introduces the concept of OVD and pre-trains a vision-language model with image-caption pairs. The subsequent works make use of the robust multi-modal representation of CLIP [24], and transfer its knowledge to object detectors through knowledge distillation [9, 36], exploiting extra data [5, 41] and text prompt tuning [2, 5]. In this paper, we propose to expand the ability of object detectors, *e.g.*, expanding categories or improving detection performance, by training on synthetic dataset.

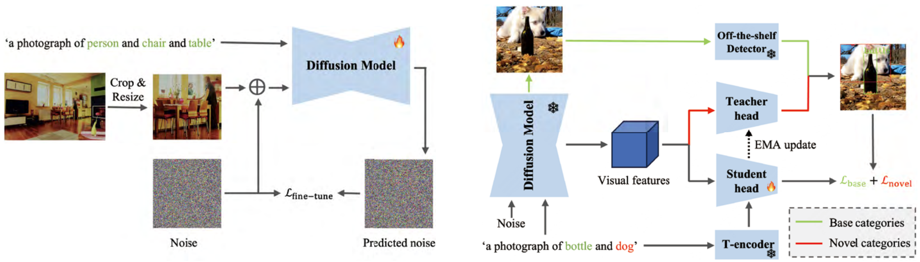
Generative Models. Image generation has been considered as a task of interest in computer vision for decades. In the recent literature, significant progress has been made, for example, the generative adversarial networks (GANs) [8], variational autoencoders (VAEs) [15], flow-based models [14], and autoregressive models (ARMs) [32]. More recently, there has been a growing research interest in diffusion probabilistic models (DPMs), which have shown great promise in generating high-quality images across diverse datasets. For examples, GLIDE [23] utilizes a pre-trained language model and a cascaded diffusion structure for text-to-image generation. DALL-E 2 [25] is trained to generate images by inverting the CLIP image space, while Imagen [29] explores the advantages of using pre-trained language models. Stable Diffusion [28] proposes the diffusion process in VAE latent spaces rather than pixel spaces, effectively reducing resource consumption. In general, the rapid development of generative models opens the possibility for training large models with synthetic dataset.

3. Methodology

In this section, we present details for constructing a *dataset synthesizer*, that enables to generate photo-realistic images with bounding boxes for each object instance, and train an object detector on the combined real and synthetic datasets.

3.1. Problem Formulation

Given a detection dataset of real images with manual annotations, *i.e.*, $\mathcal{D}_{\text{real}} = \{(x_1, \mathcal{B}_1, \mathcal{Y}_1), \dots, (x_N, \mathcal{B}_N, \mathcal{Y}_N)\}$, where $\mathcal{B}_i = \{b_1, \dots, b_m | b_j \in \mathbb{R}^{2\times 2}\}$ denotes the set of box coordinates for the annotated instances in one image, and $\mathcal{Y}_i = \{y_1, \dots, y_m | y_j \in \mathcal{R}^{\mathcal{C}_{\text{base}}}\}$ refers to the categories of the instances. Our goal is thus to exploit the given real dataset ($\mathcal{D}_{\text{real}}$), to steer a generative diffusion model into *dataset synthesizer*, that enables to augment the existing detection dataset, *i.e.*, $\mathcal{D}_{\text{final}} = \mathcal{D}_{\text{real}} + \mathcal{D}_{\text{syn}}$. As a result, detectors trained on the combined dataset demonstrate enhanced ability, *i.e.*, extending the detection categories or improving the detection performance.



(a) Fine-tuning diffusion model on detection dataset.

(b) Supervised training and self-training for grounding head (*i.e.* student).

Figure 2. Illustration of the process for finetuning diffusion model and training the grounding head: (a) stable diffusion model is fine-tuned on the detection dataset on base categories. (b) The grounding head is trained on synthetic images, with supervised learning on base categories and self-training on novel categories.

In the following sections, we first describe the procedure for constructing an *image synthesizer*, that can generate images suitable for training object detector (Section 3.2). To simultaneously generate the images and object bounding boxes, we propose a novel instance-level grounding module, which aligns the text embedding of category name with the regional visual features from *image synthesizer*, and infers the coordinates for the objects in synthetic images. To further improve the alignment towards objects of arbitrary category, we adopt self-training to tune the grounding module on object categories not existing in $\mathcal{D}_{\text{real}}$ (Section 3.3). As a result, the proposed model, termed as **InstaGen**, can automatically generate images along with bounding boxes for object instances, and construct *synthetic dataset* ($\mathcal{D}_{\text{syn}}$) at scale, leading to improved ability when training detectors on it (Section 3.4).

3.2. Image Synthesizer for Object Detection

Here, we build our *image synthesizer* based on an off-the-shelf stable diffusion model (SDM [28]). Despite of its impressive ability in generating photo-realistic images, it often outputs images with only one or two objects on simplistic background with the text prompts, for example, 'a photograph of a [category1 name] and a [category2 name]', as demonstrated in Figure 4b. As a result, object detectors trained on such images may exhibit reduced robustness when dealing with complex real-world scenarios. To bridge such domain gap, we propose to construct the *image synthesizer* by fine-tuning the SDM with an existing real-world detection dataset ($\mathcal{D}_{\text{real}}$).

Fine-tuning procedure. To fine-tune the stable diffusion model (SDM), one approach is to naïvely use the sample from detection dataset, for example, randomly pick an image and construct the text prompt with all categories in the image. However, as the image often contains multiple objects, such approach renders significant difficulty for fine-tuning the SDM, especially for small or occluded objects. We adopt a mild strategy by taking random crops from the images, and construct the text prompt with categories in the image crops, as shown in Figure 2a. If an image crop contains multiple objects of the same category, we only use this category name once in the text prompt.

Fine-tuning loss. We use the sampled image crop and constructed text prompt to fine-tune SDM with a squared error loss on the predicted noise term as follows:

$$\mathcal{L}_{\text{fine-tune}} = \mathbb{E}_{z,\epsilon\sim\mathcal{N}(0,1),t,y}\left[||\epsilon - \epsilon_\theta(z^t, t, y)||_2^2\right], \quad (1)$$

where z denotes a latent vector mapped from the input image with VAE, t denotes the denoising step, uniformly sampled from $\{1, \ldots, T\}$, T refers to the length of the diffusion Markov chain, and ϵ_θ refers to the estimated noise from SDM with parameters θ being updated. We have experimentally verified the necessity of this fine-tuning step, as shown in Table 4.

3.3. Dataset Synthesizer for Object Detection

In this section, we present details for steering the *image synthesizer* into *dataset synthesizer* for object detection, which enables to simultaneously generate images and object bounding boxes. Specifically, we propose an instance-level grounding module that aligns the text embedding of object category, with the regional visual feature of the diffusion model, and infers the coordinates for bounding boxes, effectively augmenting the *image synthesizer* with instance grounding, as shown in Figure 3. To further improve the alignment in large visual diversity, we propose a self-training scheme that enables the grounding module to generalise towards arbitrary categories, including those not exist in real detection dataset ($\mathcal{D}_{\text{real}}$). As a result, our *data synthesizer*, termed as **InstaGen**, can be used to construct synthetic dataset for training object detectors.

3.3.1 Instance Grounding on Base Categories

To localise the object instances in synthetic images, we introduce an open-vocabulary grounding module, that aims

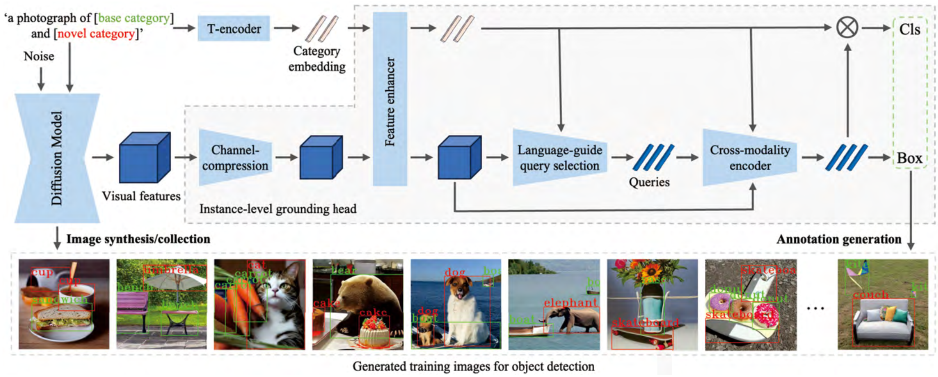


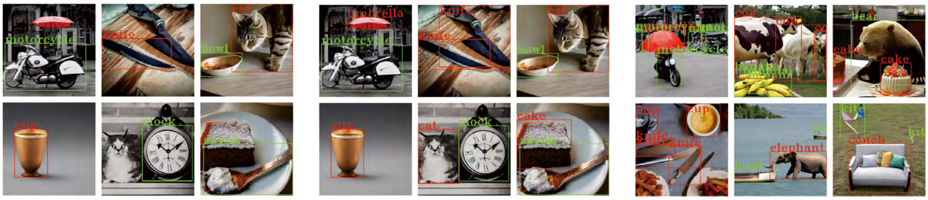
Figure 3. Illustration of the dataset generation process in InstaGen. The data generation process consists of two steps: (i) **Image collection**: given a text prompt, SDM generates images with the objects described in the text prompt; (ii) **Annotation generation**: the instance-level grounding head aligns the category embedding with the visual feature region of SDM, generating the corresponding object bounding-boxes.

to simultaneously generate image (x) and the corresponding instance-level bounding boxes ($\mathcal{B}$) based on a set of categories ($\mathcal{Y}$), *i.e.*, $\{x, \mathcal{B}, \mathcal{Y}\} = \Phi_{\text{InstaGen}}(\epsilon, \mathcal{Y})$, where $\epsilon \sim \mathcal{N}(0, I)$ denotes the sampled noise.

To this end, we propose an instance grounding head, as shown in Figure 3, it takes the intermediate representation from *image synthesizer* and the text embedding of category as inputs, then predicts the corresponding object bounding boxes, *i.e.*, $\{\mathcal{B}_i, \mathcal{Y}_i\} = \Phi_{\text{g-head}}(\mathcal{F}_i, \Phi_{\text{t-enc}}(g(\mathcal{Y}_i)))$, where $\mathcal{F}_i = \{f_i^1, \ldots, f_i^n\}$ refers to the multi-scale dense features from the *image synthesizer* at time step $t = 1$, $g(\cdot)$ denotes a template that decorates each of the visual categories in the text prompt, *e.g.*, 'a photograph of [category1 name] and [category2 name]', $\Phi_{\text{t-enc}}(\cdot)$ denotes the text encoder.

Inspired by GroundingDINO [22], our grounding head $\Phi_{\text{g-head}}(\cdot)$ mainly contains four components: (i) a channel-compression layer, implemented with a 3×3 convolution, for reducing the dimensionality of the visual features; (ii) a feature enhancer, consisting of six feature enhancer layers, to fuse the visual and text features. Each layer employs a deformable self-attention to enhance image features, a vanilla self-attention for text feature enhancers, an image-to-text cross-attention and a text-to-image cross-attention for feature fusion; (iii) a language-guided query selection module for query initialization. This module predicts top-N anchor boxes based on the similarity between text features and image features. Following DINO [38], it adopts a mixed query selection where the positional queries are initialized with the anchor boxes and the content queries remain learnable; (iv) a cross-modality decoder for classification and box refinement. It comprises six decoder layers, with each layer utilizing a self-attention mechanism for query interaction, an image cross-attention layer for combining image features, and a text cross-attention layer for combining text features. Finally, we apply the dot product between each query and the text features, followed by a Sigmoid function to predict the classification score $\hat{s}$ for each category. Additionally, the object queries are passed through a Multi-Layer Perceptron (MLP) to predict the object bounding boxes $\hat{b}$, as shown in Figure 3. We train the grounding head by aligning the category embedding with the regional visual features from diffusion model, as detailed below. *Once trained, the grounding head is open-vocabulary*, *i.e.*, given any categories (even beyond the training categories), the grounding head can generate the corresponding bounding-boxes for the object instances.

Training triplets of base categories. Following [18], we apply an automatic pipeline to construct the {visual feature, bounding-box, text prompt} triplets, with an object detector trained on base categories from a given dataset ($\mathcal{D}_{\text{real}}$). In specific, assuming there exists a set of base categories $\{c^1_{\text{base}}, \ldots, c^N_{\text{base}}\}$, *e.g.*, the classes in MS-COCO [20]. We first select a random number of base categories to construct a text prompt, *e.g.*, 'a photograph of [base category1] and [base category2]', and generate both the visual features and images with our *image synthesizer*. Then we take an off-the-shelf object detector, for example, pre-trained Mask R-CNN [12], to run the inference procedure on the synthetic images, and infer the bounding boxes of the selected categories. To acquire the confident bounding-boxes for training, we use a score threshold α to filter out the bounding-boxes with low confidence (an ablation study on the selection of the score threshold has been conducted in Section 4.5). As a result, an infinite number of training triplets



(a) Stable Diffusion + Grounding head w/ **Supervised training**.

(b) Stable Diffusion + Grounding head w/ Supervised- and **Self-training**.

(c) Stable Diffusion w/ **Fine-tuning** + Grounding head w/ Supervised- and Self-training.

Figure 4. Visualization of the synthetic images and bounding-boxes generated from different models. The bounding-boxes with green denote the objects from base categories, while the ones with red denote the objects from novel categories.

for the given base categories can be constructed by repeating the above operation.

Training loss. We use the constructed training triplets to train the grounding head:

$$\mathcal{L}_{\text{base}} = \sum_{i=1}^{N} [\mathcal{L}_{\text{cls}}(\hat{s}_i, c_i) + \mathbb{1}_{\{c_i \neq \varnothing\}} \mathcal{L}_{\text{box}}(\hat{b}_i, b_i)], \quad (2)$$

where the ith prediction $(\hat{s}_i, \hat{b}_i)$ from the N object queries is assigned to a ground-truth (c_i, b_i) or $\varnothing$ (no object) with bipartite matching. $\mathcal{L}_{\text{cls}}$ and $\mathcal{L}_{\text{box}}$ denote the classification loss (*e.g.* Focal loss) and box regression loss (*e.g.* L1 loss and GIoU loss), respectively.

3.3.2 Instance Grounding on Novel Categories

Till here, we have obtained a diffusion model with open-vocabulary grounding, which has been only trained with base categories. In this section, we propose to further leverage the synthetic training triplets from a wider range of categories to enhance the alignment for novel/unseen categories. Specifically, as shown in Figure 2b, we describe a framework that generates the training triplets for novel categories using the grounded diffusion model, and then self-train the grounding head.

Training triplets of novel categories. We design the text prompts of novel categories, *e.g.*, 'a photograph of [novel category1] and [novel category2]', and pass them through our proposed *image synthesizer*, to generate the visual features. To acquire the corresponding bounding-boxes for novel categories, we propose a self-training scheme that takes the above grounding head as the student, and apply a mean teacher (an exponential moving average (EMA) of the student model) to create pseudo labels for update. In contrast to the widely adopted self-training scheme that takes the image as input, the student and teacher in our case only take the visual features as input, thus *cannot* apply data augmentation as for images. Instead, we insert dropout module within each feature enhancer layer and decoder layer in the student. During training, we run inference (without dropout module) with teacher model on the visual features to produce bounding boxes, and then use a score threshold β to filter out those with low confidence, and use the remaining training triplets $(\mathcal{F}_i, \hat{b}_i, y_i^{\text{novel}})$ to train the student, *i.e.*, grounding head.

Training loss. Now, we can also train the grounding head on the mined triplets of novel categories (that are unseen in the existing real dataset) with the training loss $\mathcal{L}_{\text{novel}}$ defined similar to Eq. 2. Thus, the total training loss for training the grounding head can be: $\mathcal{L}_{\text{grounding}} = \mathcal{L}_{\text{base}} + \mathcal{L}_{\text{novel}}$.

3.4. Training Detector with Synthetic Dataset

In this section, we augment the real dataset ($\mathcal{D}_{\text{real}}$), with synthetic dataset ($\mathcal{D}_{\text{syn}}$), and train popular object detectors, for example, Faster R-CNN [27] with the standard training loss:

$$\mathcal{L}_{\text{det}} = \mathcal{L}_{\text{rpn\_cls}} + \mathcal{L}_{\text{rpn\_box}} + \mathcal{L}_{\text{det\_cls}} + \mathcal{L}_{\text{det\_box}}, \quad (3)$$

where $\mathcal{L}_{\text{rpn\_cls}}$, $\mathcal{L}_{\text{rpn\_box}}$ are the classification and box regression losses of region proposal network, and $\mathcal{L}_{\text{det\_cls}}$, $\mathcal{L}_{\text{det\_box}}$ are the classification and box regression losses of the detection head. Generally speaking, the synthetic dataset enables to improve the detector's ability from two aspects: (i) expanding the original data with more categories, (ii) improve the detection performance by increasing data diversity.

Expanding detection categories. The grounding head is designed to be open-vocabulary, that enables to generate object bounding boxes for novel categories, even though it is trained with a specific set of base categories. This feature enables **InstaGen** to construct a detection dataset for any category. Figure 4 demonstrates several synthetic images and object bounding boxes for novel categories, *i.e.*, the object with red bounding box. We evaluate the effectiveness of training on synthetic dataset through experiments on open-vocabulary detection benchmark. For more details, please refer to Figure 1b and Section 4.2.

Increasing data diversity. The base diffusion model is trained on a large corpus of image-caption pairs, that enables to generate diverse images. Taking advantage of such

Method	Supervision	Detector	Backbone	$AP50^{box}_{all}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$
Detic [41]	CLIP	Faster R-CNN	R50	45.0	47.1	27.8
PromptDet [5]	CLIP	Faster R-CNN	R50	-	50.6	26.6
BARON [34]	CLIP	Faster R-CNN	R50	53.5	60.4	34.0
OADP [33]	CLIP	Faster R-CNN	R50	47.2	53.3	30.0
ViLD [9]	CLIP	Mask R-CNN	R50	51.3	59.5	27.6
F-VLM [16]	CLIP	Mask R-CNN	R50	39.6	-	28.0
RO-ViT [13]	CLIP	Mask R-CNN	ViT-B [1]	41.5	-	30.2
VLDet [19]	CLIP	CenterNet2 [40]	R50	45.8	50.6	32.0
CxORA [35]	CLIP	DAB-DETR [21]	R50	35.4	35.5	35.1
DK-DETR [17]	CLIP	Deformable DETR [42]	R50	-	61.1	32.3
EdaDet [31]	CLIP	Deformable DETR [42]	R50	52.5	57.7	37.8
InstaGen	Stable Diffusion	Faster R-CNN	R50	52.3	55.8	**42.3**

Table 1. Results on open-vocabulary COCO benchmark. $AP50^{box}_{novel}$ is the main metric for evaluation. Our detector, trained on synthetic dataset from **InstaGen**, significantly outperforms state-of-the-art CLIP-based approaches on novel categories.

capabilities, **InstaGen** is capable of generating dataset with diverse images and box annotations, which can expand the original dataset, *i.e.*, increase the data diversity and improve detection performance, particularly in data-sparse scenarios. We conducted experiments with varying proportions of COCO [20] images as available real data, and show the effectiveness of training on synthetic dataset when the number of real-world images is limited. We refer the readers for more details in Section 4.3, and results in Figure 1c.

4. Experiment

In this section, we use the proposed **InstaGen** to construct synthetic dataset for training object detectors, *i.e.*, generating images with the corresponding bounding boxes. Specifically, we present the implementation details in Section 4.1. To evaluate the effectiveness of the synthetic dataset for training object detector, we consider three protocols: open-vocabulary object detection (Section 4.2), data-sparse object detection (Section 4.3) and cross-dataset object detection (Section 4.4). Lastly, we conduct ablation studies on the effectiveness of the proposed components and the selection of hyper-parameters (Section 4.5).

4.1. Implementation details

Network architecture. We build *image synthesizer* from the pre-trained Stable Diffusion v1.4 [28], and use the CLIP text encoder [24] to get text embedding for the category name. The channel compression layer maps the dimension of visual features to 256, which is implemented with a 3×3 convolution. For simplicity, the feature enhancer, language-guided query selection module and cross-modality decoder are designed to the same structure as the ones in [22]. The number of the object queries is set to 900.

Constructing image synthesizer. In our experiments, we first fine-tune the stable diffusion model on a real detection dataset, *e.g.*, the images of base categories. During training, the text encoder of CLIP is kept frozen, while the remaining components are trained for 6 epochs with a batch size of 16 and a learning rate of 1e-4.

Instance grounding module. We start by constructing the training triplets using base categories *i.e.*, the categories present in the existing dataset. The text prompt for each triplet is constructed by randomly selecting one or two categories. The regional visual features are taken from the *image synthesizer* time step $t = 1$, and the oracle ground-truth bounding boxes are obtained using a Mask R-CNN model trained on base categories, as explained in Section 3.3.1.

Subsequently, we train the instance grounding module with these training triplets for 6 epochs, with a batch size of 32. In the 6th epoch, we transfer the weights from the student model to the teacher model, and proceed to train the student for an additional 6 epochs. During this training, the student receives supervised training on the base categories and engages in self-training on novel categories, and the teacher model is updated using exponential moving average (EMA) with a momentum of 0.999. The initial learning rate is set to 1e-4 and is subsequently reduced by a factor of 10 at the 11th epoch, and the score thresholds α and β are set to 0.8 and 0.4, respectively.

Training object detector on combined dataset. In our experiment, we train an object detector (Faster R-CNN [27]) with ResNet-50 [11] as backbone, on a combination of the existing real dataset and the synthetic dataset. Specifically, for synthetic dataset, we randomly select one or two categories at each iteration, construct the text prompts, and feed them as input to generates images along with the corresponding bounding boxes with β of 0.4. Following the standard implementation [27], the detector is trained for 12 epochs ($1\times$ learning schedule) unless specified. The initial learning rate is set to 0.01 and then reduced by a factor of 10 at the 8th and the 11th epochs.

InstaGen	10%	25%	50%	75%	100%
✗	23.3	29.5	34.1	36.1	37.5
✓	28.5	32.6	35.8	37.3	38.5

Table 2. Results on data-sparse object detection. We employ Faster R-CNN with the ResNet-50 backbone as the default object detector and evaluate its performance using the AP metric on MS COCO benchmark. Please refer to the text for more details.

Method	Supervision	Detector	Extra Data	Object365	LVIS
Gao *et al.* [7]	CLIP	CenterNet2	✓	6.9	8.0
VL-PLM [39]	CLIP	Mask R-CNN	✓	10.9	22.2
InstaGen	Stable Diffusion	Faster R-CNN	✗	**11.4**	**23.3**

Table 3. Results on generalizing COCO-base to Object365 and LVIS. All detectors utilize the ResNet-50 backbone. The evaluation protocol follows [7] and reports AP50. Extra data refers to an additional dataset that encompasses objects from the categories within the target dataset. In both experiments, the extra data consists of all the images from COCO, which has covered the majority of categories in Object365 and LVIS.

G-head	ST	FT	$AP50^{box}_{all}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$
✓			50.6	55.3	37.1
✓	✓		51.1	55.0	40.3
✓	✓	✓	52.3	55.8	42.3

Table 4. The effectiveness of the proposed components. G-head, ST and FT refer to the grounding head, self-training the grounding head and fine-tuning SDM, respectively.

4.2. Open-vocabulary object detection

Experimental setup. Following the previous works [5, 39], we conduct experiments on the open-vocabulary COCO benchmark, where 48 classes are treated as base categories, and 17 classes as the novel categories. More results for LVIS can be found in the **supplementary material**. To train the grounding head, we employ 1250 synthetic images per category per training epoch. While for training the object detector, we use 3000 synthetic images per category, along with the original real dataset for base categories. The object detector is trained with input size of 800×800 and scale jitter. The performance is measured by COCO Average Precision at an Intersection over Union of 0.5 (AP50).

Comparison to SOTA. As shown in Table 1, we evaluate the performance by comparing with existing CLIP-based open-vocabulary object detectors. It is clear that our detector trained on synthetic dataset from **InstaGen** outperforms existing state-of-the-art approaches significantly, *i.e.*, around +5AP improvement over the second best. In essence, through the utilization of our proposed open-vocabulary grounding head, **InstaGen** is able to generate detection data for novel categories, enabling the detector to attain exceptional performance. To the best of our knowledge, this is the first work that applies generative diffusion model for dataset synthesis, to tackle open-vocabulary object detection, and showcase its superiority in this task.

4.3. Data-sparse object detection

Experimental setup. Here, we evaluate the effectiveness of synthetic dataset in data-spare scenario, by varying the amount of real data. We randomly select subsets comprising 10%, 25%, 50%, 75% and 100% of the COCO training set, this covers all COCO categories. These subsets are used to fine-tune stable diffusion model for constructing *image synthesizer*, and train a Mask R-CNN for generating oracle ground-truth bounding boxes in synthetic images. We employ 1250 synthetic images per category to train a Faster R-CNN in conjunction with the corresponding COCO subset. The performance is measured by Average Precision [20].

Comparison to baseline. As shown in Table 2, the Faster R-CNN trained with synthetic images achieves consistent improvement across various real training data budgets. Notably, as the availability of real data becomes sparse, synthetic dataset plays even more important role for performance improvement, for instance, it improves the detector by +5.2 AP (23.3→28.5 AP) when only 10% real COCO training subset is available.

4.4. Cross-dataset object detection

Experimental setup. In this section, we assess the effectiveness of synthetic data on a more challenging task, namely cross-dataset object detection. Following [39], we evaluate the COCO-trained model on two unseen datasets: Object365 [30] and LVIS [10]. Specifically, we consider the 48 classes in the open-vocabulary COCO benchmark as the source dataset, while Object365 (with 365 classes) and LVIS (with 1203 classes) serve as the target dataset. When training the instance grounding module, we acquire 1250 synthetic images for base categories from the source dataset, and 100 synthetic images for the category from the target dataset at each training iteration. In the case of training the object detector, we employ 500 synthetic images per category from the target dataset for each training iteration. The detector is trained with input size of 1024×1024 and scale jitter [39].

Comparison to SOTA. The results presented in Table 3 demonstrate that the proposed **InstaGen** achieves superior performance in generalization from COCO-base to Object365 and LVIS, when compared to CLIP-based methods such as [7, 39]. It is worth noting that CLIP-based methods require the generation of pseudo-labels for the categories from the target dataset on COCO images, and subsequently train the detector using these images. These methods necessitate a dataset that includes objects belonging to the categories of the target dataset. In contrast, **InstaGen** possesses the ability to generate images featuring objects of any cat-

#Images	$AP50^{box}_{all}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$
1000	51.6	55.9	39.7
2000	51.7	55.4	41.1
3000	52.3	55.8	42.3

Table 5. Number of generated images.

α	$AP50^{box}_{all}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$
0.7	51.3	55.1	40.6
0.8	52.3	55.8	42.3
0.9	51.8	55.6	41.1

Table 6. α for bounding-box filtration.

β	$AP50^{box}_{all}$	$AP50^{box}_{base}$	$AP50^{box}_{novel}$
0.3	46.4	53.3	26.9
0.4	52.3	55.8	42.3
0.5	51.2	55.4	39.2

Table 7. β for bounding-box filtration.

egory without the need for additional datasets, thereby enhancing its versatility across various scenarios.

4.5. Ablation study

To understand the effectiveness of the proposed components, we perform thorough ablation studies on the open-vocabulary COCO benchmark [20], investigating the effect of fine-tuning stable diffusion model, training instance grounding module, self-training on novel categories. Additionally, we investigate other hyper-parameters by comparing the effectiveness of synthetic images and different score thresholds for base and novel categories.

Fine-tuning diffusion model. We assess the effectiveness of fine-tuning stable diffusion model, and its impact for synthesizing images for training object detector. Figure 4c illustrates that **InstaGen** is capable of generating images with more intricate contexts, featuring multiple objects, small objects, and occluded objects. Subsequently, we employed these generated images to train Faster R-CNN for object detection. The results are presented in Table 4, showing that *image synthesizer* from fine-tuning stable diffusion model delivers improvement detection performance by 2.0 AP (from 40.3 to 42.3 AP).

Instance grounding module. To demonstrate the effectiveness of the grounding head in open-vocabulary scenario, we exclusively train it on base categories. Visualization examples of the generated images are presented in Figure 4a. These examples demonstrate that the trained grounding head is also capable of predicting bounding boxes for instances from novel categories. Leveraging these generated images to train the object detector leads to a 37.1 AP on novel categories, surpassing or rivaling all existing state-of-the-art methods, as shown in Table 1 and Table 4.

Self-training scheme. We evaluate the performance after self-training the grounding head with novel categories. As shown in Table 4, training Faster R-CNN with the generated images of novel categories, leads to a noticeable enhancement in detection performance, increasing from 37.1 to 40.3 AP. Qualitatively, it also demonstrates enhanced recall for novel objects after self-training, as shown in Figure 4b.

Number of synthetic images. We investigate the performance variation while increasing the number of the generated images per category for detector training. As shown in Table 5, when increasing the number of generated images from 1000 to 3000, the detector's performance tends to be increasing monotonically, from 39.7 to 42.3 AP on novel categories, showing the scalability of the proposed training mechanism.

Score thresholds for bounding box filtration. We compare the performance with different score thresholds α and β for filtering bounding boxes on base categories and novel categories, respectively. From the experiment results in Table 6, we observe that the performance is not sensitive to the value of α, and $\alpha = 0.8$ yields the best performance. The experimental results using different β are presented in Table 7. With a low score threshold ($\alpha = 0.3$), there are still numerous inaccurate bounding boxes remaining, resulting in an AP of 26.9 for novel categories. by increasing β to 0.4, numerous inaccurate bounding boxes are filtered out, resulting in optimal performance. Hence, we set $\alpha = 0.8$ and $\beta = 0.4$ in our experiments.

5. Limitation

Using synthetic or artificially generated data in training AI algorithms is a burgeoning practice with significant potential. It can address data scarcity, privacy, and bias issues. However, there remains two limitations for training object detectors with synthetic data, (i) synthetic datasets commonly focus on clean, isolated object instances, which limits the exposure of the detector to the complexities and contextual diversity of real-world scenes, such as occlusions, clutter, varied environmental factors, deformation, therefore, models trained on synthetic data struggle to adapt to real-world conditions, affecting their overall robustness and accuracy, (ii) existing diffusion-based generative model also suffers from long-tail issue, that means the generative model struggles to generate images for objects of rare categories, resulting in imbalanced class representation during training and reduced detector performance for less common objects.

6. Conclusion

This paper proposes a dataset synthesis pipeline, termed as **InstaGen**, that enables to generate images with object bounding boxes for arbitrary categories, acting as a annotation-free approach for constructing large-scale synthetic dataset to train object detector. We have conducted thorough experiments to show the effectiveness of training on synthetic data, on improving detection performance, or expanding the number of detection categories. Significant improvements have been shown in various detection scenarios, including open-vocabulary (+4.5 AP) and data-sparse (+1.2 $\sim$ 5.2 AP) detection.

References

[1] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 6

[2] Yu Du, Fangyun Wei, Zihe Zhang, Miaojing Shi, Yue Gao, and Guoqi Li. Learning to prompt for open-vocabulary object detection with vision-language model. In *CVPR*, pages 14084–14093, 2022. 2

[3] Chengjian Feng, Yujie Zhong, Yu Gao, Matthew R Scott, and Weilin Huang. Tood: Task-aligned one-stage object detection. In *ICCV*, pages 3490–3499. IEEE Computer Society, 2021. 1, 2

[4] Chengjian Feng, Yujie Zhong, and Weilin Huang. Exploring classification equilibrium in long-tailed object detection. In *ICCV*, pages 3417–3426, 2021. 2

[5] Chengjian Feng, Yujie Zhong, Zequn Jie, Xiangxiang Chu, Haibing Ren, Xiaolin Wei, Weidi Xie, and Lin Ma. Promptdet: Towards open-vocabulary detection using uncurated images. In *ECCV*, pages 701–717. Springer, 2022. 1, 2, 6, 7

[6] Chengjian Feng, Zequn Jie, Yujie Zhong, Xiangxiang Chu, and Lin Ma. Aedet: Azimuth-invariant multi-view 3d object detection. In *CVPR*, pages 21580–21588, 2023. 2

[7] Mingfei Gao, Chen Xing, Juan Carlos Niebles, Junnan Li, Ran Xu, Wenhao Liu, and Caiming Xiong. Open vocabulary object detection with pseudo bounding-box labels. In *ECCV*, pages 266–282. Springer, 2022. 7

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. 2

[9] Xiuye Gu, Tsung-Yi Lin, Weicheng Kuo, and Yin Cui. Open-vocabulary object detection via vision and language knowledge distillation. *arXiv preprint arXiv:2104.13921*, 2021. 2, 6, 1

[10] Agrim Gupta, Piotr Dollar, and Ross Girshick. Lvis: A dataset for large vocabulary instance segmentation. In *CVPR*, pages 5356–5364, 2019. 7, 1

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[12] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017. 1, 4

[13] Dahun Kim, Anelia Angelova, and Weicheng Kuo. Region-aware pretraining for open-vocabulary object detection with vision transformers. In *CVPR*, pages 11144–11154, 2023. 6

[14] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *NeurIPS*, 31, 2018. 2

[15] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. 2

[16] Weicheng Kuo, Yin Cui, Xiuye Gu, AJ Piergiovanni, and Anelia Angelova. F-vlm: Open-vocabulary object detection upon frozen vision and language models. 2022. 6, 2

[17] Liangqi Li, Jiaxu Miao, Dahu Shi, Wenming Tan, Ye Ren, Yi Yang, and Shiliang Pu. Distilling detr with visual-linguistic knowledge for open-vocabulary object detection. In *ICCV*, pages 6501–6510, 2023. 6

[18] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Open-vocabulary object segmentation with diffusion models. In *CVPR*, pages 7667–7676, 2023. 4

[19] Chuang Lin, Peize Sun, Yi Jiang, Ping Luo, Lizhen Qu, Gholamreza Haffari, Zehuan Yuan, and Jianfei Cai. Learning object-language alignments for open-vocabulary object detection. 2022. 6

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *ECCV*, pages 740–755. Springer, 2014. 1, 4, 6, 7, 8

[21] Shilong Liu, Feng Li, Hao Zhang, Xiao Yang, Xianbiao Qi, Hang Su, Jun Zhu, and Lei Zhang. Dab-detr: Dynamic anchor boxes are better queries for detr. *arXiv preprint arXiv:2201.12329*, 2022. 6

[22] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 4, 6

[23] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2112.10741*, 2021. 2

[24] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. pages 8748–8763. PMLR, 2021. 2, 6

[25] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 1 (2):3, 2022. 2

[26] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1, 2

[27] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015. 1, 2, 5, 6

[28] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *CVPR*, pages 10684–10695, 2022. 1, 2, 3, 6

[29] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. *NeurIPS*, 35:36479–36494, 2022. 1, 2

[30] Shuai Shao, Zeming Li, Tianyuan Zhang, Chao Peng, Gang Yu, Xiangyu Zhang, Jing Li, and Jian Sun. Objects365:

A large-scale, high-quality dataset for object detection. In *ICCV*, pages 8430–8439, 2019. 1, 7

[31] Cheng Shi and Sibei Yang. Edadet: Open-vocabulary object detection using early dense alignment. In *ICCV*, pages 15724–15734, 2023. 6

[32] Aaron Van den Oord, Nal Kalchbrenner, Lasse Espeholt, Oriol Vinyals, Alex Graves, et al. Conditional image generation with pixelcnn decoders. *NeurIPS*, 29, 2016. 2

[33] Luting Wang, Yi Liu, Penghui Du, Zihan Ding, Yue Liao, Qiaosong Qi, Biaolong Chen, and Si Liu. Object-aware distillation pyramid for open-vocabulary object detection. In *CVPR*, pages 11186–11196, 2023. 6

[34] Size Wu, Wenwei Zhang, Sheng Jin, Wentao Liu, and Chen Change Loy. Aligning bag of regions for open-vocabulary object detection. In *CVPR*, pages 15254–15264, 2023. 6, 2

[35] Xiaoshi Wu, Feng Zhu, Rui Zhao, and Hongsheng Li. Cora: Adapting clip for open-vocabulary detection with region prompting and anchor pre-matching. In *CVPR*, pages 7031–7040, 2023. 6

[36] Johnathan Xie and Shuai Zheng. Zsd-yolo: Zero-shot yolo detection using vision-language knowledge distillation. *arXiv preprint arXiv:2109.12066*, 2(3):4, 2021. 2

[37] Alireza Zareian, Kevin Dela Rosa, Derek Hao Hu, and Shih-Fu Chang. Open-vocabulary object detection using captions. In *CVPR*, pages 14393–14402, 2021. 2

[38] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 4

[39] Shiyu Zhao, Zhixing Zhang, Samuel Schulter, Long Zhao, BG Vijay Kumar, Anastasis Stathopoulos, Manmohan Chandraker, and Dimitris N Metaxas. Exploiting unlabeled data with vision and language models for object detection. In *ECCV*, pages 159–175. Springer, 2022. 7

[40] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Probabilistic two-stage detection. *arXiv preprint arXiv:2103.07461*, 2021. 6

[41] Xingyi Zhou, Rohit Girdhar, Armand Joulin, Philipp Krähenbühl, and Ishan Misra. Detecting twenty-thousand classes using image-level supervision. In *ECCV*, pages 350–368. Springer, 2022. 2, 6, 1

[42] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. *arXiv preprint arXiv:2010.04159*, 2020. 6

InstaGen: Enhancing Object Detection by Training on Synthetic Dataset
Supplementary Material

In this supplementary document, we present the experimental results of the LVIS-OVD benchmark in Section S1. Additionally, we perform an ablation study to evaluate the coupling between the diffusion model and the grounding head in Section S2. Furthermore, we evaluate the quality of the pseudo-labels generated by the grounding head in Section S3. Lastly, we provide more qualitative results in Section S4.

S1. Open-vocabulary setting on LVIS

Experimental setup. We conduct experiments on the LVIS-OVD benchmark. The latest LVIS v1.0 [10] consists of 1203 categories, each with bounding box and instance mask annotations. The categories are divided into three groups based on the number of images in which each category appears in the training set: rare (1-10 images), common (11-100 images), and frequent (more than 100 images). In line with the problem setting in ViLD [9] and Detic [41], we treat the frequent and common classes as base categories, while considering the rare classes as novel categories. For evaluation on LVIS v1.0 *minival* set, we mainly consider the mask Average Precision for novel categories, *i.e.* AP_{novel}. However, to complete the AP metric, we also report AP_{c} (for common classes), AP_{f} (for frequent classes) and AP (for all classes).

Similar to PromptDet [5], we enhance the prompt template by incorporating a more detailed description to mitigate lexical ambiguity, particularly for the rare classes in LVIS. It should be noted that the description can be easily extracted from the metadata of the dataset. Consequently, the text prompt for the selected categories is generated as follows: 'a photograph of [category1 name] ([category1 description]) and [category2 name] ([category2 description])'. During the training of the grounding head, we utilize 500 synthetic images per category per training epoch. In addition, for the training of the object detector, we employ 250 synthetic images per category per training epoch and conduct 24 epochs of training.

Comparison to SOTA. We conduct a comparison with the existing CLIP-based open-vocabulary object detectors using the Mask-RCNN model with ResNet-50, as shown in Table S2. The results indicate that our detector, trained on synthetic dataset from **InstaGen**, achieves comparable or improved performance over existing CLIP-based methods.

S2. Tight coupling *vs*. Loose coupling

To generate high-quality bounding-boxes for the synthetic images, we have designed a tight coupling between the diffusion model and the instance-level grounding head, namely, the grounding head predicts the bounding-boxes based on the SDM's internal representation. To demonstrate the effectiveness of the tight coupling design, we compare it with a loose coupling design. For the latter, we train an open-vocabulary detector (*i.e.* ResNet-101 + instance level grounding head) on the synthetic images with base categories, and generate pseudo-labels for novel categories. When training detectors on such synthetic dataset, it gives 31.9 AP on novel categories on the COCO-OVD benchmark, 10.4 AP lower than tight coupling, showing the benefits of rich semantic and positional information encoded in SDM's visual features.

$\mathcal{L}_{\text{base}}$	$\mathcal{L}_{\text{novel}}$	Detector AP	Precision	Recall
✓		70.2	87.9	68.3
✓	✓	79.7	89.1	90.0

Table S1. The quality of the pseudo-labels.

S3. Quality of Pseudo-labels

Here we evaluate the quality of the pseudo-labels generated by the proposed grounding head. We adopt two metrics to assess their quality: (i) Detector AP and (ii) Precision and Recall. For Detector AP, we leverage the pre-trained Mask-RCNN model on the COCO dataset to generate ground truths (GTs) for the synthetic images, and then compute the AP of the pseudo labels derived from the teacher model. In the case of Precision and Recall, we randomly select and annotate 200 synthetic images, then calculate the precision and recall of their pseudo-labels. As shown in Table S1, after self-training on novel categories, the quality of the pseudo-labels can be significantly improved in terms of Detector AP (70.2%→79.7%), Precision (87.9%→89.1%) and Recall (68.3%→90.0%).

S4. Qualitative Results

We show more qualitative results generated by our InstaGen in Figure S1. Without any manual annotations, InstaGen can generate high-quality images with object bounding-boxes of novel categories. In Figure S2, we further show the qualitative results predicted by the Faster R-CNN trained with the synthetic images form InstaGen on COCO validation set. The detector can now accurately localize and recognize the objects from novel categories.

Method	Supervision	Detector	Backbone	Input Size	AP	AP_c	AP_f	AP_{novel}
ViLD-ens. [9]	CLIP	Mask R-CNN	R50	1024×1024	25.5	24.6	30.3	16.6
Detic [41]	CLIP	Mask R-CNN	R50	1024×1024	26.8	26.3	31.6	17.8
F-VLM [16]	CLIP	Mask R-CNN	R50	1024×1024	24.2	-	-	18.6
PromptDet [5]	CLIP	Mask R-CNN	R50	800×800	21.4	18.5	25.8	19.0
DetPro [2]	CLIP	Mask R-CNN	R50	800×800	25.9	25.6	28.9	19.8
BARON [34]	CLIP	Mask R-CNN	R50	800×800	25.1	24.4	28.9	18.0
BARON [34]†	CLIP	Mask R-CNN	R50	800×800	27.6	27.6	29.8	22.6
InstaGen	Stable Diffusion	Mask R-CNN	R50	800×800	23.0	20.6	27.1	20.3

Table S2. Results on open-vocabulary LVIS benchmark. † indicates using ensembling strategy for classification scores and learned prompts for the category's names.

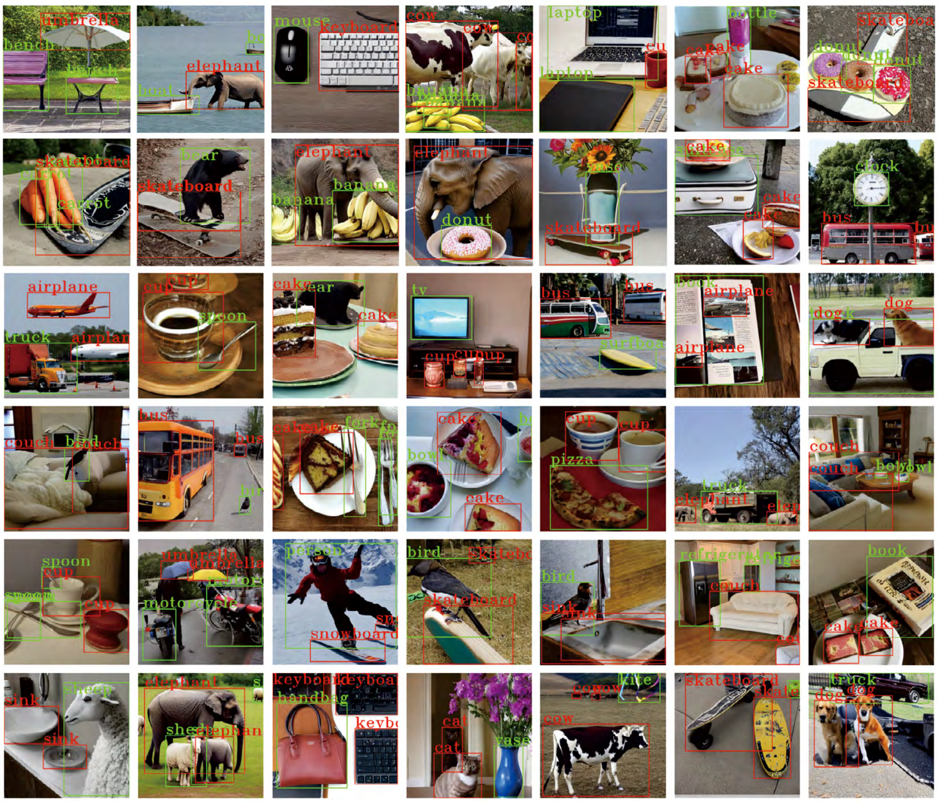
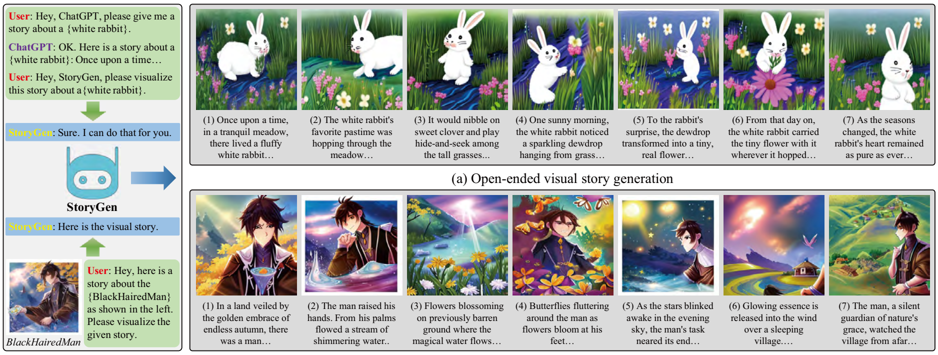


Figure S1. Qualitative results generated by our InstaGen. The bounding-boxes with green denote the objects from base categories, while the ones with red denote the objects from novel categories.

Figure S2. Qualitative results from our Faster R-CNN trained with the synthetic images from InstaGen on COCO validation set. The bounding-boxes with green denote the objects from base categories, while the ones with red denote the objects from novel categories.

Intelligent Grimm – Open-ended Visual Storytelling via Latent Diffusion Models

Chang Liu[1,3*], Haoning Wu[1*], Yujie Zhong[2], Xiaoyun Zhang[1], Yanfeng Wang[1,3], Weidi Xie[1,3]
[1]Coop. Medianet Innovation Center, Shanghai Jiao Tong University, China
[2]Meituan Inc., China [3]Shanghai AI Laboratory, China



(a) Open-ended visual story generation

(b) Open-ended visual story continuation

Figure 1. **An illustration of open-ended visual storytelling.** In practice, users can feed a unique and engaging story synthesized by a large language model into our proposed **StoryGen** model to generate a sequence of images coherently, denoted as *open-ended visual story generation*. And they can also provide a pre-defined character with its corresponding storyline, to perform *open-ended visual story continuation*. We recommend the reader to zoom in and read the story.

Abstract

Generative models have recently exhibited exceptional capabilities in text-to-image generation, but still struggle to generate image sequences coherently. In this work, we focus on a novel, yet challenging task of generating a coherent image sequence based on a given storyline, denoted as ***open-ended visual storytelling****. We make the following three contributions: (i) to fulfill the task of visual storytelling, we propose a learning-based auto-regressive image generation model, termed as* ***StoryGen****, with a novel vision-language context module, that enables to generate the current frame by conditioning on the corresponding text prompt and preceding image-caption pairs; (ii) to address the data shortage of visual storytelling, we collect paired image-text sequences by sourcing from online videos and open-source E-books, establishing processing pipeline for constructing a large-scale dataset with diverse characters, storylines, and artistic styles, named* ***StorySalon****; (iii) Quantitative experiments and human evaluations have validated the superiority of our StoryGen, where we show StoryGen can generalize to unseen characters without any optimization, and generate image sequences with coherent content and consistent character. Code, dataset, and models are available at* `https://haoningwu3639.github.io/StoryGen_Webpage/`.

"Mirror mirror on the wall, who's the fairest of them all?"

—— Grimms' Fairy Tales

1. Introduction

This paper considers an interesting, yet challenging task, namely, *open-ended visual storytelling*. The goal is to train a generative model that effectively captures the relation between visual elements and corresponding text descriptions, to generate a sequence of images that tell a visually coherent story, as shown in Figure 1. The outcome of this task has significant potential for education, as it provides chil-

*: These authors contribute equally to this work.

dren with an engaging and interactive way to learn complex visual concepts and develop imagination, creativity, emotional intelligence, and language skills, as evidenced by research in psychology [5, 48].

The recent literature has witnessed tremendous progress in image generation, particularly with the guidance of text as prompt, such as stable diffusion [42], DALL·E [40] and Imagen [14]. However, to generalize the models for open-ended visual storytelling, we are facing three challenges: (i) previous models are designed to only generate images independently, without considering context, for example, preceding frames or overall narrative, resulting in a lack of visual consistency; (ii) most methods generate images by only conditioning on text, which potentially leads to ambiguities or requires unnecessarily long descriptions to maintain character appearances; (iii) existing datasets are limited to a few animations, covering a closed set of vocabulary or characters [25, 31, 36]. Training on such datasets suffers from severe overfitting on seen characters, leading to unsatisfactory generalization capability for open-ended generation.

This paper describes a learning-based model for open-ended visual storytelling, termed as **StoryGen**, that enables to generate unseen characters without any further optimization, while having character consistency. At inference, StoryGen can synthesize frames either by taking text prompts, or along with preceding image-text pairs as conditions, *i.e.*, iteratively creating visual sequences that are aligned with language description, while being consistent with preceding frames in both style and character perspectives. Specifically, to achieve consistency within the generated image sequence, we incorporate a novel **vision-language context module** into the pre-trained stable diffusion model, which provides visual context by conditioning the generation process on extracted diffusion denoising feature of previous frames under the guidance of corresponding captions.

As for training, we construct a dataset called **StorySalon**, that features a rich source of coherent images and stories, primarily comprising children's storybooks collected from videos and E-books. As a result, our dataset includes a diverse vocabulary with different characters, storylines, and artistic styles. The scale and diversity of our collected dataset enable the model for open-vocabulary visual storytelling, *i.e.*, generating new image sequences that are not limited to pre-defined storylines, characters, or scenes. For example, we can prompt a large language model to create unique and engaging stories, then feed them into StoryGen for generation, as shown in Figure 1.

To summarize, we make the following contributions in this paper: (i) we initiate a fun yet challenging task, namely, *open-ended visual storytelling*, that involves generating engaging image sequences aligned to a given storyline; (ii) we propose a learning-based open-ended visual storytelling model, termed as **StoryGen**, which can generalize to unseen characters without any further optimization and generate coherent visual stories, utilizing a novel vision-language context module; (iii) we establish a data processing pipeline and collect a large-scale dataset of storybooks, called **StorySalon**, from online videos and open-source E-books, resulting in a diverse vocabulary with various characters, storylines, and artistic styles; (iv) we conduct quantitative experiments and human evaluations to validate the effectiveness of our proposed modules, demonstrating the superiority of our model, in terms of image quality, consistency, and visual-language alignment of generated contents.

2. Related Works

Text-to-image Generation has been tackled using various generative models, with GAN [8] as the first widely used model. Several GAN-based methods [53, 56, 57] have achieved notable success, and auto-regressive transformers [49], such as DALL·E [40], have also demonstrated the ability to generate high-quality images based on text prompts. Recently, diffusion models, such as Imagen [44] and DALL·E 2 [41], have emerged as a popular approach. Stable Diffusion Models [42] performs diffusion process in latent space, and can generate impressive images after pre-training on a large-scale text-image dataset.

Diffusion Models learn to model a data distribution via iterative denoising and are trained with denoising score matching. Notably, DDPM [13] demonstrates improved performance over other generative models, while DDIM [46] significantly boosts efficiency. In view of their superior generative capabilities, diffusion models have found extensive utility in various downstream applications besides image generation, such as video generation [6, 14, 15, 45], image manipulation [2, 10, 18, 33], grounded generation [26], image restoration [4], and image inpainting [1, 28, 35, 51].

Story Synthesis is first introduced as the task of story visualization by StoryGAN [25], which presents a GAN-based framework and the PororoSV dataset, derived from cartoons. Some works [29, 30] follow the GAN-based framework, whereas others [3, 21] emphasize more on text representation. StoryDALL-E [31] extends story synthesis to story continuation with the initial image given, and exploits a pre-trained DALL·E model [40] to produce coherent images. AR-LDM [36] introduces an auto-regressive latent diffusion model to generate image sequences, but only consistent within a limited character vocabulary. NUWA-XL [55] exploits hierarchical diffusion models to synthesize long videos, but still achieve character consistency by memorizing. TaleCrafter [7] proposes a story visualization system and utilizes LoRA [16] to achieve character consistency. However, large-scale applications will be constrained due to its optimization-based nature. In this paper, we target more ambitious applications, to develop an open-ended visual storytelling model, that can synthesize coherent image

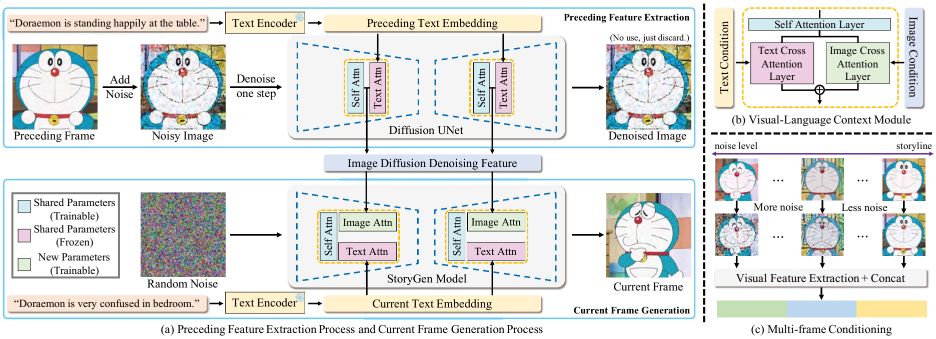


Figure 2. **Architecture Overview**. (a) Our StoryGen model utilizes current text prompt and previous visual-language contexts as conditions to generate an image, iteratively synthesizing a coherent image sequence. Note the parameters of the corresponding attention layers are shared between Diffusion UNet and StoryGen. To avoid potential ambiguity, the parameters are not shared across UNet blocks in a single model. (b) The proposed Visual-Language Context Module can effectively combine the information from current text prompt and contexts from preceding image-caption pairs. (c) We add more noise to reference frames with longer temporal distances to the current frame as positional encoding to distinguish the temporal order. The multiple features can then be directly concatenated to serve as context conditions.

sequences based on storylines of diverse topics.

3. Method

In this section, we start by formulating the problem of open-ended visual storytelling in Section 3.1; then we elaborate on the proposed StoryGen architecture in Section 3.2; lastly, we present details for model training in Section 3.3.

3.1. Problem Formulation

In this paper, we focus on a challenging task, termed as *open-ended visual storytelling*, the goal is to generate continuous image sequence from a given story in the form of natural language. Specifically, we propose a learning-based auto-regressive image generation model, called **StoryGen**, that generates the current frame $\hat{\mathcal{I}}_k$ by conditioning on the current text prompt $\mathcal{T}_k$, and image-text pairs $(\hat{\mathcal{I}}_{<k}, \mathcal{T}_{<k})$ of previous frames, as illustrated in Figure 2 (a). The model is formulated as follows:

$$
\begin{aligned}
\{\hat{\mathcal{I}}_1, \hat{\mathcal{I}}_2, \ldots, \hat{\mathcal{I}}_L\} &= \Phi_{\text{StoryGen}}(\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_L\}; \Theta) \\
\hat{\mathcal{I}}_k &:= \Phi_{\text{StoryGen}}(\hat{\mathcal{I}}_k | \mathcal{T}_k, (\hat{\mathcal{I}}_{<k}, \mathcal{T}_{<k}))
\end{aligned}
$$

Here, $\{\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_L\}$ refer to the given storylines, and $\{\hat{\mathcal{I}}_1, \hat{\mathcal{I}}_2, \ldots, \hat{\mathcal{I}}_L\}$ denote the generated image sequence. $\Phi_{\text{StoryGen}}(\cdot)$ represents our proposed StoryGen model. In one-step generation, StoryGen takes the current text prompt, and preceding image-caption pairs as conditions, and generates an image consistent with both the story's narrative and previous frames. The whole image sequence can then be synthesized with step-by-step inference.

Relation to Existing Tasks. In contrast to existing story visualization works, this paper makes improvements from two aspects: (i) conventional generation/continuation tasks are limited to training on specific characters/stories, for example, [25, 31, 36] only exploits datasets from animation *The Flintstones* and *Pororo*, while our model enables to generate visual stories based on any given storyline, such as a brand-new one generated by ChatGPT; and any pre-defined character, for example, *'Doraemon'* from the Internet; (ii) unlike existing work that requires costly character-specific optimization, for example, [7, 36] rely on LoRA-based [16] optimization to adapt to new characters, our model is learning-based and expected to generalize to any unseen character without any further optimization.

3.2. Architecture

To tackle the problem of open-ended visual storytelling, we expect the model to not only condition on the current text prompt, but also preceding image-text pairs. In this section, we describe the procedure for one-step generation, *i.e.*, generating the k-th frame ($k > 1$) by conditioning on $\{(\hat{\mathcal{I}}_1, \mathcal{T}_1), \ldots, (\hat{\mathcal{I}}_{k-1}, \mathcal{T}_{k-1}), \mathcal{T}_k\}$. Generally speaking, our proposed **StoryGen** model comprises four components: (i) Input Initialization, (ii) Context Encoding, (iii) Visual-Language Contextual Fusion, (iv) Conditional Generation.

Input Initialization. Our model is built upon the foundation of a pre-trained stable diffusion model (SDM), which randomly samples a noisy latent $\mathbf{x}$ from the latent space of the VAE [19] encoder. Moreover, for a given text prompt $\mathcal{T}_k$, the text condition will be extracted by a pre-trained CLIP [38] text encoder ϕ_{CLIP} via $\mathcal{C}^{\text{T}} = \phi_{\text{CLIP}}(\mathcal{T}_k)$.

Context Encoding. In standard SDM, the noisy latent is recursively denoised with a UNet, conditioning on the text prompt. However, in our case, it is crucial for the generation procedure to also condition on context features of preceding frames, to maintain consistency in characters and storyline.

In practice, to extract the contextual features, we add noise to the preceding frames and exploit the pre-trained SDM to denoise for one diffusion step under the guidance of their corresponding captions. The diffusion features after every self-attention layer in the UNet blocks can be directly selected to serve as the conditioning visual context features, thus constituting a pyramid of visual context features. The visual condition features for $\hat{\mathcal{I}}_k$ can be expressed as:

$$\mathcal{C}^{\mathrm{V}} = [\phi_{\mathrm{SDM}}(\hat{\mathcal{I}}_1, \phi_{\mathrm{CLIP}}(\mathcal{T}_1)), \ldots, \phi_{\mathrm{SDM}}(\hat{\mathcal{I}}_{k-1}, \phi_{\mathrm{CLIP}}(\mathcal{T}_{k-1}))]$$

Experimentally, we notice that, the magnitude of noise added to the preceding frames can greatly affect the conditional generation quality, *i.e.*, large-scale noise on preceding frames incurs severe information loss. Thus, we propose to use a much smaller diffusion timestep t' for preceding frames compared with the diffusion timestep t of the current image $\hat{\mathcal{I}}_k$, and follow a $t' = t/10$ rule. As depicted in Figure 2 (c), in case of multiple preceding image-caption pairs, we use larger t' for frames with longer temporal distances to $\hat{\mathcal{I}}_k$. Therefore, the extracted multi-frame visual context features can be directly concatenated, and their different noise level will serve as temporal positional embedding. Such design reflects the intuition that frames with longer distances will incur less effect on generating the current frame.

Vision-Language Contextual Fusion. Here, our vision-language context module is designed to fuse information from current text prompt and contextual information from preceding image-caption pairs. This is achieved by augmenting the transformer decoder in SDM with an additional image cross-attention layer. Note that, the math expression in this section is not strict, we omit the footnote of diffusion timestep t and UNet block level l for simplicity.

Specifically, on visual context conditioning, the noisy latent $\mathbf{x}$ is projected into query, and cross-attends to the visual context features from the corresponding-level UNet block that act as key and value, denoted as:

$$\mathbf{Q}_I = \mathbf{x}\mathbf{W}_I^Q, \quad \mathbf{K}_I = \mathcal{C}^{\mathrm{V}}\mathbf{W}_I^K, \quad \mathbf{V}_I = \mathcal{C}^{\mathrm{V}}\mathbf{W}_I^V$$

where $\mathbf{W}_I^Q$, $\mathbf{W}_I^K$, and $\mathbf{W}_I^V$ represent different projection matrices, respectively.

On text conditioning, the noisy latent $\mathbf{x}$ is again projected to query, and cross-attends to the text features of the current prompt encoded by CLIP text encoder, *i.e.*,

$$\mathbf{Q}_T = \mathbf{x}\mathbf{W}_T^Q, \quad \mathbf{K}_T = \mathcal{C}^{\mathrm{T}}\mathbf{W}_T^K, \quad \mathbf{V}_T = \mathcal{C}^{\mathrm{T}}\mathbf{W}_T^V$$

where $\mathbf{W}_T^Q$, $\mathbf{W}_T^K$, and $\mathbf{W}_T^V$ also represent corresponding projection matrices.

As depicted in Figure 2 (b), the image cross-attention layer is inserted in parallel to the text cross-attention layer in the transformer decoder of UNet blocks. Drawing inspiration from ControlNet [58], the results from these two cross-attention layers are simply summed up as the final output $\mathbf{O}$. The final output can thus be expressed as:

$$\mathbf{O} = \mathrm{Softmax}(\frac{\mathbf{Q}_I(\mathbf{K}_I)^\top}{\sqrt{d}})\mathbf{V}_I + \mathrm{Softmax}(\frac{\mathbf{Q}_T(\mathbf{K}_T)^\top}{\sqrt{d}})\mathbf{V}_T$$

Conditional Generation. With the fused vision-language condition features from above, our StoryGen can now generate visual stories that achieve both content coherence and character consistency. Here, our conditional generation procedure can be represented as:

$$\hat{\mathcal{I}}_k = \Phi_{\mathrm{StoryGen}}(\hat{\mathcal{I}}_k | \mathcal{T}_k, (\hat{\mathcal{I}}_{<k}, \mathcal{T}_{<k})) = \Phi_{\mathrm{StoryGen}}(\mathbf{x}, \mathcal{C}^{\mathrm{T}}, \mathcal{C}^{\mathrm{V}})$$

With the new conditioning modality introduced, we also adopt another classifier-free guidance term [12], as has been done in [2]. Concretely, we exploit two different guidance scales, w_v and w_t for the visual condition and the text condition. The relation between the final noise for inference $\bar{\epsilon}_\theta$ and UNet-predicted noise ϵ_θ is now expressed as:

$$\begin{aligned}\bar{\epsilon}_\theta(\mathbf{x}_t, t, \mathcal{C}^{\mathrm{V}}, \mathcal{C}^{\mathrm{T}}) &= \epsilon_\theta(\mathbf{x}_t, t, \varnothing, \varnothing) \\ &+ w_v(\epsilon_\theta(\mathbf{x}_t, t, \mathcal{C}^{\mathrm{V}}, \varnothing) - \epsilon_\theta(\mathbf{x}_t, t, \varnothing, \varnothing)) \\ &+ w_t(\epsilon_\theta(\mathbf{x}_t, t, \mathcal{C}^{\mathrm{V}}, \mathcal{C}^{\mathrm{T}}) - \epsilon_\theta(\mathbf{x}_t, t, \mathcal{C}^{\mathrm{V}}, \varnothing))\end{aligned}$$

Discussion. Our work differs from previous ones from two aspects. First, our StoryGen is a learning-based method, which can directly generalize to unseen characters by attending to reference images. Second, we propose to condition the generation process on diffusion features of preceding image-text pairs from the same SDM, which preserves more visual details, greatly differing from existing works [22, 52, 54] using CLIP, BLIP [24], or VAE features.

3.3. Model Training

Training Objective. At training stage, we randomly sample a triplet each time, *i.e.*, $\{\mathcal{I}_k, \mathcal{T}_k, (\mathcal{I}_{<k}, \mathcal{T}_{<k})\}$. The objective function can be expressed as:

$$\mathcal{L}_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\epsilon_t,\mathcal{C}^{\mathrm{V}},\mathcal{C}^{\mathrm{T}}}\left[\|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t, \mathcal{C}^{\mathrm{V}}, \mathcal{C}^{\mathrm{T}})\|^2\right]$$

Two-stage Training Strategy. Our two-stage training strategy includes single-frame pre-training and multiple-frame fine-tuning. To be specific, at the first stage, we do not introduce additional image cross-attention layers, and only train self-attention layers in standard SDM to ensure the single-frame generation ability. In multiple-frame fine-tuning, we train additional image cross-attention layers in vision-language context module on our dataset, with all other parameters frozen. This enables the generation procedure to

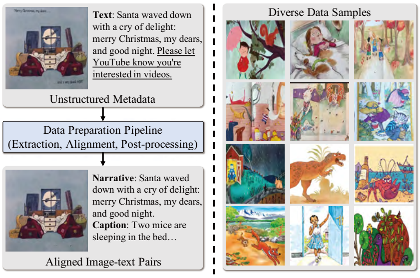


Figure 3. **Dataset Pipeline and Visualization**. **Left**: Metadata sourced from the Internet undergoes a three-step pipeline including frame extraction, visual-language alignment and post-processing, resulting in properly aligned image-text pairs. **Right**: Our StorySalon dataset contains diverse styles and characters.

utilize information from not only current prompt, but also preceding image-caption pairs.

Inference. As shown in Figure 1, at inference time, we can prompt ChatGPT to generate novel storylines, and synthesize the first image directly or attending to a pre-defined character. Then the previously synthesized frames, along with the story descriptions, are treated as conditions to synthesize the image sequence in an auto-regressive manner. Experimentally, our proposed StoryGen is shown to generate images that align with the storyline, as well as maintain consistency with previously generated frames.

4. StorySalon Dataset

In order to train our proposed *open-ended visual storytelling* model, we construct a large-scale dataset, termed as **StorySalon**. The dataset contains videos and E-books with diverse characters, storylines, and artistic styles. Specifically, we download a large number of videos and subtitles from YouTube, by querying keywords related to story-telling for children, for instance, *storytime*. Additionally, we collect E-books (partially with corresponding audios available) from six open-source libraries which are all registered under the Creative Commons 4.0 International Attribution (CC BY 4.0) license. In the following, we elaborate on the data processing pipeline and statistics of our collected dataset.

Visual Frame Extraction. We extract keyframes from the videos, along with the corresponding subtitles and their timestamps. To remove duplicate frames, we extract ViT features for each frame using pre-trained DINO [32]. For the image groups with high similarity scores, we only keep one of each. Then, we use YOLOv7 [50] to segment and remove real-person frames and headshots, as they often correspond to the story-teller and are unrelated to the content of the storybook. Similarly, we extract images from the

Dataset	Style	#Frames	Avg.Length	#Categories
PororoSV [25]	Animation	73,665	5	9
FlintstonesSV [9]	Animation	122,560	5	7
DiDeMoSV [31]	Real	52,905	3	-
VIST [17]	Real	145,950	5	-
StorySalon	Animation	**159,778**	**14**	**446**

Table 1. **Dataset Statistics.** Our StorySalon dataset far exceeds previous story generation datasets in terms of the total number of images, average length, and categories of characters included.

downloaded E-books, except for those with extraneous information, for example, the authorship page. We acquire the corresponding text description with Whisper [39] from the audio file, and for E-books that do not have corresponding audio files, but with available storyline text, we use OCR algorithms, to directly recognize the text on each page.

Visual-Language Alignment. As shown in Figure 3, for each of the image, we can collect two types of text descriptions, *e.g.*, story-level narration, and descriptive captions. This is based on our observation that there actually exists a semantic gap between narrative storyline and descriptive text, for example, the same image can be well described as *"The cat is isolated by others, sitting alone in front of a village."* in the story, or *"A black cat sits in front of a number of houses."* as descriptive caption, therefore, directly fine-tuning stable diffusion models with story narration may be detrimental to its pre-trained text-image alignment. In practice, to get story-level paired image-text samples, we align the subtitles with visual frames by using Dynamic Time Warping (DTW) algorithm [34]. To get visual descriptions, we use TextBind [23] to generate captions for each image, with both the image and the corresponding narrative text as inputs. At training time, this allows us to substitute the original story with more accurate and descriptive captions.

Visual Frame Post-processing. In practice, we observe that book pages and borders in images can potentially interfere with our generative model by having story texts printed on them. To tackle this, we use an OCR detector to identify text regions in images and an image inpainting model [42] to fill in the text and headshot regions, resulting in more precise image-text pairs that are suitable for model training.

Discussion. After the three-step pipeline above, we obtain our StorySalon dataset. As shown in Table 1, our dataset has nearly 160K animation-style images in total with an average length of 14 frames per story, which is conducive to building long-range semantic correspondence. Finally, we query MiniGPT-4 [59] about the main character category of each image in our dataset, like *Dog* and *Cat*, then count the categories and filter out those appear less than 3 times. Compared with previous datasets with less than 10 characters, our dataset comprises hundreds of character categories, and even more character instances, which provides a data basis

Model	FID ↓	CLIP-I ↑	CLIP-T ↑
GT	-	1.0	0.2668
SDM	73.50	0.6155	0.3218
Prompt-SDM	67.35	0.6272	**0.3225**
Finetuned-SDM	42.01	0.6970	0.3005
StoryDALL·E	38.34	0.6823	0.2366
AR-LDM	39.55	0.6864	0.2614
StoryGen	**33.90**	**0.7467**	0.2875

Table 2. **Comparison of automatic metrics** on StorySalon test set. Prompt-SDM denotes Stable Diffusion model with cartoon-style-directed prompts and Finetuned-SDM represents a Stable Diffusion model with all parameters fine-tuned on our StorySalon dataset.

Model	Align. ↑	Style ↑	Cont. ↑	Char. ↑	Qual. ↑	Pref. ↑
Story Generation						
GT	4.04	4.66	4.41	4.54	4.29	–
SDM	3.61	2.88	2.90	2.51	3.74	14.05%
Prompt-SDM	3.39	2.56	2.68	2.10	3.44	8.57%
StoryGen-S	3.50	2.73	2.81	2.21	3.19	10.24%
StoryGen	**3.78**	**4.79**	**4.26**	**4.64**	**3.76**	**67.14%**
Story Continuation						
StoryDALL·E	1.18	1.55	1.20	1.14	1.19	0.63%
AR-LDM	2.47	2.82	2.40	1.87	2.54	2.50%
StoryGen	**4.23**	**4.70**	**4.35**	**4.38**	**4.18**	**96.87%**

Table 3. **Comparison results of human evaluation.** GT stands for ground truth from the test set. StoryGen-S represents StoryGen without context conditions. The abbreviated metrics are Text-image alignment, Style consistency, Content consistency, Character consistency, image quality, and Preference, respectively.

for training open-ended visual storytelling models, showing a significantly broader range of visual styles and character appearances over existing datasets.

5. Experiments

In this section, we start by describing our experimental settings, then compare with other models from three different perspectives: image-text alignment, consistency and image quality with subjective human evaluation and quantitative metrics. Additionally, we present results for ablation experiments to prove the effectiveness of our proposed modules.

5.1. Experimental Settings

Training Details. Our model is built on the stable diffusion v1.5 model, and trained with a learning rate of 1×10^{-5} and a batch size of 256. We begin with a single-frame self-attention pre-training stage, which involves 3,000 iterations on 8 NVIDIA RTX3090. Next, we incorporate our proposed vision-language context module, and train it for 5,000 iterations using a single preceding image-caption pair as context condition, then continue to train it for another 5,000 iterations with multiple image-caption pairs for multi-frame conditioning. To maintain our model's unconditional denoising ability for classifier-free guidance, we randomly drop the current text and the context image-caption pairs with a probability of 5% and 15%, respectively. During inference, we utilize DDIM [46] with 40 steps of sampling and select the guidance weight $w_v = 7.0$ and $w_t = 3.5$.

Baselines. We consider two scenarios of our proposed open-ended storytelling task, namely, story generation and story continuation. For **story generation**, we need the model to be able to generate a complete visual story only based on a given storyline. So we present a comparison with Stable Diffusion Model (**SDM**) and **Prompt-SDM**, which conditions on an additional cartoon-style-directed prompt *"A cartoon style image"*. For **story continuation**, the first frame or the main character is given, and the model is expected to generate coherent images based on the storyline. In this scenario, we compare our model with two closed-set story continuation models: namely, **StoryDALL·E** [31] and **AR-LDM** [36] re-trained on our StorySalon dataset.

Automatic Metrics. To evaluate the quality of generated image sequences, we adopt three widely-used metrics, including Fréchet Inception Distance score (FID) [11], CLIP image-image similarity (CLIP-I), and CLIP text-image similarity (CLIP-T). Notably, in order to avoid the impact of randomness in synthesis quality, we utilize a CLIP-based scoring function trained exclusively on text-to-image generated images, namely, PickScore [20], to automatically select the generated images with better quality. Each chosen image is selected from a pool of 10 candidates.

5.2. Quantitative Evaluation Results

We compare our StoryGen model with other baselines on StorySalon test set, which contains 5% of total data (nearly 7K pairs). Each contains a current prompt and the image-text context of the previous frame. The models are expected to generate the current frame based on given conditions.

The quantitative results in Table 2 demonstrate that our StoryGen model exhibits significant performance improvement in terms of FID score and CLIP-I similarity compared to existing models, while maintaining comparable CLIP-T similarity. This confirms that our model can effectively exploit contextual information, thus generating animation-style visual stories based on the given storyline. Notably, CLIP trained on natural images tends to have an understanding bias towards animation-style images, and the slight decline in CLIP-T is an inevitable result of the conflict between text condition and newly introduced image condition.

5.3. Human Evaluation Results

Considering that the above metrics may not reflect the quality of the generated stories accurately, and there is no stan-

(a) *Open-ended story generation* for: **a story of a {white dog}**: (1) Once upon a time, in a peaceful countryside, there lived a white dog… (2) The white dog had an adventurous spirit, always eager to discover… (3) One afternoon, the white dog was staring at a sunflower… (4) The white dog ventured into a sunflower field… (5) The white dog discovered a bird's nest in the field… (6) From that day on, the white dog became a guardian… (7) The white dog's spirit remained steadfast and bright, as the seasons changed and the leaves fell…

(b) *Open-ended story continuation* for: **a story of a {a white-haired man}**: (1) In a perpetual twilight, the white-haired man reached towards the twilight sky, stars appearing at his touch… (2) One twilight, the white-haired man looked concerned at a dark void in the sky... (3) The white-haired man drew stars in the sky with a silver quill… (4) The white-haired man was observing new constellations shining where the void once was… (5) The white-haired man with a serene expression was watching the peaceful starry sky... (6) The white-haired man walked towards a tower observatory under the starry sky… (7) Alone but content, the white-haired man's gaze traversed the depths of space…

Figure 4. **Qualitative Comparison with other methods**. The image sequences in orange, green, and blue boxes are generated by Prompt-SDM, AR-LDM and StoryGen respectively. Our synthesis results exhibit impressive performance superiority in terms of style, content and character consistency, text-image alignment, and image quality. Please refer to the Appendix for more qualitative results.

dardized metric for evaluating the consistency within the visual story, we further include human evaluation for comparison of image-text alignment, image style, story consistency, character consistency and synthesis quality.

For the two scenarios mentioned above, we respectively conduct two types of human evaluation to assess the quality of generated visual stories. To mitigate bias, participants are unaware of the type of storybooks they are evaluating. Concretely, we prompt GPT-4 to produce multiple storylines for both test modes, and for story continuation, we search the Internet for multiple characters that have never appeared in our dataset. Then we utilize our StoryGen along with other baselines to generate corresponding sequences of images.

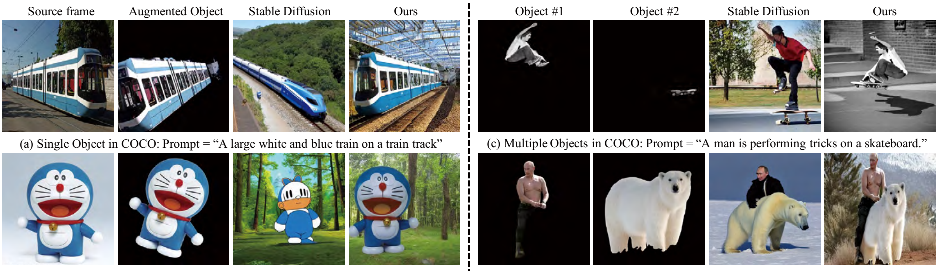
Protocol-I. We randomly select an equal number of samples from the generated results of our StoryGen and other baselines. Each time we randomly sample a visual story from these sources, and participants are then invited to rate the sample with a score ranging from 1 to 5, taking into account text-image alignment, style consistency, content consistency, character consistency and image quality. Higher scores indicate better samples. We also evaluate the same number of samples from StorySalon test set as a reference.

Protocol-II. Each time we randomly sample a storyline and its corresponding visual storybooks generated by StoryGen and other methods. Participants are invited to select their preferred generated result among these different image sequences of the same storyline.

Results. The results of human evaluation presented in Table 3 illustrate that our StoryGen model demonstrates excellent performance in overall score, especially in terms of consistency and quality. This indicates that our model can generate coherent image sequences that are highly consistent with given text prompts and visual-language contexts.

5.4. Qualitative Results

In Figure 4, we present visualization results of both open-ended visual story generation and visual story continuation,



(a) Single Object in COCO: Prompt = "A large white and blue train on a train track"

(b) Single Object in Real-Data: Prompt = "Doraemon is standing in a forest."

(c) Multiple Objects in COCO: Prompt = "A man is performing tricks on a skateboard."

(d) Multiple Objects in Real-Data: Prompt = "Putin is riding a polar bear in the wild."

Figure 5. **Ablation studies on consistency**. We incorporate our proposed Visual-Language Context Module into a pre-trained SDM, and train it on MS-COCO [27] with other parameters frozen. The content consistency of single-object and multi-object generation on COCO and real data has demonstrated the effectiveness of our module. Please refer to the Appendix for experiment details and quantitative results.

showing that our StoryGen can generate visual stories with a broad vocabulary, while maintaining content coherence and character consistency throughout the narrative, whereas other methods fail to do so. Moreover, our model can stably maintain the animation style of generated images, which satisfies the requirements of visual storytelling for children. More results can be found in the supplementary material.

5.5. Ablation Studies

In order to demonstrate the effectiveness of our proposed modules, we conduct ablation studies from both quantitative metrics and qualitative visualization.

On Variants of StoryGen. We evaluate the performance of multiple model variants on the StorySalon test set, including (i) our model without the context module, marked as **StoryGen-Single**, which solely fine-tunes the self-attention layers on our dataset. (ii) our model with context features encoded by the VAE of SDM as context condition, without text-guided diffusion process, denoted as **StoryGen-VAE**; (iii) our model with CLIP image embedding as context condition (**StoryGen-CLIP**); (iv) our model with context features extracted by BLIP image encoder (**StoryGen-BLIP**); (v) our model with naive denoising features at **L**arge-scale diffusion **T**imestep, satisfying $t' = t$, as condition (**StoryGen-LT**); and (vi) our full model (**StoryGen**). We also employ PickScore to filter generation results of all these models. The findings presented in Table 4 illustrate the inclusion of our context module can significantly improve the model performance, in terms of CLIP-I and FID. As for the slight inferiority in CLIP-T, we have claimed above that this is due to the understanding bias towards animation-style images for CLIP trained on natural images.

Qualitative Visualization. As mentioned above, consistency is a crucial factor in visual story generation. We hope to more intuitively demonstrate that our proposed context module can accurately capture the image content of the previous frame. To this end, we incorporate our context module into SDM and train it from scratch on the MS-COCO [27] with other parameters frozen. Specifically, we crop the object and perform data augmentations such as translation and rotation to use it as image condition. The category of the cropped object is used as its corresponding text, and the caption of the original image serves as the text prompt. We expect the model to reconstruct the original image relying on the conditions above, which enables the context module to learn how to leverage the previous image. As shown in Figure 5, our model can make full use of the objects in the reference frame and generate new images that are consistent with them, while SDM fails to do so. In addition, this can also be transferred to any real-world reference image, which strongly illustrates the robustness and capability of our context module to assist diffusion models in generating images based on any given object.

Model	FID ↓	CLIP-I ↑	CLIP-T ↑
StoryGen-Single	38.81	0.6869	**0.3140**
StoryGen-VAE	36.98	0.6846	0.3061
StoryGen-CLIP	36.66	0.6934	**0.3140**
StoryGen-BLIP	34.78	0.7026	0.2838
StoryGen-LT	36.41	0.7141	0.3025
StoryGen	**33.90**	**0.7467**	0.2875

Table 4. **Ablation studies** on Visual-Language Context Module.

6. Conclusion

In this paper, we consider an interesting, yet challenging task, termed as *open-ended visual storytelling*, which involves generating a sequence of images that tell a coherent visual story based on the given storyline. Our proposed learning-based **StoryGen** model can take input from the preceding image-caption context along with the text prompt to generate coherent image sequences in an auto-regressive manner, *i.e.*, without test-time optimization. On the data side, we establish a data processing pipeline to collect a large-scale dataset named **StorySalon** that comprises storybooks with diverse characters, storylines, and artistic styles sourced from videos and E-books. Extensive human evaluation and quantitative comparison have illustrated that our proposed model substantially outperforms existing models, from the perspective of image quality, content coherence, character consistency, and visual-language alignment.

Acknowledgments

This work is supported by National Key R&D Program of China (No. 2022ZD0161400), National Natural Science Foundation of China (62271308), STCSM (22511105700, 22DZ2229005), 111 plan (BP0719010), and State Key Laboratory of UHD Video and Audio Production and Presentation.

References

[1] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[2] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2, 4

[3] Hong Chen, Rujun Han, Te-Lin Wu, Hideki Nakayama, and Nanyun Peng. Character-centric story visualization via visual planning and token alignment. In *Proceedings of the Conference on Empirical Methods in Natural Language Processinng*, 2022. 2

[4] Zheng Chen, Yulun Zhang*, Ding Liu, Bin Xia, Jinjin Gu, Linghe Kong*, and Xin Yuan. Hierarchical integration diffusion model for realistic image deblurring. In *Advances in Neural Information Processing Systems*, 2023. 2

[5] K. Dickinson David, A. Griffith Julie, Golinkoff Roberta, Michnick, and Hirsh-Pasek Kathy. How reading books fosters language development around the world. *Child Development Research*, 2012. 2

[6] Patrick Esser, Johnathan Chiu, Parmida Atighehchian, Jonathan Granskog, and Anastasis Germanidis. Structure and content-guided video synthesis with diffusion models. In *Proceedings of the International Conference on Computer Vision*, 2023. 2

[7] Yuan Gong, Youxin Pang, Xiaodong Cun, Menghan Xia, Yingqing He, Haoxin Chen, Longyue Wang, Yong Zhang, Xintao Wang, Ying Shan, and Yujiu Yang. Talecrafter: Interactive story visualization with multiple characters. *SIGGRAPH Asia*, 2023. 2, 3

[8] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 2020. 2

[9] Tanmay Gupta, Dustin Schwenk, Ali Farhadi, Derek Hoiem, and Aniruddha Kembhavi. Imagine this! scripts to compositions to videos. In *Proceedings of the European Conference on Computer Vision*, 2018. 5

[10] Amir Hertz, Ron Mokady, Jay Tenenbaum, Kfir Aberman, Yael Pritch, and Daniel Cohen-Or. Prompt-to-prompt image editing with cross attention control. In *Proceedings of the International Conference on Learning Representations*, 2023. 2

[11] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. In *Advances in Neural Information Processing Systems*, 2017. 6

[12] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 4

[13] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems*, 2020. 2

[14] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Gritsenko, Diederik P Kingma, Ben Poole, Mohammad Norouzi, David J Fleet, et al. Imagen video: High definition video generation with diffusion models. *arXiv preprint arXiv:2210.02303*, 2022. 2

[15] Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*, 2022. 2

[16] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models. In *Proceedings of the International Conference on Learning Representations*, 2022. 2, 3

[17] Ting-Hao Huang, Francis Ferraro, Nasrin Mostafazadeh, Ishan Misra, Aishwarya Agrawal, Jacob Devlin, Ross Girshick, Xiaodong He, Pushmeet Kohli, Dhruv Batra, et al. Visual storytelling. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016. 5

[18] Bahjat Kawar, Shiran Zada, Oran Lang, Omer Tov, Huiwen Chang, Tali Dekel, Inbar Mosseri, and Michal Irani. Imagic: Text-based real image editing with diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[19] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations*, 2014. 3

[20] Yuval Kirstain, Adam Polyak, Uriel Singer, Shahbuland Matiana, Joe Penna, and Omer Levy. Pick-a-pic: An open dataset of user preferences for text-to-image generation. In *Advances in Neural Information Processing Systems*, 2023. 6, 5

[21] Bowen Li. Word-level fine-grained story visualization. In *Proceedings of the European Conference on Computer Vision*, 2022. 2

[22] Dongxu Li, Junnan Li, and Steven CH Hoi. Blip-diffusion: Pre-trained subject representation for controllable text-to-image generation and editing. In *Advances in Neural Information Processing Systems*, 2023. 4

[23] Huayang Li, Siheng Li, Deng Cai, Longyue Wang, Lemao Liu, Taro Watanabe, Yujiu Yang, and Shuming Shi. Textbind: Multi-turn interleaved multimodal instruction-following. *arXiv preprint arXiv:2309.08637*, 2023. 5

[24] Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning*, 2022. 4

[25] Yitong Li, Zhe Gan, Yelong Shen, Jingjing Liu, Yu Cheng, Yuexin Wu, Lawrence Carin, David Carlson, and Jianfeng Gao. Storygan: A sequential conditional gan for story visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 2, 3, 5

[26] Ziyi Li, Qinye Zhou, Xiaoyun Zhang, Ya Zhang, Yanfeng Wang, and Weidi Xie. Guiding text-to-image diffusion model towards grounded generation. In *Proceedings of the International Conference on Computer Vision*, 2023. 2

[27] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Proceedings of the European Conference on Computer Vision*, 2014. 8, 4

[28] Andreas Lugmayr, Martin Danelljan, Andres Romero, Fisher Yu, Radu Timofte, and Luc Van Gool. Repaint: Inpainting using denoising diffusion probabilistic models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2

[29] Adyasha Maharana and Mohit Bansal. Integrating visuospatial, linguistic, and commonsense structure into story visualization. In *Proceedings of the Conference on Empirical Methods in Natural Language Processinng*, 2021. 2

[30] Adyasha Maharana, Darryl Hannan, and Mohit Bansal. Improving generation and evaluation of visual stories via semantic consistency. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2021. 2

[31] Adyasha Maharana, Darryl Hannan, and Mohit Bansal. Storydall-e: Adapting pretrained text-to-image transformers for story continuation. In *Proceedings of the European Conference on Computer Vision*, 2022. 2, 3, 5, 6

[32] Caron Mathilde, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision*, 2021. 5

[33] Chenlin Meng, Yutong He, Yang Song, Jiaming Song, Jiajun Wu, Jun-Yan Zhu, and Stefano Ermon. Sdedit: Guided image synthesis and editing with stochastic differential equations. In *Proceedings of the International Conference on Learning Representations*, 2021. 2

[34] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, 2007. 5

[35] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *Proceedings of the International Conference on Machine Learning*, 2022. 2

[36] Xichen Pan, Pengda Qin, Yuhong Li, Hui Xue, and Wenhu Chen. Synthesizing coherent story with auto-regressive latent diffusion models. In *Winter Conference on Applications of Computer Vision*, 2024. 2, 3, 6

[37] Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. Sdxl: Improving latent diffusion models for high-resolution image synthesis. *arXiv preprint arXiv:2307.01952*, 2023. 6

[38] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Proceedings of the International Conference on Machine Learning*, 2021. 3

[39] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. Robust speech recognition via large-scale weak supervision. In *Proceedings of the International Conference on Machine Learning*, 2023. 5

[40] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *Proceedings of the International Conference on Machine Learning*, 2021. 2

[41] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *arXiv preprint arXiv:2204.06125*, 2022. 2

[42] Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Bjorn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022. 2, 5, 3, 6

[43] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, 2015. 2

[44] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L Denton, Kamyar Ghasemipour, Raphael Gontijo Lopes, Burcu Karagol Ayan, Tim Salimans, et al. Photorealistic text-to-image diffusion models with deep language understanding. In *Advances in Neural Information Processing Systems*, 2022. 2

[45] Uriel Singer, Adam Polyak, Thomas Hayes, Xi Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, et al. Make-a-video: Text-to-video generation without text-video data. In *Proceedings of the International Conference on Learning Representations*, 2023. 2

[46] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *Proceedings of the Inter-*

national Conference on Learning Representations, 2020. 2, 6

[47] Yang Song, Prafulla Dhariwal, Mark Chen, and Ilya Sutskever. Consistency models. In *Proceedings of the International Conference on Machine Learning*, 2023. 6

[48] Gabrielle A. Strouse, Angela Nyhout, and Patricia A. Ganea. The role of book features in young children's transfer of information from picture books to real-world contexts. *Frontiers in Psychology*, 2018. 2

[49] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, 2017. 2

[50] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 5

[51] Shaoan Xie, Zhifei Zhang, Zhe Lin, Tobias Hinz, and Kun Zhang. Smartbrush: Text and shape guided object inpainting with diffusion model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2023. 2

[52] Li Xin, Chu Wenqing, Wu Ye, Yuan Weihang, Liu Fanglong, Zhang Qi, Li Fu, Feng Haocheng, Ding Errui, and Wang Jingdong. Videogen: A reference-guided latent diffusion approach for high definition text-to-video generation. *arXiv preprint arXiv:2309.00398*, 2023. 4

[53] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 2

[54] Hu Ye, Jun Zhang, Sibo Liu, Xiao Han, and Wei Yang. Ip-adapter: Text compatible image prompt adapter for text-to-image diffusion models. *arXiv preprint arxiv:2308.06721*, 2023. 4

[55] Shengming Yin, Chenfei Wu, Huan Yang, Jianfeng Wang, Xiaodong Wang, Minheng Ni, Zhengyuan Yang, Linjie Li, Shuguang Liu, Fan Yang, et al. Nuwa-xl: Diffusion over diffusion for extremely long video generation. In *Association for Computational Linguistics*, 2023. 2

[56] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the International Conference on Computer Vision*, 2017. 2

[57] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaogang Wang, Xiaolei Huang, and Dimitris N Metaxas. Stackgan++: Realistic image synthesis with stacked generative adversarial networks. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[58] Lvmin Zhang, Anyi Rao, and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. In *Proceedings of the International Conference on Computer Vision*, 2023. 4

[59] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*, 2023. 5, 4

Intelligent Grimm - Open-ended Visual Storytelling via Latent Diffusion Models

Supplementary Material

Contents

A. Preliminaries on Diffusion Models

Diffusion models are a type of generative models that undergo a denoising process, converting input noise into meaningful data samples. Diffusion models comprise a forward diffusion process that incorporates Gaussian noise into an image sample $\mathbf{x}_0$, accomplished via a Markov process over T steps. If we denote the noisy image at step t as $\mathbf{x}_t$, the transition function $q(\mathbf{x}_t|\mathbf{x}_{t-1})$ connecting $\mathbf{x}_{t-1}$ and $\mathbf{x}_t$ can be expressed as follows:

$$q(\mathbf{x}_t|\mathbf{x}_{t-1}) = \mathcal{N}(\mathbf{x}_t; \sqrt{1-\beta_t}\mathbf{x}_{t-1}, \beta_t\mathbf{I}) \quad q(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} q(\mathbf{x}_t|\mathbf{x}_{t-1})$$

where $\beta_t \in (0, 1)$ is the variance schedule controlling the step size.

Using Gaussian distribution property and reparameterization, if we define $\alpha_t = 1 - \beta_t$ and $\bar{\alpha}_t = \prod_{i=1}^{t} \alpha_i$, we can write the equation above as follows:

$$q(\mathbf{x}_t|\mathbf{x}_0) = \mathcal{N}(\mathbf{x}_t; \sqrt{\bar{\alpha}_t}\mathbf{x}_0, (1-\bar{\alpha}_t)\mathbf{I})$$

Diffusion models also comprise a reverse diffusion process that learns to restore the initial image sample from noise. A UNet-based model [43] is utilized in the diffusion model to learn the reverse diffusion process p_θ. The process p_θ can be expressed using the following equation.

$$p_\theta(\mathbf{x}_{0:T}) = p(\mathbf{x}_T)\prod_{t=1}^{T} p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) \quad p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t) = \mathcal{N}(\mathbf{x}_{t-1}; \boldsymbol{\mu}_\theta(\mathbf{x}_t, t), \boldsymbol{\Sigma}_\theta(\mathbf{x}_t, t))$$

where $\boldsymbol{\mu}_\theta$ is the predicted Gaussian distribution mean value.

As we compute the loss function by taking the mean absolute error of the noise term $\boldsymbol{\epsilon}_\theta$ into account, we can express the mean value $\boldsymbol{\mu}_\theta$ in terms of the noise term $\boldsymbol{\epsilon}_\theta$ as follows:

$$\boldsymbol{\mu}_\theta(\mathbf{x}_t, t) = \frac{1}{\sqrt{\alpha_t}}\Big(\mathbf{x}_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\Big)$$

Therefore, the objective can be written as:

$$\mathcal{L}_t = \mathbb{E}_{t\sim[1,T],\mathbf{x}_0,\boldsymbol{\epsilon}_t}\Big[\|\boldsymbol{\epsilon}_t - \boldsymbol{\epsilon}_\theta(\mathbf{x}_t, t)\|^2\Big]$$

B. Further Architecture Details

In this section, we will provide a more comprehensive illustration about more design details of our model.

B.1. Parameter Sharing Strategy

Initially, we will discuss the strategy of parameter sharing between the standard diffusion UNet and our StoryGen. As illustrated in Figure 2 of the main paper, the standard diffusion UNet is exploited in *Preceding Feature Extraction* to extract diffusion context features, and StoryGen is utilized in *Current Frame Generation* to generate new frames with consistency and coherence.

Specifically, the parameters of the corresponding attention layers are shared between the standard diffusion UNet and our StoryGen, including self-attention layers and text cross-attention layers. In practise, the standard diffusion UNet here is a modified version of our StoryGen, without the image cross-attention layers, and all other parameters are shared. This design allows the UNet to extract contextual diffusion features within the same latent space as StoryGen.

B.2. Multi-frame Condition Strategy

As illustrated in Figure 2 and Section 3.2 of the main paper, when dealing with multiple preceding image-caption pairs, we add more noise (corresponding to a larger diffusion timestep t') to reference frames with longer temporal distances to the current frame. Such design effectively serves two purposes, *first*, it is based on the observation that frames with longer temporal distances will incur less effect on the generation of the current frame; *second*, the different noise level also serves as positional encoding, allowing for the differentiation of temporal order, which enables us to directly concatenate these diffusion context features.

Specifically, we use t to represent the diffusion timestep of the current image $\hat{\mathcal{I}}_k$, and use t'_j to represent the diffusion timestep of the preceding image-text pair $(\hat{\mathcal{I}}_j, \mathcal{T}_j)$. When generating image $\hat{\mathcal{I}}_k$, we use $t'_{k-1} = t/10$ for $(\hat{\mathcal{I}}_{k-1}, \mathcal{T}_{k-1})$ pair, $t'_{k-2} = 2t/10$ for $(\hat{\mathcal{I}}_{k-2}, \mathcal{T}_{k-2})$ pair, and so on. In summary, the diffusion timestep t'_{k-i} for $(\hat{\mathcal{I}}_{k-i}, \mathcal{T}_{k-i})$ pair will follow a $t'_{k-i} = i * t/10$ rule.

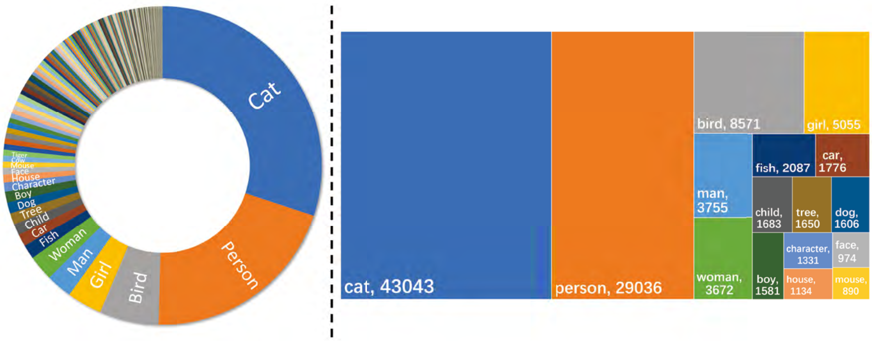


Figure 6. **Dataset Statistics Results. Left:** Distribution of text-image pairs classified by the main character categories in our collected StorySalon dataset. **Right:** The top 16 character categories and corresponding numbers in StorySalon, cover a wide range of character types.

B.3. Two-stage Training Strategy

As illustrated in Section 3.3 of the main paper, we exploit a two-stage training strategy, including single-frame pre-training and multiple-frame fine-tuning. In **single-frame pre-training stage**, we do not introduce additional image cross-attention layers, and only train the self-attention layers in standard SDM [42] on our dataset. The goal of this stage is to train the model for single-frame generation in the style of storybooks. In **multiple-frame fine-tuning stage**, we train the additional image cross-attention layers in vision-language context module on our dataset, with all other parameters frozen. We first train image cross-attention layers with a single preceding image-caption pair, and then continue with multiple image-caption pairs. Consequently, StoryGen acquires the capability to condition on multiple preceding image-caption pairs and generate image sequences in an auto-regressive manner. Throughout the entire two-stage training, the text cross-attention layers remain frozen, preserving the vision-language alignment inherited from the pre-trained stable diffusion models.

C. Dataset Details

In Section C.1, we present additional details about the data sources of our StorySalon dataset. Subsequently, we show the detailed statistics of our dataset in Section C.2.

C.1. Data Sources

Our StorySalon dataset mainly comprises of two components, *e.g.*, online videos collected from YouTube, and open-source E-books collected from six online libraries. For online videos, we download a large number of videos and corresponding subtitles from YouTube, by querying keywords related to story-telling for children, for instance, *storytime*. For open-source E-books, we collect E-books (partially with corresponding audios available) from six open-source online libraries which are all registered under the Creative Commons 4.0 International Attribution (CC BY 4.0) license. These online libraries have consistently dedicated themselves to assisting children in underdeveloped regions. We extend our appreciation for their ongoing endeavors and contributions. Specifically, these open-source online libraries include:

- **African Storybook.** https://africanstorybook.org;
- **Bloom Library.** https://bloomlibrary.org/;
- **Book Dash.** https://bookdash.org/;
- **Global Digital Library.** https://digitallibrary.io/topic/library-books/;
- **Room to Read.** https://literacycloud.org/;
- **Digital Library of Illustrated Storybooks.** https://storyweaver.org.in/en.

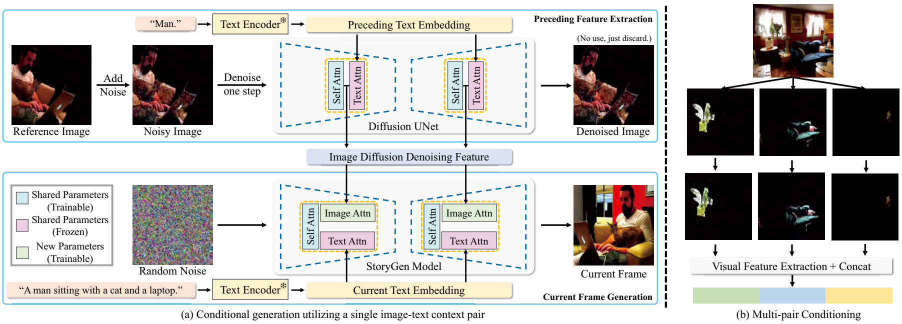


Figure 7. **Architecture Overview**. (a) Our StoryGen model utilizes current text prompt and previous visual-language contexts as conditions to generate an image. (b) In case of multiple image-text context pairs, the multiple features can be directly concatenated to serve as context conditions.

C.2. Dataset Statistics

Our StorySalon dataset comprises a total of $11,280$ storybooks and $159,778$ text-image pairs, with approximately 160K animation-style images, averaging 14 frames per narrative, as demonstrated in Table 1 of the main paper. We divide the dataset into train and test sets following a $9:1$ ratio. Both the video and E-book components are randomly split into train and test sets according to this proportion.

To categorize the characters in storybooks, we use MiniGPT-4 [59] to infer the predominant character category in each image of our dataset, such as *Dog* and *Cat*. Subsequently, we count these categories and exclude those occurring fewer than three times. The distribution of these character categories is depicted in Figure 6. In contrast to preceding datasets featuring fewer than ten characters, our dataset encompasses hundreds of character categories, with rich appearances. Consequently, StorySalon offers the data foundation for training open-ended visual storytelling models.

D. Consistency Ablation on COCO

In Section D.1, we will give a brief illustration on further details about our qualitative consistency ablation experiment on COCO [27]. Subsequently, in Section D.2, we will design a new quantitative ablation experiment on COCO, and present additional results. Finally, in Section D.3, we will provide more visualization results of this consistency ablation on COCO.

D.1. Experiment Details

Motivation. This experiment serves as an ablation study, designed to show StoryGen's ability in utilizing image conditions, and preserving visual details. Specifically, at training time, we train StoryGen on images from COCO datasets, through a self-supervised learning, by reconstructing the input image, with the image caption as text prompt, and cropped objects as reference. At inference time, the model enables to directly generate images with cropped objects as reference.

Experiment Settings. Our experiments on COCO include two scenarios: conditional generation utilizing a single image-text context pair, and alternatively, employing multiple image-text context pairs, as shown in Figure 7.

Training on single image-text context pair. In this case, we randomly select an image-text pair from the COCO dataset. Initially, we extract all objects with their respective masks, collage them together, and then apply data augmentation, such as translation and rotation to create a composite reference image. The categories of the extracted objects serve as the reference text prompt candidates, while the captions of the initial image are employed as the text prompt candidates. If multiple candidates for the text prompt or reference text prompt exist, we will randomly choose one. We expect the model to reconstruct the original image based on these conditions, which enables the context module to learn how to leverage the given reference objects. We fine-tune our StoryGen on the train set of COCO2017 for $5,000$ iterations. We only fine-tune the additional image cross-attention layers, and keep self-attention layers and text cross-attention layers frozen.

Training on multiple image-text context pairs. In this case, instead of collaging the objects from an image together, we individually apply data augmentation to these objects, thereby generating several reference images for a single image-text pair. The categories of these objects are consistently selected as the corresponding reference text prompts. Note that, we employ same diffusion noise scales across these multiple reference images, given that they lack a temporal sequence and hold equal significance. Taking the model pre-trained for single image-text pair, we continue to fine-tune the image cross-attention layers for another $5,000$ iterations, with all other parameters frozen.

D.2. Quantitative Results

To measure consistency, we compute the similarity between the generated image and reference image with a pre-trained DINO [32] model. Specifically, we compare our StoryGen and variants with the original Stable Diffusion model on the validation set of COCO2017. Notably, the only difference between original SDM and our StoryGen here, is that StoryGen is augmented with additional image cross-attention layers trained on COCO train set, and all other parameters remain identical between the models. Thus, we utilize StoryGen to synthesize the original images with the reference objects, reference text prompts, and image captions, in both single and multiple image-text context pair scenarios; and we also exploit the original SDM to generate images with the image captions of current frames. For every image in COCO validation set, ten candidate images are generated. We use PickScore [20] to automatically identify those with better visual quality, then calculate the average DINO feature similarity between the ground truth images and the generated images for both StoryGen and original Stable Diffusion models.

The quantitative results in Table 5 demonstrate that our StoryGen model exhibits significant performance improvement in terms of consistency between the generated image and given references. Both StoryGen and StoryGen (Multiple) outperform the standard SDM. Moreover, compared to utilizing CLIP or BLIP features as visual conditions, our StoryGen model using diffusion-denoising features as condition demonstrates significant performance advantage, showing its effectiveness for retaining visual details from the reference image.

Model	SDM	StoryGen-CLIP	StoryGen-BLIP	StoryGen	StoryGen (Multiple)
Consistency Score ↑	0.4804	0.5103	0.5475	0.7076	**0.7317**

Table 5. Quantitative results on measuring consistency. SDM stands for standard stable diffusion models. StoryGen-CLIP and StoryGen-BLIP represent StoryGen with features extracted by CLIP and BLIP image encoder as context condition, as stated in our ablation study. StoryGen and StoryGen (Multiple) stand for StoryGen in single image-text context pair scenario and multiple image-text context pairs scenario, respectively. Consistency Score stands for the average DINO feature similarity, and the higher score yields better results.

D.3. Qualitative Results

We provide more visualization samples of our consistency ablation on COCO in this section. The results of StoryGen are synthesized with the reference objects, reference text prompts, and image captions, in both single and multiple image-text context pair scenarios. As the original SDM cannot utilize reference images to exploit contextual visual information, the results of SDM are generated with the image captions only. The visualization results are all selected from the generated samples on the validation set of COCO2017.

Single-pair COCO Visualization. The qualitative results of our StoryGen on COCO, in single image-text context pair scenario, are depicted in Figure 9. Compared with the results of SDM, our model demonstrates obvious consistency in its generation results. Despite SDM can generate images that satisfy the text prompts, *i.e.*, good image-text alignment, the generated images are unable to maintain consistency with the reference image.

Multi-pair COCO Visualization. The qualitative results of our StoryGen on COCO, in multiple image-text context pairs scenario, are depicted in Figure 10. Despite multiple context pairs lead to more potential compositions, StoryGen is still able to generate high-quality images with the given reference objects, maintaining strong object consistency and semantic coherence, showing the ability of our architecture to exploit reference images for generation.

E. Broader Impacts

Our storytelling model also has some positive impacts on the industry of creation and education: The widespread application of our visual storytelling model has the potential to inspire creators and artists to create a large number of visual storybooks rich in basic knowledge, which will have a profound impact on children's early education, as demonstrated by related work in psychology. Our work draws inspiration from those open-source online libraries assisting children in underdeveloped regions, can potentially assist artists in creating educational storybooks tailored to these young learners.

F. Limitations

The principal constraint of our storytelling model lies in the selection of stable diffusion models as its foundational architecture. Stable diffusion models are known to grapple with significant issues, notably the generation of images with inaccuracies in limb counts (such as legs, arms, or fingers) and decreased quality in the synthesis of images with multiple objects. Regrettably, our storytelling model inherits these limitations from the stable diffusion model. We anticipate addressing these shortcomings in future research endeavours by considering the adoption of more robust architectures, such as DALL-E 3, SD-XL [37], or consistency models [47].

G. More Experiments

We will provide more quantitative evaluations and visualization samples of our *open-ended visual storytelling* in this section.

G.1. Analysis on multi-frame conditioning

We conduct the multi-frame condition experiments on a test subset with 5,400 samples. As shown in Table 6, we find that: (i) Conditioning on previous frame is critical, (ii) the number of conditioning frames gives similar results. However, we do find differences in qualitative results, so we use the 3 closest frames as conditions in auto-regressive generation. As for frames with less than 3 previous frames, we use all previous frames instead.

	0-frame	1-frame	2-frame	3-frame
FID ↓	40.29	**32.34**	33.27	33.65
CLIP-I ↑	0.6841	0.7368	0.7419	**0.7435**

Table 6. **Comparison of multi-frame conditioning.**

G.2. Multi-object conditioned Story Continuation

As illustrated in Figure 8, benefiting from our diverse StorySalon dataset and training strategy, our StoryGen model also demonstrates excellent performance on multi-object story continuation.

G.3. Story Generation Visualization

We conduct a comparative analysis of results from StoryGen against those generated by SDM [42], Prompt-SDM [42], and StoryGen-Single. As illustrated in Figure 11, Figure 12, and Figure 13, the story generation results of our proposed models demonstrate significantly better consistency in style and character, as well as improved alignment between text and image.

While the results from SDM and Prompt-SDM are visually appealing, they exhibit a lack of stylistic and character consistency. The results of StoryGen-Single also display inconsistency, which proves that our StoryGen effectively utilizes the additional image condition to achieve consistency, rather than naive memorization.

G.4. Story Continuation Visualization

We undertake another comparative analysis between the results from StoryGen and those from StoryDALL-E [31] and AR-LDM [36]. As depicted in Figure 14, Figure 15, and Figure 16, the story continuation results of our proposed models exhibit superior proficiency in maintaining style and character consistency, achieving stronger alignment between story and image, and enhancing image quality. Note that, all characters in the given reference image are unseen in our StorySalon datasets.

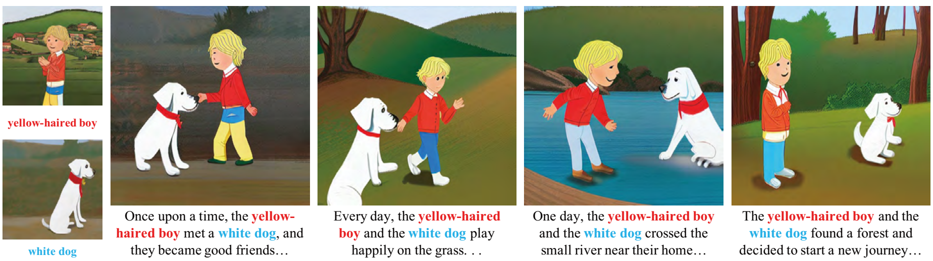


Figure 8. **Example of Multi-object Story Continuation**.

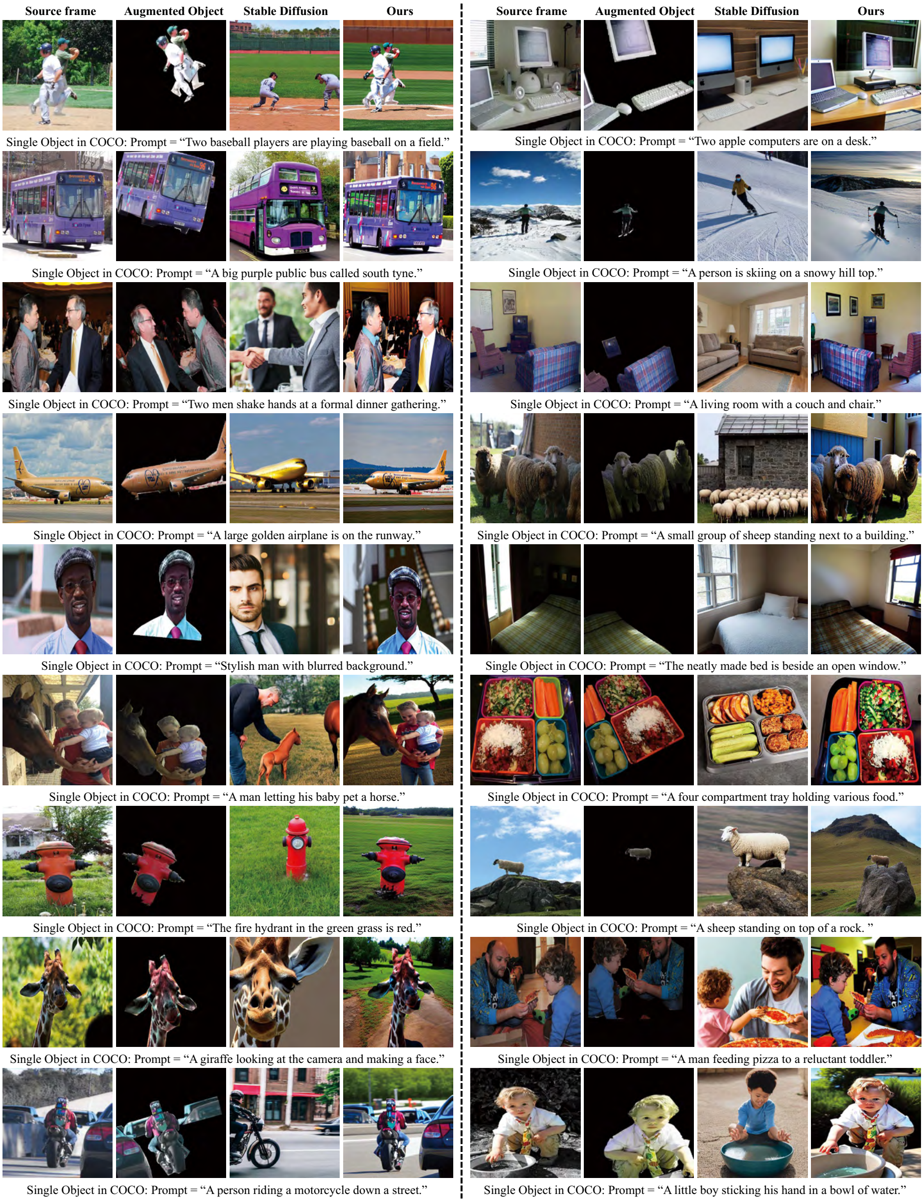
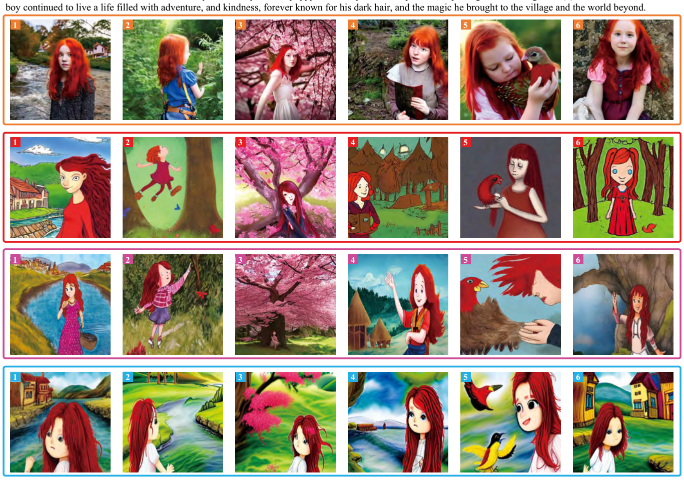


Figure 9. **Visualization results of StoryGen on COCO in single image-text context pair scenario.**

Figure 10. **Visualization results of StoryGen on COCO in multiple image-text context pairs scenario.**

(a) *Open-ended story generation* for: **a story of a {black-haired boy}**: (1) Once upon a time, in a charming village nestled among rolling hills, there lived a curious black-haired boy. His hair was as dark as a moonless night, and his bright eyes sparkled with a thirst for adventure. The boy was known throughout the village for his boundless curiosity and his unquenchable spirit. (2) Every morning, the boy would set out on grand adventures. He would explore the lush forests, climb the towering mountains, and wade through babbling brooks. His black hair would glisten with dew as he ventured deeper into the heart of nature, eager to discover its secrets. (3) One sunny afternoon, the boy stumbled upon a hidden glen bathed in golden sunlight. In the center of the glen stood a majestic oak tree, its branches heavy with ripe, red apples. The boy couldn't resist the temptation and climbed the tree to savor the sweetest apples he had ever tasted. (4) As the seasons changed, the boy's reputation for his adventurous spirit grew. He became the village's most beloved explorer, sharing tales of his journeys and the wonders he had encountered in the wilderness. (5) One chilly winter's night, when the village was blanketed in snow, the boy found a lost, shivering puppy. Without hesitation, he scooped up the tiny creature and nestled it in the warmth of his coat. From that day on, the boy and the puppy were inseparable companions, exploring the world together. (6) And so, the black-haired boy continued to live a life filled with adventure, and kindness, forever known for his dark hair, and the magic he brought to the village and the world beyond.



(b) *Open-ended story generation* for: **a story of a {red-haired girl}**: (1) In a picturesque village nestled by a meandering river, there lived a spirited red-haired girl. Her fiery locks flowed like molten lava, and her bright green eyes sparkled with curiosity. The girl was known throughout the village for her boundless energy and her insatiable appetite for adventure. (2) Every morning, the red-haired girl would set out on grand explorations. She would climb the tallest trees, leap across babbling streams, and chase butterflies through blooming meadows. Her red hair would catch the golden sunlight as she ventured deeper into the beauty of nature, eager to uncover its secrets. (3) One sunny afternoon, the girl stumbled upon a hidden grove, bathed in a warm, dappled light. In the center of the grove stood a magnificent cherry blossom tree, its branches laden with delicate pink blossoms. (4) As the seasons changed, the girl's reputation for her adventurous spirit grew. She became the village's most beloved explorer, sharing stories of her journeys and the wonders she had uncovered in the natural world. (5) One starry night, when the village was bathed in the soft glow of the moon, the girl made a discovery that would forever change her life. She found a small, injured bird with feathers as red as her hair. With gentle care, she nursed the bird back to health, and from that day on, they became inseparable companions, exploring the world together. (6) And so, the red-haired girl continued to live a life filled with curiosity, and compassion, forever known for her fiery hair, and the magic she brought to the village and the world beyond.

Figure 11. **Visualization results of Story Generation**. The images in orange, red, pink, and blue boxes are generated by SDM, Prompt-SDM, StoryGen-Single, StoryGen, respectively.

(a) *Open-ended story generation* for: **a story of a {black girl}**: (1) Once upon a time, in a vibrant forest brimming with emerald leaves and silver streams, there lived a young black girl with hair as curly as the ferns and a smile as bright as the sun. She wore a dress spun from the purple petals of the wildflowers that grew in abundance around her tiny, ivy-covered cottage. (2) Every morning, the girl would step outside, her basket in hand, to gather the most colorful fruits the forest had to offer. She loved the way the morning dew made the world look like it was sprinkled with diamonds, and she'd often dance, twirling amidst the shimmering mist. (3) One afternoon, while exploring deeper into the forest, she discovered a clear, tranquil pond that mirrored the sky so perfectly it seemed as if it held the clouds and the sun within its depths. Here, she would sit and daydream, tossing pebbles to create ripples that carried her thoughts to the stars. (4) As the seasons changed, so did the forest, and the girl witnessed the leaves painting themselves in oranges and reds. She collected these leaves, pressing them into a book, creating a mosaic of memories, each leaf a reminder of the day's joy and wonder. (5) When winter whispered in, the girl found beauty in the silence of the forest covered in snow. She built sculptures of snow, each one more fantastical than the last, giving life to the winter's quiet. (6) Spring brought a chorus of blooms, and the girl, now with a crown of flowers in her hair, joined in the celebration, planting seeds that she had gathered, ensuring the cycle of growth and beauty continued. (7) The girl grew, and with each passing year, she learned the secrets of the trees, the whispers of the wind, and the dance of the seasons. In harmony with the forest, she became its guardian, a symbol of the enduring dance of life.

(b) *Open-ended story generation* for: **a story of a {black wolf}**: (1) In the depths of a snowy wilderness, there was a solitary black wolf whose coat shimmered against the stark white of the frozen landscape. The wolf had a majestic presence, with eyes that glinted like the first stars of the evening sky. His powerful paws left a trail of footprints as the only evidence of his passage through the thick blanket of snow. (2) Each day, the black wolf would climb to the peak of a great mountain, letting out a deep, resounding howl that echoed through the valleys. The sound would carry for miles, a song of strength and solitude that resonated with the whispering pines and the crisp, winter air. (3) With the arrival of spring, the black wolf watched as the snow melted, revealing a carpet of wildflowers. He roamed through the blossoming terrain, his black fur juxtaposed against the riot of colors, a guardian of the waking world. (4) Summer brought with it an abundance of life, and the black wolf would spend his nights chasing the golden orb of the moon, racing through forests where the fireflies lit his path, a living embodiment of the night's spirit. (5) When autumn arrived, the wolf found joy in the crunch of the leaves beneath his paws. The forest was a cascade of oranges, reds, and yellows, and he moved through it like a shadow, part of the tapestry of the changing seasons. (6) The wolf, now older and wiser, took to resting by a tranquil lake during the quiet afternoons, reflecting on the cycles of nature. The calm waters mirrored his noble figure and the fiery sunsets, a scene of perfect peace and solitude. (7) As years passed, the black wolf became a legend of the wilderness, a solitary figure that moved with the grace of the seasons. His story was written in the earth, a tale of harmony with the world, a spirit both wild and free.

Figure 12. **Visualization results of Story Generation**. The images in orange, red, pink, and blue boxes are generated by SDM, Prompt-SDM, StoryGen-Single, StoryGen, respectively.

(a) *Open-ended story generation* for: **a story of a {princess}**: (1) Once upon a time, in a land ruled by the rhythm of seasons, there lived a beautiful princess who was in love with the stars. Her castle was adorned with towers that reached towards the heavens, and every night, she would climb to the highest balcony to converse with the twinkling dots scattered across the night sky. (2) One evening, as a shooting star sliced through the darkness, the princess made a wish. She longed to visit the stars and dance among them, to learn their ancient secrets and to see the world from their eternal vantage point. (3) The next morning, the princess discovered a mysterious, silvery seed had fallen from the sky and landed in her royal garden. She planted the seed in the earth, watering it with water from the enchanted spring that ran through the castle grounds. (4) To the princess's wonder, the seed grew rapidly, unfurling into a magnificent vine with leaves that shimmered like stars and flowers that glowed with the luminescence of the moon. The vine spiraled up one of the castle towers, beckoning the princess to climb. (5) With her heart pounding with excitement, the princess began to ascend the vine. As she climbed higher, the air grew thinner, and the sky seemed to embrace her. The vine ended at the threshold of the stars. (6) The princess stepped off the vine and found herself walking on a path of stardust. Each step she took was lighter than the last, and she danced among the stars, just as she had wished. They whispered to her in the language of light, sharing stories of distant worlds and the dance of the cosmos. (7) As dawn approached, the princess knew it was time to return to her own world. She glided back down the stardust path, down the vine, and stepped onto her balcony as the first rays of sunlight kissed the horizon. Her heart was full of starlight, and her eyes shone with the reflection of her night among the stars.

(b) *Open-ended story generation* for: **a story of a {prince}**: (1) Once upon a time, in a kingdom draped in emerald valleys and crystal lakes, there lived a young prince who was fascinated by the secrets of nature. The prince spent his days wandering the expansive gardens of the palace. He was particularly enchanted by a single rosebush that grew at the edge of the garden. (2) One day, as the prince watched the rosebush, he noticed that it began to wilt despite his careful attention. Worried, he consulted the ancient library in his castle, scouring old texts and botany books for a cure. He learned of a rare water source, hidden deep in the forest, whose waters were said to rejuvenate any plant. (3) Determined to save his cherished rosebush, the prince set out alone into the forest. He traversed through thickets and over streams, guided by the chirps of birds and the rustling of leaves. His journey was long, and the forest seemed to whisper secrets as he passed. (4) Finally, after several days, he arrived at a glade where the sunlight shimmered down like warm gold. There, at the center, was a spring that sparkled with water so clear it looked like liquid diamonds. The prince filled his flask with the water, feeling its coolness and vitality. (5) On his way back to the castle, the prince encountered a variety of creatures. A wise owl nodded at him from a tree branch, a family of rabbits watched curiously from the bushes, and a graceful deer bowed its head as he passed. The prince realized he was not alone in his quest; the forest itself was guiding and protecting him. (6) Upon his return, the prince immediately watered the ailing rosebush with the magical spring water. Overnight, the rosebush regained its vigor, its petals unfolding with colors so vivid and fragrant that they seemed to glow in the moonlight. (7) The prince's dedication to his rosebush became a legend in the kingdom. He went on to create the most magnificent garden, filled with plants and flowers from all over the world, each thriving under his care. The prince became known not just as a ruler, but as a guardian of nature, with a garden that was a testament to his love for all living things.

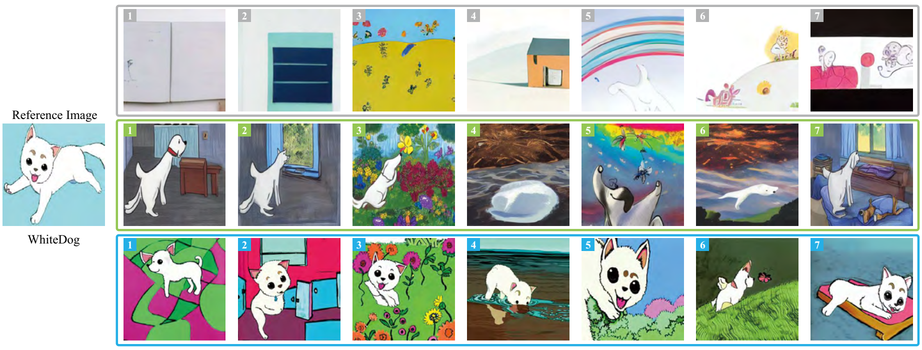
Figure 13. **Visualization results of Story Generation**. The images in orange, red, purple, and blue boxes are generated by SDM, Prompt-SDM, StoryGen-Single, StoryGen, respectively.

(a) *Open-ended story continuation* for: **a story of a {a brown-haired girl}**: (1) Once upon a time, in a village where the night sky always shimmered with stars, there lived a brown-haired girl known for her enchanting hats. She wore a unique hat every day, each adorned with symbols of dreams and hopes, like flowers and stars. This girl had a special gift; she could weave the essence of the night sky into her hats, making them glow with a soft, comforting light. (2) One evening, as the crescent moon hung low, the girl decided to craft a hat that would capture the beauty of the night forever. She worked under the starlight, her fingers dancing with threads that glimmered like constellations. She poured her joy and the whispers of the night wind into her creation, a hat that seemed to hold the entire galaxy within its folds. (3) When she placed the hat upon her head, the village was bathed in a celestial glow. The villagers looked out from their homes in awe, as gentle light cascaded down the cobblestone streets. They felt peace and wonder, and the night seemed less dark, less fearful, as if the girl's hat had captured the heart of the night itself. (4) The girl realized that her gift was more than just crafting hats; she could bring hope to her people. She began to wander the streets each night, her hat glowing ever so brightly, a beacon for those who were lost or in need of comfort. She didn't need words; her presence was enough to lift spirits. (5) As the seasons changed, so did her hats. The girl and her hats became a symbol of the village's identity, a reminder that beauty and magic could be found in the simplest of things. (6) Years passed, and the girl grew older, but her legacy remained timeless. The hats she had crafted were passed down through generations, each one a treasured heirloom that continued to glow with a piece of the night. (7) And so, the brown-haired girl with her magical hats lived on in the hearts of the people, a legend woven into the fabric of the village. They would look up at the stars and remember the girl who walked with the night, who showed them that even in darkness, there is light to be found.

(b) *Open-ended story continuation* for: **a story of a {a red-haired man}**: (1) In a land where colors held magic, there lived a red-haired man who was the guardian of the Flame of Creation. His hair, the color of burning embers, was a symbol of the fire that he protected — a fire that had the power to ignite inspiration and passion in the hearts of the people. (2) Despite the grandeur of his task, the red-haired man felt a growing sense of solitude. His life was an endless cycle of tending to the flame. He longed to see the world beyond his hall, to witness the wonders his flame brought to life. Yet, he dared not leave, for the flame required constant care, and there was no one else to shoulder the burden. (3) One evening, as shadows danced along the walls of his hall, the red-haired man noticed that the flame flickered unusually. It whispered to him of a possibility he had never considered. The flame could divide, sharing a spark that could be carried out into the world without letting the original fire die. (4) With a mixture of trepidation and excitement, the red-haired man fashioned a lantern from the hall's curtains and an old chair. He captured the wayward spark in the lantern, ensuring that the main flame continued to burn strong. Now he held a piece of the Flame of Creation, a portable spark that would allow him to venture into the world. (5) As the red-haired man stepped outside, the lantern's glow seemed to brighten the world in hues he had only imagined. Wherever he walked, life sprang forth: flowers bloomed, trees bore fruit, and the night sky shimmered with new stars. (6) In time, the red-haired man realized that while the flame's magic was powerful, it was the actions of the people that truly created change. The man's solitude was replaced by a sense of connection to the world, fulfilled by the knowledge that the flame's inspiration was at the heart of all creativity. (7) The guardian of the Flame of Creation returned to his hall, understanding now that his duty was not only to protect the flame but also to share its gift. He would continue to venture out into the world, carrying the spark that brought dreams to life.

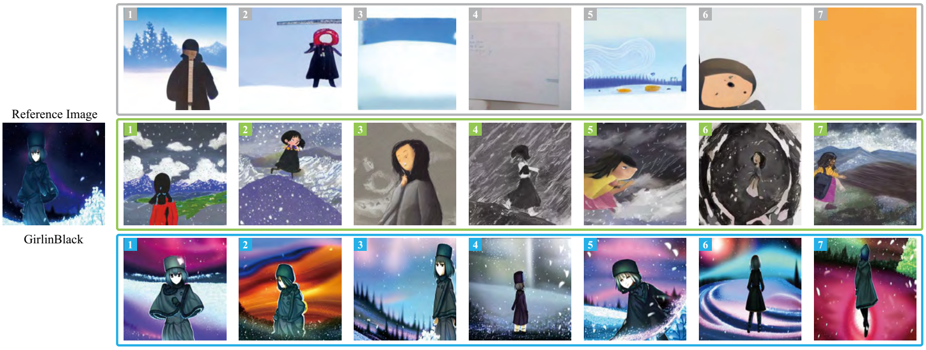
Figure 14. **Visualization results of Story Continuation**. The images in gray, green, and blue boxes are generated by StoryDALL-E, AR-LDM, StoryGen, respectively.

(a) *Open-ended story continuation* for: **a story of a {a white cat}**: (1) Once upon a time, in a cozy little house bathed in sunshine, there lived a small white cat with big, bright eyes that sparkled like gems. Every morning, the cat would find a sunbeam streaming through the window, jump onto the sill, and bask in the golden glow, feeling the warmth on its soft fur. (2) One afternoon, while the cat was lounging in its favorite spot, a gentle breeze carried in the sweet scent of flowers from the garden. The cat, curious and playful, leapt off the sill and followed the fragrance, finding itself in the midst of a colorful flower bed, with petals dancing in the wind. (3) In the garden, the cat discovered a little fountain, its water sparkling in the sunlight. With a paw, the cat gently tapped the surface, causing ripples to spread across the water. Fascinated, the cat watched the ripples, feeling as if the water was playing a game with it. (4) As the day turned to dusk, and the sky painted itself in hues of orange and pink, the cat climbed atop a wooden fence, gazing at the setting sun. There, it sat, silhouette outlined against the sky, feeling the cool evening breeze ruffling its fur. (5) When the stars began to twinkle in the night sky, the white cat returned inside, finding a cozy spot in front of the fireplace. The flickering flames cast warm, dancing shadows, and the cat curled up, closing its eyes, the warmth lulling it into a peaceful sleep. (6) Throughout the night, in its dreams, the cat journeyed to fantastical places, flying among the clouds and walking on the moon, each dream more vivid and wondrous than the last, until the soft rays of dawn signaled a new day. (7) And so the days went by, with the little white cat finding joy in the simplest things: a ray of sunshine, the scent of flowers, the playfulness of water, the serenity of sunsets, the warmth of fire, and the boundless adventures found in dreams.

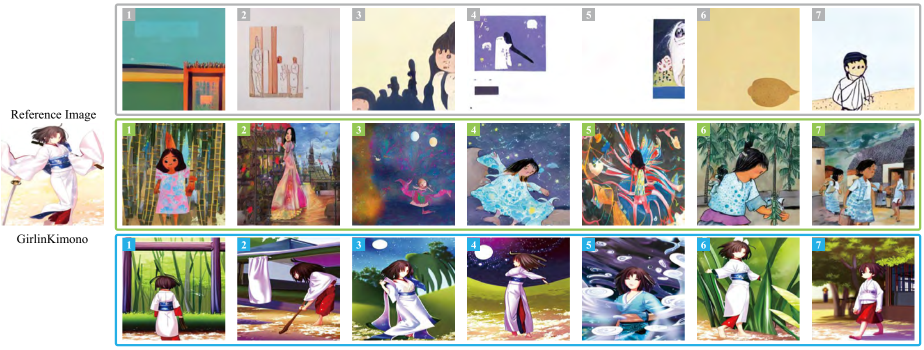


(b) *Open-ended story continuation* for: **a story of a {a white dog}**: (1) Once upon a time in a colorful world, there was a small white dog with playful spots over its fur. This curious little dog loved to explore every corner of its bright and cheerful home. It would scamper around with boundless energy, its tongue lolling out in a happy pant, and its tail wagging like a fluffy pendulum. (2) One sunny morning, the white dog found a mysterious blue butterfly fluttering near the window. It was unlike any butterfly it had seen before. The dog tilted its head, eyes wide with wonder, and decided to follow the butterfly wherever it might lead. With a joyful bounce, the dog leapt towards the fluttering creature, embarking on a new adventure. (3) The butterfly led the dog through the garden where flowers bloomed in every shade imaginable. The dog marveled at the sights, sniffing the fragrant air filled with the scent of fresh blooms. It chased the butterfly around the garden, over the green grass, under the flowering bushes, and around the stone path. (4) As they journeyed together, the white dog and the butterfly came across a clear, bubbling stream. The dog had never seen water so clear, and it watched in amazement as the sunlight danced upon the water's surface. Feeling adventurous, the dog dipped its paws into the cool stream, sending ripples across the water. (5) Suddenly, the butterfly soared up high, with the white dog gazing after it. The dog noticed a rainbow arching across the sky, its colors reflecting the vibrant world below. The dog felt a surge of joy and decided to race along the stream, as if it were racing the colors of the rainbow. (6) The day turned to evening, and the sky painted itself with the hues of sunset. The white dog found itself on a hill, watching the sun dip below the horizon. The butterfly landed gently on the dog's nose, as if to say goodbye. The dog sat peacefully, feeling grateful for the day's journey and the beauty it had seen. (7) As the stars began to twinkle in the night sky, the white dog returned home, its heart full of the day's wonders. It curled up in its cozy bed, dreaming of the gardens, the stream, the rainbow, and the butterfly. The white dog knew that tomorrow was another day for adventure, but for now, it rested, wrapped in the warmth of its memories.

Figure 15. **Visualization results of Story Continuation**. The images in gray, green, and blue boxes are generated by StoryDALL-E, AR-LDM, StoryGen, respectively.



(a) *Open-ended story continuation* for: **a story of a {a girl in black}**: (1) In a realm where winter reigned eternal, the girl in the black coat walked alone, her presence the only warmth in the icy world. She was the Whisperer of the Wind, a gentle spirit who could speak to the cold breezes and soothe their icy fury. (2) Each morning, as the sun struggled to pierce the wintry gloom, she would climb the highest hill and listen to the stories the wind told. Tales of distant lands, of sun-soaked shores, and of children playing under the warmth of a softer sun. (3) One day, the wind spoke of a coming storm, a tempest that could bury her world in snow and silence forever. The Whisperer knew she had to calm the storm's heart, or springtime's hope would never return to her frozen home. (4) With courage in her step, she walked into the heart of the storm, her black coat fluttering like a banner of night. She spoke to the blizzard, her voice a melody that rivaled the storm's howl, a plea for peace and tranquility. (5) The storm, taken aback by the girl's bravery and the sweetness of her voice, began to lessen its wrath. Snowflakes slowed their dance, and the icy gales held their breath, listening to the Whisperer's song. (6) As the storm's heart calmed, the snow ceased, and the winds carried the girl's song across the land. Wherever her voice reached, ice melted, revealing the first glimpses of the soil beneath—a promise of the spring to come. (7) The girl in the black coat became the legend of the winter world, the one who conversed with the wind and turned the fiercest of storms into a peaceful slumber. And though she wandered alone, her song of warmth and the hope of spring lived on in the hearts of all those who yearned for the thaw.

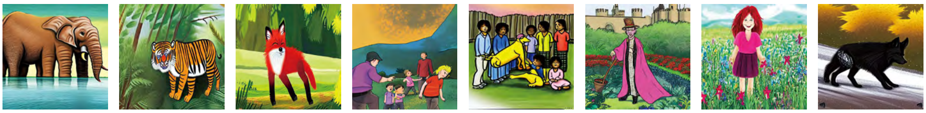


(b) *Open-ended story continuation* for: **a story of a {a girl in kimono}**: (1) In the heart of a dense bamboo forest, there stood a solitary shrine, its red torii gate a stark contrast against the sea of green. It was here that the girl in the white kimono found solace and purpose. She was the shrine's keeper, tasked with ensuring that the balance between the human realm and the spirits was maintained. (2) Each morning, with the first light of dawn casting a soft glow over the land, the girl would sweep the shrine's grounds with a handmade broom, her white kimono glimmering in the sun's gentle rays. She took great care in her work, for she knew that cleanliness was a gesture of respect to the spirits. (3) It was during the night of the full moon that the girl's responsibilities took on a magical turn. The air would thrum with energy, and the border between worlds grew thin. On these nights, the girl would perform a sacred dance, a ritual to honor the spirits and ensure their goodwill towards the villagers. (4) With each precise step and wave of her sleeve, the girl's dance would draw luminescent orbs from the moonlit sky, each one a spirit coming to witness her devotion. The orbs hovered around her, pulsating with the serene energy of the unseen world. (5) As the dance reached its crescendo, the spirits would begin to swirl around the girl, creating a vortex of otherworldly light. The girl's connection to the spirits was strongest at this moment, and she would whisper her wishes for the village's safety and prosperity. (6) When the dance ended and the first light of dawn approached, the spirits would depart, leaving behind a trail of sparkling dew on the bamboo leaves. This dew was said to have healing properties, and the girl would collect it carefully, a gift from the spirits to the villagers. (7) The villagers rarely saw the mysterious events that took place at the shrine, but they felt the peace and prosperity that the girl's rituals brought to their lives. The girl in the white kimono remained ever vigilant, a silent guardian whose dance with the spirits kept their world in harmony.

Figure 16. **Visualization results of Story Continuation**. The images in gray, green, and blue boxes are generated by StoryDALL-E, AR-LDM, StoryGen, respectively.

G.5. Failure Case Visualization

Figure 17 presents some instances where StoryGen did not perform optimally. These failure cases primarily stem from the inherent limitations of SDM. Figures (a), (b), and (c) illustrate occurrences where StoryGen is prone to generating images with limb count inaccuracies, such as incorrect numbers of legs. Figures (d) and (e) show scenarios where the generation of multiple objects results in each object being of subpar quality. Figures (e), (f), and (g) depict instances of StoryGen producing low-quality human faces. Regarding Figure (h), despite the visual prompt being "*A black wolf walking through a forest with autumn leaves falling*", the generated image erroneously includes snowfall, due to the winter setting of the reference image. This discrepancy arises from the conflict between the image and text conditions.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure 17. **Some failure cases of StoryGen.**

Learning or Self-aligning? Rethinking Instruction Fine-tuning

Mengjie Ren[1,3*], **Boxi Cao**[1,3], **Hongyu Lin**[1†], **Cao Liu**[4], **Xianpei Han**[1,2,5]
Ke Zeng[4], **Guanglu Wan**[4], **Xunliang Cai**[4], **Le Sun**[1,2,5]
[1]Chinese Information Processing Laboratory [2]State Key Laboratory of Computer Science
Institute of Software, Chinese Academy of Sciences
[3]University of Chinese Academy of Sciences
[4]Meituan
[5]Key Laboratory of System Software, Chinese Academy of Sciences
{renmengjie2021,boxi2020,hongyu,xianpei,sunle}@iscas.ac.cn
{liucao,zengke02,wanguanglu,caixunliang}@meituan.com

Abstract

Instruction Fine-tuning (IFT) is a crucial phase in building large language models (LLMs). Previous works mainly focus on the IFT's role in the transfer of behavioral norms and the learning of additional world knowledge. However, the understanding of the underlying mechanisms of IFT remains significantly limited. In this paper, we design a knowledge intervention framework to decouple the potential underlying factors of IFT, thereby enabling individual analysis of different factors. Surprisingly, our experiments reveal that attempting to learn additional world knowledge through IFT often struggles to yield positive impacts and can even lead to markedly negative effects. Further, we discover that maintaining internal knowledge consistency before and after IFT is a critical factor for achieving successful IFT. Our findings reveal the underlying mechanisms of IFT and provide robust support for some very recent and potential future works. We release our experimental dataset and codes to facilitate future work [1].

1 Introduction

The advent and evolution of large language models (LLMs) have marked a significant milestone in natural language processing (NLP) (Brown et al., 2020; Touvron et al., 2023b,a). As one of the core steps in the construction of LLMs, instruction fine-tuning (IFT) employs supervised instruction-response pairs to fine-tune LLMs, thereby facilitating the transformation of LLM from a continuous writing model to a question-answering agent (Chung et al., 2024; Iyer et al., 2022; Jang et al., 2023).

[*] Work was partially done during Ren's internship at Meituan.
[†]Corresponding Author
[1]https://github.com/renmengjie7/Self-Aligning

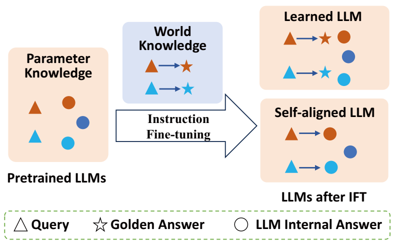


Figure 1: Two potential mechanisms for instruction fine-tuning. 1) Learning, which injects world knowledge in IFT data into LLMs; 2) Self-aligning, which aligns queries with knowledge already in LLMs with similar behavioral norms. Elements with the same color are related.

Despite the crucial role of IFT in the construction of LLMs, there is a significant lack of in-depth research on the mechanisms by which IFT operates. In traditional machine learning, supervised learning aims to fit models to specific tasks and data distributions (Goodfellow et al., 2016; Bishop, 2006), whereas the impact of IFT on LLMs is markedly different. As shown in Figure 1, on one hand, one of the most apparent effects of IFT is its ability to align the output of LLMs more closely with the latent behavioral norms contained within the IFT data, thereby enabling more effective parameter knowledge expression (Zhou et al., 2024; Chen et al., 2023a). On the other hand, many existing studies (Li et al., 2023; Cui et al., 2023; Chen et al., 2023b) aim to facilitate domain-specific adaptation of LLMs through IFT, by injecting the world knowledge contained in IFT data into LLMs.

Unfortunately, both the transfer of behavioral norms and the enhancement of domain knowledge are closely coupled with the corpus applied in IFT, rendering the analysis of IFT's true effects exceedingly challenging. Due to the interconnection of these two effects, it is challenging to discern whether the benefits derived are due to the promotion of more effective expression of parameter knowledge or the injection of additional world knowledge. The coupling between the above two factors, along with a lack of in-depth analysis of the IFT's mechanism, hampers our comprehension of the effectiveness of IFT. This limitation hinders our ability to develop robust strategies for IFT data construction, model training, and model evaluation due to insufficient theoretical support. Therefore, a thorough and comprehensive analysis of the underlying core factors that drive the effectiveness of IFT is crucial for achieving more effective IFT.

To this end, this paper designs a knowledge intervention framework for analyzing the underlying mechanisms of IFT. The main idea of our framework is to control the consistency between the knowledge in IFT data and the existing parameter knowledge of LLM, in order to decouple the injection of domain knowledge from the transfer of behavioral norms during IFT. This allows for a separate analysis of the roles of these two crucial factors. Specifically, we first employ in-context learning (ICL) (Dong et al., 2022; Brown et al., 2020) to probe the internal parameter knowledge of LLMs. Building on this, we intervene in the composition of existing parameter knowledge and the newly introduced world knowledge within IFT data and then observe the differences in model behavior after IFT using different intervention groups. Based on the framework, we conduct an in-depth analysis to answer the following two critical research questions (RQ):

- **RQ1:** *How does the world knowledge within IFT data affect LLMs?*
- **RQ2:** *What is the underlying cause of the above impact?*

For RQ1, we initially discover that significant discrepancies between the world knowledge contained in IFT data and the existing parameter knowledge within LLM can substantially undermine the model's abilities. Performance derived from a set of IFT data that contains incorrect world knowledge but aligns with the model's parameter knowledge is significantly better than that from a set containing correct world knowledge but inconsistent with the model's internal parameter knowledge. To dive into this phenomenon, we explicitly supply LLMs with the world knowledge necessary for answering the instruction, integrated into the context. This strategy allows the model to focus on transforming the output behavioral norms instead of jointly learning the inconsistent world knowledge. We discover that the detrimental effects caused by the inconsistency between parameter knowledge and world knowledge can be significantly mitigated by explicitly providing such self-contained IFT data points. These two findings indicate that attempting to introduce world knowledge through IFT that is inconsistent with the model's parameter knowledge can severely undermine the model, which suggests that the injection of world knowledge does not lie at the center of a successful IFT.

For RQ2, we further analyze the model performance under varying degrees of consistency between the knowledge contained in IFT data and the parameter knowledge in the original LLM. We find that while the consistency between the two has a significant impact on model performance, a higher degree of consistency does not necessarily correlate with better model performance. However, our further research reveals a strong correlation between the model's ultimate performance after IFT and the consistency of the model's internal knowledge before and after IFT. That is to say, for the model after IFT, if its responses are more consistent with the responses produced by the original model from in-context learning probing, then the performance of the model after IFT is also better. The validity of the finding is independent of whether the test data belongs to the same domain as the training data, and is also unrelated to the original performance of the base LLM. This implies that the phenomenon is solely influenced by the consistency of the knowledge before and after IFT. Furthermore, we discover that using IFT data that is either too consistent or too inconsistent with the original parameter knowledge can lead to a divergence in the model's internal knowledge before and after IFT, thereby resulting in a decline in performance.

Our experiments reveal the fundamental role of IFT in the construction of LLMs. Essentially, IFT is not a supervised, domain-specific **learning** process, but a process of **self-aligning** instruction with existing internal knowledge of LLMs that can be obtained through few-shot in-context learning probing. Our findings not only provide robust theoretical support for the very recently emerged research on self-alignment (Sun et al., 2024) and super-alignment (Burns et al., 2023), etc., but also shed light on the future direction of data construction, model training, and model evaluation for IFT.

2 Related Work

By observing output token distribution shift of models before and after IFT, Lin et al. (2024) found that most shifts occur with stylistic tokens, strongly supporting the superficial alignment hypothesis (Zhou et al., 2024), false promise (Gudibande et al., 2023) and related works on IFT data construction (Chen et al., 2023a; Shen, 2024) and proxy-guided decoding (Liu et al., 2024). While providing intuitive insights, they fall short of providing a comprehensive analysis of IFT's underlying mechanisms.

Meanwhile, recent efforts have focused on achieving automated alignment, such as self-instruction-tuning (Sun et al., 2024; Guo et al., 2024), self-rewarding (Yuan et al., 2024) and super-alignment (Burns et al., 2023). Despite repeated validations of their effectiveness, there remains limited understanding of their success.

3 Knowledge Intervention Framework

During the process of IFT, the potential transfer of behavioral norms and the injection of world knowledge are coupled together. Consequently, prior research on IFT has struggled to distinguish the relative effects of these two. To further investigate the underlying mechanisms of IFT, this paper designs a knowledge intervention framework to decouple these two factors. The main idea behind our framework is to control the association between knowledge in IFT data and the existing parameter knowledge in LLM, thereby managing the degree of potential world knowledge that would be injected during IFT. Through this, we can decouple the injection of world knowledge and the transfer of behavioral norms, by observing the effects of IFT at varying degrees of world knowledge injection.

Specifically, we select four multi-choice datasets from different domains. For each question in each dataset, we employ in-context learning to probe the internal parameter knowledge for each base LLM. Then, we construct multiple instruction datasets by adjusting the consistency of the knowledge within IFT data and model parameter knowledge for each base model. Finally, we analyze the underlying impact of different degrees of world knowledge injection on IFT by fine-tuning LLMs under different settings and examining their performance on the homogeneous, in-domain, and out-of-domain test sets. In the following, we will introduce our knowledge intervention framework in detail.

3.1 Domain IFT Corpus Setup

In order to facilitate more efficient knowledge consistency identification and IFT evaluation, we select all IFT corpus in the form of multiple choice from four domains: medicine, history, engineering, and jurisprudence. For medicine, we craft a dataset with 20,000 training, 2,206 testing, and 10 development instances by filtering MedMCQA (Pal et al., 2022) for entries with explanations and one correct answer, and then applying random sampling. For the other three domains, we procure the relevant items from Xiezhi Benchmark (Gu et al., 2024) and held 10 for development and 250 for test [2].

To make a comprehensive evaluation, for each domain's IFT, we construct three types of test sets: **1) homogeneous test set (HOMO)**, which is held out from the same multiple-choice dataset of the domain; **2) in-domain test set (ID)**, including data from MMLU (Hendrycks et al., 2021) that belong to the domain; **3) an out-of-domain test set (OOD)**, comprising instances in MMLU that are from distinct domains. By observing the accuracy performance differences across the three types of test sets, we aim to examine the impact of IFT on various scenarios. Please refer to the Appendix A for more details about our data processing.

3.2 Parameter Knowledge Probing via Few-shot In-context Learning

Our knowledge intervention framework relies on effectively detecting the parameter knowledge of pre-trained LLMs. To this end, this paper leverages few-shot in-context learning (Dong et al., 2022; Brown et al., 2020), which is a widely-used approach for probing the abilities and internal knowledge of pre-trained LLMs (Zhang et al., 2023; Wan et al., 2024), to identify the parameter knowledge of our base LLMs. Specifically, we utilize in-context learning to probe the base model's response to each data item in domain multi-choice dataset and regard the response as the model's parameter knowledge for this question.

[2] The availability of multiple-choice datasets tailored to specific domains is extremely limited. Consequently, from this benchmark, we select the three domains with the largest volume of data.

3.3 Construction of Instruction Data

Upon probing the internal knowledge of pre-trained LLMs, we build instances based on the consistency between world knowledge contained within the domain data and parameter knowledge, thereby constructing different IFT datasets. Specifically, for each domain and base model, we construct IFT data under three settings, including:

- **Harmonious setting**, which consists of data where the embedded world knowledge is consistent with model parameter knowledge. This means the pre-trained LLM can answer correctly under in-context learning. In the learning process under this setting, the model only needs to transfer behavioral norms, without the need to acquire additional world knowledge due to the above consistency.

- **Incompatible setting**, which comprises instances where the pre-trained LLM cannot correctly answer under in-context learning. Due to the complete inconsistency, the model needs to learn both the behavioral norms and the world knowledge during its training phase.

- **Self-aligning setting**, which consists of data in which the queries are exactly the same as those in the incompatible set, but we modify the answers corresponding to each query to match the pre-trained LLM's internal knowledge. Therefore, under this setting, all responses are incorrect, and the model will not learn any additional world knowledge.

To ensure a fair comparison, we maintain the same size across the three groups of data. Meanwhile, to prevent the potential collapse during model training due to the exclusive use of multiple-choice questions, we generate an explanation for the answer and incorporate an equal proportion of general instruction data sampled from alpaca-gpt4-en (Peng et al., 2023), thereby ensuring a more stable and real IFT.

3.4 Experiment Setup

Base Model We use LLaMA-2-7B, LLaMA-2-13B, LLaMA-2-70B (Touvron et al., 2023b), and Mistral-7B (Jiang et al., 2023) as the base models of our experiments.

Training Details We only calculate loss on outputs, setting epoch to 3, learning rate to $2e^{-5}$ and batch size to 256. For Mistral-7B, we set learning rate to $1e^{-5}$ [3]. We use DeepSpeed ZeRO3 (Rasley et al., 2020) for LLaMA-2-70B and FSDP (Zhao et al., 2023) for the other three. All experiments are implemented on Nvidia A100-80GB GPUs.

4 Exp-I: Does Learning Domain-specific World Knowledge Matter for IFT?

In our first group of experiments, we would like to examine how the additional world knowledge in the IFT corpus affects LLMs. To this end, we conduct experiments under three settings including harmonious, incompatible, and self-aligning. By observing the performance discrepancies among these settings, we analyze the effects of injecting world knowledge into the IFT.

Finding 1. *When encompassing correct world knowledge, IFT data congruent with model parameter knowledge can lead to superior IFT outcomes.*

To show this, we compare the experimental results under two settings: harmonious and incompatible. Note that the datasets of both two settings contain correct world knowledge, meaning each query-response pair aligns with correct knowledge. Therefore, the core distinction between these two settings lies in whether the entailed world knowledge is consistent with the parameter knowledge of the LLM. Specifically, training in the harmonious setting only requires learning behavioral norms without the need for learning any additional knowledge, while training in the incompatible setting requires learning both.

Table 1 reveals that models fine-tuned under the harmonious setting outperform those fine-tuned under the incompatible setting across homogeneous, in-domain, and out-of-domain evaluations[4]. Specifically, the harmonious setting yields mean performance gains of 11.27%, 14.58%, and 14.57% over the incompatible setting for homogeneous, in-domain, and out-of-domain tests, respectively. The results indicate that utilizing IFT data, which is consistent with model parameter knowledge and does not inject any additional domain-specific world knowledge, yields superior fine-tuned models. This

[3]Training loss of Mistral-7B using learning rate $2e^{-5}$ does not converge and even spikes.

[4]Regarding the evaluations of HOMO, ID, and OOD: The homogeneous, in-domain, and out-of-domain evaluations for each domain cover different subsets and the level of difficulty varies across these subsets. Therefore, the absolute performance differences between HOMO, ID, and OOD are not directly comparable.

Eval	Medicine			History			Engineering			Jurisprudence		
	HAR	INC	SELF	HAR	INC	SELF	HAR	INC	SELF	HAR	INC	SELF
LLaMA-2-7B												
HOMO	$\mathbf{40.22}_{11.77\uparrow}$	28.45	$\underline{37.00}_{8.55\uparrow}$	$\mathbf{38.80}_{9.20\uparrow}$	29.60	$\underline{33.60}_{4.00\uparrow}$	$\mathbf{48.40}_{16.00\uparrow}$	32.40	$\underline{32.80}_{0.40\uparrow}$	$\mathbf{37.60}_{3.60\uparrow}$	$\underline{34.00}$	$33.20_{0.80\downarrow}$
ID	$\underline{39.82}_{2.56\uparrow}$	37.26	$\mathbf{41.46}_{4.20\uparrow}$	$\mathbf{54.30}_{23.22\uparrow}$	31.08	$\underline{46.02}_{14.94\uparrow}$	$\mathbf{42.07}_{11.04\uparrow}$	$\underline{31.03}$	$26.21_{4.82\downarrow}$	$\mathbf{38.86}_{3.16\uparrow}$	35.70	$\underline{36.34}_{0.64\uparrow}$
OOD	$\underline{39.97}_{3.22\uparrow}$	36.75	$\mathbf{40.94}_{4.19\uparrow}$	$\mathbf{39.64}_{8.95\uparrow}$	30.69	$\underline{37.22}_{6.53\uparrow}$	$\mathbf{40.38}_{12.12\uparrow}$	28.26	$\underline{29.17}_{0.91\uparrow}$	$\mathbf{38.49}_{3.93\uparrow}$	34.56	$\underline{34.88}_{0.32\uparrow}$
LLaMA-2-13B												
HOMO	$\mathbf{40.83}_{4.78\uparrow}$	$\underline{36.05}$	$34.41_{1.64\downarrow}$	$\mathbf{48.40}_{16.00\uparrow}$	32.40	$\underline{43.60}_{11.20\uparrow}$	$\mathbf{58.00}_{20.80\uparrow}$	37.20	$\underline{55.20}_{18.00\uparrow}$	$\mathbf{44.00}_{11.60\uparrow}$	32.40	$\underline{37.60}_{5.20\uparrow}$
ID	$\mathbf{55.43}_{20.37\uparrow}$	35.06	$\underline{52.13}_{17.07\uparrow}$	$\mathbf{68.28}_{22.15\uparrow}$	46.13	$\underline{64.09}_{17.96\uparrow}$	$\mathbf{45.52}_{15.86\uparrow}$	29.66	$\underline{40.00}_{10.34\uparrow}$	$\mathbf{54.77}_{16.22\uparrow}$	38.55	$\underline{52.77}_{14.22\uparrow}$
OOD	$\mathbf{54.21}_{18.44\uparrow}$	35.77	$\underline{50.98}_{15.21\uparrow}$	$\mathbf{51.30}_{13.32\uparrow}$	37.98	$\underline{49.06}_{11.08\uparrow}$	$\mathbf{52.15}_{16.21\uparrow}$	35.94	$\underline{51.12}_{15.18\uparrow}$	$\mathbf{50.83}_{11.57\uparrow}$	39.26	$\underline{48.27}_{9.01\uparrow}$
LLaMA-2-70B												
HOMO	$\mathbf{47.95}_{5.41\uparrow}$	42.54	$\underline{46.03}_{3.49\uparrow}$	$\mathbf{59.20}_{17.20\uparrow}$	42.00	$\underline{51.60}_{9.60\uparrow}$	$\mathbf{62.40}_{7.20\uparrow}$	55.20	$\underline{57.60}_{2.40\uparrow}$	$\mathbf{55.20}_{7.60\uparrow}$	47.60	$\underline{51.60}_{4.00\uparrow}$
ID	$\mathbf{65.37}_{3.97\uparrow}$	61.40	$\underline{63.11}_{1.71\uparrow}$	$\mathbf{82.37}_{11.08\uparrow}$	71.29	$\underline{81.29}_{10.00\uparrow}$	$\mathbf{55.17}_{15.86\uparrow}$	39.31	$\underline{54.48}_{15.17\uparrow}$	$\mathbf{67.69}_{5.48\uparrow}$	62.21	$\underline{67.52}_{5.31\uparrow}$
OOD	$\mathbf{65.34}_{4.99\uparrow}$	60.35	$\underline{63.93}_{3.58\uparrow}$	$\mathbf{63.63}_{5.69\uparrow}$	57.94	$\underline{63.54}_{5.60\uparrow}$	$\mathbf{65.62}_{6.41\uparrow}$	59.21	$\underline{64.75}_{5.54\uparrow}$	$\mathbf{61.90}_{4.87\uparrow}$	57.03	$\underline{61.45}_{4.42\uparrow}$
Mistral-7B												
HOMO	$\mathbf{49.80}_{15.12\uparrow}$	34.68	$\underline{35.02}_{0.34\uparrow}$	$\mathbf{46.80}_{13.60\uparrow}$	33.20	$\underline{40.80}_{7.60\uparrow}$	$\mathbf{59.60}_{11.20\uparrow}$	48.40	$\underline{55.20}_{6.80\uparrow}$	$\mathbf{48.00}_{9.20\uparrow}$	38.80	$\underline{43.60}_{4.80\uparrow}$
ID	$\mathbf{58.17}_{16.40\uparrow}$	41.77	$\underline{51.83}_{10.06\uparrow}$	$\mathbf{67.74}_{38.39\uparrow}$	29.35	$\underline{50.11}_{20.76\uparrow}$	$\mathbf{44.83}_{13.80\uparrow}$	31.03	$\underline{42.07}_{11.04\uparrow}$	$\mathbf{55.21}_{13.78\uparrow}$	41.43	$\underline{49.38}_{7.95\uparrow}$
OOD	$\mathbf{54.48}_{14.01\uparrow}$	40.47	$\underline{47.81}_{7.34\uparrow}$	$\mathbf{53.07}_{20.09\uparrow}$	32.98	$\underline{45.07}_{12.09\uparrow}$	$\mathbf{50.49}_{8.60\uparrow}$	41.89	$\underline{44.51}_{2.62\uparrow}$	$\mathbf{52.42}_{11.49\uparrow}$	40.93	$\underline{48.88}_{7.95\uparrow}$

Table 1: The performance of the four base LLMs after fine-tuning under harmonious (HAR), incompatible (INC), and self-aligning (SELF) settings. For each domain and base model, models fine-tuned on the harmonious dataset and on the self-aligning dataset achieve superior performance compared to those fine-tuned on the incompatible dataset, across all scenarios including homogeneous (HOMO), in-domain (ID), and out-of-domain (OOD) evaluations. The best/second-best performance for each domain and base model in each evaluation is highlighted in bold/underline. The arrows indicate the differences compared to the incompatible setting.

conclusion holds true across homogeneous, in-domain, and out-of-domain evaluations.

Finding 2. *Using IFT data that aligns with model parameter knowledge yet is erroneous yields better performance than employing those that are correct but incongruent with model parameter knowledge.*

To further investigate the impact of learning domain-specific world knowledge on IFT, we conduct a more direct comparative experiment between the self-aligning and incompatible settings. For each domain and pre-trained LLM, the two settings' datasets use identical queries that the model cannot answer correctly under in-context learning and have different responses: the incompatible dataset's responses are correct, reflecting world knowledge, while the self-aligning dataset's responses represent model parameter knowledge, are incorrect.

Table 1 compares results under the self-aligning and incompatible settings. Surprisingly, despite the self-aligning dataset containing only incorrect answers, models fine-tuned on it significantly outperform those using the incompatible dataset, which requires learning inconsistent world knowledge. The performance difference is notable, with the former achieving an average increase of 5.25%, 9.78%, and 6.97% in homogeneous, in-domain, and out-of-domain evaluations, respectively.

This finding emphatically indicates that injecting additional domain knowledge through IFT also fails to bring effective improvements to LLMs even in homogeneous and in-domain evaluations. Conversely, maintaining consistency with model parameter knowledge, that is, without injecting any additional world knowledge through the self-aligning setting, can yield superior results. Moreover, this advantage holds true across all evaluations, as well as different model sizes and architectures.

The above results demonstrate a significant decline of performance in models fine-tuned using data that contain correct world knowledge but conflict with model parameter knowledge, compared to using consistent IFT data aligned with parameter knowledge. This suggests that introducing additional world knowledge through IFT, in cases where it is inconsistent with the parameter knowledge, may not yield the benefits we anticipate. Therefore, the core role of IFT may lie in facilitating the transfer of behavioral norms, rather than injecting additional domain-specific world knowledge. To further validate this conclusion, in the next section, we design a novel method to decouple the conflict knowledge contained in IFT data and present further analysis of this issue.

5 Exp-II: IFT with Contextualized Knowledge

In this section, we introduce a new analysis method called contextualized knowledge decoupling to fur-

Model	IFT	HOMO	ID	OOD	Overall
LLaMA-2-7B	Vanilla	31.11	33.77	32.56	32.48
	Contextualized	**37.62**	**43.70**	**40.61**	**40.65**
LLaMA-2-13B	Vanilla	34.51	37.35	37.24	36.37
	Contextualized	**41.47**	**49.48**	**46.59**	**45.84**
Mistral-7B	Vanilla	**38.77**	35.90	39.07	37.91
	Contextualized	37.22	**43.47**	**44.99**	**41.89**
Average	Vanilla	34.80	35.67	36.29	35.59
	Contextual	**38.77**	**45.55**	**44.06**	**42.79**

Table 2: The performance comparison between models fine-tuned with vanilla IFT and models fine-tuned with contextualized IFT respectively.

ther investigate the impact of inconsistent knowledge during IFT. This approach involves explicitly providing relevant world knowledge needed to answer a query within the context of the query itself. Under this paradigm, the model no longer needs to learn knowledge during IFT, but only needs to use the knowledge in the context to answer in the expected behavioral norm. This method helps prevent the model from learning additional world knowledge during IFT, which is inconsistent with parameter knowledge, thus separating knowledge injection from behavioral norm transfer.

To this end, we start with the data of the incompatible setting to construct a dataset with contextualized knowledge. Specifically, given an instruction-answer pair in the incompatible group, we employ GPT-3.5[5] to generate the world knowledge that is required to answer the instruction. The knowledge is then concatenated with the original instruction, as well as the answer, to construct an augmented pair. Finally, we use the constructed data to fine-tune LLMs and compare them with the models fine-tuned with vanilla IFT.

Finding 3. *Ensuring that the model does not learn world knowledge conflicting with parameter knowledge during IFT enhances the effectiveness of IFT.*

Table 2 shows the results of contextualized knowledge decoupling in three base models [6]. From the table, it can be observed that fine-tuning the model with data using explicit contextualized knowledge significantly mitigates the adverse effects caused by inconsistencies between parameter knowledge and world knowledge in IFT data. Compared to vanilla IFT using incompatible data, our method achieves an average improvement of 8.16% on LLaMA-2-7B, 9.48% on LLaMA-2-13B, and 3.98% on Mistral-7B. Except in rare cases, IFT using data with contextualized knowledge can significantly improve the effect of IFT in the incompatible setting across homogeneous, in-domain, and out-of-domain evaluations.

The results indicate that we should not force the model to learn additional inconsistent knowledge when the world knowledge in IFT data is inconsistent with model parameter knowledge. Instead, by decoupling knowledge learning and the transfer of behavioral norms, the problems caused by the above knowledge conflicts can be effectively alleviated. This further verifies our observation in Section 4 that additional world knowledge injection may be ineffective or even harmful for IFT. During IFT, the model should focus on transferring the behavioral norms relying on the existing parameter knowledge and the regularity in the IFT data, rather than learning additional world knowledge. Therefore, for RQ1, we conclude:

Conclusion 1. *For IFT, there is little, if not even causing additional damage, benefits from the learning of world knowledge incongruent with parameter knowledge.*

6 Exp-III: Is Consistency All You Need?

The above findings appear to suggest a conclusion: For better transfer of behavioral norms, we should employ IFT data that completely aligns with model parameter knowledge without any additional world knowledge. To substantiate this hypothesis, we design a new set of experiments.

Specifically, by adjusting the proportion of samples derived from the incompatible and the self-aligning group, we aim to adjust the ratios of consistency between the world knowledge in the IFT data and model parameter knowledge, thereby observing the variations in IFT outcomes under different consistency ratios.

Finding 4. *Employing IFT data that is fully consistent with model parameter knowledge does not necessarily result in optimal performance.*

Figure 4 displays the results on three base models under different consistency ratios across four domains. From the figure, we can see that: 1) The performance of the incompatible group (i.e.,

[5] https://openai.com/

[6] In this experimental setting, "vanilla" is equivalent to "incompatible" group.

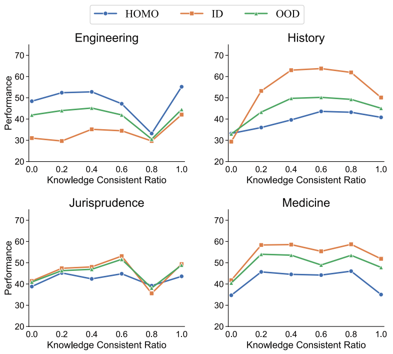


Figure 2: The performance of Mistral-7B fine-tuned with instruction datasets of varying consistency ratios. Each dataset is composed of a mixture of incompatible and self-aligning data, and the consistency ratio represents the proportion of self-aligning samples. Note that a consistency ratio of 0 signifies that all data samples are incompatible, whereas a ratio of 1 indicates exclusively self-aligning data. The results of other base models are presented in the Appendix C due to page limitations.

Model	HOMO		ID		OOD	
	r	p-value	r	p-value	r	p-value
Mistral-7B	0.78	0.00	0.81	0.00	0.82	0.00
LLaMA-2-7B	0.27	0.14	0.19	0.30	0.21	0.24
LLaMA-2-13B	0.56	0.00	0.78	0.00	0.87	0.00
All	0.43	0.00	0.57	0.00	0.48	0.00

Table 3: The Spearman partial correlation analysis between the model performance after fine-tuning, and the knowledge consistency between base model and fine-tuned model. The analysis is controlled with the base model's performance on each test set. r and p-value denote partial correlation coefficient and significance respectively. For LLaMA-2-13B and Mistral-7B, p-values significantly lower than 0.05 indicates a high level of confidence.

ratio=0) is indeed poor, which aligns with our previous conclusion; 2) Conversely, relying solely on IFT data that completely aligns with the model's parameter knowledge (i.e., ratio=1) fails to ensure superior performance across a broad range of scenarios; 3) Optimal performance is most frequently achieved through a balanced integration of incompatible and self-aligning data. The ideal proportion of this combination varies across different base models and domains.

The observations suggest that while there is a significant impact of the consistency between the knowledge within the IFT data and the parameter knowledge of the original model on the performance of the fine-tuned model, this consistency is not the fundamental factor influencing IFT performance. Therefore, a deeper exploration into the mechanisms underlying knowledge consistency is essential to identify the true determinants of IFT's effectiveness.

7 Exp-IV: Rethinking Consistency: What Really Matters for IFT?

To further explore the underlying mechanisms of IFT, we analyze the knowledge discrepancies of different base LLMs before and after IFT on various evaluation data to observe the extent of internal knowledge alternation triggered by IFT. Specifically, for each sample within the test data, we initially compute the Pearson correlation coefficient between the ranking of the original model's predictions on choices through in-context learning probing, and those provided by the fine-tuned model. Building upon this, we calculate the average Pearson correlation coefficient across each test set and subsequently compare it with the performance on the same test set of the fine-tuned model. Through this experimental analysis, we aim to observe how the knowledge changes induced by IFT affect the ultimate effect of IFT.

To this end, we employ a partial correlation assessment (Spearman, 1961) to analyze the aforementioned Pearson correlation and model performance. Specifically, we utilize a total of 96 models, which include 72 models from Exp-III, in addition to models fine-tuned with consistency ratios of 0.05 and 0.1, conducting tests on homogeneous, in-domain, and out-of-domain data. To eliminate the potential influence of the base model's performance on the correlation analysis, ensuring that the results only reflect the differences brought about by varying IFT data, we treat the base model's performance as a control variable. By employing partial correlation assessment under the constraint of removing this control variable's influence, we analyze the correlation between the Pearson correlation coefficients of model predictions' rankings before and after IFT and their ultimate performance. Please refer to the Appendix D for more details about our analysis.

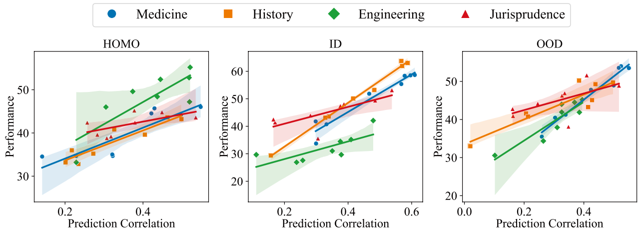


Figure 3: The regression analysis between the model performance after fine-tuning, and the knowledge consistency between base model and fine-tuned model. We show the results of Mistral-7B in three evaluations. The grouped linear regression demonstrate the positive correlations between the model performance after IFT and model internal knowledge consistency before and after IFT. Points in the same regression line indicate the results of the same base model fine-tuned with different IFT data of the same domain on the same test set (HOMO, ID, or OOD).

Finding 5. *The consistency of internal knowledge within a model before and after IFT is the key factor affecting the performance of the fine-tuned model.*

Figure 3 presents the results of grouped regression analysis conducted on Mistral-7b. Table 3 displays the correlation coefficients and significance for the partial correlation analysis across different base models and different evaluations. From the aforementioned figure and table, a significant trend can be observed, namely that the correlation of predictions made by models before and after IFT on a given evaluation has a substantial impact on the final performance of the fine-tuned models on that evaluation. It is noteworthy that this phenomenon holds true across homogeneous, in-domain, and out-of-domain evaluations. This implies that even for homogeneous and in-domain test sets, which are within the same field as the IFT data, the injection of additional world knowledge and the alteration of internal parameter knowledge through IFT do not contribute to enhancing the final performance of the models. Instead, maintaining consistency in the knowledge of models before and after IFT significantly positively influences the performance of the fine-tuned models.

To further investigate whether this finding is the underlying mechanism responsible for Finding 4, we explore the impact of different consistency data on the correlation of model knowledge before and after IFT in Exp-III. We analyze the probability distribution of model outputs under different IFT data conditions. Table 4 presents the related experimental results, which reveal that: IFT using data containing world knowledge completely inconsistent with the parameter knowledge evidently leads to a divergence in the internal knowledge of the fine-tuned model from that of the original model, thereby impairing the performance of the fine-tuned model. Furthermore, IFT using data containing world knowledge completely consistent with the parameter knowledge also may result in a divergence in the knowledge distribution between the original and fine-tuned models. Specifically, fine-tuning with only consistent IFT data may steer the model towards an sharp knowledge distribution, whereas the original model's parameter knowledge exists as a relatively smooth distribution. Conversely, using a middle setting that mixes incompatible and self-aligning data allows the optimization process to maintain model parameter knowledge unchanged while preserving the distribution's smoothness, thereby enabling the fine-tuned model outputs to more closely resemble those of the original model, ultimately yielding better performance.

Model	Best	Self-aligning	Incompatible
Mistral-7B	0.24	0.34	0.37
LLaMA-2-7B	0.16	0.93	0.51
LLaMA-2-13B	0.18	0.68	0.34

Table 4: KL divergence between the prediction distribution of fine-tuned model under zero-shot setting and the prediction distribution of base model using in-context learning probing. "Best" denotes the model that exhibits the best average performance across three evaluations.

The experimental results mentioned above demonstrate that the key to the performance of fine-tuned models lies in the consistency of model parameter knowledge before and after IFT. In fact, the core mechanism underlying the superior performance observed in both Finding 1 and Finding 2 for the IFT data is attributed to the maintenance of consistency in model knowledge before and after IFT. Therefore, for RQ2 raised above, we conclude:

Conclusion 2. *The essence of an effective IFT lies in maintaining the consistency of model parameter knowledge before and after IFT.*

8 Conclusion and Discussion

Our experiments and conclusions indicate that the core function of IFT is not to learn domain-specific world knowledge. Instead, learning world knowledge that is inconsistent with model parameter knowledge actually undermines the performance of the model in all evaluations from homogeneous, in-domain to out-of-domain. Furthermore, we discover that the consistency of model parameter knowledge before and after IFT (i.e., the knowledge probed through in-context learning before IFT and the knowledge exhibited under zero-shot setting after IFT) plays a crucial role in determining the ultimate performance of the fine-tuned model. These two findings unveil a fundamental mechanism of IFT, that is, IFT is not a supervised, domain-specific knowledge *learning* process, but a process of *self-aligning* instruction with the already existing parameter knowledge of LLMs. Therefore, the ultimate determinant of IFT effect is not the extent of domain knowledge injection, but rather whether the IFT process can facilitate more effective self-aligning, thereby enhancing the expression of the model's parameter knowledge under the zero-shot question-answering paradigm after IFT.

Our discovery not only provides guidance for future IFT data construction, model training, and model evaluation but also provide robust support for some very recent studies. For instance, super-alignment (Burns et al., 2023) aims to use a weak model to guide a strong model's alignment. Our conclusion proves that it is entirely possible to use a weak model with less knowledge to guide a strong, more knowledgeable model for IFT. Our conclusions elucidate the viable underlying factors for self-instruction-tuning (Sun et al., 2024; Guo et al., 2024), self-rewarding (Yuan et al., 2024) and consistent alignment (Wan et al., 2024) and provide a solid foundation for the future development of these studies.

Limitations

In order to facilitate probing model parameter knowledge, we currently focus on multiple-choice questions. In the future, we plan to extend our framework to free-style generation. Besides, due to hardware limitations, the vast majority of experiments are conducted on models with about 10B parameters, we only explore 70B models in several experiments. Repeating our study on larger models in more domains will contribute to a deeper understanding of the IFT for larger models.

Acknowledgements

We sincerely thank all anonymous reviewers for their insightful comments and valuable suggestions. This research work is supported by CAS Project for Young Scientists in Basic Research (Grant No.YSBR-040), the National Natural Science Foundation of China under Grants no. 62122077 and 62106251, and the Basic Research Program of ISCAS, Grant No. ISCAS-JCZD-202303.

References

Christopher M Bishop. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, and Jared D. Kaplan. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Collin Burns, Pavel Izmailov, Jan Hendrik Kirchner, Bowen Baker, Leo Gao, Leopold Aschenbrenner, Yining Chen, Adrien Ecoffet, Manas Joglekar, Jan Leike, et al. 2023. Weak-to-strong generalization: Eliciting strong capabilities with weak supervision. *arXiv preprint arXiv:2312.09390*.

Hao Chen, Yiming Zhang, Qi Zhang, Hantao Yang, Xiaomeng Hu, Xuetao Ma, Yifan Yanggong, and Junbo Zhao. 2023a. Maybe only 0.5% data is needed: A preliminary exploration of low training data instruction tuning. *arXiv preprint arXiv:2305.09246*.

Yirong Chen, Xiaofen Xing, Jingkai Lin, Huimin Zheng, and Zhenyu Wang. 2023b. Soulchat: Improving llms' empathy, listening, and comfort abilities through fine-tuning with multi-turn empathy conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 1170–1183, Singapore. Association for Computational Linguistics.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research*, 25(70):1–53.

Jiaxi Cui, Zongjian Li, Yang Yan, Bohua Chen, and Li Yuan. 2023. Chatlaw: Open-source legal large language model with integrated external knowledge bases. *arXiv preprint arXiv:2306.16092*.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, and Zhifang Sui. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234*.

Lingyu Gao, Aditi Chaudhary, Krishna Srinivasan, Kazuma Hashimoto, Karthik Raman, and Michael Bendersky. 2023. Ambiguity-aware in-context learning with large language models. *arXiv preprint arXiv:2309.07900*.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Zhouhong Gu, Xiaoxuan Zhu, Haoning Ye, Lin Zhang, Jianchen Wang, Yixin Zhu, Sihang Jiang, Zhuozhi Xiong, Zihan Li, Weijie Wu, et al. 2024. Xiezhi: An ever-updating benchmark for holistic domain knowledge evaluation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18099–18107.

Arnav Gudibande, Eric Wallace, Charlie Snell, Xinyang Geng, Hao Liu, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. The false promise of imitating proprietary llms. *arXiv preprint arXiv:2305.15717*.

Hongyi Guo, Yuanshun Yao, Wei Shen, Jiaheng Wei, Xiaoying Zhang, Zhaoran Wang, and Yang Liu. 2024. Human-instruction-free llm self-alignment with limited samples. *arXiv preprint arXiv:2401.06785*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Srinivasan Iyer, Xi Victoria Lin, Ramakanth Pasunuru, Todor Mihaylov, Daniel Simig, Ping Yu, Kurt Shuster, Tianlu Wang, Qing Liu, Punit Singh Koura, et al. 2022. Opt-iml: Scaling language model instruction meta learning through the lens of generalization. *arXiv preprint arXiv:2212.12017*.

Joel Jang, Seungone Kim, Seonghyeon Ye, Doyoung Kim, Lajanugen Logeswaran, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2023. Exploring the benefits of training expert language models over instruction tuning. In *Proceedings of the 40th International Conference on Machine Learning*, pages 14702–14729.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yunxiang Li, Zihan Li, Kai Zhang, Ruilong Dan, Steve Jiang, and You Zhang. 2023. Chatdoctor: A medical chat model fine-tuned on a large language model meta-ai (llama) using medical domain knowledge. *Cureus*, 15(6).

Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. 2024. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. In *The Twelfth International Conference on Learning Representations*.

Alisa Liu, Xiaochuang Han, Yizhong Wang, Yulia Tsvetkov, Yejin Choi, and Noah A Smith. 2024. Tuning language models by proxy. *arXiv preprint arXiv:2401.08565*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064.

Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. 2022. Medmcqa: A large-scale multi-subject multi-choice dataset for medical domain question answering. In *Conference on Health, Inference, and Learning*, pages 248–260. PMLR.

Baolin Peng, Chunyuan Li, Pengcheng He, Michel Galley, and Jianfeng Gao. 2023. Instruction tuning with gpt-4. *arXiv preprint arXiv:2304.03277*.

Jeff Rasley, Samyam Rajbhandari, Olatunji Ruwase, and Yuxiong He. 2020. Deepspeed: System optimizations enable training deep learning models with over 100 billion parameters. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 3505–3506, New York, NY, USA. Association for Computing Machinery.

Ming Shen. 2024. Rethinking data selection for supervised fine-tuning. *arXiv preprint arXiv:2402.06094*.

C. Spearman. 1961. *The Proof and Measurement of Association Between Two Things*. Studies in Individual Differences: The Search for Intelligence. Appleton-Century-Crofts, East Norwalk, CT, US.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. 2024. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Fanqi Wan, Xinting Huang, Leyang Cui, Xiaojun Quan, Wei Bi, and Shuming Shi. 2024. Mitigating hallucinations of large language models via knowledge consistent alignment. *arXiv preprint arXiv:2401.10768*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, et al. 2023. Baichuan 2: Open large-scale language models. *arXiv preprint arXiv:2309.10305*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*.

Hanning Zhang, Shizhe Diao, Yong Lin, Yi R Fung, Qing Lian, Xingyao Wang, Yangyi Chen, Heng Ji, and Tong Zhang. 2023. R-tuning: Teaching large language models to refuse unknown questions. *arXiv preprint arXiv:2311.09677*.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. 2023. Pytorch fsdp: experiences on scaling fully sharded data parallel. *arXiv preprint arXiv:2304.11277*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

A Data Construction

A.1 Train

As mentioned in Section 3, we collect four multi-choice question datasets from different domains: medicine, history, engineering, and jurisprudence. The quantity and split of the dataset for each domain are detailed in Table 5.

Domain	Dev	Test	Train
Medicine	10	1462	20000
History	10	250	8605
Jurisprudence	10	250	6510
Engineering	10	250	4805

Table 5: The number of instances in the development, test, and train sets for each domain.

For each item in each train set, we employ five demonstrations from the corresponding dev set for in-context learning to assess the parameter knowledge of each base model. The probing prompt is shown in Table 6.

The following are multiple choice questions about <domain>. Please choose the correct answer.

5-shot
<question1>
<question1 options>
Answer:<question1 answer>

<demo2>
.
.
.
<demo5>

<question>
<question options>
Answer:

Table 6: Prompt design based on Yang et al. (2023) for probing model parameter knowledge.

Because model's responses are influenced by the selection and order of demonstrations in in-context learning (Gao et al., 2023; Min et al., 2022), we regard responses with a confidence level exceeding 0.5 as reflective of the model's parameter knowledge to ensure the reliability of identification. For every domain and base model combination, we establish train datasets on the three different settings as described in Section 3. We report the amount of training data for each combination in Table 7. We employ the prompt in Table 8 for constructing instruction-response pairs.

Model	Engineering	History	Jurisprudence	Medicine
LLaMA-2-7B	738	996	1033	2507
LLaMA-2-13B	668	1236	676	1782
LLaMA-2-70B	712	1072	1002	2518
Mistral-7B	534	838	639	1400

Table 7: The number of train sets for each domain and base model combination.

For <explanation> in the prompt, if the answer is consistent with the model's parameter knowledge but incorrect, we employ prompt in Table 9 for prompting the base model to generate explanations for its

 # Input The following are multiple choice questions about domain. Please choose the correct answer. <question> <question options> **# Output** <answer> Explanation: <explanation>

Table 8: Format of instruction-response pair for Vanilla IFT.

choice; if the answer is correct, we directly use the explanation in MedMCQA for medicine and employ prompt in Table 10 for prompting GPT3.5 to generate explanations for the golden answer for the other three domains.

For stable and real IFT, we incorporate an equal proportion of general instruction data sampled from alpaca-gpt4-en (Peng et al., 2023) which is identical for each domain and base model combination. We use vicuna-v1.5 (Zheng et al., 2024) format to train models.

A.2 Test

To compare the performance across three settings for each domain and base model combination, we devise three evaluation types: homogeneous, in-domain, and out-of-domain tests. The first evaluation involves a holdout from the same multi-choice question set, while the subsequent two, in-domain and out-of-domain, are derived from MMLU's splits. The specific details of the splits for each domain are detailed in Table 11.

B Details about Contextualized IFT

In Section 5, we use the prompt in Table 12 to employ GPT-3.5 to provide the knowledge required for the inconsistent IFT data. For contextualized IFT, we employ the prompt in Table 13 for building instruction-response pairs to train models. Note we do not provide the knowledge required for the instruction during the testing phase.

C Performance of Models Fine-tuned in Different Consistency Ratios

In Exp-III, due to page limitations, we only report the performance of Mistral-7B fine-tuned with instruction datasets of varying consistency ratios. In Figure 4, we supplement the results of the other two base LLMs: LLaMA-2-7B and LLaMA-2-13B.

D Details about Spearman Partial Correlation Analysis

Partial correlation measures the degree of association between two variables (here, the model performance after IFT and the knowledge consistency between the base model and fine-tuned model) while controlling the effect of one or more additional variables (the accuracy on the same test set of the base model in our case).

To calculate the Spearman partial correlation between the model performance after IFT and the knowledge consistency between the base model and fine-tuned model, we first calculate the knowledge consistency, measured by the correlation coefficient of model prediction ranking before and after IFT using the following formula.

$$pr = \frac{1}{n}\sum_{i=1}^{|D_{\text{test}}|} \text{pearson}(\text{sort}(m_{\text{base}}, q_i, c_i), \text{sort}(m_{\text{tuned}}, q_i, c_i))$$

 # Instruction

Below is a list of conversations between a human and an AI assistant (you).
Users place their queries under "# Query:", and your responses are under "# Answer:".
You are a helpful, respectful, and honest assistant. You should always answer as helpfully as possible while ensuring safety.
Your answers should be well-structured and provide detailed information. They should also have an engaging tone.
Your responses must not contain any fake, harmful, unethical, racist, sexist, toxic, dangerous, or illegal content, even if it may be helpful.
Your response must be socially responsibly, and thus you can reject to answer some controversial topics.

 # Query:
```Can you tell me some common types of renewable energy sources? ```
 # Answer:
```
Absolutely, below are some of the most common types of renewable energy sources:

1. Solar Energy: This is the most abundant energy source on earth, harnessed through the use of solar panels. These panels convert sunlight into electricity without any moving parts, noise, pollution, or damage to the environment.
2. Wind Energy: Wind turbines convert the kinetic energy in the wind into mechanical power. This mechanical power can be used for specific tasks (such as pumping water) or converted into electricity to power homes, businesses, and schools.
3. Hydropower: Generated by using electricity generators to capture the energy from falling or fast-moving water. This renewable source can come from various ways, including a large dam on a river, tidal or wave energy from the ocean, or using small scale turbines in streams.
4. Geothermal Energy: This type of energy is generated from the heat deep within the Earth. This heat can be used directly for heating buildings or to generate electricity. It is continuously produced inside the Earth and is nearly as reliable as the tides.
5. Biomass Energy: Biomass is organic material that comes from plants and animals, and it contains stored energy from the sun. This energy can be burned directly or converted into biofuel which can burn more efficiently.

Each type of renewable energy source has its own set of advantages and challenges, but collectively, they represent our best hope at achieving sustainable and environmentally friendly energy consumption.
```

 # Query:
```
Below is a multiple-choice question and the answer. Please give the explanation.
Question: <question>
Choices: <question options>
Answer: <answer>
```

 # Answer:

Table 9: Prompt design based on URIAL (Lin et al., 2024) for employing base model to generate explanations for its predicted answer.

The following is a multi choice question about <domain>.

<question>
<choice>

The answer is "<answer>". Please explain why.

Table 10: Prompt design for employing GPT-3.5 to generate explanations for the golden answer.

where q_i and c_i represent the i-th question and its corresponding set of candidate choices. The terms m_{base} and m_{tuned} denote the base model and its fine-tuned model, respectively. D_{test} represents any test set like HOMO, ID, or OOD. $|D_{\text{test}}|$ denotes the size of the test set. The 'sort' function ranks the candidate choices based on the predicted probability distribution from the input model. pr reflects the average
```

Domain	In-domain Subcategories
Engineering	["electrical_engineering"]
History	["high_school_european_history", "high_school_us_history", "high_school_world_history", "prehistory"]
Jurisprudence	["econometrics", "high_school_geography", "high_school_government_and_politics", "high_school_macroeconomics", "high_school_microeconomics", "high_school_psychology", "human_sexuality", "international_law", "jurisprudence", "professional_law", "sociology", "public_relations", "professional_psychology", "security_studies", "us_foreign_policy"]
Medicine	["anatomy", "clinical_knowledge", "college_medicine", "human_aging", "medical_genetics", "nutrition", "professional_medicine", "virology"]

Table 11: In-domain and out-of-domain split of MMLU subcategories for our four experimental domains. The out-of-domain subcategories comprise the remaining subcategories not included in the in-domain classification.

Given a multi-choice question and the answer, please write a short piece of evidence to support it so that a layman who has read the evidence you give can answer the question correctly.
If your response contains words "listed", "option" or "choice" like "among the listed/given options', you will be penalized.

Question:
<question>
<question options>

Answer:
<answer>

Evidence:

Table 12: Prompt design for employing GPT-3.5 to generate the knowledge required for the instruction.

 # Input
The following are multiple choice questions about <domain>. Given the context. Please choose the correct answer.

<context>
<question>
<question options>

 # Output
<answer>

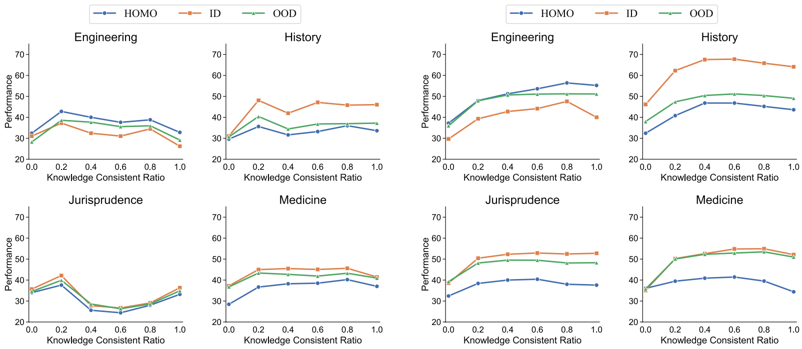
Table 13: Format of instruction-response pair for our contextualized IFT.

Pearson correlation coefficient between the rankings predicted by the base model and the fine-tuned model of each question.

For each base model, its every fine-tuned model, and each test set, we calculate the corresponding above pr and organize them along with the corresponding accuracy on the same test set of the base model and fine-tuned model to three lists, denoted as pr_s, p_{base} and p_{tuned}. Then we compute the Spearman partial correlation. The formulas are:

$$e(pr_s) = pr_s - \hat{pr}_s(p_{\text{base}})$$
$$e(p_{\text{tuned}}) = p_{\text{tuned}} - p_{\hat{\text{tuned}}}(p_{\text{base}})$$
$$r, p\text{-}value = \text{spearman}(e(pr_s), e(p_{\text{tuned}}))$$

Here, $\hat{pr}_s(p_{\text{base}})$ and $p_{\hat{\text{tuned}}}(p_{\text{base}})$ present the estimated value of pr_s and p_{tuned} predicted based on



(a) LLaMA-2-7B (b) LLaMA-2-13B

Figure 4: The performance of LLaMA-2-13B and LLaMA-2-7B fine-tuned with instruction datasets of varying consistency ratios. Each dataset is composed of a mixture of incompatible and self-aligning data, and the consistency ratio represents the proportion of self-aligning samples.

p_{base}, respectively. The residuals of pr_s and p_{tuned} relative to p_{base} are denoted as $e(pr_s)$ and $e(p_{\text{tuned}})$, respectively. We finally calculate Spearman's correlation coefficient and the corresponding p-values between the residuals, denoted as r and $p\text{-}value$.

In this way, we can find the direct relationship between the model performance after IFT, and the knowledge consistency between the base model and fine-tuned model by removing the influence of base model performance that affects them both.

ODM: A Text-Image Further Alignment Pre-training Approach for Scene Text Detection and Spotting

Chen Duan*
duanchen02@meituan.com

Pei Fu*
fupei@meituan.com

Shan Guo
guoshan05@meituan.com

Qianyi Jiang
jiangqianyi02@meituan.com

Xiaoming Wei
weixiaoming@meituan.com

Abstract

In recent years, text-image joint pre-training techniques have shown promising results in various tasks. However, in Optical Character Recognition (OCR) tasks, aligning text instances with their corresponding text regions in images poses a challenge, as it requires effective alignment between text and OCR-Text (referring to the text in images as OCR-Text to distinguish from the text in natural language) rather than a holistic understanding of the overall image content. In this paper, we propose a new pre-training method called ***O****CR-Text* ***D****estylization* ***M****odeling (ODM) that transfers diverse styles of text found in images to a uniform style based on the text prompt. With ODM, we achieve better alignment between text and OCR-Text and enable pre-trained models to adapt to the complex and diverse styles of scene text detection and spotting tasks. Additionally, we have designed a new labeling generation method specifically for ODM and combined it with our proposed Text-Controller module to address the challenge of annotation costs in OCR tasks, allowing a larger amount of unlabeled data to participate in pre-training. Extensive experiments on multiple public datasets demonstrate that our method significantly improves performance and outperforms current pre-training methods in scene text detection and spotting tasks. Code is available at ODM.*

1. Introduction

Optical Character Recognition (OCR) has garnered significant attention in the field of computer vision for its remarkable performance in automated data entry, document analysis, instant translation, and more. Most existing methods for obtaining OCR results involve a two-stage process, including a text detection model and a text recognition model, or directly utilizing an end-to-end text spotting model. Furthermore, many of these methods initialize the model through pre-training on the ImageNet dataset [37].

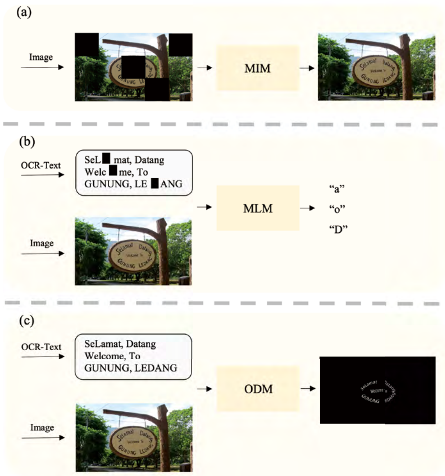


Figure 1. Comparisons of different pre-training strategies. (a) Obtain the pre-trained model through mask image modeling, taking only image embeddings as inputs. (b) Obtain the pre-trained model through mask language modeling, which simultaneously takes both OCR-Text and image as inputs. (c) Our approach obtains the pre-trained model through OCR-Text destylization modeling.

Pre-training techniques have recently gained significant attention for their outstanding performance across a wide range of computer vision tasks. Two commonly used pre-training methods in computer vision are: (1) Masked Image Modeling (MIM) pre-training [1, 3, 10, 49], which focuses on learning visual contextualized representations

*First Author and Second Author contribute equally to this work.

solely from images. This method is typically applied to vision-dominated tasks, such as image classification. (2) Masked Language Modeling (MLM) [6, 51, 52], which utilizes both text and image as inputs and extracts semantic information from both modalities. However, when applying these pre-training methods in the OCR field, two specific issues arise: (1) With the MIM-based method, there are instances where the text in the image is completely obscured by the masked patch, primarily due to the relatively small proportion of the text. This has the potential to hinder the pre-trained model's ability to effectively learn textual feature information. (2) The MLM-based method, although achieving weakly supervised training by masking the text input, does not explicitly exploit text location information during training. This can lead to ineffective alignment between text and image features, as well as inadequate handling of image information. These challenges highlight the need for innovative approaches that specifically address the unique requirements of OCR.

In this paper, we propose a novel pre-training technique called OCR-Text Destylization Model (ODM) to address the challenges of text-image alignment in OCR tasks. As illustrated in Fig. 1, unlike existing MIM and MLM methods, ODM introduces a new pixel-level image reconstruction modeling based on text prompts. Since OCR tasks primarily focus on the text within the image while considering other pixels irrelevant, ODM aims to reconstruct a binary image that removes the text style and enforces alignment between the text and OCR-Text. This is achieved by utilizing a pixel-level reconstruction approach instead of the traditional three-channel reconstruction. To further enhance the model's understanding of the text, we propose a Text-Controller module. This module guides the image encoder to identify and interpret the OCR-Text, facilitating the alignment between the text and OCR-Text. Additionally, we have designed a novel method for generating ODM labels, effectively addressing the issue of inadequate pixel-level labels in the dataset. By leveraging font files, text, and location labels, we generate binary images with a unified font style, as illustrated in Fig. 2, which showcases some examples of OCR-Text destylization images. These advancements in ODM and the Text-Controller module, along with the novel label generation method, contribute to improved text-image alignment in OCR tasks.

In summary, the main contributions are three-fold:

(1) We propose a simple yet effective pre-training method called ODM, which focuses on learning features specifically for OCR-Text. By using pixel-level labels with a uniform style, we successfully destylize OCR-Text, improving text comprehension. This crucial feature information enables the pre-trained model to adapt well to various scenarios in text detection and spotting tasks.

(2) We introduce a novel Text-Controller module that helps regulate the model's output, enhancing its understanding of OCR-Text. With this module, our method does not require a perfect match between the input image and the text pair. As a result, we can utilize weakly annotated data (i.e., using other OCR recognition engines to obtain the text and location in the image and filter it based on recognition confidence, text size, etc.), which can greatly reduce the annotation cost.

(3) Experimental results on public datasets demonstrate that ODM delivers outstanding performance and surpasses existing pre-training techniques across a range of scene text detection and spotting datasets.

(a) (b) (c) (d)

Figure 2. The upper row and lower row represent the original images and their corresponding destylized labels, respectively. (a), (b), (c), and (d) are taken from the ICDAR15 [15], CTW1500 [26], TotalText [5], and LSVT [41] datasets, respectively.

2. Related Work

Scene Text Detection. Scene text detectors based on deep learning can primarily be categorized into regression based [11, 18, 32, 38, 42, 57] and segmentation based [22, 23, 46, 47, 50, 55, 58] methods. Regression-based methods perceive scene text as an object and directly regress the bounding box of the text instance. Segmentation-based methods treat the text detection task as a semantic segmentation problem. They obtain a segmentation map by directly segmenting the text instance and subsequently group the segments into a box through post-processing.

Scene Text Spotting. Scene text spotting represents the unification of detection and recognition processes within a singular framework. Numerous studies have improved performance by simultaneously learning detectors and recognizers. In [2, 7, 12, 19, 20, 25], end-to-end implementation is achieved by training the detection and recognition separately. The Mask TextSpotter [21, 30, 31] series performs character segmentation during the recognition process. The ABCNet [27, 28] series obtains detection coordinates through control points of the Bezier curve. SwinTextSpotter [13] and ESTextSpotter [14] use separate detection-recognition heads corresponding to different annotation formats.

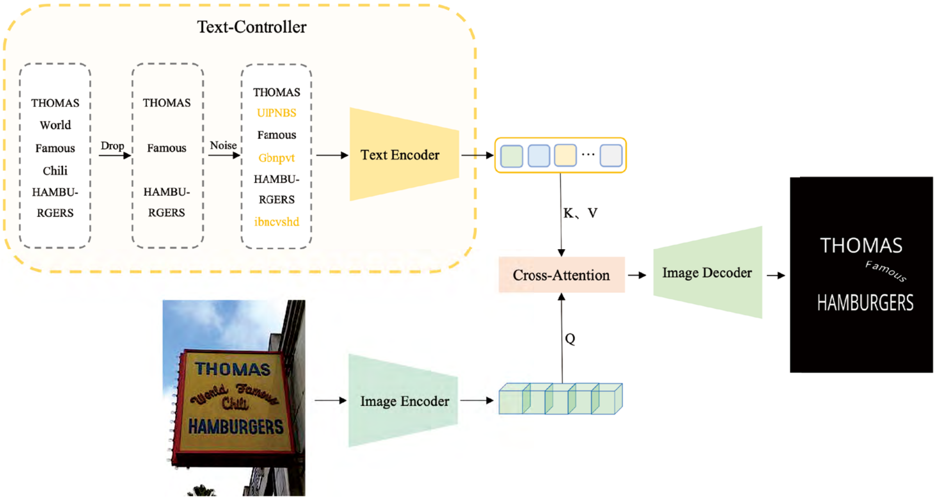


Figure 3. The overall architecture of ODM. The text is encoded by the Text-Controller to get the encoded text features, and the image is encoded by the image encoder to get the encoded image features. The text features and image features interact through cross-attention, and finally output destylization binary image.

Furthermore, some methods directly decode coordinates and recognition results directly by predicting sequence, utilizing the structure of transformer encoder and decoder. Both SPTS [34] and SPTS V2 [29] obtain coordinates by predicting the central point of the text instance, employing an auto-regressive approach to predict the central point and word transcription tokens. UNITS [16] can handle various types of detection formats through prompts, and they can extract text beyond the number of trained text instances. DeepSolo [54] inspired by DETR, enables the decoder to simultaneously perform text detection and recognition.

Vision-Language Pre-training. Modern pre-training methods generally involve MLM [6] or MIM [3, 10, 49], or a combination of both. The MLM task randomly masks a set of text tokens from the input and reconstructs them based on the context around the masked tokens. MIM, on the other hand, randomly masks a percentage of image patches and predicts the RGB values of raw pixels. STKM [45] learns text knowledge from datasets with image-level text annotations, the acquired text knowledge can subsequently be transferred to various text detectors. Inspired by CLIP [35], VLPT [40] adopts fine-grained cross-modality interaction to align unimodal embeddings for learning better representations of backbone via carefully designed pre-training tasks. oCLIP [52] proposes an MLM-based vision-language pre-training method, which has achieved excellent performance in text detection and text spotting tasks. StrucTexTv2 [56] implements pre-training through MIM, where it randomly masks some text word regions in the input images and feeds them into the encoder.

While these methods have shown the effectiveness of pre-training in enhancing OCR performance, they either do not utilize higher-level textual semantic information as input or do not explicitly utilize the positional information of OCR-Text, making it difficult to achieve effective alignment between text and OCR-Text. Inspired by MaskFeat [48], which achieves pre-training by reconstructing the HOG features of masked image regions, and considering that glyph has been employed in some OCR tasks [4, 33, 43, 53] with proven efficacy. We propose utilizing the Text Controller module to reconstruct the corresponding destylized glyph of the text. This approach enables visible alignment between text and OCR-Text. As shown in Fig. 4, our method can better attend to OCR-Text, highlighting its superiority in learning visual text representations for scene text image tasks.

3. Methodology

We introduce ODM, a pre-training technique that effectively aligns text and OCR-Text, leveraging the intrinsic characteristics of OCR-Text and explicitly learning the lo-

(a) Origin (b) No change (c) Drop-Text (d) Noise-Text

Figure 4. Illustration of the proposed Text-Controller Module: The attention heatmap (from the cross-attention layer) of the text branch under different input scenarios is depicted. (a)The original image. (b) The text input consists of three instances: "Rootin", "Ridge", and "Toymakers". (c) The "Toymakers" instance is discarded. (d) A non-existent instance "Sjehf" is added.

Figure 5. The upper row and lower row represent the original images and their corresponding predicted results, respectively.

cation information. The overall pipeline of our method is depicted in Fig. 3. Our network consists of two input branches: the image encoder, which employs ResNet50 [9] to extract features from input images, and the text encoder, designed to extract textual features. By applying cross-attention between the extracted textual and visual features, we generate image features that align with textual features. These aligned features are then processed by a decoder to generate destylization binary images.

3.1. The Text-Controller Module

To address the challenge of the image encoder-decoder architecture in comprehending the significance of characters, we introduce the Text-Controller module to regulate the feature extraction of the image and align the OCR-Text features with textual features in the hidden space.

Drop-Text. General text encoders, such as CLIP [35], are designed to process text descriptions as input, with the primary objective of establishing alignment between text and image features. In the Text-Controller Module, control over the decoder is executed through prompts. Specifically, when a portion of the OCR-Text is input, the model is expected to only reconstruct that part of the binary image, treating the remaining OCR-Text as the background. During the training phase, the input OCR-Text is selected randomly, with a ratio varying from 0% to 100%. This strategy encourages the model to focus more on aligning the text with the corresponding OCR-Text.

Noise-Text. Noise, such as inaccurate labels, can potentially impact the model's training effectiveness. In contrast, we have introduced a concept termed Noise-Text, which leverages noise to augment the model's performance. This approach entails adding noise to the text encoder's input, introducing non-existent OCR-Text by altering its Token encoding value. The integration of these disruptive elements empowers the model to align the features of the text and OCR-Text more effectively, even in more intricate and challenging scenarios.

In Fig. 4, we demonstrate the performance of cross-attention on images when employing different strategies in Text-Controller Module. Our approach achieves alignment between the text and corresponding OCR-Text while remaining unaffected by noisy text.

3.2. OCR-Text Destylization

To ensure that the image encoder can learn the fundamental features of OCR-Text, we have designed a simple decoder to reconstruct the destylized glyph of the text. This decoder comprises only basic FPN [24] layer upsampling and 1*1 convolution.

Fig. 5 demonstrates the predicted results on some real images using our proposed method. The training dataset only consists of SynthText [8] and does not include these real images. From the results, it can be observed that our proposed method effectively achieves OCR-Text destylization.

3.3. Loss Function

The ODM produces a binary image, conceptualizing supervised training as a pixel-level segmentation task. As a result, we optimize the model using a binary cross-entropy loss function for training:

$$\mathcal{L}_{seg} = \frac{1}{N}\sum_{i=1}^{N} -[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \quad (1)$$

where N represents the number of pixels, p_i represents the predicted result, and y_i represents the Ground Truth.

On the other hand, the training output of ODM is to generate a destylization binary image, rather than a text segmentation map that maintains proportional consistency with the original image. Consequently, relying exclusively on pixel-level cross-entropy loss poses a challenge in effectively guiding the model to learn the destylization of characters. To optimize at the feature level, we incorporate the OCR LPIPS loss function proposed by OCR-VQGAN [36]. Specifically, we utilize a well-trained detector (Unet-VGG16[39]) to input both the Ground Truth and the model's predicted binary images. Subsequently, the $\mathcal{L}1$ Loss is computed to foster the learning of a rich latent space and the destylized character glyph images, which is defined as follows:

$$\mathcal{L}_{ocr} = \sum_{l} \frac{1}{H_l W_l} \|VGG_l(\hat{y}) - VGG_l(y)\|_1 \tag{2}$$

where H_l and W_l respectively represent the height and width of the output feature map in the l-th layer, VGG represents a well-trained detector, $\hat{y}$ represents the predicted value, and y represents the Ground Truth.

At the same time, drawing upon the batch-level contrastive loss proposed by CLIP, we expedite the mapping of text and images into the same semantic space by maximizing the similarity among positive samples and minimizing that among negative samples, which is defined as follows:

$$\begin{aligned}\mathcal{L}_{bc} = CE(&\frac{exp(I,T_+)}{\sum_{i=1}^{B} exp(I,T_i)}, y(I)) \\ + CE(&\frac{exp(T,I_+)}{\sum_{i=1}^{B} exp(T,I_i)}, y(T))\end{aligned} \tag{3}$$

where CE represents the cross-entropy loss, I represents image features, T represents text features, I_+ represents the image feature corresponding to the current text feature, T_+ represents the text feature corresponding to the current image feature, and $y(*)$ represents the Ground Truth.

The loss function $\mathcal{L}_{\text{total}}$ is the sum of segmentation loss $\mathcal{L}_{\text{seg}}$, OCR LPIPS loss $\mathcal{L}_{\text{ocr}}$, and batch contrastive loss $\mathcal{L}_{\text{bc}}$, formulated as follows:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_{seg} + \beta\mathcal{L}_{ocr} + \gamma\mathcal{L}_{bc} \tag{4}$$

where the weights α, β, and γ are empirically set to 1, 1, and 0.5, respectively.

3.4. Label Generation

One of the challenges in supervised training for ODM is the creation of data labels, particularly the acquisition of fine-grained pixel-level labels, which is both challenging and costly. To address this, we propose a method for generating pixel-level labels.

For four-point annotations, we calculate the dimensions of the quadrilateral and estimate the size and position of each character based on the number of characters. We then populate these characters using a font file, such as "NotoSans-Regular". For multi-point annotations, we employ the image synthesis approach from ABCNet [27] and utilize the Bezier curves provided by the labels to compute the curvature of the text. Similar to four-point annotations, we determine the size and position of each character from the multi-point annotations and calculate the slope using the top-left point of each character and the subsequent point. Once we obtain the slope, we adjust the orientation of the characters accordingly. Some examples of the generated labels are shown in Fig. 2.

During the process of acquiring pixel-level labels, there may be discrepancies in the spacing between the original OCR-Text and the pixel-level labels. As a result, there might be instances where individual characters in our generated labels do not align precisely with the position of OCR-Text characters in the original image. However, this discrepancy does not impact our task since our approach involves transforming the original image into a destylization binary image rather than performing pixel-by-pixel text segmentation.

4. Experiment

4.1. Datasets

In our experiments, we employ a substantial quantity of publicly accessible datasets, executing pre-training on the SynthText [8] dataset and fine-tuning on text detection and text spotting tasks utilizing datasets like ICDAR15 [15], CTW1500 [26], and TotalText [5].

SynthText is a synthetic dataset for text detection and recognition, consisting of more than 800K synthesized images. Each image is annotated with text regions and their corresponding textual content. The synthetic texts in the dataset are generated by integrating real texts into natural images. The positions, orientations, sizes, and appearances of the synthetic texts are randomly varied to simulate text in real-world scenarios.

ICDAR2015 consists of 1500 images, allocates 1000 images for training, and the remaining 500 for testing. The ICDAR2015 contains different types of text, including horizontal text and multi-oriented text.

CTW1500 is a dataset for curved text instances, consisting of 1500 images, comprising 1500 images, of which 1000 are allocated for training and the remaining 500 for testing.

TotalText is a dataset for curved text instances, consisting of 1555 images, with 1255 designated for training and the remaining 300 for testing.

4.2. Implementation Details

Training. In our proposed method, we employ ResNet50 as the image encoder, with a 6-layer transformer [44] serving as the text encoder. The reconstruction decoder is constructed from FPN and 1x1 convolutions. During the training phase, we configure the image size to 512x512, set the text length for the Text-Controller to 25, and cap the maximum number of text instances at 32. The learning rate is

Table 1. The performance of different models on ICDAR15, CTW1500, and TotalText for scene text detection is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	ICDAR 15			CTW1500			TotalText		
		P	R	H	P	R	H	P	R	H
FCENet[58]†	-	82.43	**88.34**	85.28	**85.7**	80.1	83.1	78.1	**84.85**	81.34
FCENet + ODM	Syn	**91.38**	82.19	**86.54**	84.92	**82.33**	**83.60**	**84.05**	80.67	**82.32**
Δ				+ 1.26			+ 0.5			+ 0.98
DBNet++[23]†	-	91.61	81.46	86.24	79.06	79.50	79.28	81.84	80.22	81.02
DBNet++ + ODM	Syn	**91.87**	**85.94**	**88.81**	**81.99**	**81.98**	**81.97**	**88.01**	**82.25**	**85.03**
Δ				+ 2.57			+ 2.69			+ 4.01

Table 2. Comparison with existing scene text pre-training techniques on PSENet. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	ICDAR 15			CTW1500			TotalText		
		P	R	H	P	R	H	P	R	H
PSENet[46]†	-	83.96	76.36	79.98	80.6	75.6	78.0	75.1	81.8	78.3
PSENet[46]	Syn	86.2	79.4	82.7	81.8	77.8	79.7	87.8	79.0	82.6
PSENet + STKM[45]	Syn	87.78	84.06	85.88	85.08	78.23	81.51	85.08	78.23	81.51
PSENet + oCLIP[52]	Syn	**88.95**	80.98	84.78	86.3	79.6	82.8	90.7	80.8	85.5
PSENet + oCLIP[52]	Web	-	-	-	**87.5**	79.9	83.5	**92.2**	82.4	**87.0**
PSENet + ODM	Syn	88.43	**85.41**	**86.90**	85.86	**85.38**	**85.62**	88.56	**83.37**	85.94
Δ				+ 6.92			+ 7.62			+ 7.64

established at 1e-4. The network training is conducted on 8 A100 GPUs, with each card handling a batch size of 64, resulting in a total batch size of 512. The entire training process spans 100 epochs.

Fine-tuning. We executed evaluations on a variety of OCR tasks, encompassing text detection methods such as DB++ [23][1], FCENet [58][2], and PSENet [46][3], as well as text spotting techniques including ABCNet [27][4], DeepSolo [54][5], and SPTS [34][6]. In the comparative experiments, the only variable altered is the backbone weights, with all other settings maintained consistently.

[1] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/dbnetpp

[2] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/fcenet

[3] https://github.com/open-mmlab/mmocr/tree/main/configs/textdet/psenet

[4] https://github.com/aim-uofa/AdelaiDet/blob/master/configs/BAText

[5] https://github.com/ViTAE-Transformer/DeepSolo

[6] https://github.com/shannanyinxiang/SPTS

Evaluation protocol. We use intersection over union (IoU) to determine whether the model correctly detects the region of text, and we calculate precision (P), recall (R), and Hmean (H) for comparison following ICDAR2015 [15].

4.3. Comparison with Detection Methods

To evaluate the effectiveness of our proposed method on text detection tasks, we conducted comparable experiments with DB++, PSENet, and FCENet. Our model underwent initial pre-training on the SynthText dataset and was subsequently fine-tuned on different datasets. The results of these experiments are shown in Tab. 1 and Tab. 2. The networks utilizing our pre-trained backbone demonstrated improvements compared to their original counterparts, with PSENet showing particularly notable improvement. This suggests that our pre-trained model weights can effectively focus on text regions within the scene, showcasing the significant potential of our pre-trained weights.

Table 3. The performance of different models on ICDAR15 for scene text spotting is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement. 'D' and 'E' represent Detection and End-to-End, respectively. 'S', 'W', and 'G' donate using strong, weak, and generic lexicons, respectively.

Method	PD	D-H	E-S	E-W	E-G
ABCNet[27]†	-	85.22	78.01	73.86	68.09
ABCNet + ODM	Syn	**87.56**	**80.87**	**75.75**	**70.01**
Δ		+ 2.34	+ 2.86	+ 1.89	+ 1.92
DeepSolo[54]†	-	86.44	83.50	79.36	74.19
DeeoSolo + ODM	Syn	**88.08**	**85.83**	**80.67**	**75.2**
Δ		+ 1.64	+ 2.3	+ 1.31	+ 1.01
SPTS[34]†	-	-	77.5	70.2	65.8
SPTS + ODM	Syn	-	**78.8**	**72.1**	**67.8**
Δ			+ 1.3	+ 1.9	+ 2.0

Table 4. The performance of different models on CTW1500 for scene text spotting is presented in the table. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement. 'D' and 'E' represent Detection and End-to-End, respectively. 'N' and 'F' represent None and Full, respectively.

Method	PD	D-H	E-N	E-F
ABCNet[27]†	-	83.69	64.17	78.66
ABCNet + ODM	Syn	**85.40**	**65.06**	**79.79**
Δ		+ 1.71	+ 0.89	+ 1.13
DeepSolo[54]†	-	86.31	76.35	84.31
DeeoSolo + ODM	Syn	**86.58**	**78.07**	**85.65**
Δ		+ 0.27	+ 1.72	+ 1.34
SPTS[34]†	-	-	74.2	82.4
SPTS + ODM	Syn	-	**78.2**	**84.2**
Δ			+ 4.0	+ 1.8

4.4. Comparison with Spotting Methods

To evaluate the effectiveness of our proposed method on text spotting tasks, we conducted comparable experiments with ABCNet, DeepSolo, and SPTS, covering a wide range of task scenarios. Our model was initially pre-trained on the SynthText dataset and then fine-tuned on different datasets. The results of these experiments are presented in Tab. 3 and Tab. 4. The experimental results demonstrate that utilizing our pre-trained weights for fine-tuning significantly improves the performance of the models across various datasets. This indicates that the pre-trained weights acquired through our proposed pre-training method accurately locate the positions of OCR-Text and extract image features that capture the semantic information of the text instance.

Table 5. The performance of different models on LSVT for scene text detection is presented in the table. "PD" and "LSVT" refer to the pre-training dataset and LSVT dataset, respectively. "+ODM" refers to our pre-trained model with 400,000 pseudo-label images in the LSVT dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model, and Δ represents the improvement.

Method	PD	P	R	H
PSENet[46]†	-	64.61	72.29	68.23
PSENet + ODM	LSVT	**66.33**	**73.60**	**69.78**
Δ				+ 1.55
DBNet++[23]†	-	71.21	**79.28**	75.02
DBNet++ + ODM	LSVT	**75.81**	77.16	**76.48**
Δ				+ 1.46

4.5. Weakly Supervised Pre-training

To evaluate the effectiveness of our proposed method in aligning text with images in weakly labeled data, we assess its efficacy. We first employ PPOCRv3 [17] to generate pseudo labels (i.e., text and position in the image) on the 400,000 weakly annotated images from the LSVT [41] dataset. We then generate the corresponding pixel-level destylized annotations using our proposed label generation method. To ensure the quality of pre-training, we only select labels with inference confidence exceeding 0.9 and a text size larger than 32 pixels. We pre-train our model with these generated labels and subsequently fine-tune different scene text detectors using 30,000 fully annotated images from the LSVT dataset. The results presented in Tab. 5 demonstrate that our proposed method, when exclusively utilizing the pseudo labels for pre-training, achieves approximately a 1.5% improvement in the Hmean score after fine-tuning on both DBNet++ and PSENet models. This demonstrates that our method can perform consistently even under weakly supervised scenarios and partially addresses the issue of a large amount of unlabeled data not being able to participate in pre-training.

4.6. Comparison with Pre-training Methods

We compare our proposed method with existing scene text pre-training strategies, including STKM and oCLIP. To investigate the effectiveness of different pre-training objectives, we conducted ablative experiments with PSENet on three datasets: ICDAR15, CTW1500, and TotalText. As shown in Tab. 2, when pre-training on the same set of data, our proposed method outperforms the existing pre-training techniques. Furthermore, when fine-tuning on the

Table 6. Proportion ablation study of the proposed Text-Controller module on CTW1500. "PD" and "Syn" refer to the pre-training dataset and SynthText dataset, respectively. "TP" refers to the selected text proportion. "+ODM" refers to our pre-trained model on the SynthText dataset, which is adopted for fine-tuning. † represents the scores obtained after reproducing the model.

Method	PD	TP	P	R	H
PSENet[46]	-	-	80.6	75.6	78.0
PSENet[46]	Syn	-	81.8	77.8	79.7
PSENet + ODM	Syn	0%	82.41	84.19	83.29
PSENet + ODM	Syn	30%	84.83	82.18	83.48
PSENet + ODM	Syn	50%	83.76	**84.78**	**84.27**
PSENet + ODM	Syn	70%	**85.32**	82.37	83.82

Table 7. Ablation study of our proposed components on TotalText. We fine-tune PSENet by using the pre-trained models with different modules. "TE", "DT", "NT", and "OL" refer to Text Encoder, Drop-Text, Noise-Text, and OCR Loss, respectively.

Method	TE	DT	NT	OL	P	R	H
PSENet[46]					75.10	81.80	78.30
PSENet + ODM					85.08	78.55	81.68
PSENet + ODM	✓				86.66	82.75	84.66
PSENet + ODM	✓	✓			88.06	82.97	85.44
PSENet + ODM	✓		✓		88.10	82.57	85.24
PSENet + ODM	✓	✓	✓		88.17	83.21	85.62
PSENet + ODM	✓	✓	✓	✓	**88.56**	**83.37**	**85.94**

CTW1500 dataset, our proposed method, which was pre-trained on SynthText alone, even surpasses the performance of oCLIP, which was pre-trained on 40 million web images.

4.7. Ablation Experiments

Proportion Ablation We conducted experiments to assess the influence of selected proportions in our Drop-Text and Noise-Text strategies. Four groups of experiments were performed with selected proportions of 0%, 30%, 50%, and 70% respectively. The selected proportions for these strategies were the only variables adjusted, while all other configurations remained constant. We pre-trained the models on the SynthText dataset with varying proportions and transferred the pre-trained weights to fine-tune PSENet on the CTW1500 dataset. The results are presented in Tab. 6. The performance of the model was effectively enhanced by facilitating text-image alignment through the Drop-Text and Noise-Text strategies. The impact of varying proportions on performance was substantial. A small proportion of text instances resulted in minimal performance improvement, possibly due to the insufficient changes in a small number of words to achieve effective text-image alignment. This minimal improvement can be attributed to the errors introduced during training due to the lack of significant changes in the text instances. On the other hand, using a large proportion of text instances also had a limited effect on performance. This can be attributed to the fact that when a substantial number of words are either eliminated or added as noise, the model captures fewer valid features during training, leading to difficulties in convergence and potentially biased learning. Hence, it is important to select an appropriate proportion that ensures the model captures an adequate number of valid features. This helps the model align text with OCR-Text in more complex scenarios and enhances the robustness of the model.

Module Ablation: In our research, we investigated the contributions of several proposed modules. We trained the models with different combinations of these modules on the SynthText dataset and subsequently fine-tuned the pre-trained weights using PSENet on the TotalText dataset, as shown in Tab. 7. The empirical results indicated that for the task of OCR-Text Desytlization, the text feature furnished by the Text-Controller module aids the model in better understanding and locating OCR-Text. Additionally, our proposed Drop-Text and Noise-Text strategies effectively bolster the model's performance by intensifying the alignment between text and OCR-Text.

5. Conclusion

This paper introduces ODM, a novel pre-trained method that aims to transform diverse styles of text found in images into a uniform style based on the text prompt. By leveraging the pre-trained model generated through ODM, it can seamlessly integrate into existing detection and spotting networks, resulting in significant performance improvements. To address the challenge of annotation costs in OCR tasks, we propose a new labeling generation method designed specifically for ODM. Additionally, we introduce the Text-Controller module, which helps regulate the model output and improves its understanding of OCR-Text. By combining these approaches, we enable a larger amount of unlabeled data to be used in the pre-training process, effectively reducing annotation costs. Our extensive ablation and comparative experiments demonstrate the effectiveness and robustness of our model. The results highlight the potential of ODM in OCR pre-training and its valuable contributions to the advancement of scene text detection and spotting tasks. Looking ahead, we plan to explore the potential of this method in other domains, such as document analysis, handwriting recognition, and other complex scene text scenarios.

References

[1] Hangbo Bao, Li Dong, Songhao Piao, and Furu Wei. Beit: Bert pre-training of image transformers. *arXiv preprint arXiv:2106.08254*, 2021.

[2] Fedor Borisyuk, Albert Gordo, and Viswanath Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *Proceedings of the 24th ACM SIGKDD in-*

ternational conference on knowledge discovery & data mining, pages 71–79, 2018.

[3] Jun Chen, Ming Hu, Boyang Li, and Mohamed Elhoseiny. Efficient self-supervised vision pretraining with local masked reconstruction. *arXiv preprint arXiv:2206.00790*, 2022.

[4] Jingye Chen, Yupan Huang, Tengchao Lv, Lei Cui, Qifeng Chen, and Furu Wei. Textdiffuser: Diffusion models as text painters. *Advances in Neural Information Processing Systems*, 36, 2024.

[5] Chee-Kheng Ch'ng, Chee Seng Chan, and Cheng-Lin Liu. Total-text: toward orientation robustness in scene text detection. *International Journal on Document Analysis and Recognition (IJDAR)*, 23(1):31–52, 2020.

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[7] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9076–9085, 2019.

[8] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2315–2324, 2016.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Mae: Masked autoencoders are scalable vision learners. *arXiv preprint arXiv:2111.06377*, 2021.

[11] Minghang He, Minghui Liao, Zhibo Yang, Humen Zhong, Jun Tang, Wenqing Cheng, Cong Yao, Yongpan Wang, and Xiang Bai. Most: A multi-oriented scene text detector with localization refinement. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8813–8822, 2021.

[12] Tong He, Zhi Tian, Weilin Huang, Chunhua Shen, Yu Qiao, and Changming Sun. An end-to-end textspotter with explicit alignment and attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5020–5029, 2018.

[13] Mingxin Huang, Yuliang Liu, Zhenghao Peng, Chongyu Liu, Dahua Lin, Shenggao Zhu, Nicholas Yuan, Kai Ding, and Lianwen Jin. Swintextspotter: Scene text spotting via better synergy between text detection and text recognition. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4593–4603, 2022.

[14] Mingxin Huang, Jiaxin Zhang, Dezhi Peng, Hao Lu, Can Huang, Yuliang Liu, Xiang Bai, and Lianwen Jin. Estextspotter: Towards better scene text spotting with explicit synergy in transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19495–19505, 2023.

[15] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th international conference on document analysis and recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[16] Taeho Kil, Seonghyeon Kim, Sukmin Seo, Yoonsik Kim, and Daehee Kim. Towards unified scene text spotting based on sequence generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15223–15232, 2023.

[17] Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, et al. Pp-ocrv3: More attempts for the improvement of ultra lightweight ocr system. *arXiv preprint arXiv:2206.03001*, 2022.

[18] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[19] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: A fast text detector with a single deep neural network. In *Proceedings of the AAAI conference on artificial intelligence*, 2017.

[20] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *IEEE transactions on image processing*, 27(8):3676–3690, 2018.

[21] Minghui Liao, Guan Pang, Jing Huang, Tal Hassner, and Xiang Bai. Mask textspotter v3: Segmentation proposal network for robust scene text spotting. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI 16*, pages 706–722. Springer, 2020.

[22] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai. Real-time scene text detection with differentiable binarization. In *Proceedings of the AAAI conference on artificial intelligence*, pages 11474–11481, 2020.

[23] Minghui Liao, Zhisheng Zou, Zhaoyi Wan, Cong Yao, and Xiang Bai. Real-time scene text detection with differentiable binarization and adaptive scale fusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):919–931, 2022.

[24] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017.

[25] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5676–5685, 2018.

[26] Yuliang Liu, Lianwen Jin, Shuaitao Zhang, Canjie Luo, and Sheng Zhang. Curved scene text detection via transverse and longitudinal sequence connection. *Pattern Recognition*, 90: 337–345, 2019.

[27] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9809–9818, 2020.

[28] Yuliang Liu, Chunhua Shen, Lianwen Jin, Tong He, Peng Chen, Chongyu Liu, and Hao Chen. Abcnet v2: Adaptive bezier-curve network for real-time end-to-end text spotting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(11):8048–8064, 2021.

[29] Yuliang Liu, Jiaxin Zhang, Dezhi Peng, Mingxin Huang, Xinyu Wang, Jingqun Tang, Can Huang, Dahua Lin, Chunhua Shen, Xiang Bai, et al. Spts v2: single-point scene text spotting. *arXiv preprint arXiv:2301.01635*, 2023.

[30] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In *Proceedings of the European conference on computer vision (ECCV)*, pages 67–83, 2018.

[31] Pengyuan Lyu, Minghui Liao, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(2): 532–548, 2021.

[32] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *IEEE transactions on multimedia*, 20(11):3111–3122, 2018.

[33] Jian Ma, Mingjun Zhao, Chen Chen, Ruichen Wang, Di Niu, Haonan Lu, and Xiaodong Lin. Glyphdraw: Learning to draw chinese characters in image synthesis models coherently. *arXiv preprint arXiv:2303.17870*, 2023.

[34] Dezhi Peng, Xinyu Wang, Yuliang Liu, Jiaxin Zhang, Mingxin Huang, Songxuan Lai, Jing Li, Shenggao Zhu, Dahua Lin, Chunhua Shen, et al. Spts: single-point text spotting. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4272–4281, 2022.

[35] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021.

[36] Juan A Rodriguez, David Vazquez, Issam Laradji, Marco Pedersoli, and Pau Rodriguez. Ocr-vqgan: Taming text-within-image generation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3689–3698, 2023.

[37] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115:211–252, 2015.

[38] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2550–2558, 2017.

[39] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[40] Sibo Song, Jianqiang Wan, Zhibo Yang, Jun Tang, Wenqing Cheng, Xiang Bai, and Cong Yao. Vision-language pre-training for boosting scene text detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15681–15691, 2022.

[41] Yipeng Sun, Zihan Ni, Chee-Kheng Chng, Yuliang Liu, Canjie Luo, Chun Chet Ng, Junyu Han, Errui Ding, Jingtuo Liu, Dimosthenis Karatzas, et al. Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1557–1562. IEEE, 2019.

[42] Jingqun Tang, Wenqing Zhang, Hongye Liu, MingKun Yang, Bo Jiang, Guanglong Hu, and Xiang Bai. Few could be better than all: Feature sampling and grouping for scene text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4563–4572, 2022.

[43] Yuxiang Tuo, Wangmeng Xiang, Jun-Yan He, Yifeng Geng, and Xuansong Xie. Anytext: Multilingual visual text generation and editing. *arXiv preprint arXiv:2311.03054*, 2023.

[44] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[45] Qi Wan, Haoqin Ji, and Linlin Shen. Self-attention based text knowledge mining for text detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5983–5992, 2021.

[46] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9336–9345, 2019.

[47] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8440–8449, 2019.

[48] Chen Wei, Haoqi Fan, Saining Xie, Chao-Yuan Wu, Alan Yuille, and Christoph Feichtenhofer. Masked feature prediction for self-supervised visual pre-training. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14668–14678, 2022.

[49] Zhenda Xie, Zheng Zhang, Yue Cao, Yutong Lin, Jianmin Bao, Zhuliang Yao, Qi Dai, and Han Hu. Simmim: A simple framework for masked image modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9653–9663, 2022.

[50] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *IEEE Transactions on Image Processing*, 28(11):5566–5579, 2019.

[51] Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, et al. Layoutlmv2: Multi-modal pre-training

for visually-rich document understanding. *arXiv preprint arXiv:2012.14740*, 2020.

[52] Chuhui Xue, Wenqing Zhang, Yu Hao, Shijian Lu, Philip HS Torr, and Song Bai. Language matters: A weakly supervised vision-language pre-training approach for scene text detection and spotting. In *European Conference on Computer Vision*, pages 284–302. Springer, 2022.

[53] Mingkun Yang, Biao Yang, Minghui Liao, Yingying Zhu, and Xiang Bai. Class-aware mask-guided feature refinement for scene text recognition. *Pattern Recognition*, 149:110244, 2024.

[54] Maoyuan Ye, Jing Zhang, Shanshan Zhao, Juhua Liu, Tongliang Liu, Bo Du, and Dacheng Tao. Deepsolo: Let transformer decoder with explicit points solo for text spotting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19348–19357, 2023.

[55] Wenwen Yu, Yuliang Liu, Wei Hua, Deqiang Jiang, Bo Ren, and Xiang Bai. Turning a clip model into a scene text detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6978–6988, 2023.

[56] Yuechen Yu, Yulin Li, Chengquan Zhang, Xiaoqiang Zhang, Zengyuan Guo, Xiameng Qin, Kun Yao, Junyu Han, Errui Ding, and Jingdong Wang. Structextv2: Masked visual-textual prediction for document image pre-training. *arXiv preprint arXiv:2303.00289*, 2023.

[57] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 5551–5560, 2017.

[58] Yiqin Zhu, Jianyong Chen, Lingyu Liang, Zhanghui Kuang, Lianwen Jin, and Wayne Zhang. Fourier contour embedding for arbitrary-shaped text detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3123–3131, 2021.

Speculative Decoding via Early-exiting for Faster LLM Inference with Thompson Sampling Control Mechanism

Jiahao Liu[1], Qifan Wang[2], Jingang Wang[1]*, Xunliang Cai[1]
[1]Meituan; [2]Meta AI
{liujiahao12,wangjingang02,caixunliang}@meituan.com
wqfcr@fb.com

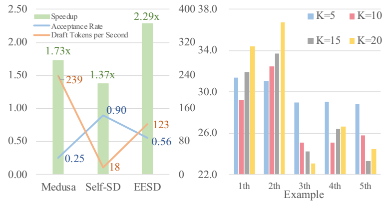
Abstract

The recent advancements in large language models (LLMs) have been extraordinary, yet the escalating inference costs associated with them present challenges in real-world applications. To address these challenges, we propose a novel approach called Early-exiting Speculative Decoding (EESD) with lossless acceleration. Specifically, EESD utilizes a segment of the LLM to generate draft tokens, incorporating Early-exiting structures after the first N layers. To enhance the quality of draft tokens, a self-distillation method is integrated. This early-exiting design not only reduces deployment and training costs but also significantly accelerates the token generation speed. Moreover, we introduce a novel sampling mechanism that leverages Thompson Sampling to regulate the generation processes, automatically determining the quantity of draft tokens in each round. The original LLM is then employed to validate these draft tokens through a single forward pass, and thus guarantees that the final output text maintains a distribution consistent with vanilla auto-regressive decoding. The experimental results on both 13B and 70B models demonstrate that our approach decodes tokens at a markedly accelerated rate compared to prior methods, showing the effectiveness of our approach.

1 Introduction

Large Language Models (LLMs) excel in various NLP tasks due to their immense parameters and complex network (OpenAI, 2023; Chowdhery et al., 2023; Touvron et al., 2023a,b). However, these models generate tokens one-by-one in an auto-regressive manner during inference, making the generation exceedingly resource-intensive and time-consuming. To overcome this bottleneck, researchers have introduced an effective decoding technique - Speculative Decoding (SD) (Leviathan et al., 2023; Chen et al., 2023; Miao et al., 2023).

* Jingang Wang is the corresponding author.



(a) End-to-End Speedup (b) Cost vs. drafting steps

Figure 1: Experimental results using LLaMA-2-70B on the Gsm8k. (a) Speedup comparison with Medusa (Cai et al., 2023) and Self-SD (Zhang et al., 2023b). EESD achieves a highest speedup with a best tradeoff between draft token generation speed and acceptance rate. (b) Generation costs (seconds) with different drafting steps (K) in randomly select five samples from Gsm8k. The optimal value of K varies across different samples, indicating that a fixed K value for all samples is not ideal.

SD essentially introduces two models, a small model (the draft model) which is used to concurrently generate multiple draft tokens, and the original LLM (the target model) which is employed for draft token verification. In this way, SD maintains the same performance as the auto-regressive decoding while boosting the inference speed.

Compared to vanilla Speculative Decoding (Leviathan et al., 2023; Chen et al., 2023), several advanced models such as Medusa (Cai et al., 2023) and Self-SD (Zhang et al., 2023b) have been introduced, which only require deploying one LLM instead of two models, resulting in fewer resources required for both training and deployment. While these approaches achieve promising results, there are two main limitations. *First*, they fail to optimize the trade-off between the quality and speed of the draft token generation. For example, as shown in Figure 1a, while Medusa can generate draft tokens rapidly, the quality of these tokens tends to be subpar[1]. On the other hand, Self-SD manages to produce high quality draft tokens but does so at a much slower speed, resulting in lower overall speedup. *Second*, most SD approaches commence verification after generating a pre-defined length of draft tokens (referred to as drafting steps K). The choice of K significantly influences the acceleration of the inference process. Typically, larger drafting steps result in faster end-to-end generation, but there is a potential trade-off as the acceptance rate may decrease if the quality of the longer draft sequence is not high. As illustrated in Figure 1b, the optimal value of K varies across different examples. This variation suggests that utilizing a fixed K may not yield the most effective strategy. Instead, an adaptive method is preferable to determine when to terminate the drafting process.

To address these challenges, in this paper, we propose a novel Early-Exiting Speculative Decoding method, named EESD, to facilitate efficient and qualitative generation of draft tokens. Specifically, EESD introduce an Early-exiting layer that is superimposed on the first-N layers of the LLM, which has shown powerful predictive potential in previous research (Bae et al., 2023; Schuster et al., 2022). A self-distillation method is further employed to enhance the learning of the Early-exiting layer. To identify the optimal drafting steps, we reformulate the task of determining the length of draft token generation as a multi-armed bandit (MAB) problem, and propose a novel Control Mechanism based on Thompson Sampling (TS) that is well-studied for estimating unknown parameters and facilitating optimal decision making. Comprehensive evaluations on both 13B and 70B models demonstrate the superior performances of our approach over several baselines. The main contributions of this paper are summarized as follows:

- We introduce a novel Early-exiting framework for generating draft tokens, which allows a single LLM to fulfill the drafting and verification stages. We train it using self-distillation. Our investigations indicate that this framework strikes an excellent balance between the quality and speed of draft token generation.
- We conceptualize the generation length of draft tokens as a MAB problem and propose a novel control mechanism based on Thompson Sampling, which leverages sampling to devise an optimal strategy.
- We conducted extensive experiments on three benchmarks. The results affirm that EESD can significantly improve the model's inference speed, outperforming existing SD methods.

[1]We use the acceptance rate to represent the quality of the draft tokens, which is percent of draft tokens are accepted by the target model during the verification.

2 Related Work

LLM Compression The central objective of model compression is to alleviate computational demands and enhance inference speed. The research on LLM compression mainly includes three directions, including knowledge distillation (Zhang et al., 2023a; Li et al., 2023; Gu et al., 2023), network pruning (Ma et al., 2023; Xia et al., 2023; Frantar and Alistarh, 2023) and quantization (Xiao et al., 2023; Liu et al., 2023; Frantar et al., 2022; Lin et al., 2023; Gong et al., 2023). The methods mentioned above work by reducing the model's footprint, thereby decreasing memory demand and enhancing the speed of inference. However, these methods sacrifices a degree of LLM's capability.

Efficient Decoding Leviathan et al. (2023); Chen et al. (2023) propose a method that uses a small model to generate draft tokens and then uses LLM for verification, which accelerates the decoding process while guaranteeing lossless outputs, named as Speculative Decoding. However, some researchers suggest that the extra small model is not essential for SD. For instance, Medusa (Cai et al., 2023) generates draft tokens by leveraging additional parameters instead of small model, while Self-SD (Zhang et al., 2023b) uses the substructure of LLM to generate draft tokens. In addition, He et al. (2023) unveil a method that replaces the generation of draft tokens with a large text database. In the other hand, some researchers introduce an Early-exiting method. This method dynamically modifies the depth of the decoder for each token generation, making predictions at an intermediate layer (Teerapittayanon et al., 2016; Elbayad et al., 2020). Furthermore, Bae et al. (2023) propose a new Early-exiting method that incorporates a shallow-deep module and synchronized parallel decoding.

3 Methodology

The overall model architecture of EESD is illustrated in Figure 2. Essentially, our model is com-

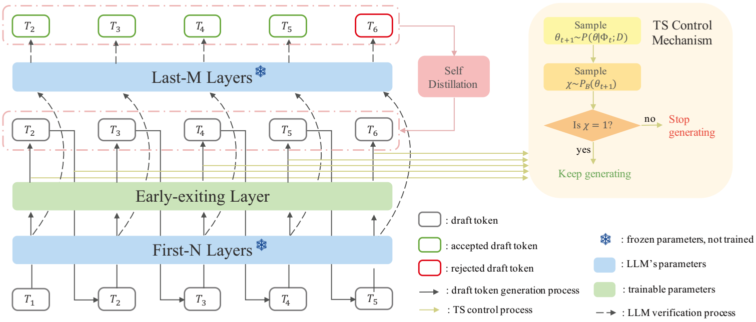


Figure 2: The framework of EESD which consists of three components: (1) Early-exiting layer which generate draft tokens efficiently and effectively; (2) Self-distillation which distills knowledge from the LLM (the target model); (3) TS control mechanism which can predict the optimal timing of terminating the draft token generation in each round. We divide the LLM (the target model) into two parts: the first-N layers and the last-M layers.

posed of three key components. 1) the Early-exiting layer that built on top of the first few layers of the LLM as a draft model; 2) the self-distillation method to enhance the learning of the draft model and boost the text generation quality; and 3) the Thompson Sampling control mechanism that adaptively determines the length of the draft tokens conditioned on its quality. We present the details of these components in the following sections.

3.1 Early-exiting Layer

Most previous methods (Bae et al., 2023; Kavehzadeh et al., 2023) use non-continuous sub-network of original LLM (target model) as their draft model. In this work, we utilize the continuous first-N layers approach, which yields one significant advantage: the kv-cache of the draft model and the target model can share the first-N layers, thereby trimming redundant computation. Concreately, we formulated an Early-exiting Layer with the computation process as elucidated below,

$$p(y_t) = softmax(W^T Transformer^e(H_t^N)), \quad (1)$$

where H_t^N represents the hidden state of the N-th layer, which is calculated from the first-N layers of the origin LLM. And t represents t-th token. $p(y_t)$ is obtained from H_t^N through one layer of transformer. For LLaMA model, we also add RMSNorm layer before the output prediction head, which is $RMSNorm(Transformer^e(H_t^N))$.

As mentioned above, we incorporate an learnable Transformer layer subsequent to the first-N layers, and train this layer and W in Eq.(1) (RMSNorm parameters are trained as well for LLaMA model). In order to speed up the model convergence, we initialize the $Transformer^e$ and W with the last layer and predict head of the original LLM respectively. Since the training is confined to only a single Transformer layer and W, with the first-N layers being frozen, this approach dramatically reduce the computational resources.

3.2 Self-Distillation

To further enhance the effectiveness of the draft model, we employ self-distillation to learn the knowledge from the LLM. The key idea is that there is a large amount of valuable data used during the training of the LLM. However, it is usually impossible to obtain these original data as most of them are not directly accessible. Therefore, we propose to bridge the gap with self-distillation, which guides the learning of the early-exiting layer by transfering the knowledge from the generated text of the LLM. Specifically, Liu et al. (2023) suggest that a hybrid generation strategy of greedy and sampling is effective, and we adopt this approach for text generation from LLM. It is worth noting that the text generated by the LLM may contain certain lower-quality samples. We thus retain a subset of the open-source data for training purposes. The parameters of the Early-exiting Layer are trained utilizing an amalgamation of data generated by the LLM and open-source datasets, with the cross-entropy loss between the prediction of it and the ground truth of mixed datasets.

3.3 Thompson Sampling Control Mechanism

The above methods can effectively improve the quality and speed of draft token generation. However, as we mentioned in the introduction, a predetermined drafting step is not a good strategy. Consequently, we view the controlling draft model generation as a MAB problem. Specifically, we view it as a Bernoulli process, where the model independently determines whether to continue draft token generating, denoted as $P_B(\theta)$. And the probability θ is uncertain, related to the input sample. The Thompson Sampling (TS) method can better estimate unknown variables through balancing exploration and exploitation (Slivkins, 2019). As illustrated in Algorithm 1, we utilize TS algorithm to adaptively estimate θ. The crux of the TS algorithm involves modeling uncertain parameters θ as a posterior distribution using Bayesian theory, i.e. $P(\theta|D)$, with D representing observed environmental samples. The core of this algorithm lies in designing a reasonable posterior distribution, which we will detail in the following chapters.

TS with Beta Distribution Considering that the sample adheres to the Bernoulli distribution, we adopt the conjugate distribution approach and select the Beta distribution as posterior distribution. This setup means the prior and posterior distributions share the same distribution function but with different parameters, which greatly simplifies the computational process. The probability density function of the Beta distribution is as follows:

$$Beta(\theta; \alpha, \beta) = \frac{1}{B(\alpha, \beta)} \theta^{\alpha-1}(1-\theta)^{\beta-1}, \quad (2)$$

where $B(\alpha, \beta)$ is a standardization function. The Beta distribution has two parameters, α and β, so $\Phi = \{\alpha, \beta\}$ and $\Phi_0 = \{\alpha_0, \beta_0\}$ in Algorithm 1. According to Bayesian inference, we can update parameters α and β according to the following formula, which is the 18-th step in Algorithm 1,

$$\alpha_t = \alpha_{\{t-|Q_v|\}} + r, \quad (3)$$

$$\beta_t = \beta_{\{t-|Q_v|\}} + (n - r), \quad (4)$$

where r represents the number of successful experiments in the observed samples, and n represents the total number of experiments. And in Algorithm 1, the value of r is set to $|Q_v| - 1$, indicating that the draft model should continue to generate on this set of tokens (i.e., $\chi = 1$). And the value of n is set to $min(|Q_v| + 1, |Q_d|)$, indicating the number of tokens that have been validated by the target model. Because we stop verifying when we encounter an inconsistent token, subsequent tokens are not considered.

Algorithm 1 TS Control Algorithm

Require: Target Model M_t; Draft Model M_d; Max Generation Length L; Hyperparameters Φ_0; Input Prompt $\{x_0, ..., x_n\}$.
1: Initialize prior probability $P(\theta|\Phi_0)$ according to user-set hyperparameters Φ_0.
2: Initialize the result set $Q_g \leftarrow \{x_0, ..., x_n\}$ and $t \leftarrow 0$.
3: **while** t < L **do**
4:   Initialize the draft model result set and i, $Q_d \leftarrow Null, i \leftarrow 0$.
5:   **while** t + i < L **do**
6:     Get new token $x_i \leftarrow M_d(Q_g \bigcup Q_d)$.
7:     Add token x_i to set Q_d.
8:     Sample θ_{t+i} from $P(\theta|\Phi_t; D)$.
9:     Sample $\chi \in \{0, 1\}$ from Bernoulli distribution $P_B(\theta_{t+i})$.
10:     $i \leftarrow i + 1$.
11:     **if** $\chi = 0$ **then**
12:       Break
13:     **end if**
14:   **end while**
15:   Verify the results Q_d by Target Model M_t and get Q_v received by M_t, $Q_v \subseteq Q_d$.
16:   Add the set Q_v to Q_g, $Q_g \leftarrow Q_g \bigcup Q_v$.
17:   Update t according to length of Q_v, $t \leftarrow t + |Q_v|$.
18:   Update the parameters Φ_t of the posterior distribution.
19: **end while**
20: **return** Q_g

TS with Calibration In the prior segment, we introduce a TS algorithm with Beta distribution to improve the estimation of θ. However, according to MAB theory, initial phases are more exploration-focused, which may result in less accurate θ estimations (Ou et al., 2019; Peng et al., 2019). To alleviate this issue, we propose a novel hybrid method that combines *Model Prediction* and *Sampling Prediction*. We rely more on model prediction to mitigate inaccuracies from initial exploration. As the sampling prediction begins to converge later, we calibrate the model prediction with sampling prediction to achieve a more precise θ.

We train a single-layer to predict the value of θ. The computation formula for this is as follows,

$$\theta_M^{t+i} = Sigmoid(W_p(W^i H_t^{(T)}, H_{t+i}^{(D)})), \quad (5)$$

where t is the number of tokens that have already been generated, and i is the number of new tokens generated by the draft model in the current loop. $H_t^{(T)}$ represents the hidden state of the LLM (target model) at position t in the last layer, while the corresponding $H_{t+i}^{(D)}$ represents the hidden state of the draft model at position $t+i$ in the last layer. W^i is the transformation matrix at the i-th position for target model, and considering that i might be very large, we have restricted the number of $W^i \in \mathbb{R}^{d\times d}$, i.e. $i = min(i, 10)$. We sample a portion of training dataset to train the parameters $\{W^1, W^2, ..., W^{10}\}$ and $W_p \in \mathbb{R}^{2\times 2d}$. The labels for this data are acquired by comparing the tokens produced by both the target and draft models, signifying the true value of χ. Following this, we update the parameters using cross-entropy loss.

According to the central limit theorem, when the sample size is sufficiently large, the sample mean adheres to a Gaussian distribution. Therefore, we make an assumption that sample mean $\widetilde{\chi}$ of χ in one drafting round follows a Gaussian distribution, i.e. $\widetilde{\chi} \sim \mathcal{N}(\mu, \sigma_S^2)$. As supported by Bayesian theory, when the random variable follows a Gaussian distribution with a known variance but an unknown mean and the prior distribution is also a Gaussian distribution, it satisfies the conjugate distribution. Consequently, we define μ to follow a Gaussian distribution with the model's predict score as mean and predict error as variance, $\mu \sim \mathcal{N}(\theta_M, \sigma_M^2)$. In Algorithm 1, we set $\Phi_0 = \{\sigma_M, \sigma_S, \hat{\theta_0}\}$, where σ_M , σ_S and $\hat{\theta_0}$ are hyperparameters set by the user. In Step 8 of Algorithm 1, we sample θ value from Gaussian distribution, $\theta_{t+i} \sim \mathcal{N}(\mu_{t+i}, \sigma_{t+i}^2)$, and compute the values of μ and σ using the formula provided,

$$\mu_{t+i} = \frac{\sigma_S^2}{n\sigma_M^2 + \sigma_S^2}\theta_M^{t+i} + \frac{n\sigma_M^2}{n\sigma_M^2 + \sigma_S^2}\hat{\theta}_t, \tag{6}$$

$$\frac{1}{\sigma_{t+1}^2} = \frac{1}{\sigma_M^2} + \frac{n}{\sigma_S^2}, \tag{7}$$

Where n is verification times. We update parameter Φ based on the following formula in step 18,

$$\hat{\theta}_t = \frac{\hat{\theta}_{\{t-|Q_v|\}} * (t - |Q_v| + 1) + |Q_v|}{t+1}, \tag{8}$$

For a more detail, please refer to Appendix B.

4 Experiment

4.1 Setup

Training stage We randomly extract 100,000 samples from the SlimPajama (Soboleva et al., 2023) to train LLaMA-2-70B, LLaMA-2-13B and CodeLLaMA-2-13B. And use the ShareGPT[2] dataset to train LLaMA-2-70B-chat and Vicuna-13B (Zheng et al., 2023a). We choose the first-5 layers as the draft model for 70B models and first-3 for 13B models. For fair comparison, we train our model and Medusa (Cai et al., 2023) using the same data and set Medusa head at 4. More training details can be found in Appendix C

Evaluation stage We conduct experiments on three benchmarks under 1-shot setting: Gsm8k (Cobbe et al., 2021), XSum (Narayan et al., 2018) and Humaneval (Chen et al., 2021). And we also evaluate EESD on the MT-bench (Zheng et al., 2023b). We randomly select 500 instances from the test set for evaluation. And we set final output length at 512 and batch size at 1. We set the drafting step K at 10 for Vanilla SD (Chen et al., 2023) and Self SD (Zhang et al., 2023b). The reported results are the average of 10 different runs. We have only conducted on greedy generation[3], as the findings from the top-p sampling exhibit similar trends.

Metrics We propose a Harmonic Mean (HM) to assess the quality of the draft tokens and strategy for generating them, while the specific calculation formula is as $S = \frac{2 * v_d * r_d}{v_d + r_d} * 100\%$, where v_d indicates the percent of draft tokens that are accepted by the target model, and r_d represents the proportion of tokens that come from the draft model. More detailed explanation in Appendix D. Due to the verification process, all baseline and EESD methods can assure that the generation results are identical to the original LLM, hence we only need to compare their speedup.

All experiments are conducted on NVIDIA A100-80GB GPUs.

4.2 Main Results

We report evaluation results for Gsm8k and XSum in Table 1, for Humaneval in Table 2 and for MT-bench in Table 3. As shown in Table 1, 2 and 3, it is clear that EESD significantly outperforms the previous methods on both 13B and 70B models, especially on LLaMA-2-70B, which demonstrates the effectiveness of our approach. There are several key observations from these results. **First**, we observe that EESD can yield 2.45× times speedup on CodeLLaMA-2-13B for coding task, suggesting

[2]https://huggingface.co/datasets/Aeala/ShareGPT_Vicuna_unfiltered

[3]For all experiments, we only generate one top-1 draft token candidate in Step 6 of Algorithm 1, and retain the result consistent with the target model's top-1 token on verifying at Step 15 of Algorithm 1.

Target Model	Method	Draft Model	Trainable Params	Deployment Params	Gsm8k HM	Gsm8k Speedup	XSum HM	XSum Speedup
LLaMA-2-70B	Vanilla SD	LLaMA-2-7B	7B†	77B	80.35	1.88×	67.30	1.46×
	Self SD*	Self	-	70B	78.64	1.37×	68.60	1.23×
	Medusa*	Self	1.32B	71.3B	33.75	1.73×	25.69	1.42×
	EESD (+Beta-TS)	Self	1.12B	71.1B	58.79	2.13×	51.91	1.80×
	EESD (+Cali-TS)	Self	1.12B+0.67B‡	71.8B	62.25	**2.29**×	53.41	**1.86**×
LLaMA-2-70B-chat	Vanilla SD	LLaMA-2-7B-chat	7B†	77B	63.90	1.44×	62.75	1.39×
	Self SD*	Self	-	70B	67.99	1.13×	68.38	1.16×
	Medusa*	Self	1.32B	71.3B	22.86	1.42×	17.02	1.20×
	EESD (+Beta-TS)	Self	1.12B	71.1B	47.76	1.79×	40.73	1.51×
	EESD (+Cali-TS)	Self	1.12B+0.67B‡	71.8B	48.23	**1.82**×	41.85	**1.55**×
LLaMA-2-13B	Vanilla SD	LLaMA-2-7B	7B†	20B	84.59	0.96×	75.63	0.77×
	Vanilla SD	TinyLLaMA-1.1B[4]	1.1B†	14.1B	82.49	1.19×	75.42	1.05×
	Self SD*	Self	-	13B	80.53	1.37×	77.61	1.35×
	Medusa*	Self	760M	13.8B	31.71	1.77×	25.03	1.53×
	EESD (+Beta-TS)	Self	481M	13.5B	57.22	1.91×	55.46	1.84×
	EESD (+Cali-TS)	Self	481M+262M‡	13.7B	58.97	**2.04**×	56.45	**1.92**×
Vicuna-13B	Vanilla SD	Vicuna-7B	7B†	20B	64.22	0.69×	53.77	0.55×
	Vanilla SD	Vicuna-68M[5]	68M†	13.1B	26.87	1.28×	23.05	1.17×
	Self SD*	Self	-	13B	68.07	1.24×	60.38	1.12×
	Medusa*	Self	760M	13.8B	26.08	1.53×	15.55	1.21×
	EESD (+Beta-TS)	Self	481M	13.5B	43.01	1.57×	32.23	1.25×
	EESD (+Cali-TS)	Self	481M+262M‡	13.7B	43.53	**1.59**×	32.50	**1.27**×

Table 1: Evaluation on Gsm8k and XSum with different methods. **Speedup** signifies the acceleration effect in comparison with the auto-regression method. † For all Vanilla SD, we use the Homologous small model as the draft model, and we think that this small model needs to be trained. ‡ Model prediction requires the training of additional parameters. * Due to differences in experimental setups, our results are slightly different from their paper. Nevertheless, all experimental results for both the baselines and EESD are obtained under the same settings, ensuring a fair and consistent comparison. Results are statistically significant with respect to all baselines (all p-value < 0.005).

Target Model	Method	Trainable Params	Humaneval HM	Humaneval Speedup
LLaMA-2-13B	Vanilla SD	7B†	87.32	0.97×
	Self SD	-	79.93	1.36×
	Medusa	760M	26.67	1.61×
	EESD (+Beta-TS)	481M	61.43	2.08×
	EESD (+Cali-TS)	481M+262M	62.87	**2.15**×
CodeLLaMA-2-13B	Vanilla SD	7B†	91.12	1.09×
	Self SD	-	83.51	1.38×
	Medusa	761M	49.14	1.97×
	EESD (+Beta-TS)	481M	68.94	2.21×
	EESD (+Cali-TS)	481M+262M	70.15	**2.45**×

Table 2: Evaluation on Humaneval with different speculative decoding methods. † We use LLaMA-2-7B and CodeLLaMA-2-7B as draft models, respectively. Results are statistically significant with respect to all baselines (all p-value < 0.005).

Target Model	Method	Trainable Params	Humaneval HM	Humaneval Speedup
LLaMA-2-13B	Vanilla SD	7B†	90.01	1.02×
	Self SD	-	89.25	1.58×
	Medusa	760M	43.08	1.89×
	EESD (+Beta-TS)	481M	62.15	2.03×
	EESD (+Cali-TS)	481M+262M	64.31	**2.11**×
Vicuna-13B	Vanilla SD	7B†	81.50	0.91×
	Self SD	-	86.20	1.46×
	Medusa	761M	30.11	1.71×
	EESD (+Beta-TS)	481M	52.82	1.82×
	EESD (+Cali-TS)	481M+262M	54.83	**1.89**×

Table 3: Evaluation on MT-bench with different speculative decoding methods. † We use LLaMA-2-7B and Vicuna-7B as draft models, respectively. Results are statistically significant with respect to all baselines (all p-value < 0.005).

our method exhibits particular effectiveness within this domain. **Second**, compared to Vanilla SD and Medusa, EESD shows superior results with fewer training and deployment parameters. For instance, EESD achieves up to 2.13× and 1.80× times faster speeds on llama-2-70b model with just 1.12B parameters being trained. While we introduce an additional training process as compared to Self-SD, we manage to significantly improve speed effectiveness, utilizing minimal training resources. **Third**, we discover that a stronger capability of the draft model, indicated by a higher HM value, does not necessarily result in higher speedup. It is essential to consider the generation speed of draft tokens, and our approach can strike an optimal balance between the two to achieve higher speedup (detailed in Appendix D).

5 Analysis and Discussion

5.1 Ablation Study

To elucidate the impact of different components within our approach, we conduct a series of ablation studies. In Table 4, we exhibit experimental

[4] https://huggingface.co/TinyLlama/TinyLlama-1.1B-intermediate-step-1431k-3T

[5] https://huggingface.co/double7/vicuna-68m

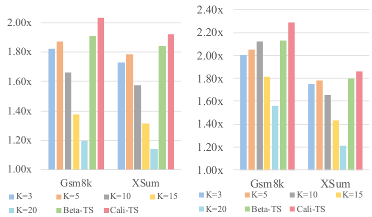
Method	Gsm8k (HM)	XSum (HM)
Vanilla SD (LLaMA-2-7B)	0.96×	0.77×
Vanilla SD (TinyLLaMA-1.1B)	1.19×	1.05×
EESD (Beta-TS)	1.91× (57.22)	1.84× (55.46)
w/o Early-exiting Layer	1.18× (23.69)	1.15× (23.10)
w/o Self-Distillation	1.82× (54.12)	1.73× (51.84)
EESD (Cali-TS)	2.04× (58.97)	1.92× (56.45)
w/o Sampling-Prediction	1.82× (54.03)	1.78× (53.49)
w/o Model-Prediction	1.88× (57.10)	1.82× (55.38)
EESD w/o TS†	1.66× (44.76)	1.58× (41.77)

Table 4: Ablation studies of different components based on LLaMA-2-13B. We exhibit the Speedup on Gsm8k and XSum, and also release the HM value in parentheses. † We set the drafting step K at 10. Other models yield similar patterns to LLaMA-2-13B.

results, and several significant insights can be inferred. **First**, we notice a substantial decrement in the model's performance when we replace the TS control with a fixed K value, which signifies the effectiveness of our proposed method for managing the generation of draft tokens. **Second**, similar to the prior approach, we introduce a trainable lm head just after the first-N layers, dispensing with the Early-exiting Layer. However, such a modification result in a significant decline in the model's performance, strongly indicating the fundamental role of the Early-exiting Layer in maintaining the quality of draft tokens. **Third**, a noteworthy observation is that our approach attains commendable results solely with only open-source data, especially on XSum. Furthermore, the performance can be improved with the addition of self-distillation, demonstrating the utility of data generated by original LLM. **Fourth**, within the Cali-TS approach, the role of sample prediction surpasses that of model prediction, and an integration of both can yield more optimal results.

5.2 Can the TS control mechanism predict the optimal drafting steps?

To investigate the ability of the TS control mechanism to automatically determine the quantity of draft tokens in each round, we conducted experiments on the effects of varying drafting steps K. As illustrated in Figure 3, the optimal K value differs across models and datasets, but the TS with Beta distribution consistently slightly exceeds the effect of the optimal K value. Moreover, with the boost from the model prediction, the TS with calibration can achieve a better acceleration effect. The experiment confirmed that the TS control mechanism can adaptively predict the optimal length for generating draft tokens in each round.



(a) LLaMA-2-13B (b) LLaMA-2-70B

Figure 3: We evaluate the speedup in generating 512 tokens using the EESD method at varying K values.

Model	Method	Gsm8k	XSum
LLaMA-2-70B	Vanilla SD	1.88×	1.46×
	+ Beta-TS	2.02× (+0.14)	1.67× (+0.21)
	Self SD	1.37×	1.23×
	+ Beta-TS	1.44× (+0.07)	1.25× (+0.04)
LLaMA-2-13B	Vanilla SD	0.96×	0.77×
	+ Beta-TS	0.99× (+0.03)	0.85× (+0.08)
	Self SD	1.37×	1.35×
	+ Beta-TS	1.41× (+0.04)	1.40× (0.05)

Table 5: Speedup of other SD methods with TS control mechanism. We use LLaMA-2-7B as draft model for Vanilla SD.

5.3 The generality of TS control mechanism

To verify the generality and effectiveness of the proposed TS control mechanism, we further apply it to other SD models instead of a pre-defined K. The results are reported in Table 5. According to the results, we can observe that TS control mechanism could be easily integrated into other SD methods to lift their performances. Note that the results in Table 5 are different from the results of w/o TS in ablation study. In ablation study, we set K at 10, which is not a superior setting, and as shown in Figure 3, K=5 is a better setting for the EESD of LLaMA-2-13B. However, for vanilla SD and self SD, K=10 is a suitable setting.

5.4 Effect of the first-N layers

Our experiments explore the impact of varying the number of first-N layers. As shown in Figure 4, using more layers improves the quality of the draft tokens, as measured by a higher HM value. However, end-to-end speedup does not correspondingly increase along with draft quality. This suggests that the extra time required to generate draft tokens with more layers offsets some of the end-to-end speedup.

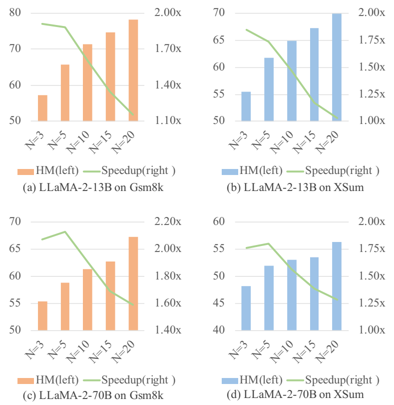


Figure 4: Effect of the different first-N layers. We valuate EESD (+Beta-TS) across varying N values of the first-N layers.

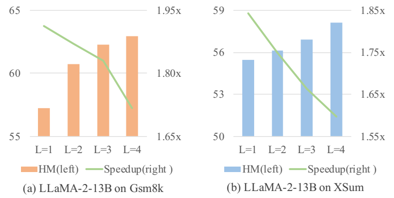


Figure 5: Effect of varying the number of Early-exiting layers.

The results indicate that layer augmentation only leads to slight improvements in the quality of draft tokens. Therefore, utilizing fewer layers for generating draft tokens proves to be an effective strategy. In addition, for larger models, such as 70B, the value of N needs to be slightly larger. And it is empirically suggested that N should be 5%-10% of the total number of LLM layers.

5.5 One Transformer layer is best for Early-exiting layer?

As demonstrated in Table 4, it has been proven that adding one Transformer layer after the First-N layers significantly improves the draft model's performance. To further investigate, we evaluate the effect of increasing the number of Transformer layers. Figure 5 illustrates that augmenting the number of Transformer layers following the First-N layers does yield an improvement in draft token quality.

Model	Method	Seq.	Loaded Params	Trainable Params	Num. of GPU	Batch Size	Time per batch
70B	Vanilla SD	4k	7B	7B	8	64	4.1s
	Medusa	4k	70.0B	1.32B	8	64	30.0s
	EESD	4k	5.6B	1.12B	8	64	2.3s
13B	Vanilla SD	2k	7B	7B	2	64	15.5s
	Medusa	2k	13.6B	760M	2	64	9.5s
	EESD	2k	1.6B	481M	2	64	1.6s

Table 6: Training efficiency of three methods on A100-80GB for LLaMA-2-70B and LLaMA-2-13B. And we use LLaMA-2-7B as draft model for Vanilla SD.

Model	Method	Gsm8k	XSum
LLaMA-2-70B	EESD (Beta-TS)	2.13×	1.80×
	+ Tree Attention	2.31× (+0.18)	1.91× (+0.11)
	EESD (Cali-TS)	2.29×	1.86×
	+ Tree Attention	2.48× (+0.19)	1.96× (+0.10)
LLaMA-2-13B	EESD (Beta-TS)	1.91×	1.84×
	+ Tree Attention	2.04× (+0.13)	1.89× (+0.05)
	EESD (Cali-TS)	2.04×	1.92×
	+ Tree Attention	2.18× (+0.14)	2.12× (+0.20)

Table 7: Speedup of EESD with implementing tree attention. We generate multiple draft token candidates and only retain the result consistent with the target model's top-1 token on verifying process.

However, because degree of this improvement is relatively small, it results in a reduction in the overall end-to-end speedup. Therefore, the experiment indicates that a single Transformer layer is enough to ensure the quality of draft tokens, and perfectly balance the quality and generation speed of draft tokens to achieve the optimal end-to-end speedup.

5.6 Training Efficiency

We compare training efficiency of three methods, which are tested on NVIDIA A100-80G GPUs. We set batch size to 64 and used SlimPajama datasets to train these models. As shown in Table 6, EESD only requires loading the parameters of First-N layers and Early-exiting layer, while Medusa requires loading all parameters of LLM and Medusa heads. Notably, during training, although both Medusa and EESD only update a portion of parameters, Medusa requires each sample to be computed across the whole LLM network. In contrast, EESD only needs to compute across First-N layers. Consequently, compared to Medusa and Vanilla SD, EESD significantly reduces in both training time and memory consumption.

5.7 Implement Tree Attention

Tree attention has been a prevalent technique in inference acceleration (Miao et al., 2023; Spector and Ré, 2023). This technique functions by structuring numerous draft token candidates within a

Model	Method	Drafting	Verification	Sampling	Others
LLaMA-2-70B	EESD(+Beta-TS)	5.517s	25.203s	0.017s	4.076s
	EESD(+Cali-TS)	5.022s	24.106s	0.426s	4.023s
LLaMA-2-13B	EESD(+Beta-TS)	3.656s	7.986s	0.014s	0.526s
	EESD(+Cali-TS)	3.436s	7.392s	0.355s	0.507s

Table 8: Breakdown of computational time (seconds) for EESD on 200 instances randomly sampled from XSum. We set final output sequence length at 512.

tree framework, allowing the LLM to concurrently verify several potential draft sequences through parallel decoding. It significantly increases the acceptance rate, thereby augmenting the overall speed of end-to-end generation. As shown in Table 7, we can easily implement the tree attention mechanism to EESD, resulting in significant increases in speed. It can achieve up to 2.48× and 1.96× times speedup on LLaMA-2-70B, as well as up to 2.18× and 2.12× times speedup on LLaMA-2-13B.

5.8 Breakdown of Computation

Table 8 presents an analysis of the computational time required for EESD generating on 200 instances randomly selected from XSum. The results indicate that Cali-TS exhibits higher time consumption compared to Beta-TS during the sampling phase. However, Cali-TS significantly diminishes the time usage in the drafting and verification stages, due to its superior control over the draft token generation process. Consequently, Cali-TS can yield a lower total time consumption.

6 Conclusion

In this work, we propose EESD, a novel method designed to lossless accelerate LLM by leveraging its first-N layers for generating draft tokens and employing Thompson Sampling to regulate this process. Specifically, we introduce an Early-exiting layer after first-N layers and train it using self-distillation, which strike an optimal balance between efficiency and performance of draft token generation. Furthermore, we devise a novel hybrid method that effectively combines model prediction and sampling prediction, resulting in remarkable generation speed enhancement. After conducting exhaustive experiments, the results demonstrate that EESD not only achieves a significant inference speedup but also substantially reduces both training time and memory consumption, compared to previous speculative decoding methods.

Limitations

In this section, we discuss the limitations of our work as follows. First, while we have given an empirical suggestion for the setting of N value in the First-N layers, we have not thoroughly studied the function of these first layers and how they affect the final outputs. We believe that a more detailed investigation of this is helpful for choosing the optimal N value. As such, we will conduct this research in future work. Second, we propose a model for predicting whether the draft token is consistent with the LLM's token in Section 3.3. However, this model has a large number of parameters, which is not very friendly for training and deployment. Therefore, we plan to refine the model's structure to improve its efficiency in future work.

Acknowledgements

Jingang Wang is funded by Beijing Nova Program(Grant NO. 20220484098). We sincerely thank all reviewers for their valuable comments and suggestions, which are crucial for improving our work.

References

Sangmin Bae, Jongwoo Ko, Hwanjun Song, and Se-Young Yun. 2023. Fast and robust early-exiting framework for autoregressive language models with synchronized parallel decoding. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 5910–5924. Association for Computational Linguistics.

Tianle Cai, Yuhong Li, Zhengyang Geng, Hongwu Peng, and Tri Dao. 2023. Medusa: Simple framework for accelerating llm generation with multiple decoding heads. `https://github.com/FasterDecoding/Medusa`.

Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and John Jumper. 2023. Accelerating large language model decoding with speculative sampling. *CoRR*, abs/2302.01318.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen

Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating large language models trained on code.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2023. Palm: Scaling language modeling with pathways. *J. Mach. Learn. Res.*, 24:240:1–240:113.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

Maha Elbayad, Jiatao Gu, Edouard Grave, and Michael Auli. 2020. Depth-adaptive transformer. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 10323–10337. PMLR.

Elias Frantar, Saleh Ashkboos, Torsten Hoefler, and Dan Alistarh. 2022. GPTQ: accurate post-training quantization for generative pre-trained transformers. *CoRR*, abs/2210.17323.

Zhuocheng Gong, Jiahao Liu, Qifan Wang, Yang Yang, Jingang Wang, Wei Wu, Yunsen Xian, Dongyan Zhao, and Rui Yan. 2023. Prequant: A task-agnostic quantization approach for pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 8065–8079. Association for Computational Linguistics.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2023. Knowledge distillation of large language models. *CoRR*, abs/2306.08543.

Zhenyu He, Zexuan Zhong, Tianle Cai, Jason D. Lee, and Di He. 2023. REST: retrieval-based speculative decoding. *CoRR*, abs/2311.08252.

Parsa Kavehzadeh, Mojtaba Valipour, Marzieh Tahaei, Ali Ghodsi, Boxing Chen, and Mehdi Rezagholizadeh. 2023. Sorted llama: Unlocking the potential of intermediate layers of large language models for dynamic inference using sorted fine-tuning (soft). *CoRR*, abs/2309.08968.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 19274–19286. PMLR.

Liunian Harold Li, Jack Hessel, Youngjae Yu, Xiang Ren, Kai-Wei Chang, and Yejin Choi. 2023. Symbolic chain-of-thought distillation: Small models can also "think" step-by-step. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 2665–2679. Association for Computational Linguistics.

Ji Lin, Jiaming Tang, Haotian Tang, Shang Yang, Xingyu Dang, and Song Han. 2023. AWQ: activation-aware weight quantization for LLM compression and acceleration. *CoRR*, abs/2306.00978.

Zechun Liu, Barlas Oguz, Changsheng Zhao, Ernie Chang, Pierre Stock, Yashar Mehdad, Yangyang Shi, Raghuraman Krishnamoorthi, and Vikas Chandra. 2023. LLM-QAT: data-free quantization aware training for large language models. *CoRR*, abs/2305.17888.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *CoRR*, abs/2305.11627.

Xupeng Miao, Gabriele Oliaro, Zhihao Zhang, Xinhao Cheng, Zeyu Wang, Rae Ying Yee Wong, Zhuoming Chen, Daiyaan Arfeen, Reyna Abhyankar, and Zhihao Jia. 2023. Specinfer: Accelerating generative LLM serving with speculative inference and token tree verification. *CoRR*, abs/2305.09781.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 1797–1807. Association for Computational Linguistics.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

Mingdong Ou, Nan Li, Cheng Yang, Shenghuo Zhu, and Rong Jin. 2019. Semi-parametric sampling for stochastic bandits with many arms. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7933–7940. AAAI Press.

Yi Peng, Miao Xie, Jiahao Liu, Xuying Meng, Nan Li, Cheng Yang, Tao Yao, and Rong Jin. 2019. A practical semi-parametric contextual bandit. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 3246–3252. ijcai.org.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Aleksandrs Slivkins. 2019. Introduction to multi-armed bandits. *CoRR*, abs/1904.07272.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R Steeves, Joel Hestness, and Nolan Dey. 2023. SlimPajama: A 627B token cleaned and deduplicated version of RedPajama.

Benjamin Spector and Christopher Ré. 2023. Accelerating LLM inference with staged speculative decoding. *CoRR*, abs/2308.04623.

Surat Teerapittayanon, Bradley McDanel, and H. T. Kung. 2016. Branchynet: Fast inference via early exiting from deep neural networks. In *23rd International Conference on Pattern Recognition, ICPR 2016, Cancún, Mexico, December 4-8, 2016*, pages 2464–2469. IEEE.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023a. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023b. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288.

Mengzhou Xia, Tianyu Gao, Zhiyuan Zeng, and Danqi Chen. 2023. Sheared llama: Accelerating language model pre-training via structured pruning. *CoRR*, abs/2310.06694.

Guangxuan Xiao, Ji Lin, Mickaël Seznec, Hao Wu, Julien Demouth, and Song Han. 2023. Smoothquant: Accurate and efficient post-training quantization for large language models. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 38087–38099. PMLR.

Chen Zhang, Yang Yang, Jiahao Liu, Jingang Wang, Yunsen Xian, Benyou Wang, and Dawei Song. 2023a. Lifting the curse of capacity gap in distilling language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4535–4553. Association for Computational Linguistics.

Jun Zhang, Jue Wang, Huan Li, Lidan Shou, Ke Chen, Gang Chen, and Sharad Mehrotra. 2023b. Draft & verify: Lossless large language model acceleration via self-speculative decoding. *CoRR*, abs/2309.08168.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric. P Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023a. Judging llm-as-a-judge with mt-bench and chatbot arena.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023b. Judging llm-as-a-judge with mt-bench and chatbot arena. *CoRR*, abs/2306.05685.

A Thompson Sampling with Beta Distribution

Given that the samples follow a Bernoulli distribution, we can infer using Bayes' theorem that when the prior distribution is Beta, the posterior distribution is also Beta. This phenomenon, known as a conjugate distribution, means that the prior and posterior share the same distribution function but with different parameters. Using a conjugate distribution greatly simplifies the computational derivation, leading us to select the Beta distribution as the prior. As shown in Algorithm 2, we implement Thompson Sampling algorithm with Beta distribution to iteratively estimate the value of θ. The prior distribution of θ is Beta, and as previously described, the posterior distribution of θ is also Beta.

Algorithm 2 Thompson Sampling with Beta Distribution Algorithm

Require: Target Model M_t; Draft Model M_d; Max Generation Length L; Hyperparameters α_0,β_0; Input Prompt $\{x_0, ..., x_n\}$.

Initialize prior probability $Beta(\theta; \alpha_0, \beta_0)$.
Initialize the result set $Q_g \leftarrow \{x_0, ..., x_n\}$ and $t \leftarrow 0$.
while t < L **do**
  Initialize the draft model result set and i, $Q_d \leftarrow Null, i \leftarrow 0$.
  while t + i < L **do**
    Get new token $x_i \leftarrow M_d(Q_g \bigcup Q_d)$.
    Add token x_i to set Q_d.
    Sample θ_{t+i} from $Beta(\theta; \alpha_t, \beta_t)$.
    Sample $\chi \in \{0, 1\}$ from Bernoulli distribution $P_B(\theta_{t+i})$.
    $i \leftarrow i + 1$.
    if $\chi = 0$ **then**
      Break
    end if
  end while
  Verify the results Q_d by Target Model M_t and get Q_v received by M_t, $Q_v \subseteq Q_d$.
  Add the set Q_v to Q_g, $Q_g \leftarrow Q_g \bigcup Q_v$.
  Update t according to length of Q_v, $t \leftarrow t + |Q_v|$.
  Calculate $r \leftarrow |Q_v| - 1$.
  Calculate $n \leftarrow min(|Q_v| + 1, |Q_d|)$
  Update α_t, $\alpha_t \leftarrow \alpha_{\{t-|Q_v|\}} + r$.
  Update β_t, $\beta_t \leftarrow \beta_{\{t-|Q_v|\}} + (n - r)$.
end while
return Q_g

B Thompson Sampling with Calibration

In the previous section, we unveiled a Thompson Sampling algorithm with Beta distribution (Beta-TS). This method progressively update the parameters of Beta distribution to enhance the precision of the estimated θ. However, based on the Multi-Arm Bandit (MAB) theory, the early phase is more exploration-oriented, this predisposition can lead to a less accurate initial estimation of the θ. To escalate the efficiency of the Thompson Sampling method, we further propose a hybrid approach of model prediction and sampling prediction. In early stage, we rely more on model prediction to curtail the inaccuracies introduced by exploration. In later stages, as the sampling prediction converges, we calibrate the model prediction with the result of sampling to obtain an accurate estimate of θ. The details are illustrated in Algorithm 3.

Algorithm 3 Thompson Sampling with Calibration Algorithm

Require: Target Model M_t; Draft Model M_d; Max Generation Length L; Hyperparameters σ_M,σ_S,μ_0,σ_0; Input Prompt $\{x_0, ..., x_n\}$.

Initialize prior probability $\mathcal{N}(\mu_0, \sigma_0^2)$.
Initialize the result set $Q_g \leftarrow \{x_0, ..., x_n\}$, $n \leftarrow 0$ and $t \leftarrow 0$.
while t < L **do**
  Initialize the draft model result set and i, $Q_d \leftarrow Null, i \leftarrow 0$.
  while t + i < L **do**
    Get new token $x_i \leftarrow M_d(Q_g \bigcup Q_d)$.
    Add token x_i to set Q_d.
    Get model predict score, $\theta_M^{t+i} \leftarrow Sigmoid(MLP(W^i H_t^{(T)}, H_{t+i}^{(D)}))$.
    Calculate $\mu_{t+i} \leftarrow \frac{\sigma_S^2}{n\sigma_M^2+\sigma_S^2}\theta_M^{t+i} + \frac{n\sigma_M^2}{n\sigma_M^2+\sigma_S^2}\hat{\theta}_t$.
    Calculate $\sigma_{t+i}^2 \leftarrow \frac{\sigma_M^2\sigma_S^2}{\sigma_S^2+n\sigma_M^2}$.
    Sample θ_{t+i} from Gaussian distribution $\mathcal{N}(\mu_{t+i}, \sigma_{t+i}^2)$.
    Sample $\chi \in \{0, 1\}$ from Bernoulli distribution $P_B(\theta_{t+i})$.
    $i \leftarrow i + 1$.
    if $\chi = 0$ **then**
      Break
    end if
  end while
  Verify the results Q_d by Target Model M_t and get Q_v received by M_t, $Q_v \subseteq Q_d$.
  Add the set Q_v to Q_g, $Q_g \leftarrow Q_g \bigcup Q_v$.
  Update t according to length of Q_v, $t \leftarrow t + |Q_v|$.
  Update $n \leftarrow n + 1$.
  Update $\hat{\theta}_t$, $\hat{\theta}_t \leftarrow \frac{\hat{\theta}_{\{t-|Q_v|\}} * (t-|Q_v|+1)+|Q_v|}{t+1}$.
end while
return Q_g

According to the central limit theorem, when the sample size is sufficiently large, the sample mean adheres to a Gaussian distribution. Therefore, we make an assumption that sample mean $\widetilde{\chi}$ of χ in one drafting iteration follows a Gaussian distribution, i.e $\widetilde{\chi} = \frac{\chi_1+\chi_2+\chi_3+...+\chi_k}{k} \sim \mathcal{N}(\mu, \sigma_S^2)$. And According to the central limit theorem, σ_S equals $\sigma_\chi / sqrt(k)$, where σ_χ is the standard deviation of the random variable χ and we assume σ_χ

is known. In this case, we make the assumption that k is a fixed value and pre-determined by the user, which guarantees that $\widetilde{\chi}$ is independently and identically distributed. Furthermore, we employ a model to estimate the value of μ. It can be posited that μ follows a Gaussian distribution, characterized by the model's predicted value as the mean and the model's predicted error as the variance, i.e. $\mu \sim \mathcal{N}(\theta_M, \sigma_M^2)$, where θ_M is model's predicted value. Here, we presume that the model's predicted error is a known entity. As supported by Bayesian theory, when the random variable follows a Gaussian distribution with a known variance but an unknown mean and the prior distribution is also a Gaussian distribution, it satisfies the conjugate distribution. Therefore, the posterior distribution is following,

$$\begin{aligned} &P(\mu|D) \\ &\propto P(D|\mu, \sigma_S^2)P(\mu|\theta_M, \sigma_M^2) \\ &\propto \mathcal{N}\left(\frac{\sigma_S^2}{n\sigma_M^2+\sigma_S^2}\theta_M^{t+i} + \frac{n\sigma_M^2}{n\sigma_M^2+\sigma_S^2}\hat{\theta}_t, \frac{\sigma_M^2\sigma_S^2}{\sigma_S^2+n\sigma_M^2}\right) \end{aligned} \tag{9}$$

where n is the number of verification by LLM, $\{\sigma_M, \sigma_S, \hat{\theta}_0\}$ is pre-determined by the user and $\hat{\theta}_t$ is the observed sample mean of the random variable $\widetilde{\chi}$. Due to k is a constant, we set $\hat{\theta}_t$ to observed sample mean of the random variable χ, i.e.

$$\hat{\theta}_t = \frac{\hat{\theta}_{\{t-|Q_v|\}} * (t - |Q_v| + 1) + |Q_v|}{t+1}, \tag{10}$$

Carefully thinking Eq.(6) and Eq.(9), we can observe that when n is small, the mean μ tends to θ_M^{t+i}, and when n is large, μ tends to $\hat{\theta}_t$. This achieves the reduction of uncertainty in sampling prediction through model prediction in the early exploration stage. In the later stage, as the number of observed samples increases, the accuracy of $\hat{\theta}_t$ markedly enhances. Concurrently, μ draws closer to $\hat{\theta}_t$, thereby yielding more accurate prediction. Therefore, our proposed method of mixing model prediction and sampling prediction can outperform the Beta-TS algorithm.

C Training Details

We implement all experiments with the deep learning framework PyTorch on NVIDIA A100-80G GPUs. We set the learning rate to 1e-3 and the batch size to 64, for training EESD and Medusa. The hyperparameter settings we adopt are shown in Table 9

D Harmonic Mean Metrics

We believe that two indicators, the acceptance rate and the proportion of draft tokens, will affect the end-to-end acceleration effect. The acceptance rate, denoted as v_d, indicates the percentage of draft tokens that are accepted by the target model. It is calculated as follows,

$$v_d = \frac{N_{right}}{N_{all\_draft}}, \tag{11}$$

where N_{right} is the number of draft tokens that are accepted by the target model, and N_{all_draft} denotes the total count of tokens generated by the draft model. The proportion of draft tokens, denoted as r_d, represents the proportion of tokens that come from the draft model, which is calculated as follows,

$$r_d = \frac{N_{right}}{L}, \tag{12}$$

where L is the total number of tokens in the final output sequence.

Once we have computed the aforementioned two metrics, we can infer the speedup. The inference time of end-to-end generation can be calculated according to the following formula,

$$T = \frac{r_d * L}{v_d}T_d + (1 - r_d) * L * T_t, \tag{13}$$

Where T_d represents the time taken by the draft model to generate one token, and T_t represents the time taken by the target model to generate one token. We can compute speed of the method by $sp = \frac{L}{T}$ and speedup by $speedup = \frac{sp}{T_t}$. Therefore, by integrating Eq.(13), we obtain the following formula,

$$speedup = \frac{v_d}{(\alpha - v_d) * r_d + v_d}, \tag{14}$$

where α equals $\frac{T_d}{T_t}$. To achieve speedup greater than 1.00×, the term $\alpha - v_d$ must be less than zero, given that both v_d and r_d are positive values. The speedup of the method is influenced by the factors α, v_d and r_d. Under the premise that $\alpha < v_d$, ideally, α should be as low as possible, while v_d and r_d should be as high as possible. Both v_d and r_d are influenced by the quality of the draft tokens and the strategy for generating the draft tokens. A well-devised strategy for draft token generation can increase the values of v_d and r_d, but the upper bound is restricted by the inherent quality of the draft tokens themselves. In our experiments, we observe that as the increasing of the drafting steps

Model	Method	Train Dataset	Seq.	# Epoch	Learning Rate	Batch Size	# GPUs
LLaMA-2-70B	Medusa	SlimPajama	4k	6	1e-3	64	8
	EESD	SlimPajama	4k	6	1e-3	64	8
LLaMA-2-70B-chat	Medusa	ShareGPT	4k	6	1e-3	64	8
	EESD	ShareGPT	4k	6	1e-3	64	8
LLaMA-2-13B	Medusa	SlimPajama	2k	4	1e-3	64	2
	EESD	SlimPajama	2k	4	1e-3	64	2
Vicuna-13B	Medusa	ShareGPT	2k	4	1e-3	64	2
	EESD	ShareGPT	2k	4	1e-3	64	2
CodeLLaMA-2-13B	Medusa	SlimPajama	16k	4	1e-3	64	8
	EESD	SlimPajama	16k	4	1e-3	64	8

Table 9: The hyperparameter values for EESD and Medusa training.

Target Model	Method	Draft Model	Harmonic Mean v_d	r_d	HM	Inference Time (/s)	Speed (token/s)	Speedup
LLaMA-2-70B	Auto-regressive	-	-	-	-	56.32	9.10	1.00×
	Vanilla SD	LLaMA-2-7B	0.74	0.88	80.35	29.89	17.13	1.88×
	Self SD	Self	0.90	0.70	78.64	41.09	12.46	1.37×
	Medusa	Self	0.25	0.50	33.75	32.56	15.72	1.73×
	EESD (+Beta-TS)	Self	0.52	0.68	58.79	26.44	19.36	2.13×
	EESD (+Cali-TS)	Self	0.56	0.71	62.25	24.61	20.80	**2.29**×
LLaMA-2-70B-chat	Auto-regressive	-	-	-	-	56.53	9.06	1.00×
	Vanilla SD	LLaMA-2-7B-chat	0.52	0.83	63.90	39.17	13.07	1.44×
	Self SD	Self	0.66	0.70	67.99	49.89	10.26	1.13×
	Medusa	Self	0.16	0.39	22.86	39.97	12.81	1.41×
	EESD (+Beta-TS)	Self	0.39	0.62	47.76	31.57	16.22	1.79×
	EESD (+Cali-TS)	Self	0.39	0.62	48.23	31.04	16.49	**1.82**×
LLaMA-2-13B	Auto-regressive	-	-	-	-	21.93	23.35	1.00×
	Vanilla SD	LLaMA-2-7B	0.81	0.89	84.59	22.78	22.48	0.96×
	Vanilla SD	TinyLLaMA-1.1B	0.76	0.90	82.49	18.43	27.78	1.19×
	Self SD	Self	0.90	0.73	80.53	16.04	31.92	1.37×
	Medusa	Self	0.24	0.47	31.71	12.42	41.22	1.77×
	EESD (+Beta-TS)	Self	0.50	0.67	57.22	11.46	44.68	1.91×
	EESD (+Cali-TS)	Self	0.52	0.69	58.97	10.77	47.54	**2.04**×
Vicuna-13B	Auto-regressive	-	-	-	-	22.26	23.00	1.00×
	Vanilla SD	Vicuna-7B	0.52	0.83	64.22	32.40	15.80	0.69×
	Vanilla SD	Vicuna-68M	0.17	0.60	26.87	17.39	29.44	1.28×
	Self SD	Self	0.76	0.62	68.07	17.91	28.59	1.24×
	Medusa	Self	0.19	0.42	26.08	14.57	35.14	1.53×
	EESD (+Beta-TS)	Self	0.34	0.58	43.01	14.16	36.16	1.57×
	EESD (+Cali-TS)	Self	0.35	0.59	43.53	14.00	36.57	**1.59**×

Table 10: The detailed evaluation results on Gsm8k with different methods of Table 1. We present the result from our assessment of Harmonic Mean, inference time, and the speed of end-to-end generation. Additionally, we also present the speedup in comparison with the auto-regression method.

K, the acceptance rate v_d usually decreases, while the proportion of draft tokens r_d increases in most cases. Therefore, we hope to come up with a metric that can help us understand the optimality of K. And then, we use the harmonic mean of v_d and r_d to assess it.

As shown in Table 10, 11, 12 and 13, we present the v_d and r_d scores of each baseline method as well as our method across three benchmark evaluations, as detailed explanations of Table 1, 2 and 3.

E Inference Time and Speed up

We present the comprehensive results of end-to-end inference time and tokens generated per second for the auto-regressive method, three speculative decoding methods and EESD. These results provide detailed explanations of the data shown in Table 1, 2 and 3. These results, gathered from evaluations using the Gsm8k benchmark, are detailed in Table 10. Furthermore, we provide the additional results from the XSum dataset in Table 11, the Humaneval dataset in Table 12 and the MT-bench dataset in Table 13.

F Case Study

As shown in Figure 6, we demonstrated two Xsum samples. As described in Section 4.1, we adopt a 1-shot setting and use a greedy generation strategy.

[INPUT]

Article: Masers were invented before the laser, but have languished in obscurity because they required high magnetic fields and difficult cooling schemes.

……

this type of maser could be used to detect some extraterrestrial intelligence that hasn't been detected."

Summary: Researchers have shown off a microwave-emitting version of the laser, called a maser, that works at room temperature.

Article: The 26-year-old England tight-head prop, who was sent off, pleaded guilty at a disciplinary hearing on Tuesday.

……

Brookes was dismissed in the 38th minute of Saints' 22-16 defeat when he charged into the ruck and struck the head of Newcastle hooker Scott Lawson with his shoulder.

Summary:

[Auto-regression OUTPUT]

The 26-year-old England tight-head prop, who was sent off, pleaded guilty at a disciplinary hearing on Tuesday. </s>

[EESD Generation Process]

- ROUND 1: The 26-year-old England
- ROUND 2: -head prop, who was dismissed
- ROUND 3: off for a year in the 201
- ROUND 4: pleaded guilty to a misdemean
- ROUND 5: the London
- ROUND 6: disciplinary hearing on Tuesday. </s>

[EESD Final OUTPUT]

The 26-year-old England tight-head prop, who was sent off, pleaded guilty at a disciplinary hearing on Tuesday. </s>

(a) EXAMPLE 1

[INPUT]

Article: Masers were invented before the laser, but have languished in obscurity because they required high magnetic fields and difficult cooling schemes.

……

this type of maser could be used to detect some extraterrestrial intelligence that hasn't been detected."

Summary: Researchers have shown off a microwave-emitting version of the laser, called a maser, that works at room temperature.

Article: Liberal Democrat AM Eluned Parrott said the Welsh government had "wasted" more than £52,000 on them at railway stations in south Wales.

She accused ministers of spending public money to "promote themselves before an election".

……

Formal consultation for the £600m programme to develop an integrated network of rail, bus and light rail services begins in 2016.

Summary:

[Auto-regression OUTPUT]

The Welsh government has been accused of wasting more than £52,000 on posters promoting the south Wales Metro. </s>

[EESD Generation Process]

- ROUND 1: The UK's largest broadband service provider
- ROUND 2: government has been working on
- ROUND 3: of spending £100m
- ROUND 4: taxpay
- ROUND 5: than £52,000 of
- ROUND 6: the rail system
- ROUND 7: reforms. </s>
- ROUND 8: the idea of a
- ROUND 9: Wales Government's plans
- ROUND 10: . </s>

[EESD Final OUTPUT]

The Welsh government has been accused of wasting more than £52,000 on posters promoting the south Wales Metro. </s>

(b) EXAMPLE 2

Figure 6: A visualization of the generation process of EESD with Cali-TS on LLaMA-2-70B. We present two examples from the XSum dataset, and demonstrate the input text, the text generated by the original LLM using auto-regression strategy, the generation process of EESD, and the final result generated by EESD. In EESD generation process, the green color represents the draft token that are accept by LLM, and the red color represents the rejected draft tokens, and gray color represents the draft tokens that will be discarded after the rejected token. And in the EESD final output, the green color represents tokens generated by the draft model, and the red color represents the tokens generated by the original LLM.

Target Model	Method	Draft Model	Harmonic Mean v_d	r_d	HM	Inference Time (/s)	Speed (token/s)	Speedup
LLaMA-2-70B	Auto-regressive	-	-	-	-	62.59	8.18	1.00×
	Vanilla SD	LLaMA-2-7B	0.57	0.83	67.30	42.85	11.95	1.46×
	Self SD	Self	0.80	0.60	68.60	51.72	9.90	1.21×
	Medusa	Self	0.19	0.42	25.69	44.21	11.58	1.42×
	EESD (+Beta-TS)	Self	0.44	0.63	51.91	34.81	14.71	1.80×
	EESD (+Cali-TS)	Self	0.46	0.65	53.41	33.58	15.25	**1.86**×
LLaMA-2-70B-chat	Auto-regressive	-	-	-	-	62.57	8.18	1.00×
	Vanilla SD	LLaMA-2-7B-chat	0.70	0.66	62.75	44.98	11.38	1.39×
	Self SD	Self	0.66	0.70	68.38	53.93	9.49	1.16×
	Medusa	Self	0.12	0.31	17.02	52.01	9.85	1.20×
	EESD (+Beta-TS)	Self	0.32	0.55	40.73	41.32	12.39	1.51×
	EESD (+Cali-TS)	Self	0.33	0.57	41.85	40.43	12.66	**1.55**×
LLaMA-2-13B	Auto-regressive	-	-	-	-	22.46	22.80	1.00×
	Vanilla SD	LLaMA-2-7B	0.67	0.86	75.63	29.09	17.60	0.77×
	Vanilla SD	TinyLLaMA-1.1B	0.67	0.86	75.42	21.39	23.94	1.05×
	Self SD	Self	0.87	0.70	77.61	16.63	30.79	1.35×
	Medusa	Self	0.18	0.41	25.03	14.67	34.90	1.53×
	EESD (+Beta-TS)	Self	0.48	0.66	55.46	12.18	42.04	1.84×
	EESD (+Cali-TS)	Self	0.49	0.67	56.45	11.69	43.80	**1.92**×
Vicuna-13B	Auto-regressive	-	-	-	-	22.82	22.44	1.00×
	Vanilla SD	Vicuna-7B	0.41	0.79	53.77	40.71	12.58	0.55×
	Vanilla SD	Vicuna-68M	0.15	0.56	23.05	19.50	26.26	1.17×
	Self SD	Self	0.69	0.54	60.38	20.14	25.42	1.12×
	Medusa	Self	0.11	0.29	15.55	18.28	28.01	1.21×
	EESD (+Beta-TS)	Self	0.24	0.47	32.23	17.96	28.51	1.25×
	EESD (+Cali-TS)	Self	0.25	0.48	32.50	17.75	28.85	**1.27**×

Table 11: The detailed evaluation results on XSum with different methods of Table 1. We present the result from our assessment of Harmonic Mean, inference time, and the speed of end-to-end generation. Additionally, we also present the speedup in comparison with the auto-regression method.

We observe that during the generation process of EESD, those draft tokens that are inconsistent with the original LLM's output will be discarded. This ensures that the final result generated by EESD is the same as auto-regression. Furthermore, we find that for samples with a high draft token acceptance rate, the EESD tends to generate longer draft sequence in one drafting round, as shown in example 1. Conversely, for samples with a lower acceptance rate, the EESD displays a tendency to generate shorter draft sequence, minimizing the quantity of discarded draft tokens, as shown in example 2. The examples in Figure 6 shows that our method is effective in adaptively determining the length of draft token generation, leading to significant improvement in the final end-to-end generation speed.

Target Model	Method	Draft Model	Harmonic Mean v_d	r_d	HM	Inference Time (/s)	Speed (token/s)	Speedup
LLaMA-2-13B	Auto-regressive	-	-	-	-	21.83	23.45	1.00×
	Vanilla SD	LLaMA-2-7B	0.86	0.89	87.32	22.60	22.65	0.97×
	Self SD	Self	0.89	0.72	79.93	16.09	31.82	1.36×
	Medusa	Self	0.19	0.42	26.67	13.57	37.73	1.61×
	EESD (+Beta-TS)	Self	0.54	0.72	61.43	10.52	48.67	2.08×
	EESD (+Cali-TS)	Self	0.55	0.73	62.87	10.14	50.49	**2.15**×
CodeLLaMA-2-13B	Auto-regressive	-	-	-	-	22.82	22.44	1.00×
	Vanilla SD	CodeLLaMA-2-7B	0.92	0.90	91.12	20.84	24.57	1.09×
	Self SD	Self	0.92	0.76	83.51	16.58	30.88	1.38×
	Medusa	Self	0.45	0.55	49.14	11.56	44.29	1.97×
	EESD (+Beta-TS)	Self	0.64	0.75	68.94	10.31	49.66	2.21×
	EESD (+Cali-TS)	Self	0.65	0.76	70.15	9.32	54.94	**2.45**×

Table 12: The detailed evaluation results on Humaneval with different methods of Table 2. We present the result from our assessment of Harmonic Mean, inference time, and the speed of end-to-end generation. Additionally, we also present the speedup in comparison with the auto-regression method.

Target Model	Method	Draft Model	Harmonic Mean v_d	r_d	HM	Inference Time (/s)	Speed (token/s)	Speedup
LLaMA-2-13B	Auto-regressive	-	-	-	-	21.61	23.69	1.00×
	Vanilla SD	LLaMA-2-7B	0.91	0.89	90.01	21.19	24.16	1.02×
	Self SD	Self	0.94	0.85	89.25	13.68	37.43	1.58×
	Medusa	Self	0.38	0.50	43.08	11.43	44.79	1.89×
	EESD (+Beta-TS)	Self	0.56	0.70	62.15	10.65	48.08	2.03×
	EESD (+Cali-TS)	Self	0.58	0.72	64.31	10.24	49.99	**2.11**×
Vicuna-13B	Auto-regressive	-	-	-	-	21.38	23.95	1.00×
	Vanilla SD	Vicuna-7B	0.79	0.85	81.50	23.49	21.80	0.91×
	Self SD	Self	0.93	0.80	86.20	14.64	34.97	1.46×
	Medusa	Self	0.22	0.46	30.11	12.50	40.96	1.71×
	EESD (+Beta-TS)	Self	0.45	0.63	52.82	11.75	43.57	1.82×
	EESD (+Cali-TS)	Self	0.47	0.66	54.83	11.31	45.27	**1.89**×

Table 13: The detailed evaluation results on MT-bench with different methods of Table 3. We present the result from our assessment of Harmonic Mean, inference time, and the speed of end-to-end generation. Additionally, we also present the speedup in comparison with the auto-regression method.

STATE: A Robust ATE Estimator of Heavy-Tailed Metrics for Variance Reduction in Online Controlled Experiments

Hao Zhou*
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
Meituan
Beijing, China
zhouhao29@meituan.com

Kun Sun*†
Meituan
Beijing, China
sunkun07@meituan.com

Shaoming Li
Meituan
Beijing, China
shaoming.li@outlook.com

Yangfeng Fan
Meituan
Beijing, China
fanyangfeng@meituan.com

Guibin Jiang
Meituan
Beijing, China
jiangguibin@meituan.com

Jiaqi Zheng
State Key Laboratory for Novel Software Technology
Nanjing University
Nanjing, China
jzheng@nju.edu.cn

Tao Li
Meituan
Beijing, China
litao19@meituan.com

Abstract

Online controlled experiments play a crucial role in enabling data-driven decisions across a wide range of companies. Variance reduction is an effective technique to improve the sensitivity of experiments, achieving higher statistical power while using fewer samples and shorter experimental periods. However, typical variance reduction methods (e.g., regression-adjusted estimators) are built upon the intuitional assumption of Gaussian distributions and cannot properly characterize the real business metrics with heavy-tailed distributions. Furthermore, outliers diminish the correlation between pre-experiment covariates and outcome metrics, greatly limiting the effectiveness of variance reduction.

In this paper, we develop a novel framework that integrates the Student's t-distribution with machine learning tools to fit heavy-tailed metrics and construct a robust average treatment effect estimator in online controlled experiments, which we call STATE. By adopting a variational EM method to optimize the loglikehood function, we can infer a robust solution that greatly eliminates the negative impact of outliers and achieves significant variance reduction. Moreover, we extend the STATE method from count metrics to ratio metrics by utilizing linear transformation that preserves unbiased estimation, whose variance reduction is more complex but less investigated in existing works. Finally, both simulations on synthetic data and long-term empirical results on Meituan experiment platform demonstrate the effectiveness of our method. Compared with the state-of-the-art estimators (CUPAC/MLRATE), STATE achieves over 50% variance reduction, indicating it can reach the same statistical power with only half of the observations, or half the experimental duration.

CCS Concepts

• **Mathematics of computing** → *Probability and statistics*; • **General and reference** → *Experimentation.*

Keywords

Controlled Experiments, Variance Reduction, Heavy-Tailed, Robust Estimation, Causal Inference

ACM Reference Format:
Hao Zhou, Kun Sun, Shaoming Li, Yangfeng Fan, Guibin Jiang, Jiaqi Zheng, and Tao Li. 2024. STATE: A Robust ATE Estimator of Heavy-Tailed Metrics for Variance Reduction in Online Controlled Experiments. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3672352

*Both authors contributed equally to this research.
†Corresponding author.

KDD '24, August 25–29, 2024, Barcelona, Spain.

ACM ISBN 979-8-4007-0490-1/24/08
https://doi.org/10.1145/3637528.3672352

1 Introduction

Online controlled experiments (also known as A/B tests) are the most widely adopted method for measuring causal effects, and play a crucial role in enabling data-driven decisions across a wide range of companies, including Facebook [3, 13], Airbnb [8, 10], Google [14], Microsoft [9] and LinkedIn [15, 26]. These experiments are critical for businesses, as even small differences detected in key metrics can have significant implications for the total revenue [9]. For example, a strategy that increases one user's revenue by $0.1 can result in millions of dollars in the total revenue for ten million users.

In the typical settings of controlled experiments, online traffic is randomly partitioned into two groups: a treatment group and a control group. They keep almost the same configuration except that the treatment group receives an additional intervention (e.g., a new production version or a promotion email). The average treatment effect (ATE) measures the causal effect of a treatment or intervention, but its groundtruth is unknown. Thanks to the central limit theorem, the difference-in-means estimator (DIM) produces an unbiased estimation of ATE, which is calculated by the difference between the treatment and control group outcomes.

Although DIM offers unbiased estimates, it still suffers from high variance, further leading to poor sensitivity and low statistical power. For online businesses, the sensitivity of experiments is particularly important. With thousands of experiments run each year, any benefits of increased sensitivity will be amplified due to economies of scale. On the other hand, the improvement in sensitivity allows experiments to be run on a smaller user population or for shorter durations while achieving the same statistical power. It is significant for improving the product feedback cycle and agility.

In mathematics, variance reduction techniques are used to obtain higher precision for the metric of interest and have been introduced to online controlled experiments recently. For examples, CUPED [9] utilizes pre-experiment covariates to reduce metric variability between the treatment and control groups, constructing an unbiased estimator with lower variance. However, the effectiveness is limited by the linear assumption and the correlation between the covariates and outcome metrics. Hence, to develop a nonlinear multi-covariate proxy that is highly correlated with the outcome metric, several machine learning (ML) based estimators have been explored [2, 13, 15, 24, 25]. CUPAC [23] and MLRATE [13] are two state-of-the-art (SOTA) estimators in this class, both of which are based on the regression-adjusted method. To tackle the optimality, Yin [15] has demonstrated that if ML predicators converge to the conditional mean function, the asymptotic variance of ATE estimators can reach the semi-parametric variance lower bound.

Notably, the discussion on optimality is conducted under the assumption that the regression residuals follow a Gaussian distribution. However the business metrics of interest are often heavy-tailed (e.g., watch time on video sites, user Gross Merchandise Volume(GMV) on e-commerce platforms, the amount of live broadcast rewards, etc.). The observations for these metrics usually contain some outliers (observations far away from the bulk of the probability density). The significant impact of outliers on the squared loss may lead to bias and high variance in the ATE estimates. As shown in Fig. 1, the residuals of user GMV on the Meituan food delivery platform exhibit a heavy-tailed distribution due to the presence of users with extremely high-priced orders. In the Gaussian distribution, the probability of observations falling beyond six standard deviations should be extremely small (less than 0.0000002%). However, it is obvious that a larger portion of the observations exceeds six standard deviation in Fig. 1. Therefore, the Gaussian distribution does not properly characterize heavy-tailed metrics.

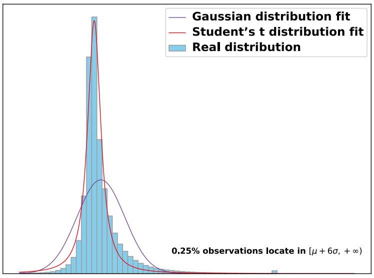


Figure 1: The distribution of residuals for user GMV on the Meituan food delivery platform. The Y-axis represents the sample count. The X-axis represents the residuals between the real value and the model predicted value, where the extremely large values are clipped to the upper bound.

To address this problem, we introduce the Student's t-distribution to variance reduction. The Gaussian distribution is a special case of the t-distribution, corresponding to the situation where the degree of freedom tend to infinity. As the degree of freedom decreases, the t-distribution has heavier tails, giving non-zero probability to observations that fall outside the bulk of the density. This characteristic endows the t-distribution with an important property called robustness. As is shown in Fig. 1, the GMV metric can be better characterized when the regression residuals are modeled as a generalized t-distribution.

In this paper, we mainly focus on the variance reduction of ATE estimation of heavy-tailed metrics, and propose an easy-to-implement robust ATE estimator, called STATE. Specifically, our work has the following key contributions.

- We integrate the machine learning regression adjustment method with the Student's t-distribution to estimate ATE for count metrics and derive a variational EM framework to infer parameters. The estimation procedure takes full advantage of both the powerful fitting ability of machine learning tools and the robustness of the t-distribution to outliers, thereby significantly reducing the variance of ATE estimation in online controlled experiments.
- We extend this method to ratio metrics, which are more complex but less investigated. We introduce linear transformation for ratio metrics while preserve unbiased estimation and consistent variance, under which regression-adjusted methods and the Student's t-distribution can be introduced to reduce variance and improve sensitivity.
- We conduct large-scale experiments on synthetic data and real business data on the Meituan food delivery platform to

verify the effectiveness of our method in this paper. Extensive experimental results demonstrate that when the metrics follow Gaussian distribution, the STATE method performs equivalently to the state-of-the-art estimators (CUPAC/MLRATE). When the metric is a heavy-tailed distribution, our method can reduce the variance by about 50% compared to the CUPAC and MLRATE. This indicates that we can achieve the same statistical power in online controlled experiments with only half of the observations, or half the experimental duration.

2 Setup and Related Work

2.1 Setup

Let X be the pre-experiment covariates, T be the binary treatment variable, and Y, Z be two outcome variables. In an online controlled experiment, suppose that there are N samples denoted by (X_i, T_i, Y_i, Z_i), which are drawn independently from an identical distribution. The treatment $T \in \{0, 1\}$ is assigned randomly and independently of the covariates X, where $T_i = 1$ represents that the i-th sample receives the treatment. Denote the number of samples with $T_i = 1$ and $T_i = 0$ by N_t and N_c, repectively. Following the Rubin Causal Model (RCM) [21], let $Y_i(1), Z_i(1), Y_i(0), Z_i(0)$ be the corresponding potential outcome when the sample is assigned with the treatment or not.

The count metrics are defined as the sample means (e.g., $\frac{\sum_{T_i=1} Y_i}{N_t}$), whose analysis unit is exactly the randomization unit in experiments. The average treatment effect (ATE) for count metrics is

$$\tau_c = E[Y_i(1)] - E[Y_i(0)].$$

The common difference-in-mean (DIM) estimator is taken as

$$\Delta Y = \frac{\sum_{T_i=1} Y_i}{N_t} - \frac{\sum_{T_i=0} Y_i}{N_c}. \tag{1}$$

According to the central limit theorem, it gives an unbiased estimation of τ_c with the variance $D[\Delta Y] = \sigma_t^2/N_t + \sigma_c^2/N_c$, where σ_t^2, σ_c^2 are the variance of samples in the treatment group and control group respectively.

The ratio metrics are usually regarded as the ratio of two count metrics, e.g., $\frac{\sum_{T_i=1} Y_i}{\sum_{T_i=1} Z_i}$. For ratio metrics, ATE is defined as

$$\tau_r = \frac{E[Y_i(1)]}{E[Z_i(1)]} - \frac{E[Y_i(0)]}{E[Z_i(0)]}.$$

The corresponding DIM estimator is

$$\Delta R = \frac{\sum_{T_i=1} Y_i}{\sum_{T_i=1} Z_i} - \frac{\sum_{T_i=0} Y_i}{\sum_{T_i=0} Z_i}. \tag{2}$$

Notice that DIM is not an unbiased estimator because

$$E[\Delta R] \neq \tau_r.$$

Fortunately, it gives a consistent estimation for τ_r according to the delta method [7].

2.2 Related Work

Variance reduction is a key technology and a longstanding challenge to improve the sensitivity of online control experiments. The original literatures mainly discuss the univariate linear adjustment method [9, 12, 17, 27]. For instance, the CUPED [9] estimator, which is widely applied in industry, utilizes relevant covariate from the pre-experiment period to reduce the variability of the outcome metric, and shows that the higher correlation between covariate and outcome, the better the performance of variance reduction. Alternatively, an equivalent technique to CUPED is the linear regression method, $Y_i = a_0 + a_1 T_i + a_2 X_i + \epsilon_i$, which assumes the outcome is a linear combination of the treatment effect and the covariate term.

To overcome the limitations of the linear model, researchers have explored multivariate adjustment methods [2, 13, 15, 23–25] by introducing the cross-fitting technique [6, 28] and "agnostic" regression [12, 17]. Typically, they utilize many covariates in a machine learning(ML) model g predicting Y from X to develop a proxy variable $g(X)$. Subsequently, estimate the ATE in a linear regression model $Y_i = a_0 + a_1 T_i + a_2 g(X_i) + \epsilon_i$. Since the proxy incorporates more prior information about the outcome, it generates further variance reduction gains.

However, the regression-adjusted methods assume that the residual ϵ follows a Gaussian distribution, which is not applicable to the heavy-tailed metrics in online businesses. If outliers are not properly dealt with, they may lead to bias and high variance in parameter estimation.

Furthermore, the majority of variance reduction methods focus on count metrics. In fact, ratio metrics are equally important in practice, and are more complex but less investigated. Existing solutions for ratio metric are mostly extensions to those of count metrics. For examples, researchers utilize the delta method to extend the CUPED estimator to the variance reduction of ratio metrics. The study conducted by [15] focuses on ratio metrics by separately minimizing the variances of the numerator and denominator, but disregards their correlation. A novel work proposed in [4] is the consistent transformation, which transforms ratio metrics into user-level linear metrics. However, it is based on a strong hypothesis where the denominator in ratio metrics is extremely large and can be approximatively taken as a constant.

In this paper, we develop more efficient estimators for count metrics and ratio metrics respectively. For count metrics, we propose a robust estimator called STATE by modeling the regression residual as a Student's t-distribution which was put forth and adopted as a robust building block, for clustering [20, 22] and robust projections[1]. The heavy-tailed nature of the t-distribution makes it much less sensitive to outliers compared to the Gaussian distribution, resulting in a substantial reduction in the variance of the ATE estimate. For ratio metrics, we adopt the main idea of the transformation method [4] while relaxing its assumptions, and introduce STATE method for ratio metrics to decrease the negative influence of outliers.

3 Robust Estimation for Count Metrics with STATE

3.1 Robust Modeling

Following the framework of typical regression adjustment techniques, our proposed ATE estimation procedure can be summarized in two stages: machine learning stage and linear regression stage. **Machine learning stage:** utilize machine learning tools g to capture the relationship between the outcome metric Y and the covariates X. Then, we can obtain a proxy variable $\hat{Y}_i$ [13, 23] composed of ML predictions $g(x_i)$. Notably, we employ the cross-fitting technique [6, 28] here to ensure that the proxy $\hat{Y}_i$ and treatment T_i are independent.

Linear regression stage: include the proxy $\hat{Y}_i$ as a regressor in the linear regression step. The ATE estimation is the estimate of a_1.

$$Y_i = a_0 + a_1 T_i + a_2 \hat{Y}_i + \epsilon_i \tag{3}$$

for simplicity, we denote the Eq. (3) as $Y_i = a^T x_i + \epsilon_i$, where $a = (a_0, a_1, a_2)^T$, and $x_i = (1, T_i, \hat{Y}_i)^T$.

To capture the structure of the typical observations while dealing with outliers automatically, we model the residuals using the Student's t-distribution instead of the Gaussian distribution, here.

$$\epsilon_i \sim S_t(\epsilon | u, \sigma^2, v) \tag{4}$$

As noted in [18], the t-distribution can be re-written as a convolution of a Gaussian distribution with a Gamma distribution placed on its precisions by introducing a latent variable η_i.

$$S_t(\epsilon | u, \sigma^2, v) = \int_0^\infty \mathcal{N}(\epsilon | u, \frac{\sigma^2}{\eta_i}) \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) d\eta_i \tag{5}$$

without loss of generality, the expectation u of the residuals is set to 0. $\mathcal{G}$ is the Gamma density, $\mathcal{G}(\eta | a, b) = \frac{b^a}{\Gamma(a)} \eta^{a-1} e^{-b\eta}$, and $\Gamma(\cdot)$ represents the Gamma function.

It should be noted that the maximum likelihood estimate of a_1 based on Eq. (3),(4) is the ATE estimation. Now we derive the log-likelihood function of the observations in the following form.

$$\begin{aligned}
\mathcal{L} &= \ln \prod_{i=1}^{N} P(Y_i | x_i, \theta) \\
&= \sum_{i=1}^{N} \ln S_t(Y_i | a^T x_i, \sigma^2, v) \\
&= \sum_{i=1}^{N} \ln \int_0^\infty \mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) d\eta_i \\
&\geq \sum_{i=1}^{N} \int_0^\infty q(\eta_i) \ln \frac{\mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2})}{q(\eta_i)} d\eta_i \\
&\equiv \mathcal{F}(Y | x, \theta, q)
\end{aligned} \tag{6}$$

where $q(\eta_i)$ is the posterior of latent variable η_i, $\theta = \{a^T, \sigma^2, v\}$ is the set of parameters of the model. $\mathcal{F}$ is called the variational free energy function. The inequality holds due to Jensen Inequality.

3.2 The Variational Likelihood Bound

The variational free energy $\mathcal{F}$, which is also called the Evidence Lower Bound(ELBO) of the likelihood function. Optimizing the likelihood function $\mathcal{L}$ directly is intractable, we can optimize the ELBO instead by the EM algorithm according to Minorize-Maximization optimization[16, 19]. $\mathcal{F}$ can be evaluated as follows:

$$\mathcal{F}(Y | x, \theta, q) = \sum_{i=1}^{N} (\langle \ln \mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \rangle + \langle \ln \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) \rangle - \langle \ln q(\eta_i) \rangle) \tag{7}$$

where

$$\langle \ln \mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \rangle = -\frac{1}{2} \ln 2\pi\sigma^2 + \frac{1}{2} \langle \ln \eta_i \rangle - \frac{\langle \eta_i \rangle}{2\sigma^2} (Y_i - a^T x_i)^2 \tag{8}$$

$$\langle \ln \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) \rangle = \frac{v}{2} \ln \frac{v}{2} - \ln \Gamma(\frac{v}{2}) + (\frac{v}{2} - 1) \langle \ln \eta_i \rangle - \frac{v}{2} \langle \eta_i \rangle \tag{9}$$

$$\langle \ln q(\eta_i) \rangle = \xi_i \ln \zeta_i - \ln \Gamma(\xi_i) + (\xi_i - 1) \langle \ln \eta_i \rangle - \xi_i \tag{10}$$

where $\langle \cdot \rangle$ represents the expectation of the term conditional on the posterior distribution $q(\eta_i)$, $\langle \ln \eta_i \rangle = \Psi(\xi_i) - \ln \zeta_i$ and $\Psi(\cdot)$ is the di-gamma function.

In the E-step of the $(k+1)$-th iteration of EM algorithm, we maximize $\mathcal{F}$ w.r.t the variational distribution q while fixing the parameters in the k-th iteration, θ^k:

$$q^{k+1}(\eta_i) = argmax_q \mathcal{F}(Y_i | x_i, \theta^k, q)$$

In the M-step, we maximize $\mathcal{F}$ w.r.t the parameters θ to obtain the new parameter values θ^{k+1}

$$\theta^{k+1} = argmax_\theta \mathcal{F}(Y_i | x_i, \theta, q^{k+1})$$

3.3 Deriving the EM Algorithm

3.3.1 Variational E-step.

The posterior distribution of latent variable η_i can be derived by taking functional derivative of $\mathcal{F}(Y_i | x_i, \theta, q(\eta_i))$ w.r.t. the term of $q(\eta_i)$,

$$q(\eta_i) = \frac{exp \langle \ln \mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) \rangle}{\int_0^\infty exp \langle \ln \mathcal{N}(Y_i | a^T x_i, \frac{\sigma^2}{\eta_i}) \mathcal{G}(\eta_i | \frac{v}{2}, \frac{v}{2}) \rangle d\eta_i}$$

After simplification, we can find $q(\eta_i)$ is Gamma density.

$$q(\eta_i) = \mathcal{G}(\eta_i | \xi_i, \zeta_i)$$

where

$$\xi_i = \frac{v}{2} + \frac{1}{2}, \zeta_i = \frac{v}{2} + \frac{(Y_i - a^T x_i)^2}{2\sigma^2}, \langle \eta_i \rangle = \frac{\xi_i}{\zeta_i} \tag{11}$$

3.3.2 M-step.

The parameter $\theta = \{a^T, \sigma^2, v\}$ are obtained by solving the stationary equations of $\mathcal{F}$ w.r.t $a^T = \{a_0, a_1, a_2\}$, σ^2, v respectively,

$$a_0 = \frac{\sum_{i=1}^{N} (Y_i - a_1 T_i - a_2 \hat{Y}_i) \langle \eta_i \rangle}{\sum_{i=1}^{N} \langle \eta_i \rangle} \tag{12}$$

$$a_1 = \frac{\sum_{i=1}^{N} (Y_i - a_0 - a_2 \hat{Y}_i) T_i \langle \eta_i \rangle}{\sum_{i=1}^{N} \langle \eta_i \rangle T_i^2} \tag{13}$$

$$a_2 = \frac{\sum_{i=1}^{N} (Y_i - a_0 - a_1 T_i) \hat{Y}_i \langle \eta_i \rangle}{\sum_{i=1}^{N} \langle \eta_i \rangle \hat{Y}_i^2} \tag{14}$$

$$\sigma^2 = \frac{1}{N} \sum_{i=1}^{N} (Y_i - a_0 - a_1 T_i - a_2 \hat{Y}_i)^2 \langle \eta_i \rangle \tag{15}$$

v is the solution of the following non-linear equation.

$$\sum_{i=1}^{N} (\ln \frac{v}{2} + 1 + \langle \ln \eta_i \rangle - \langle \eta_i \rangle - \Psi(\frac{v}{2})) = 0 \tag{16}$$

In summary, the inference procedure of the STATE method for count metrics is shown in Algorithm 1.

Algorithm 1 Variational EM Inference with STATE

1: **Input:** Data$(X_i, T_i, Y_i)_{i=1}^N$.
2: **Output:** STATE $\hat{a}_1$
3: **ML stage:** train a model $g(x)$ to predict $E[Y|X]$ by cross-fitting.
4: **EM stage:** estimate $\hat{a}_1$ as the ATE estimation.

$$Y_i = a_0 + a_1 T_i + a_2 \hat{Y}_i + \epsilon_i$$

initialize: $(a_0, a_1, a_2, \sigma^2, v) \leftarrow (a_0^{(0)}, a_1^{(0)}, a_2^{(0)}, \sigma^{2(0)}, v^{(0)})$.
5: **while** Free energy $\mathcal{F}$ not converged **do**
6: **E-step** obtain the optimal posterior distribution of η_i:

$$q^{(k+1)}(\eta_i) = \mathcal{G}(\eta_i | \xi_i^{(k+1)}, \zeta_i^{(k+1)})$$

7: where $\xi_i^{(k+1)}, \zeta_i^{(k+1)}$ are computed by Eq. (11).
8: **M-step** estimate the model parameters a_0^{k+1}, a_1^{k+1}, a_2^{k+1},
9: $\sigma^{2(k+1)}$, $v^{(k+1)}$ by Eq. (12)-(16).
10: **Evaluate** $\mathcal{F}$ by Eq. (7)-(10)
11: **end while**
12: **return** STATE: $\hat{a}_1$

3.4 Computational Complexity

Compared to the CUPAC and MLRATE estimators, STATE incurs extra computational costs, primarily due to the implementation of the EM algorithm. Specifically, in each iteration, calculating the posterior distribution parameters $\{\xi_i, \zeta_i\}_{i=1}^n$ and model parameters $\theta = \{a^T, \sigma^2, v\}$ each takes $O(N)$ operations. Consequently, the overall computational complexity of EM algorithm amounts to $O(KN)$, where K denotes the number of iterations. According to the empirical experience on the Meituan experimental platform, the EM procedure typically requires 30 to 60 seconds to process 500,000 observations.

4 Consistent Transformation of Ratio Metrics

The ratio metrics are also common in the industry, which can be regarded as the ratio of two count metrics. For example, the click-through rate is a ratio metric, which is a ratio of the click number and the pageview number. Addressing the variance reduction of ratio metrics is more complex but much less investigated in existing works. In this section, we will introduce consistent transformation so that STATE can be utilized to reduce variance of ratio metrics.

Following the notations in Sec. 2.1, let Y_t, Y_c, Z_t, Z_c be the corresponding count metrics, i.e.,

$$Y_t = \frac{\sum_{T_i=1} Y_i}{N_t}, Y_c = \frac{\sum_{T_i=0} Y_i}{N_c}, Z_t = \frac{\sum_{T_i=1} Z_i}{N_t}, Z_c = \frac{\sum_{T_i=0} Z_i}{N_c}.$$

Let R_t and R_c be the ratio metrics of the treatment group and the control group respectively, i.e.,

$$R_t = \frac{\sum_{T_i=1} Y_i}{\sum_{T_i=1} Z_i} = \frac{Y_t}{Z_t},\ R_c = \frac{\sum_{T_i=0} Y_i}{\sum_{T_i=0} Z_i} = \frac{Y_c}{Z_c}.$$

According to the delta method [7], the variance of ratio metrics can be approximately computed by

$$\begin{aligned} DR_t &= \frac{1}{[EZ_t]^2} DY_t + \frac{[EY_t]^2}{[EZ_t]^4} DZ_t - \frac{2EY_t}{[EZ_t]^3} \mathrm{cov}(Y_t, Z_t) \\ &= \frac{D[Y_t - \alpha_t Z_t]}{[EZ_t]^2}, \end{aligned} \tag{17}$$

where $\alpha_t = EY_t / EZ_t$. Therefore, we have

$$\begin{aligned} D[\Delta R] &= D[R_t - R_c] \\ &= DR_t + DR_c \\ &= \frac{D[Y_t - \alpha_t Z_t]}{[EZ_t]^2} + \frac{D[Y_c - \alpha_c Z_c]}{[EZ_c]^2}, \end{aligned}$$

where $\alpha_c = EY_c / EZ_c$ and the second equality holds due to $R_t \perp R_c$.

Since the analysis unit in ratio metrics does not match the randomization unit, the common regression-adjusted method and robust estimation cannot directly perform variance reduction for ratio metrics [15]. The work [4] transformed ratio metrics $R_t = Y_t / Z_t$ to linear metrics $L_t = Y_t - \kappa Z_t$, where $\kappa = (1 - \eta) R_t + \eta R_c$. After such linear transformation, regression adjustment can be applied for variance reduction of $\Delta L = L_t - L_c$, and the significance level of ΔR can be obtained by performing student t test for ΔL. However, there are two crucial defects in this work. Firstly, it is showed that $\Delta L = ((1 - \eta) Z_c + \eta Z_t) \Delta R$. The coefficient $(1 - \eta) Z_c + \eta Z_t$ is not a constant and thus the significance level of ΔL is not equivalent to that of ΔR. Secondly, the analysis for the variance of ΔL in this work is based on the hypothesis where the parameter κ is taken as a constant, but it actually consists of random variables R_t and R_c.

In this section, we follow the main idea in [4] (transforming ratio metrics to linear metrics), but mitigate the above two defects simultaneously. Specifically, we construct a new linear metric ΔP and let $\Delta U = \frac{\Delta P}{EZ_t EZ_c}$, which preserves both unbiased estimation and consistent variance, i.e.,

Property 1. Unbiased estimation: $E[\Delta U] = \tau_r$,
Property 2. Consistent variance: $D[\Delta U] \approx D[\Delta R]$,

where $\tau_r = \frac{E[Y_i(1)]}{E[Z_i(1)]} - \frac{E[Y_i(0)]}{E[Z_i(0)]}$ is ATE of ratio metrics. Since $EZ_t EZ_c$ is a constant, Property 1 guarantees that the significance level of ΔP is equivalent to that of τ_r. Although the value of $EZ_t EZ_c$ is unknown in practice, we can obtain the significance level of τ_r by making hypothesis testing for ΔP. Based on Property 2, performing regression adjustment for ΔP will result in a consistent estimator of ΔU with smaller variance than ΔR. Furthermore, robust estimation can be introduced and eliminate the influence of the atypical and outlying observations.

Construction of ΔP**.** For each sample, construct the new label as $P_i = \kappa_1 Y_i - \kappa_2 Z_i$. Hence, we have $P_t = \frac{\sum_{T_i=1} P_i}{N_t} = \kappa_1 Y_t - \kappa_2 Z_t$, $P_c = \frac{\sum_{T_i=0} P_i}{N_c} = \kappa_1 Y_c - \kappa_2 Z_c$ and $\Delta P = P_t - P_c = \kappa_1 \Delta Y - \kappa_2 \Delta Z$.

Proof of Property 1. Follow the above construction of ΔP and set $\kappa_1 = Z_c$ and $\kappa_2 = Y_c$. We further have $\Delta P = Y_t Z_c - Z_t Y_c$ and

$$\begin{aligned} E[\Delta U] &= \frac{E[Y_t Z_c - Z_t Y_c]}{EZ_t EZ_c} \\ &= \frac{EY_t EZ_c - EZ_t EY_c}{EZ_t EZ_c} \\ &= \frac{EY_t}{EZ_t} - \frac{EY_c}{EZ_c} \\ &= \frac{E[Y_i(1)]}{E[Z_i(1)]} - \frac{E[Y_i(0)]}{E[Z_i(0)]} \\ &= \tau_r, \end{aligned}$$

where the second equality holds due to $Y_t, Z_t \perp Y_c, Z_c$.

Proof of Property 2. Define $g(Y_t, Z_t, Y_c, Z_c) = Y_t Z_c - Z_t Y_c$. According to the delta method, the variance of ΔU can be rewritten as

$$\begin{aligned} D[\Delta U] &= \frac{D[Y_t Z_c - Z_t Y_c]}{[EZ_t EZ_c]^2} \\ &= \frac{\nabla^T g(\mu) \sum \nabla g(\mu)}{[EZ_t EZ_c]^2}, \end{aligned}$$

where $\mu = (EY_t, EZ_t, EY_c, EZ_c)$, $\nabla^T g(\mu) = (EZ_c, -EY_c, -EZ_t, EY_t)$ is the Jacobian and

$$\sum = \begin{pmatrix} DY_t & \mathrm{cov}(Y_t, Z_t) & 0 & 0 \\ \mathrm{cov}(Y_t, Z_t) & DZ_t & 0 & 0 \\ 0 & 0 & DY_c & \mathrm{cov}(Y_c, Z_c) \\ 0 & 0 & \mathrm{cov}(Y_c, Z_c) & DZ_c \end{pmatrix}$$

is the covariance matrix. Therefore, we further have

$$\begin{aligned} D[\Delta U] =& \frac{\nabla^T g(\mu) \sum \nabla g(\mu)}{[EZ_t EZ_c]^2} \\ =& \frac{1}{[EZ_t]^2} \left[DY_t + \frac{[EY_c]^2}{[EZ_c]^2} DZ_t - \frac{2EY_c}{EZ_c} \mathrm{cov}(Y_t, Z_t) \right] + \\ & \frac{1}{[EZ_c]^2} \left[DY_c + \frac{[EY_t]^2}{[EZ_t]^2} DZ_c - \frac{2EY_t}{EZ_t} \mathrm{cov}(Y_c, Z_c) \right] \\ =& \frac{D[Y_t - \alpha_c Z_t]}{[EZ_t]^2} + \frac{D[Y_c - \alpha_t Z_c]}{[EZ_c]^2}, \end{aligned}$$

where $\alpha_t = EY_t/EZ_t$ and $\alpha_c = EY_c/EZ_c$. As stated in [4], in the real online experiments of the industry, the increment in ratio metrics of the treatment group is usually small and is tightly bounded with deviation of several percents. Hence, we have $\alpha_t \approx \alpha_c$, which indicates that $D[\Delta U] \approx D[\Delta R]$ holds.

5 Experimental Evaluation

In this section, we construct large-scale experiments to demonstrate the effectiveness of STATE using both simulated data and real user data from the Meituan food delivery platform. Firstly, we validate the performance of various methods in reducing the variance of count metrics and ratio metrics in simulation data. For completeness, we also investigate the impact of the proportion of outliers in the data set on STATE. Secondly, to show the magnitude of variance reduction that can be achieved in practice, we perform a series of A/A tests on key metrics of interest for Meituan food delivery business. Finally, we discuss limitations of the STATE estimator.

5.1 Experimental Setup

Simulation data. The generating process is almost the same as [13]. Denote each sample by (X_i, T_i, Y_i, Z_i). The covariates X_i is distributed as $X_i \sim \mathcal{N}(u, I_{5\times5})$, where u is generated from a uniform distribution $U(0, 10)$. The outcome variables Y_i and Z_i are constructed by $Y_i = b(X_i) + T_i * \tau_y(X_i) + \epsilon_i$ and $Z_i = c(X_i) + T_i * \tau_z(X_i) + \eta_i$ respectively, where ϵ_i and η_i are the error terms distributed as $\epsilon_i \sim \mathcal{N}(0, 25^2)$, $\eta_i \sim \mathcal{N}(0, 10^2)$. Let $b(X_i)$ and $c(X_i)$ be nonlinear functions with the forms

$$\begin{cases} b(X_i) = 10\sin(\pi X_{i1} X_{i2}) + 6X_{i3}^2 + 10|X_{i4}| + 5|X_{i5}| + 50, \\ c(X_i) = 10\sin(\pi X_{i4}) X_{i5} + 15(X_{i2} + X_{i3})^2 + 5|X_{i1}| + 30. \end{cases}$$

The treatment effect is also given by a nonlinear form as

$$\begin{cases} \tau_y(X_i) = X_{i1} X_{i3} + \log(1 + e^{X_{i2}}), \\ \tau_z(X_i) = X_{i2}^2 + 3\log(1 + e^{X_{i4}} + |X_{i5}|). \end{cases}$$

Additionally, in order to validate the performance of our STATE estimator in dealing with heavy-tailed metrics, we also add a certain proportion of outliers to the simulation data set. The outliers are randomly generated from two uniform distributions $U(\overline{Y} + 4\sigma_y, \overline{Y} + 20\sigma_y)$ and $U(\overline{Z} + 4\sigma_z, \overline{Z} + 20\sigma_z)$, where σ_y and σ_z are the standard deviation of Y_i and Z_i respectively.

We generate 200K samples in total and perform 1000 simulation experiments. In each simulation, we randomly choose 20K samples and divide them equally into the treatment group and the control group according to the treatment indicator T_i which follows $T_i \sim \mathrm{Bernoulli}(0.5)$. Finally, we compare the variance of different ATE estimators in all the simulations.

Real user data. The real user data is collected from Meituan food delivery platform, which contains over 2.3 million samples and 58 covariates. Each sample records all the transaction information for a user during an experiment in this platform. To show the effectiveness of the STATE estimator, we select two count metrics, the count of orders per user (orders in short) and GMV per user, and a ratio metric, the average price per order (AvgOrdPrice). Here, GMV refers to the total price paid by each user on the food delivery platform during the trial period. The average price per order is calculated by dividing the sum of GMV of all users by the total number of orders of all users.

For each metric, we perform A/A tests by selecting 200k users randomly and assigning the treatment indicator $T_i \sim \mathrm{Bernoulli}(0.5)$ for each user. Similarly, the A/A test is also repeated 1000 times. Notice that the A/A test is a controlled experiment where treatment is identical to control, hence the ground truth of ATE is 0.

Benchmark. For both count and ratio metrics, multiple methods for variance reduction are implemented and taken as the benchmarks.

- Count metrics
  - DIM. The difference-in-mean estimator that computes ATE by Eq. (1) for count metrics.
  - CUPED. The state-of-the-art linear method proposed in [9], which reduces variance of count metrics by utilizing the pre-experiment covariates and gives an unbiased estimation of ATE.
  - CUPAC. Similar to CUPED, but replace the pre-experiment covariates with a proxy highly correlated with the outcome variable [23].

  - MLRATE. The machine learning regression-adjusted estimator proposed in [13].
  - STATE. The robust regression-adjusted estimator for variance reduction proposed in this paper.
- Ratio metrics
  - DIM. The difference-in-mean estimator that computes ATE of ratio metrics by Eq. (2).
  - CUPED. The generalization to ratio metrics proposed in Appendix B of [9].
  - CTRM. The consistent transformation method of ratio metrics proposed in [4].
  - STATE. The robust regression-adjusted estimator for ratio metrics after consistent transformation.

Evaluation Metrics. We compare various variance reduction techniques on two primary evaluation metrics: bias and variance. Consistent with previous studies[13, 15], we assess the bias by utilizing the empirical coverage rate of the 95% confidence intervals (CI). The closer the empirical coverage rate is to the nominal coverage rate(95%), the smaller the bias of the estimator. On the other hand, for the variance metric, the smaller the value, the shorter the confidence interval and the higher the efficiency of the estimator.

- Empirical coverage. The empirical coverage represents the proportion of 1000 simulations in which the 95% confidence interval covers the ground truth.
- Variance. The variance of ATE estimates across 1000 simulation experiments.

5.2 Simulation Experiment

In this section, we utilize simulation experiments to validate the effectiveness of the algorithms proposed in this paper. First of all, Table 1, 2 and Fig. 2 show the simulation results for count metrics and ratio metrics on the dataset with 0.5% outliers. Furthermore, we investigate the impact of the proportion of outliers on various variance reduction techniques. "Var.Red%" displays the variance reduction gains for each method compared to DIM estimator. "Emp.Cov%" displays the empirical coverage of the 95% CIs.

Table 1: Simulation Results for count metric

Methods	DIM	CUPED	CUPAC	MLRATE	STATE
Emp.Cov%	94.6	94.6	94.2	94.0	95.6
Var.Red%	0	28.7	32.6	32.4	84.1

Table 2: Simulation results for ratio metric

Methods	DIM	CUPED	CTRM	STATE
Emp.Cov%	95.1	94.3	94.6	94.8
Var.Red%	0	6.7	55.7	90.1

5.2.1 *Count Metrics.*

As shown in Table 1, the empirical coverage of all count metric estimators closely approximates the nominal coverage (95%), indicating that the biases associated with these estimators are minimal.

Fig. 2(a) depicts the probability density of ATE estimates, revealing that STATE's ATE distribution is markedly more compact, with its variance being substantially reduced compared to that of other methods. Specifically, the CUPED method reduces the variance by 28.7% compared to the DIM estimator; CUPAC and MLRATE perform similarly and deliver additional gains relative to CUPED. It indicates that the proxy variable constructed by the ML model has a superior correlation with the outcome than a single covariate. In contrast, the performance of the STATE estimator is remarkable, achieving an over 80% variance reduction compared to the DIM estimator and significantly enhancing the precision of ATE estimation. Considering both the empirical coverage rate and estimator variance, STATE clearly emerges as the preferred ATE estimator.

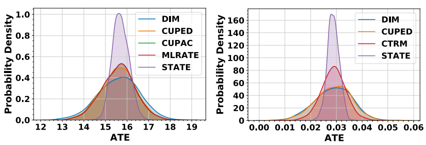
5.2.2 *Ratio Metrics.*

Fig. 2(b) and Table 2 present the detailed experimental results of variance reduction for ratio metrics in simulation data. As is shown by Table 2, the empirical coverage of all the estimators are close to the nominal coverage, in which STATE achieves the best performance on variance reduction. Specifically, CUPED only reduces 6.7% of variance compared to DIM duo to the weak correlation between the pre-experiment covariates and the ratio metric. CTRM performs much better, which achieves over 55.7% variance reduction compared with DIM. However, the effectiveness of CTRM is also limited by the correlation between the ML-based predictors and the outcome metrics, which is easily influenced by big outliers. As a robust estimator, STATE can significantly decrease the negative effects for variance caused by outliers, which contributes to over 90.1% variance reduction relative to DIM.

5.2.3 *Factors Affecting STATE Effectiveness.*

Now let's look at the factors that affect the performance of STATE. Here, we discuss the problem from a general setting to specific settings by constructing a Gaussian distribution dataset and adding different proportions of outliers from 0 to 1%. Fig. 3(a) and Fig. 3(b) display the results of empirical coverage and variance reduction respectively.

From the Fig. 3(b), we can see that when the dataset does not contain outliers, more precisely, the metrics follow a Gaussian distribution, STATE, CUPAC, and MLRATE perform similarly in variance reduction. As the proportion of outliers increases, STATE and baseline methods exhibit drastically different performance. The reason is that the degree of variance reduction of typical methods such as CUPED, CUPAC and MLRATE depends on the correlation between the constructed covariates or proxy variables and business metrics.



(a) PDF of ATE for count metric (b) PDF of ATE for ratio metric

Figure 2: Simulation results of ATE estimation

Table 3: Summary of A/A test results for count metrics

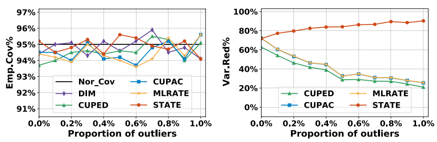
Metric	DIM	CUPED		CUPAC		MLRATE		STATE	
	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%
orders	94.9	47.2	94.9	54.6	94.5	54.6	94.5	70.5	95.1
GMV	94.8	33.3	93.9	44.3	94.5	44.4	94.4	80.7	95.2

Table 4: Summary of winsorized results at the 99.9th percentile threshold

Metric	Winsorized DIM		Winsorized CUPED		Winsorized CUPAC		Winsorized MLRATE		Huber Regression	
	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%
orders	1.1	94.7	49.0	94.5	55.9	94.4	55.9	94.3	54.9	95.0
GMV	12.4	94.7	38.3	93.9	57.3	94.3	54.6	94.4	47.3	94.5

Table 5: Summary of winsorized results at the 99th percentile threshold

Metric	Winsorized DIM		Winsorized CUPED		Winsorized CUPAC		Winsorized MLRATE		Huber Regression	
	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%
orders	9.76	94.1	53.0	94.0	60.4	94.1	60.5	94.0	54.9	95.0
GMV	21.7	94.5	49.8	93.5	71.0	94.0	67.7	93.8	47.3	94.5



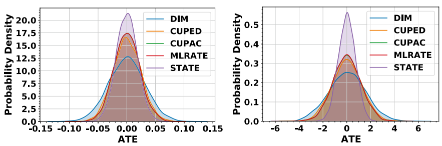
(a) Empirical coverage for all estimators (b) Variance reduction compared to DIM

Figure 3: Impact of proportions of outliers

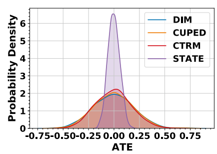
However, with the existence of outliers, correlation learning becomes increasingly difficult. The result is that the performance of these typical techniques deteriorates gradually, degrading to the variance scale of the DIM method. The STATE estimator we proposed is designed to address such challenges by modeling the noise as a Student's t-distribution. Because of the excellent robustness of the T estimate, the rate of variance increase of STATE is significantly slower than the DIM method with the increase of outliers. Consequently, the more pronounced the heavy tail phenomenon of business metrics, the more apparent the advantage of the STATE method in variance reduction.

5.3 Empirical Results

In this section, we perform two real experiments on the Meituan food delivery platform, analyzing the count metrics and ratio metrics, respectively. Here, we focus on the A/A tests, not the A/B tests running in production. This is because the true effect of the A/B tests is unknown, which makes it impossible to evaluate the empirical coverage rate of the confidence interval. Since the average treatment effect (ATE) in online experiments is typically small and unlikely to significantly change the relationship between the outcomes and covariates, the reduction in variance in A/B tests should be very similar to that in A/A tests.



(a) PDF of ATE for orders (b) PDF of ATE for GMV



(c) PDF of ATE for AvgOrdPrice

Figure 4: Empirical results of ATE estimation

5.3.1 *Count metrics.*

On the food delivery platform, business metrics with a heavy-tailed distribution are quiet common. This is typically caused by

Table 6: Summary of A/A test results for ratio metrics

Metric	DIM	CUPED		CTRM		STATE	
	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%	Var.Red%	Emp.Cov%
AvgOrdPrice	95.0	6.4	95.6	21.8	94.9	91.6	95.7

users' accidental consumption behavior. For example, a user orders takeaway to treat friends to dinner, the consumption price of her order may be a extremely large value relative to her historical consumption pattern. These sample points that fall on the tail (also known as outliers) are not affected by treatment, but has great impact on the sensitivity of measuring ATE.

To get a better sense of the magnitudes of variance reduction that the STATE method might achieve in practice, we select two key count metrics: orders per user and GMV per user, both of which have heavy-tail distributions. For each metric, we construct A/A tests. Table 3 and Fig. 4 show the empirical results. It indicates that the STATE performs substantially better than the CUPED, CUPAC and MLRATE estimators on both orders metric and GMV metric. On average, the STATE estimator reduces the variance of the order metric and GMV metric by 70.5% and 80.7% respectively compared to the DIM method, whereas the analogous figures for the state-of-the-art methods are about 54.6% and 44.4%. It demonstrates that STATE can indeed considerably improve the sensitivity of real business metrics, which is of great significance for increasing business profits and reducing the cost of experimental time.

Furthermore, we compare STATE method with the typical winsorization [11] method and Huber regression [5] for dealing with heavy-tailed metrics. Table 4 and 5 display the results of winsorization at different thresholds(99% and 99.9% quantiles of the observations). It is observed that estimation is sensitive to the threshold. A lower threshold results in more observations beyond threshold being restrained, which consequently yields a reduced variance in parameter estimation. However, there is a concomitant decline in the empirical coverage rate, suggesting an incremental increase in bias. Consequently, employing the Winsorization process necessitates a meticulous balance between the impact of bias and variance. As shown in Table 3 and 4, the variance of the parameters estimated by the Huber regression method is slightly smaller than that of the CUPAC and MLRATE methods (without winsorization). This is because Huber regression transforms the mean square loss function of extreme observations into a linear form, thereby partially reducing the impact of outliers. However, the results also show that the robustness of the Huber regression method is limited, exhibiting a considerable disparity when compared to the STATE method.

5.3.2 *Ratio metrics.*

Fig. 4(c) and Table 6 present the results for ratio metric named AvgOrdPrice in real user data, which is calculated by dividing the sum of GMV of all users by the total number of orders. As is shown by Table 6, the empirical coverage of all the estimators converges to 95%, and STATE still performs the best because of the strong capacity of resisting disturbance for outliers that exactly exists in real data. In summary, the variance reduction of STATE is about 91.6% relative to DIM, relative to CUPED is 91.0%, relative to CTRM is 89.3%.

5.4 Limitations

In the preceding discussions, STATE has achieved notable success in the application of variance reduction for both count metrics and ratio metrics. Now let's discuss the applicability of the STATE method. Inspired by section 5.2.3, we find the effectiveness of the STATE method varies from metric to metric, depending on whether the distribution of the metric has a heavy tail. When metric follows a Gaussian distribution, the STATE method is comparable with the state-of-the-art methods, but introduces extra computational cost. If the metric exhibits a heavy tail distribution, the STATE method demonstrates a strong advantage.

6 Conclusion

Variance reduction is the common technology to improve the sensitivity in online controlled experiments, which contributes to smaller user population and shorter experimental period. However, typical methods cannot characterize the heavy-tailed distributions in real business metrics, whose efficiency is greatly limited by the outlying observations. In this paper, we proposed a machine-learning-based regression adjustment method with Student's t-distributed errors to reduce the impact of outliers in variance reduction for count metrics. Furthermore, we transform ratio metrics into a linear combination but preserve unbiased estimation and consistent variance, and apply the above method to the variance reduction of ratio metrics. Both synthetic and real data demonstrate a significant decrease in variance for both count and ratio metrics compared to state-of-the-art methods. It benefits from our robust estimator that is stronger to resist disturbance of outliers. We recommend to use the method especially in the experiments where the treatment effect is not very significant. In this case, the influence of outliers is relative larger but can be effectively mitigated by our method.

As the method significantly improves the robustness of controlled experiment results, we would like to expand it to the estimation of individual treatment effects (ITE) in future works, which also suffers from the negative effects of outliers.

Acknowledgments

This work was supported in part by National Key R&D Program of China (2022YFB290180), the NSF of China (62172206), and the Xiaomi Foundation.

References

[1] Cédric Archambeau, Nicolas Delannay, and Michel Verleysen. 2006. Robust Probabilistic Projections. In *ACM International Conference on Machine Learning (ICML)*. 33–40.

- [2] Peter M Aronow and Joel A Middleton. 2013. A Class of Unbiased Estimators of the Average Treatment Effect in Randomized Experiments. *Journal of Causal Inference* 1, 1 (2013), 135–154.
- [3] Eytan Bakshy and Dean Eckles. 2013. Uncertainty in Online Experiments with Dependent Data: An Evaluation of Bootstrap Methods. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1303–1311.
- [4] Roman Budylin, Alexey Drutsa, Ilya Katsev, and Valeriya Tsoy. 2018. Consistent Transformation of Ratio Metrics for Efficient Online Controlled Experiments. In *ACM International Conference on Web Search and Data Mining (WSDM)*. 55–63.
- [5] Pauline Burke et al. 2019. Measuring Average Treatment Effect from Heavy-tailed Data. *arXiv preprint arXiv:1905.09252* (2019).
- [6] Victor Chernozhukov, Denis Chetverikov, Mert Demirer, Esther Duflo, Christian Hansen, Whitney Newey, and James Robins. 2018. Double/Debiased Machine Learning for Treatment and Structural Parameters. *The Econometrics Journal* 21, 1 (2018), C1–C68.
- [7] Anirban DasGupta. 2008. *Asymptotic Theory of Statistics and Probability*. Vol. 180. Springer.
- [8] Alex Deng, Michelle Du, Anna Matlin, and Qing Zhang. 2023. Variance Reduction Using In-Experiment Data: Efficient and Targeted Online Measurement for Sparse and Delayed Outcomes. In *ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD)*. 3937–3946.
- [9] Alex Deng, Ya Xu, Ron Kohavi, and Toby Walker. 2013. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. In *ACM International Conference on Web Search and Data Mining (WSDM)*. 123–132.
- [10] Alex Deng, Lo-Hua Yuan, Naoya Kanai, and Alexandre Salama-Manteau. 2023. Zero to Hero: Exploiting Null Effects to Achieve Variance Reduction in Experiments with One-sided Triggering. In *ACM International Conference on Web Search and Data Mining (WSDM)*. 823–831.
- [11] Wilfrid J Dixon. 1960. Simplified estimation from censored normal samples. *The Annals of Mathematical Statistics* (1960), 385–391.
- [12] David A Freedman. 2008. On Regression Adjustments to Experimental Data. *Advances in Applied Mathematics* 40, 2 (2008), 180–193.
- [13] Yongyi Guo, Dominic Coey, Mikael Konutgan, Wenting Li, Chris Schoener, and Matt Goldman. 2021. Machine Learning for Variance Reduction in Online Experiments. In *Conference on Neural Information Processing Systems (NIPS)*. 8637–8648.
- [14] Henning Hohnhold, Deirdre O'Brien, and Diane Tang. 2015. Focusing on the Long-term: It's Good for Users and Business. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 1849–1858.
- [15] Ying Jin and Shan Ba. 2023. Toward Optimal Variance Reduction in Online Controlled Experiments. *Technometrics* 65, 2 (2023), 231–242.
- [16] Kenneth Lange, David R Hunter, and Ilsoon Yang. 2000. Optimization Transfer Using Surrogate Objective Functions. *Journal of Computational and Graphical Statistics* 9, 1 (2000), 1–20.
- [17] Lin and Winston. 2013. Agnostic Notes on Regression Adjustments to Experimental Data: Reexamining Freedman's Critique. *The Annals of Applied Statistics* 7, 1 (2013), 295–318.
- [18] Chuanhai Liu and Donald B Rubin. 1995. ML Estimation of the T Distribution using EM and Its Extensions, ECM and ECME. *Statistica Sinica* 5, 1 (1995), 19–39.
- [19] Sobhan Naderi Parizi, Kun He, Reza Aghajani, Stan Sclaroff, and Pedro Felzenszwalb. 2019. Generalized Majorization-Minimization. In *ACM International Conference on Machine Learning (ICML)*. 5022–5031.
- [20] David Peel and Geoffrey J McLachlan. 2000. Robust Mixture Modelling Using the T Distribution. *Statistics and Computing* 10 (2000), 339–348.
- [21] Jasjeet S Sekhon. 2008. The Neyman-Rubin Model of Causal Inference and Estimation via Matching Methods. *The Oxford Handbook of Political Methodology* 2 (2008), 1–32.
- [22] Markus Svensén and Christopher M Bishop. 2005. Robust Bayesian Mixture Modelling. *Neurocomputing* 64 (2005), 235–252.
- [23] Yixin Tang, Caixia Huang, David Kastelman, and Jared Bauman. 2020. Control Using Predictions as Covariates in Switchback Experiments. (2020).
- [24] Stefan Wager, Wenfei Du, Jonathan Taylor, and Robert J Tibshirani. 2016. High-Dimensional Regression Adjustments in Randomized Experiments. *National Academy of Sciences* 113, 45 (2016), 12673–12678.
- [25] Edward Wu and Johann A Gagnon-Bartsch. 2018. The LOOP Estimator: Adjusting for Covariates in Randomized Experiments. *Evaluation Review* 42, 4 (2018), 458–488.
- [26] Ya Xu and Nanyu Chen. 2016. Evaluating Mobile Apps with A/B and Quasi A/B Tests. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*. 313–322.
- [27] Li Yang and Anastasios A Tsiatis. 2001. Efficiency Study of Estimators for a Treatment Effect in a Pretest–Posttest Trial. *The American Statistician* 55, 4 (2001), 314–321.
- [28] Wenjing Zheng and Mark J van der Laan. 2011. Cross-Validated Targeted Minimum-Loss-Based Estimation. *Targeted Learning: Causal Inference for Observational and Experimental Data* (2011), 459–474.

Unified Dual-Intent Translation for Joint Modeling of Search and Recommendation

Yuting Zhang*
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhangyuting21s@ict.ac.cn

Yiqing Wu*
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
iwu_yiqing@163.com

Ruidong Han
Meituan
Beijing, China
hanruidong@meituan.com

Ying Sun
Thrust of Artificial Intelligence, The Hong Kong University of Science and Technology (Guangzhou)
Guangzhou, China
yings@hkust-gz.edu.cn

Yongchun Zhu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
zhuyc0204@gmail.com

Xiang Li
Wei Lin
Meituan
Beijing, China
lixiang245@meituan.com
linwei31@meituan.com

Fuzhen Zhuang†
Institute of Artificial Intelligence, Beihang University
Beijing, China
Zhongguancun Laboratory
Beijing, China
zhuangfuzhen@buaa.edu.cn

Zhulin An†
Yongjun Xu
Institute of Computing Technology, Chinese Academy of Sciences
Beijing, China
anzhulin@ict.ac.cn
xyj@ict.ac.cn

ABSTRACT

Recommendation systems, which assist users in discovering their preferred items among numerous options, have served billions of users across various online platforms. Intuitively, users' interactions with items are highly driven by their unchanging inherent intents (e.g., always preferring high-quality items) and changing demand intents (e.g., wanting a T-shirt in summer but a down jacket in winter). However, **both types of intents are implicitly expressed in recommendation** scenario, posing challenges in leveraging them for accurate intent-aware recommendations. Fortunately, **in search** scenario, often found alongside recommendation on the same online platform, **users express their demand intents explicitly through their query words**. Intuitively, in both scenarios, a user shares the same inherent intent and his/her interactions may be influenced by the same demand intent. It is therefore feasible to utilize the interaction data from both scenarios to reinforce the dual intents for joint intent-aware modeling. But the joint modeling should deal with two problems: (1) **accurately modeling users' implicit demand intents in recommendation**; (2) **modeling the relation between the dual intents and the interactive items**. To address these problems, we propose a novel model named Unified Dual-Intents Translation for joint modeling of Search and Recommendation (UDITSR). To accurately simulate users' demand intents in recommendation, we utilize real queries from search data as supervision information to guide its generation. To explicitly model the relation among the triplet <inherent intent, demand intent, interactive item>, we propose a dual-intent translation propagation mechanism to learn the triplet in the same semantic space via embedding translations. Extensive experiments demonstrate that UDITSR outperforms SOTA baselines both in search and recommendation tasks. Moreover, our model has been deployed online on Meituan Waimai platform, leading to an average improvement in GMV (Gross Merchandise Value) of 1.46% and CTR(Click-Through Rate) of 0.77% over one month.

*Yuting Zhang and Yiqing Wu are also at University of Chinese Academy of Sciences
†Fuzhen Zhuang and Zhulin An are Corresponding authors.

KDD '24, August 25–29, 2024, Barcelona, Spain

ACM ISBN 979-8-4007-0490-1/24/08.
https://doi.org/10.1145/3637528.3671519

CCS CONCEPTS

• **Information systems → Recommender systems**.

KEYWORDS

Joint learning, Search and recommendation, Dual intent modeling, Intent translation

ACM Reference Format:
Yuting Zhang, Yiqing Wu, Ruidong Han, Ying Sun, Yongchun Zhu, Xiang Li, Wei Lin, Fuzhen Zhuang, Zhulin An, and Yongjun Xu. 2024. Unified Dual-Intent Translation for Joint Modeling of Search and Recommendation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '24), August 25–29, 2024, Barcelona, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3637528.3671519

1 INTRODUCTION

Aiming to help users discover items of interest from a vast array of options, recommendation systems have become an essential component of various online platforms, such as e-commerce [26, 48, 49] and digital news services [7, 19, 35]. Existing recommendation models [15, 16, 48, 49] typically exploit users' implicit feedback, such as click history, to predict their interests. For instance, traditional Collaborative Filtering (CF) [16] assumes that users will interact with items similar to those with which they've previously interacted. Furthermore, various models [48, 49] have been developed to capture the sequential dynamics of users' implicit feedback to model their evolving interests.

In practice, user feedback patterns in recommendation systems are highly driven by their complex intents, which can be broadly categorized into unchanging inherent intents and changing demand intents. For example, Amy and Tom may have the same noodle demand but choose different restaurants due to Amy's inherent intent for spicy flavors and Tom's for sweet. Besides, a single user's interactions can vary due to their changing demands. Yet, these intents are often implicitly expressed in the recommendation, presenting a challenge for accurate intent-aware recommendations. Existing intent-aware recommendation models [5, 23, 50] typically rely on users' implicit feedback to learn their intents. However, these models encounter a significant problem: different users may have different inherent or demand intents despite similar historical feedback. As shown in Figure 1(a), Amy's interaction with Pizza Hut might indicate a demand intent for pasta, while Tom may demand pizza instead. Ideally, recommendation systems should suggest pasta-related options to Amy and pizza-related ones to Tom. However, without any explicit intent information, existing models struggle to distinguish between these intents, resulting in inaccurate recommendations.

Fortunately, in search services, which often accompany recommendation services on the same online platform, users explicitly express their demand intents through query words, as shown in Figure 1(b). Such explicit search demand information can serve as additional explicit information to assist in learning implicit demand intents for recommendation. Indeed, both search and recommendation tasks aim to comprehend users' intents to aid them in obtaining desired items [2]. In addition, in search scenario, users' interactions are influenced not only by their explicit demand intents but also by their personalized inherent intents. Yet, search models typically focus on the match between search results and users' demand intents, often overlooking the impact of their personalized inherent intents, which are indeed significant [33]. Intuitively, in both scenarios, a user maintains the same inherent intent and his/her behaviors are likely to be determined by the same demand intent. Therefore, it is feasible to leverage interaction data from both scenarios to reinforce or complement each other's dual intents for joint intent-aware modeling. Nevertheless, this joint modeling is not trivial due to the following challenges:

(1) **How to accurately model a user's implicit demand intent in recommendation with search data?** A user's demand intent is implicit within recommendation but is explicitly indicated by search queries. If the changing demand intents in recommendation can be accurately generated, search and recommendation

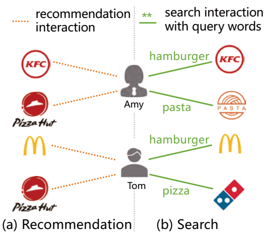


Figure 1: Examples of interaction behaviors in recommendation and search scenarios.

can be well modeled in a unified manner. The existing method, SRJGraph [47], employs the unchanging padding query in recommendation for unified modeling. This approach assumes an unchanging demand intent across all recommendation interactions, which may hinder recommendation performance. To learn demand intents, an intuitive approach is to simply incorporate users' historical queries as additional demand information into the recommendation model. However, without explicit supervision to verify the accuracy of demand intents, there may be a significant discrepancy between the learned and the actual demand intents.

(2) **How to couple the dual intents to model the relation among the intents and the interactive items?** Both inherent intent and demand intent affect the interactive item. Intuitively, the superimposition of inherent intents (e.g., preferring *cheap* items) and changing demand intents (needing a *T-shirt* in summer but a *down jacket* in winter) leads to changing interactive results (interacting with a *cheap T-shirt* and *cheap down jacket*, respectively). In essence, the demand intent can be regarded as the changing deviation from the inherent intent to the changing interactive item. A common approach is to simply feed the two intents as input features, but it cannot fully capture the relation between the dual intents and the interactive item.

To tackle these challenges, we propose a novel model named Unified Dual-Intent Translation for joint modeling of Search and Recommendation (UDITSR). Overall, UDITSR comprises a search-supervised demand intent generator and a dual-intent translation module. Specifically, in the demand intent generator, search queries serve as supervision information, allowing us to learn and understand a user's changing demand intent for recommendations both reliably and accurately. Moreover, we develop a dual-intent translation propagation mechanism. This mechanism explicitly models the interpretable relation among the triplet elements–user's <inherent intent, demand intent, interactive item>–within a shared semantic space by employing embedding translations. Particularly, we design an intent translation contrastive learning to further constrain the translation relation. Extensive offline and online experiments were conducted to demonstrate our model's effectiveness. To gain deeper insights into the effectiveness of our model, we also provide a visual analysis of relevant intents.

2 RELATED WORK

2.1 Recommendation and Search Models

Recommendation aims to filter items from vast candidate pools to match user interests. Traditional models, such as Collaborative Filtering (CF), assume users with similar behaviors share item preferences [6, 13, 15, 31]. Later studies [11, 34, 49] focus on decoding users' evolving interests from their historical behaviors, using techniques like DIN [49], which employs attention mechanism to connect past behaviors with current targets. Recognizing that users' interactions are driven by their intrinsic intents, recent studies [5, 37, 38, 50] exploit users' historical behavior sequences to understand their changing intents, aiming to better meet user needs. For instance, KA-MemNN [50] uses item categories from user behavior as intent proxies, implementing memory networks for dynamic intent modeling. However, these approaches often deduce intents from interaction behaviors or directly equate behavior with intent, without mining real intrinsic intents. In contrast, our model utilizes the user's actual demand intents in the search scenario as supervision information to imitate the intents in recommendation.

Search and recommendation services often coexist on the same platform [27]. Earlier research [2] suggests their goals are essentially equivalent−helping people get the items they want, prompting studies on their joint optimization. For example, JSR [44] introduces a shared-parameter framework, with user and item embeddings shared. USER [43] treats recommendation behavior as a form of search behavior with unchanging padding query, unifying the modeling of search and recommendation sequences. Furthermore, SRJgraph [47] constructs a unified graph from search and recommendation behaviors, incorporating search queries and a padding query for recommendation as attributes of user-item edges. These models assume the query-related intents in recommendation are unchanging while the matching degree between the query and the candidate items significantly affects search performance. This assumption creates a significant gap between the modeling of search and recommendation, greatly hindering the effectiveness of joint modeling approaches. Our model, however, adapts to learn personalized and changing query-related intents for distinct user-item pairs in recommendation, thus enhancing the unification of joint search and recommendation.

2.2 Graph Neural Network

Graph Neural Networks (GNNs) [32, 41] have gained tremendous attention in recent years due to their remarkable ability to process graph-structured data. For instance, Graph Convolutional Network (GCN) [18] employs a localized filter to aggregate information from neighbors, and Graph Attention Network (GAT) [36] leverages the attention mechanism to weigh the importance of each neighbor node during the aggregation process. Since then, numerous variants of GNNs [8, 42, 46] have been proposed to tackle various types of graphs. Nowadays, Graph Neural Networks have shown great potential in a wide range of applications, such as recommendation [14, 39, 40] and search [10, 22, 25] scenarios. In this work, we propose incorporating demand intents that are generated through search supervision in recommendation scenario, as well as explicitly stated search intents, into the construction of a unified graph. Specifically, these demand intents serve as the attributes of the edges connecting users and items. Moreover, the invariant node representations for a user across different interactions are used to indicate their inherent intents. Based on the graph, we propose a novel dual-intent translation propagation for unified dual intent-aware modeling.

3 PRELIMINARY

Let $\mathcal{U}$ and $\mathcal{I}$ denote the universal sets of users and items in both search and recommendation scenarios. In order to distinguish these two scenarios, we define the interaction records in each scenario as follows:

Definition 1.***search scenario***: In the search data $\mathcal{X}_s$, each interaction record $x_s \in \mathcal{X}_s$ can be formulated as $x_s = (u, i, q)$, which represents that user $u \in \mathcal{U}$ clicked item $i \in \mathcal{I}$ with the explicit query q. The query q can be segmented into several shorter terms as $q = [w_1, \cdots, w_{|q|}]$, where w_i denotes the i-th term and $|q|$ is the number of terms in query q.

Definition 2.***recommendation scenario***: In the recommendation data $\mathcal{X}_r$, each interaction record $x_r \in \mathcal{X}_r$ can be formulated as $x_r = (u, i)$, which represents user $u \in \mathcal{U}$ clicked item $i \in \mathcal{I}$ without an explicit query.

Thereby, the *double-scenario graph* including all user click behaviors in both scenarios can be constructed as follows:

Definition 3.***double-scenario graph***: Given the set of all user click behaviors in both scenarios, denoted as $\mathcal{X} = \mathcal{X}_s \cup \mathcal{X}_r$, the *double-scenario graph* can be formulated as $\mathcal{G} = (\mathcal{U} \cup \mathcal{I}, \mathcal{E}_s \cup \mathcal{E}_r)$. Each search edge $\epsilon \in \mathcal{E}_s$ corresponds to a record (u, i, q) in $\mathcal{X}_s$, while each recommendation edge $\epsilon \in \mathcal{E}_r$ corresponds to a record (u, i) in $\mathcal{X}_r$.

In Figure 2(a), there is an example of our *double-scenario graph*. For instance, user u_1 searches for query q_{12} and then clicks item i_2 in search scenario. Thus, an edge exists between nodes u_1 and i_2, with query q_{12} assigned as an attribute of this edge. Likewise, in recommendation scenario, when user u_1 clicks item i_1, an edge also exists between user u_1 and item i_1, but without any query attribute. Based on the above definitions, the joint modeling of search and recommendation can be defined as follows:

Problem definition. Given search data $\mathcal{X}_s$, recommendation data $\mathcal{X}_r$ and double-scenario graph $\mathcal{G}$, this task is to train a joint model of search and recommendation to predict the most appropriate items $i \in \mathcal{I}$ that user $u \in \mathcal{U}$ will interact.

4 METHODOLOGY

In this section, we introduce UDITSR for dual intent-aware joint modeling of search and recommendation, as depicted in Figure 2. We begin with the model's *embedding layer* in Section 4.1. Then, in Section 4.2, we detail a *search-supervised demand intent generator* that leverages search query data to infer recommendation intents, which allows us to convert the *double-scenario graph* into a *unified graph*. Utilizing this graph, we describe *dual-intent translation propagation* to couple inherent intents and demand intents, enhanced by a contrastive loss to constrain the translation relation. Finally, the prediction layer and optimization are illustrated in Section 4.4.

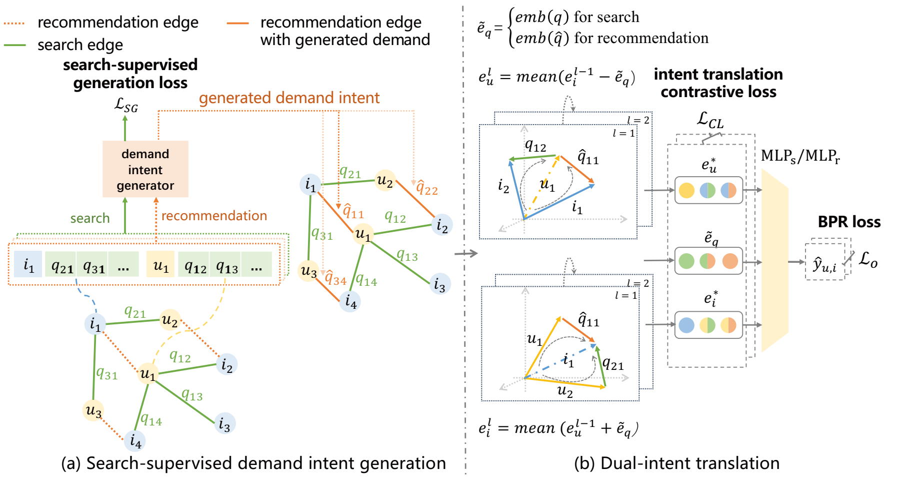


Figure 2: Overall framework of our proposed UDITSR. The *mean* in *dual-intent translation* represents the mean-pooling operation in Eq. 5. For clarity, only two interaction examples are displayed for each graph aggregation in the *dual-intent translation.*

4.1 Embedding Layer

By feeding the user ID and item ID into the user and item embedding matrices respectively, we can obtain the embeddings of user u and item i as $\mathbf{e}_u, \mathbf{e}_i$. Since each query q is a sequence of shorter terms as $[w_1, w_2, \cdots, w_{|q|}]$, we can obtain the representation of query q by combining the embeddings of its terms:

$$\mathbf{e}_q = f(\mathbf{e}_{w_1}, \mathbf{e}_{w_2}, \cdots, \mathbf{e}_{w_{|q|}}), \tag{1}$$

where $\mathbf{e}_{w_k}$ represents the embedding of the k-th query term in q and $f(\cdot)$ denotes a combination function. In this study, we choose the element-wise sum-pooling operation because it is both efficient and effective for this combination through empirical analysis.

4.2 Demand Intent Generation

4.2.1 Search-Supervised Demand Intent Generator. The notable difference between search and recommendation is that a user explicitly expresses demand intents in search, whereas recommendation lacks such explicit intents. To bridge this gap, we propose to utilize the abundant query information from search to supervise the generation of users' demand intents in recommendation. Below we describe the generator in detail.

Since the user's historical queries $q_u = [w_1^u, w_2^u, \cdots, w_{|q_u|}^u]$ and the item's historical queries $q_i = [w_1^i, w_2^i, \cdots, w_{|q_i|}^i]$ contain abundant demand intent information, we leverage them as auxiliary information to simulate the user's demand intent for recommendation. Similar to the processing of q in Eq. 1, we adopt the element-wise sum-pooling operation to obtain the representation of q_u as $\mathbf{e}_{q_u} = \sum_{k=1}^{|q_u|} \mathbf{e}_{w_k^u}$, where $\mathbf{e}_{w_k^u}$ is the embedding of the k-th query term in q_u. Since q_i contains query words from multiple users, we introduce a user-aware gate mechanism to model personalized demand intents. Particularly, the user-aware gating network g yields a distribution over the $|q_i|$ query words. The personalized representation of q_i is then formulated as the weighted sum of the embeddings of its query words, as follows:

$$\begin{aligned} \mathbf{K}_g &= \mathbf{W}_g(\mathbf{e}_u \| \mathbf{e}_{w_1^u} \| \cdots \| \mathbf{e}_{w_{|q_u|}^u}), \\ g(w_k^i) &= \frac{\exp(\mathbf{K}_g \times \mathbf{e}_{w_k^i}{}^\top)}{\sum_{k=1}^{|q_i|} \exp(\mathbf{K}_g \times \mathbf{e}_{w_k^i}{}^\top)}, \\ \mathbf{e}_{q_i} &= \sum_{k=1}^{|q_i|} g(w_k^i)\mathbf{e}_{w_k^i}, \end{aligned} \tag{2}$$

where $\cdot\|\cdot$ denotes the concatenation operation; $\mathbf{W}_g$ is used to match the dimensions of vector $\mathbf{e}_{w_k^i}$ and the concatenated vector. Then, with user-related representations $\mathbf{e}_u, \mathbf{e}_{q_u}$ and item-related representations $\mathbf{e}_i, \mathbf{e}_{q_i}$, the user's demand intent about the item can be estimated as follows:

$$\hat{\mathbf{e}}_q = \mathrm{MLP}(\mathbf{e}_u \| \mathbf{e}_i \| \mathbf{e}_{q_u} \| \mathbf{e}_{q_i}), \tag{3}$$

where MLP denotes a multi-layer perceptron. Since the ground truth queries in search data serve as the supervision information for generating demand intent, we design the generation loss as follows:

$$\mathcal{L}_{SG} = \sum_{(u,i,q)\in\mathcal{X}_s} (\mathbf{e}_q - \hat{\mathbf{e}}_q)^2. \tag{4}$$

4.2.2 Unified Graph. After generating the demand intents, each recommendation record (u, i) in $\mathcal{X}_r$ can be converted into a triplet $(u, i, \hat{q})$, where the embedding of $\hat{q}$ corresponds to the generated intents $\hat{\mathbf{e}}_q$. For simplicity, **we directly generate the representation of intent $\hat{\mathbf{e}}_q$ instead of indirectly predicting the specific query** $\hat{q}$. With the generated demand intents, the *double-scenario graph* can be converted into a *unified graph*. Specifically, an additional attribute $\hat{q}$ is attached to each recommendation edge (u, i) in $\mathcal{G}$. For brevity, we use $\widetilde{q}/\widetilde{\mathbf{e}}_q$ to uniformly represent the real $q/\mathbf{e}_q$ in search scenario and the generated $\hat{q}/\hat{\mathbf{e}}_q$ in recommendation scenario correspondingly. Based on the unified graph, we implement the unified modeling of recommendation and search below.

4.3 Dual-Intent Translation Propagation

To explicitly model the relation among the dual intents and the interactive items, we propose a dual-intent translation module inspired by the triplet-based representation learning in knowledge graphs [4]. Specifically, we use the user's embedding representation, which remains inherent for a single user, to represent their inherent intent. The search query representation and the generated demand intent in recommendation represent the user's demand intent. The representation of an interactive item is given by its embedding. We assume that a user's changing interactive item should be close to their inherent intent plus changing demand intent. Consequently, we aggregate the neighbor embeddings as follows:

$$\begin{aligned} \mathbf{e}_i^l &= mean\_pooling(\{\mathbf{e}_u^{l-1} + \widetilde{\mathbf{e}}_q, \forall u \in \mathcal{N}_i\}), \\ \mathbf{e}_u^l &= mean\_pooling(\{\mathbf{e}_i^{l-1} - \widetilde{\mathbf{e}}_q, \forall i \in \mathcal{N}_u\}), \end{aligned} \tag{5}$$

where $\mathcal{N}_u$ and $\mathcal{N}_i$ denote the neighboring nodes of user u and item i respectively, in the unified graph; $\mathbf{e}_u^0 = \mathbf{e}_u$ and $\mathbf{e}_i^0 = \mathbf{e}_i$. In particular, the subtraction aggregation operation, as opposed to the addition operation, for aggregating the embeddings of user neighboring nodes to simulate users' inherent intents. Finally, the weighted-pooling operation is applied to generate the aggregated representations by operating on the propagated L layers:

$$\mathbf{e}_i^* = \sum_{l=0}^{L} \alpha_l \mathbf{e}_i^l, \quad \mathbf{e}_u^* = \sum_{l=0}^{L} \alpha_l \mathbf{e}_u^l, \tag{6}$$

where α_l indicates the importance of the l-th layer representation in constituting the final embedding. Following LightGCN [14], we set α_l as $\frac{1}{(l+1)}$, as the focus of our work is not on its selection.

To further constrain the translation relation, we design an intent translation contrastive learning approach that adopts a margin-based ranking criterion. Specifically, we aim to ensure that $\mathbf{e}_u^* + \widetilde{\mathbf{e}}_q \approx \mathbf{e}_i^*$ (i.e., the ground truth interactive item $\mathbf{e}_i^*$ should be near to the translated intent $\mathbf{e}_u^* + \widetilde{\mathbf{e}}_q$), while the negative $\mathbf{e}_{i'}^*$ should be distant from $\mathbf{e}_u^* + \widetilde{\mathbf{e}}_q$, as follows:

$$\mathcal{L}_{CL} = \sum_{(u,i,i')\in Y} -\ln\sigma[(\mathbf{e}_u^* + \widetilde{\mathbf{e}}_q - \mathbf{e}_{i'}^*)^2 - (\mathbf{e}_u^* + \widetilde{\mathbf{e}}_q - \mathbf{e}_i^*)^2], \tag{7}$$

where $\widetilde{\mathbf{e}}_q$ denotes the representation of real query in search or the generated demand intent in recommendation for (u, i) pair; $Y = \{(u, i, i')|(u, i) \in R^+, (u, i') \in R^-\}$ denotes the pairwise training data where R^+ indicates the positive observed interaction set, and R^- represents the randomly-sampled negative set; $\sigma(\cdot)$ stands for the sigmoid function.

4.4 Model Prediction and Optimization

After obtaining the representations $\mathbf{e}_u^*, \mathbf{e}_i^*, \widetilde{\mathbf{e}}_q$, we fuse them to obtain the overall representation for the input sample $x = (u, i, \widetilde{q})$:

$$\mathbf{e}_{u,i,\widetilde{q}} = \mathbf{e}_u^* \| \mathbf{e}_i^* \| \widetilde{\mathbf{e}}_q. \tag{8}$$

Then, two different MLPs are employed to make prediction for search and recommendation tasks, respectively:

$$\hat{y}_{u,i,\widetilde{q}} = \begin{cases} \mathrm{MLP_s}(\mathbf{e}_{u,i,\widetilde{q}}) & \text{if} \quad x \in \mathcal{X}_s, \\ \mathrm{MLP_r}(\mathbf{e}_{u,i,\widetilde{q}}) & \text{if} \quad x \in \mathcal{X}_r. \end{cases} \tag{9}$$

We adopt pairwise training to train the model. Specifically, we adopt the Bayesian Personalized Ranking (BPR) [30] loss to emphasize that the observed interaction should be assigned a higher score than the unobserved one as follows:

$$\mathcal{L}_o = \sum_{(u,i,i')\in Y} -\ln\sigma(\hat{y}_{u,i,\widetilde{q}} - \hat{y}_{u,i',\widetilde{q}'}), \tag{10}$$

where the representation of $\widetilde{q}'$ denotes the demand intent for the negative pair (u, i'). Finally, the overall loss $\mathcal{L}$ is defined using hyper-parameters λ_1 and λ_2 as:

$$\mathcal{L} = \mathcal{L}_o + \lambda_1 \mathcal{L}_{SG} + \lambda_2 \mathcal{L}_{CL}. \tag{11}$$

5 EXPERIMENTS

In this section, we present empirical results to demonstrate the effectiveness of our proposed UDITSR. These experiments are designed to answer the following research questions: **RQ1** How does UDITSR perform compared with state-of-the-art search and recommendation models? **RQ2** What are the effects of the demand intent generator and dual-intent translation mechanism in UDITSR? **RQ3** Why could UDITSR perform better? **RQ4** How does UDITSR perform in real-world online recommendations with practical metrics? **RQ5** How do the hyper-parameters in UDITSR impact the search and recommendation performance?

5.1 Experimental Settings

5.1.1 Dataset Description. We conducted experiments on two real-world datasets, denoted as MT-Large and MT-Small datasets[1]. These two datasets are obtained from the Meituan platform, one of the largest takeaway platforms in China. Both datasets span eight days across two cities. Each sample in the datasets contains a user and an item, and each search sample additionally contains a query. Specifically, with 111,891 search and 65,035 recommendation interactions collected, our **MT-Small** dataset comprises 56,887 users and 4,059 items and the average number of split words per query record is 1.6801. With 1,527,869 search and 1,168,491 recommendation interactions collected, the **MT-Large** dataset contains 433,573 users and 22,967 items and the average number of split words per query is 1.5561. To evaluate model performance, we split the first six days' data for training, the seventh day's data for validation, and the

[1]We collected this dataset because there was no public dataset that includes both search and recommendation data. Our code and data will be available at https://github.com/17231087/UDITSR.

last day's data for testing. For each ground truth test record, we randomly sampled 99 items that the user did not interact with as negative samples.

Table 1: Network Configuration

Name	Value
optimizer	AdamW
batch size	256
learning rate	1e-4
weight decay	1e-5
vocab size of words in querys	5,000
dimension of embeddings	100
depth of aggregation	2
number of words per query	3
number of words per user's historical query	3
number of words per item's historical query	10
hidden sizes of MLP in demand intent generator	[200,100]
hidden sizes of $\mathrm{MLP_s}/\mathrm{MLP_r}$	[150,75]

5.1.2 Implementation Details. We implement all models using PyTorch[2], a well-known software library for deep learning. In Section 5.6, we report the impact of essential hyper-parameters in our model, including the loss weights λ_1 and λ_2, and we utilize the best settings for these hyper-parameters. The remaining network configurations are presented in Table 1. To ensure a fair comparison, we apply the above-mentioned settings across all models. Moreover, we search for optimal values of the other hyper-parameters of the baseline models as suggested in their respective original papers. Finally, we employ the early stopping strategy based on the models' performance on the validation set to avoid overfitting.

5.1.3 Evaluation Metrics. To evaluate our model's performance, we utilize four widely-used ranking metrics: Hit@K, NDCG@K [17] (we set K as 5 by default), MRR [28] and Average position of the Clicked items (Avg.C) [43]. Additionally, we adopt an accuracy metric, AUC [12] for the recommendation task.

5.1.4 Baselines. In our work, we evaluate the performance of our model with two groups of baselines to examine its effectiveness.
(1) Graph-free baselines
- **NeuMF** [15] combines traditional matrix decomposition with the MLP to extract low-dimensional and high-dimensional features simultaneously.
- **DNN** combines the embedding layer described in Section 4.1 with the prediction layer described in Section 4.4.
- **xDeepFM** [21] consists of a compressed interaction network (CIN) and an MLP for prediction, where CIN generates explicit feature interactions at the vector-wise level.
- **DIN** [49] utilizes an attention mechanism between the historical behavior sequence and the target item to model the evolving interests.
- **AEM** [1] allocates different attention values to the previous behavior sequence based on the current search queries.
- **TEM** [3] feeds the sequence of query and user behavior history into a transformer layer to extract the search intents.
- **JSR** [45] integrates neural collaborative filtering and language modeling to reconstruct query text descriptions, enabling the joint model of search and recommendation.
- **SimpleX** [24] is a simplified variant of the two-tower model with user behavior modeling.
- **MGDSPR** [20] utilizes an attention mechanism to model the relationship between users' query multi-grained semantics and their personalized behaviors for prediction.

(2) Graph-based baselines
- **GAT** [36] utilizes the attention mechanism to measure the importance of neighbor nodes during the aggregation process.
- **NGCF** [39] enhances the Graph Convolutional Networks (GCN) by incorporating user-item interactions.
- **LightGCN** [14] streamlines GCN by relying solely on neighborhood aggregation to capture collaborative filtering, omitting feature transformation and non-linear activation components.
- **GraphSRRL** [22] exploits three specific structural patterns within a user-query-item graph.
- **SRJGraph** [47] incorporates padding queries for recommendation and search queries as attributes into interaction edges, enabling joint modeling of both tasks.
- **DCCF** [29] leverages an adaptive self-supervised augmentation to disentangle intents behind user-item interactions.

Specifically, NeuMF, xDeepFM, DIN, DCCF, SimpleX, NGCF and LightGCN are proposed for the **recommendation task**, while AEM, TEM, MGDSPR and GraphSRRL are proposed for the **search task**. JSR and SRJGraph are designed for **joint learning of both tasks**. To adapt these baselines for both tasks, **real query representations for search and padding query representations for recommendation are incorporated into the prediction layer** described in Section 4.4. Previous studies [44, 47] have demonstrated that joint optimization of search and recommendation models can improve performance, so all baselines are directly trained on both search and recommendation data. All baselines use the same settings for the embedding layer and the prediction layer, and the interaction graph is built on both search and recommendation interactions.

5.2 Overall Performance Comparison (RQ1)

We present the results on the two adopted datasets in Table 2. From the results, we can observe that:
- UDITSR significantly outperforms all the competitive baselines on both tasks. Specifically, compared to the best-performing baselines, UDITSR gains an average improvement of 6.22% and 3.06% in the search and recommendation tasks, respectively.
- Most graph-based methods, such as NGCF, LightGCN, and GraphSRRL, perform well in both tasks, potentially due to their ability to effectively capture complex high-order interactive patterns.
- SRJgraph assumes that query-related intents in recommendation remain unchanging whereas in search, the matching degree between the query and the candidate items is deemed crucial.

[2]https://pytorch.org/

Dataset	Model	Search				Recommendation			
		Hit@5	NDCG@5	MRR	Avg.C↓	Hit@5	NDCG@5	MRR	AUC
MT-Small	NeuMF	0.5510	0.4264	0.4150	12.1907	0.3147	0.2306	0.2374	0.8160
	DNN	0.5877	0.4594	0.4465	9.6208	0.3241	0.2177	0.2246	0.8150
	xDeepFM	0.5053	0.3886	0.3815	14.3603	0.3184	0.2139	0.2218	0.8155
	DIN	0.5892	0.4726	0.4613	11.6023	0.3632	0.2510	0.2545	0.8213
	AEM	0.5053	0.3666	0.3568	11.7953	0.3967	0.2703	0.2686	0.7982
	TEM	0.5362	0.4185	0.4084	13.6472	0.2933	0.1970	0.2078	0.7947
	JSR	0.6143	0.4828	0.4678	8.6276	0.3460	0.2448	0.2457	0.7532
	SimpleX	0.6237	0.4864	0.4699	8.0841	0.3314	0.2288	0.2336	0.8081
	MGDSPR	0.6150	0.4743	0.4570	8.5362	0.2974	0.2032	0.2122	0.7862
	GAT	0.6025	0.4679	0.4497	10.8707	0.4202	0.3109	0.3032	0.7935
	NGCF	0.6418	0.5173	0.5000	9.7943	0.4564	0.3346	0.3284	0.8232
	LightGCN	0.6665	0.5402	0.5195	9.9139	0.4577	0.3296	0.3185	0.8174
	GraphSRRL	0.6688	0.5267	0.5042	8.0724	0.4540	0.3249	0.3159	0.7883
	SRJgraph	0.6186	0.4850	0.4647	12.1989	0.4140	0.3074	0.2997	0.7474
	DCCF	0.5013	0.3760	0.3615	19.6477	0.4323	0.3380	0.3304	0.7239
	UDITSR	**0.7008***	**0.5691***	**0.5470***	**7.5257***	**0.4841***	**0.3528***	**0.3422***	**0.8285***
	Impr.%	**4.7847**	**5.3499**	**5.2936**	**6.7725**	**5.7680**	**4.3787**	**3.5714**	**0.6438**
MT-Large	NeuMF	0.8668	0.7855	0.7682	3.3001	0.5390	0.4235	0.4129	0.8573
	DNN	0.8788	0.7874	0.7664	3.0520	0.5153	0.3962	0.3881	0.8610
	xDeepFM	0.8552	0.7417	0.7147	4.0106	0.4926	0.3897	0.3828	0.8077
	DIN	0.8914	0.7934	0.7693	2.7283	0.6005	0.4489	0.4292	0.9082
	AEM	0.8760	0.7654	0.7389	2.9007	0.5865	0.4597	0.4435	0.8940
	TEM	0.8611	0.7522	0.7269	3.4096	0.5031	0.3526	0.3419	0.8899
	JSR	0.8691	0.7748	0.7537	3.0903	0.5023	0.3789	0.3683	0.8393
	SimpleX	0.8896	0.7895	0.7651	2.6466	0.5004	0.3790	0.3691	0.8640
	MGDSPR	0.8726	0.7709	0.7473	3.0325	0.5412	0.4037	0.3888	0.8751
	GAT	0.8761	0.7796	0.7572	2.9530	0.5880	0.4540	0.4347	0.8706
	NGCF	0.8821	0.7892	0.7670	2.7377	0.6325	0.4966	0.4780	0.9096
	LightGCN	0.8937	0.8016	0.7795	2.4076	0.6158	0.4785	0.4593	0.8920
	GraphSRRL	0.8966	0.7992	0.7755	2.3504	0.6106	0.4726	0.4543	0.8891
	SRJgraph	0.8836	0.7883	0.7659	2.6849	0.5873	0.4494	0.4315	0.8942
	DCCF	0.8568	0.7459	0.7205	3.2666	0.6201	0.5007	0.4802	0.8292
	UDITSR	**0.9178***	**0.8382***	**0.8183***	**1.9819***	**0.6566***	**0.5157***	**0.4936***	**0.9146***
	Impr.%	**2.3645**	**4.5659**	**4.9775**	**15.6782**	**3.8103**	**2.9958**	**2.7905**	**0.5497**

Table 2: Overall performance on both datasets. ↓ represents that a smaller Avg.C metric value indicates better performance. *Impr.%* indicates the relative improvements of the best-performing method (bolded) over the strongest baselines (underlined). * indicates 0.05 significance level from a paired t-test comparing UDITSR with the best baselines.

Consequently, such an assumption may limit the model's performance, particularly when compared to our UDITSR, which learns and adapts to changing query-related intents.

5.3 Ablation Study (RQ2)

As the demand intent generator and dual-intent translation propagation are the core of our model, we conduct the following ablation studies to investigate their effectiveness:

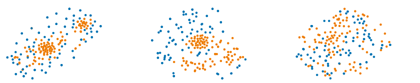
- UDITSR(w/o DeIntGen) masks all generated demand intents $\widetilde{\mathbf{e}}_q$ by assigning the embedding of the padding query to each recommended record.
- UDITSR(w/o IntTrans) replaces dual-intent translation with classical mean-pooling propagation between the user and item nodes.
- UDITSR(w/o DeIntGen & IntTrans) removes both the demand intent generator and dual-intent translation propagation, as described in the two ablation studies above.

From the results of ablation studies in Table 3, we can find that:

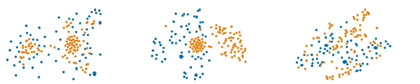
- UDITSR(w/o DeIntGen & IntTrans) performs the worst on both search and recommendation tasks, suggesting that the significant improvement of our model stems from our proposed demand intent generator and dual-intent translation propagation.
- UDITSR(w/o IntTrans) performs worse than the original UDITSR, highlighting the effectiveness of our proposed intent translation propagation mechanism.
- UDITSR(w/o DeIntGen) performs worse than UDITSR, especially for recommendation task, indicating that the search-supervised

Dataset	Ablation	Search				Recommendation			
		Hit@5	NDCG@5	MRR	Avg.C↓	Hit@5	NDCG@5	MRR	AUC
MT-Small	UDITSR(w/o DeIntGen & IntTrans)	0.6479	0.5152	0.4949	10.1960	0.4185	0.3052	0.3032	0.8179
	UDITSR(w/o IntTrans)	0.6454	0.5130	0.4924	10.5527	0.4352	0.3206	0.3154	0.8225
	UDITSR(w/o DeIntGen)	0.6959	0.5543	0.5307	8.1389	0.4510	0.3237	0.3178	0.8186
	UDITSR	**0.7008***	**0.5691***	**0.5470***	**7.5257***	**0.4841***	**0.3528***	**0.3422***	**0.8285***
MT-Large	UDITSR(w/o DeIntGen & IntTrans)	0.8660	0.7586	0.7337	3.1185	0.6183	0.4818	0.4623	0.9031
	UDITSR(w/o IntTrans)	0.8870	0.7866	0.7623	2.6399	0.6228	0.4879	0.4696	0.9061
	UDITSR(w/o DeIntGen)	0.9089	0.8192	0.7969	2.2053	0.6303	0.4890	0.4685	0.9058
	UDITSR	**0.9178***	**0.8382***	**0.8183***	**1.9819***	**0.6566***	**0.5157***	**0.4936***	**0.9146***

Table 3: Ablation study on our proposed search-supervised demand intent generator and dual-intent translation propagation.



(a) Intents in UDITSR(w/o IntTrans) for search data (b) Inherent intents in UDITSR for search data (c) Translated intents in UDITSR for search data



(d) Intents in UDITSR(w/o IntTrans) for recommendation data (e) Inherent intents in UDITSR for recommendation data (f) Translated intents in UDITSR for recommendation data

Figure 3: t-SNE visualization of learned intents and interactive items. Blue dots represent the interactive items(i.e., $\mathbf{e}_i^*$) and orange dots represent the learned intents.

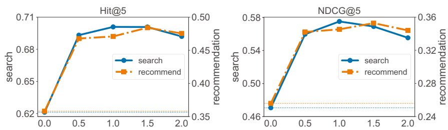


Figure 4: Performance w.r.t λ_1 of search-supervised demand intent generator for search and recommendation tasks.

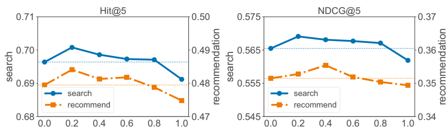


Figure 5: Performance w.r.t λ_2 of the intent translation contrastive learning for search and recommendation tasks.

demand intent generator can help UDITSR learn implicit intents more accurately in recommendation.

5.4 Intent Visualization (RQ3)

In this section, we visualize the learned intents to further investigate why our model performs better. We compare UDITSR with its ablated version without the dual-intent translation propagation (UDITSR(w/o IntTrans), detailed in Section 5.3). We employ the default setting of the t-SNE [9] provided by Scikit-learn to visualize the distribution of the learned intents and the interactive items. For clarity, we randomly sample 100 positive records from the search and recommendation test datasets respectively for plotting. Specifically, in UDITSR(w/o IntTrans), user embeddings (i.e.,$\mathbf{e}_u^*$) are regarded as the learned intents, as shown in Figures 3(a) and (d), similar to preference/intent captured by models like NGCF and LightGCN. UDITSR, however, couples inherent and demand intents via intent translation to form the final intents (i.e., $\mathbf{e}_u^*+\widetilde{\mathbf{e}}_q$), as shown in Figure 3(c) and (f). To ensure a fair comparison, we present the inherent intents (i.e., $\mathbf{e}_u^*$) learned by UDITSR in Figure 3(b) and (e).

Ideally, the distribution of learned intents should match that of interactive item representations. Figures 3(a) and (d) reveal that the intents learned by UDITSR(w/o IntTrans) are concentrated while the positive interactive items are scattered, indicating a mismatch. Meanwhile, the inherent intents learned by UDITSR are relatively scattered, indicating that our model can better learn the personalized inherent intents of different users. However, there still exist obvious gaps between the intents and items, highlighting the necessity of learning demand intents. In contrast, the translated intents learned by UDITSR are scattered in the space of the target interactive items, demonstrating its excellent intent modeling capability. The better fit of the distribution of translated intents to the target interactive distribution could be the fundamental reason for the better overall performance of UDITSR.

5.5 Online A/B test (RQ4)

Owing to the distinct architectural differences between the search and recommender systems on the Meituan Waimai platform, we have initially focused our methodological deployment on the homepage recommender systems. We conducted a month-long online A/B test from December 18, 2023, to January 17, 2024. Specifically, we utilized the search data with query information to guide the learning of user demand intent representation and leveraged the learned graph embeddings as additional features in the downstream recommendation model. The control bucket was the original online recommendation method of Meituan Waimai platform. The deployment of our method increased the **GMV**(Gross Merchandise Volume) by **1.46%** and **CTR**(Click-Through Rate) by **0.77%**, which demonstrated the effectiveness of our method. In the future, we will continue to conduct comprehensive online experiments that encompass both search and recommendation scenarios.

5.6 Hyper-Parameter Studies (RQ5)

In this section, we conduct experiments on the loss weights (λ_1, λ_2) in Eq. 11 on MT-Small dataset to explore their impact.

(1) Loss weight of the demand intent generator (λ_1). We vary λ_1 within $\{0, 0.5, 1.0, 1.5, 2.0\}$. The results in Figure 4 indicate that performance improves and then declines with increasing λ_1. With $\lambda_1 = 0$, the demand intent generator degenerates to an ordinary generator without any search-supervision information. All models with search supervision (i.e. $\lambda_1 \neq 0$) outperform models without it (i.e. $\lambda_1 = 0$). This may stem from **UDITSR's effective learning of user demand intents through explicit supervision from search**. Furthermore, our model excels across most metrics for both search and recommendation tasks at $\lambda_1 = 1.5$. Thus, we set $\lambda_1 = 1.5$ for MT-Small dataset. After a similar experiment conducted on MT-Large dataset, we adopt the best-performing setting ($\lambda_1 = 1$).

(2) Loss weight of the intent translation contrastive learning (λ_2). To investigate the impact of our proposed intent translation contrastive learning, we vary λ_2 in $\{0.0, 0.2, 0.4, 0.6, 0.8, 1.0\}$. Overall, the performance initially increases and then decreases with the increase of λ_2. Particularly, our model with λ_2 set in {0.2, 0.4, 0.6} outperforms the version without translation contrastive learning $\lambda_2 = 0$ on all metrics, **demonstrating a proper loss weight of intent translation contrastive learning can aid in intent relation modeling**. The optimal λ_2 for search is 0.2 while for recommendation task, it is 0.2 for Hit@5 and 0.4 for NDCG@5. Therefore, we set $\lambda_2 = 0.2$ for MT-Small dataset. Also, after conducting a similar experiment on MT-Large dataset, we adopt the best-performing setting $\lambda_2 = 0.4$.

6 CONCLUSION

This paper introduced a novel approach to unified intention-aware modeling for joint optimization of search and recommendation tasks. We recognized that user behaviors were motivated by their inherent intents and changing demand intents. To accurately learn users' implicit demand intents for recommendation, we innovated a demand intent generator that utilized explicit queries from search data for supervised learning. Furthermore, we proposed a dual-intent translation propagation mechanism for interpretive modeling of the relation between users' dual intents and their interactive items. In particular, we introduced an intent translation contrastive method to further constrain this relation. Our extensive offline experiments demonstrated that UDITSR outperformed the leading baselines in both search and recommendation tasks. Besides, online A/B tests further confirmed the superiority of our model. Finally, the intent visualization clearly explained the deeper reason for the remarkable improvement of our model.

ACKNOWLEDGMENTS

This research work is supported by the National Key Research and Development Program of China under Grant No.2021ZD0113602, the National Natural Science Foundation of China under Grant No.62176014 and No.62306255, the Fundamental Research Funds for the Central Universities and the Fundamental Research Project of Guangzhou under Grant No. 2024A04J4233.

REFERENCES

[1] Qingyao Ai, Daniel N Hill, SVN Vishwanathan, and W Bruce Croft. 2019. A zero attention model for personalized product search. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 379–388.
[2] Nicholas J Belkin and W Bruce Croft. 1992. Information filtering and information retrieval: Two sides of the same coin? *Commun. ACM* 35, 12 (1992), 29–38.
[3] Keping Bi, Qingyao Ai, and W Bruce Croft. 2020. A transformer-based embedding model for personalized product search. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*. 1521–1524.
[4] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems* 26 (2013).
[5] Tong Chen, Hongzhi Yin, Hongxu Chen, Rui Yan, Quoc Viet Hung Nguyen, and Xue Li. 2019. Air: Attentional intention-aware recommender systems. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*. IEEE, 304–315.
[6] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.
[7] Paul Covington, Jay Adams, and Emre Sargin. 2016. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*. 191–198.
[8] Tyler Derr, Yao Ma, and Jiliang Tang. 2018. Signed graph convolutional networks. In *2018 IEEE International Conference on Data Mining (ICDM)*. IEEE, 929–934.
[9] Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*. PMLR, 647–655.
[10] Lu Fan, Qimai Li, Bo Liu, Xiao-Ming Wu, Xiaotong Zhang, Fuyu Lv, Guli Lin, Sen Li, Taiwei Jin, and Keping Yang. 2022. Modeling user behavior with graph convolution for personalized product search. In *Proceedings of the ACM Web Conference 2022*. 203–212.
[11] Yufei Feng, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. 2019. Deep session interest network for click-through rate prediction. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. 2301–2307.
[12] Cesar Ferri, José Hernández-Orallo, and Peter A Flach. 2011. A coherent interpretation of AUC as a measure of aggregated classification performance. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. 657–664.
[13] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. 1725–1731.
[14] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*. 639–648.
[15] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*. 173–182.
[16] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*. Ieee, 263–272.
[17] Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)* 20, 4 (2002), 422–446.
[18] Thomas N Kipf and Max Welling. 2016. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations*.

[19] Lihong Li, Wei Chu, John Langford, and Robert E Schapire. 2010. A contextual-bandit approach to personalized news article recommendation. In *Proceedings of the 19th international conference on World wide web*. 661–670.

[20] Sen Li, Fuyu Lv, Taiwei Jin, Guli Lin, Keping Yang, Xiaoyi Zeng, Xiao-Ming Wu, and Qianli Ma. 2021. Embedding-based product retrieval in taobao search. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*. 3181–3189.

[21] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xdeepfm: Combining explicit and implicit feature interactions for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1754–1763.

[22] Shang Liu, Wanli Gu, Gao Cong, and Fuzheng Zhang. 2020. Structural relationship representation learning with graph embedding for personalized product search. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 915–924.

[23] Zhiwei Liu, Xiaohan Li, Ziwei Fan, Stephen Guo, Kannan Achan, and S Yu Philip. 2020. Basket recommendation with multi-intent translation graph neural network. In *2020 IEEE International Conference on Big Data (Big Data)*. IEEE, 728–737.

[24] Kelong Mao, Jieming Zhu, Jinpeng Wang, Quanyu Dai, Zhenhua Dong, Xi Xiao, and Xiuqiang He. 2021. SimpleX: A simple and strong baseline for collaborative filtering. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 1243–1252.

[25] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. 2020. A dual heterogeneous graph attention network to improve long-tail performance for shop search in e-commerce. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 3405–3415.

[26] Qi Pi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. 2020. Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 2685–2692.

[27] Chuan Qin, Le Zhang, Rui Zha, Dazhong Shen, Qi Zhang, Ying Sun, Chen Zhu, Hengshu Zhu, and Hui Xiong. 2023. A comprehensive survey of artificial intelligence techniques for talent analytics. *arXiv preprint arXiv:2307.03195* (2023).

[28] Dragomir R Radev, Hong Qi, Harris Wu, and Weiguo Fan. 2002. Evaluating Web-based Question Answering Systems.. In *LREC*. Citeseer.

[29] Xubin Ren, Lianghao Xia, Jiashu Zhao, Dawei Yin, and Chao Huang. 2023. Disentangled Contrastive Collaborative Filtering. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval* (, Taipei, Taiwan,) *(SIGIR '23)*. Association for Computing Machinery, New York, NY, USA, 1137–1146. https://doi.org/10.1145/3539618.3591665

[30] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. 452–461.

[31] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*. 285–295.

[32] Franco Scarselli, Marco Gori, Ah Chung Tsoi, Markus Hagenbuchner, and Gabriele Monfardini. 2008. The graph neural network model. *IEEE transactions on neural networks* 20, 1 (2008), 61–80.

[33] Parikshit Sondhi, Mohit Sharma, Pranam Kolari, and ChengXiang Zhai. 2018. A taxonomy of queries for e-commerce search. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1245–1248.

[34] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM international conference on information and knowledge management*. 1441–1450.

[35] Chang-You Tai, Liang-Ying Huang, Chien-Kun Huang, and Lun-Wei Ku. 2021. User-centric path reasoning towards explainable recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 879–889.

[36] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[37] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2019. Modeling multi-purpose sessions for next-item recommendations via mixture-channel purpose routing networks. In *International Joint Conference on Artificial Intelligence*. International Joint Conferences on Artificial Intelligence.

[38] Shoujin Wang, Liang Hu, Yan Wang, Quan Z Sheng, Mehmet Orgun, and Longbing Cao. 2020. Intention2basket: A neural intention-driven approach for dynamic next-basket planning. In *Twenty-Ninth International Joint Conference on Artificial Intelligence and Seventeenth Pacific Rim International Conference on Artificial Intelligence {IJCAI-PRICAI-20}*. International Joint Conferences on Artificial Intelligence Organization.

[39] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. 2019. Neural graph collaborative filtering. In *Proceedings of the 42nd international ACM SIGIR conference on Research and development in Information Retrieval*. 165–174.

[40] Shiwen Wu, Fei Sun, Wentao Zhang, Xu Xie, and Bin Cui. 2022. Graph neural networks in recommender systems: a survey. *Comput. Surveys* 55, 5 (2022), 1–37.

[41] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems* 32, 1 (2020), 4–24.

[42] Sijie Yan, Yuanjun Xiong, and Dahua Lin. 2018. Spatial temporal graph convolutional networks for skeleton-based action recognition. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 32.

[43] Jing Yao, Zhicheng Dou, Ruobing Xie, Yanxiong Lu, Zhiping Wang, and Ji-Rong Wen. 2021. USER: A unified information search and recommendation model based on integrated behavior sequence. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*. 2373–2382.

[44] Hamed Zamani and W Bruce Croft. 2018. Joint modeling and optimization of search and recommendation. *arXiv preprint arXiv:1807.05631* (2018).

[45] Hamed Zamani and W Bruce Croft. 2020. Learning a joint search and recommendation model from user-item interactions. In *Proceedings of the 13th international conference on web search and data mining*. 717–725.

[46] Chuxu Zhang, Dongjin Song, Chao Huang, Ananthram Swami, and Nitesh V Chawla. 2019. Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*. 793–803.

[47] Kai Zhao, Yukun Zheng, Tao Zhuang, Xiang Li, and Xiaoyi Zeng. 2022. Joint learning of e-commerce search and recommendation with a unified graph neural network. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*. 1461–1469.

[48] Guorui Zhou, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. 2019. Deep interest evolution network for click-through rate prediction. In *Proceedings of the AAAI conference on artificial intelligence*, Vol. 33. 5941–5948.

[49] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1059–1068.

[50] Nengjun Zhu, Jian Cao, Yanchi Liu, Yang Yang, Haochao Ying, and Hui Xiong. 2020. Sequential modeling of hierarchical user intention and preference for next-item recommendation. In *Proceedings of the 13th international conference on web search and data mining*. 807–815.