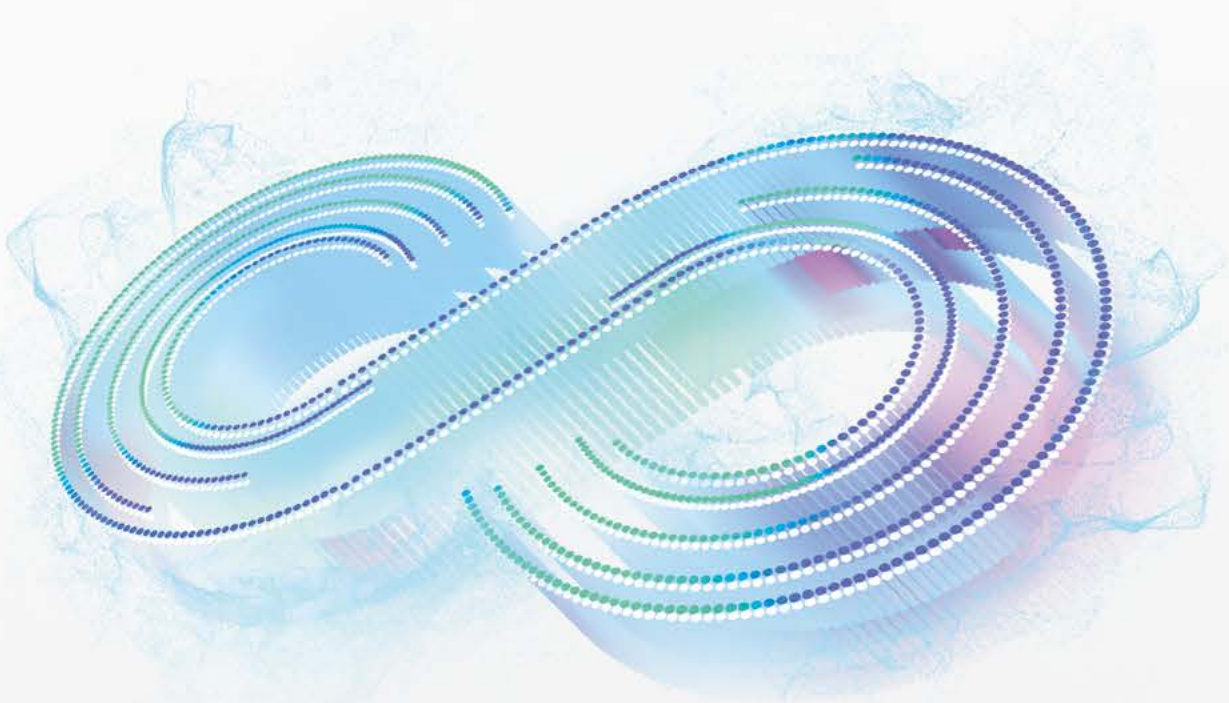
华为研究

Communications of HUAWEI RESEARCH

内部资料 免费交流
准印证号：（粤BL）022060040
2022年12月
第 3 期（总第3期）



HUAWEI

软件吞噬一切
开源吞噬软件

华为研究
内部资料，免费交流
准印证号：（粤BL）022060040

主编：
廖恒

本期责任主编：
陈海波，陆品燕

编委会：
廖恒，童文，肖新华，胡邦红，周慧慧，鲍丰，Jeff Xu，
陈海波，陆品燕，张小俊，李瑞华，白博

索阅、投稿、建议和意见反馈，请联系：
HWResearch@huawei.com

印刷数量：4000本
印刷单位：雅昌文化（集团）有限公司
印刷地址：深圳市南山区深云路19号
印刷日期：2022年12月16日

编者按

欢迎大家来到《华为研究》软件与理论分册！软件正在重塑数字世界与物理世界，成为数字世界的核心基础设施。如何构建软件基础能力并突破软件核心技术，是现代企业乃至国家拥抱数字世界的核心抓手。

与物理世界的对象不同，软件是一种“看不见摸不着”的逻辑实体，可能也是人类迄今为止所设计的最复杂的系统，一个大型软件系统的代码往往可达数亿行。软件的发展有其客观规律，不存在一劳永逸的“银弹”。软件包含基础软件、应用软件与工具软件等不同类型，面向智能终端、嵌入式设备、云与企业 IT 等诸多产业，它们的发展规律各具特色，生态现状与需求也存在差异。软件的生命周期与供应链管理具有自身的独特性，支撑软件不断发展的软件理论也在不断演进。研究与掌握这些规律和理论，是我们研发出好软件的关键。

华为公司长期以来在软件与理论领域持续投入，不仅面向终端、联接、计算、云与智能驾驶等多产业构筑了良好的竞争力，通过可信及软件工程变革不断提升软件的工程与可信能力，还构建了欧拉和鸿蒙两大操作系统生态，从而推动了 openEuler、OpenGuass、MindSpore、OpenHarmony 等基础软件开源社区的建设。

本分册汇聚了华为公司在基础软件、软件理论以及开源社区构建等领域的专家观点与新近成果。

在基础软件领域，《操作系统演进：历史、现在与展望》从产业应用演进与硬件演进两个维度来分析操作系统发展轨迹、创新机遇与技术挑战，并介绍 openEuler 和 OpenHarmony 在技术演进方面的实践；《GaussDB：云原生分布式数据库》介绍了中央软件院高斯部开发的企业级分布式数据库平台 GaussDB Kernel，以及该平台的架构和主打特性；《MindSpore：人工智能开源生态系统实践》介绍了深度学习框架 MindSpore 的主要架构与核心技术，以及在人工智能开源生态系统方面的实践探索；《华为毕昇编译器的创新与实践》在总结分析编译器技术演进的基础上，系统性地介绍了华为毕昇编译器的技术创新与实践，分析核心技术特性，并阐述了进一步演进和创新的技术方向。

在软件理论领域，《在线匹配领域的最新进展》介绍了在线匹配技术的最新学术进展；《面向组合优化的学习增强算法设计》介绍了在组合优化中运用 AI 技术的新近学术成果；求解器是一类非常重要的工具软件，也是工业软件的核心基础，《随机游走与适应度函数相结合求解布尔可满足性问题》介绍了随机游走技术在 SAT 求解器算法中的重大突破；域理论是编程语言及其分析工具最早的数学理论基础，《域理论与交互式计算》是一篇综述文章，详细介绍了域理论自上世纪六十年代以来的发展历史，以及在编程模型不断演进的过程中，如何被用于刻画交互、并发等行为。

“软件吞噬一切，开源吞噬软件”，《开源策略的制定与社区构建的度量》从企业开源策略的制定和企业开源社区的构建等方面阐述了企业如何利用开源发展生态经济，并推动产业数字化转型。

本分册还有很多优秀文章，内容面向软件领域的关键挑战，力求贴近产业热点，希望对读者有所启发。不当之处，也请多多批评指正！

陈海波
华为基础软件首席科学家

陆品燕
华为理论计算机首席科学家

基础软件

目录

软件理论

软件技术与生态

操作系统演进：历史、现在与展望

陈海波 [1]，钱栈杨 [1]，贾宁 [1]，胡欣蔚 [1]，李毅 [2]

[1] 中央软件院

[2] 终端 BG 软件部

摘要

操作系统承上启下，向上服务应用，向下管理与挖潜硬件能力，是构建硬件生态与应用生态的关键。本文分别从产业与应用演进和硬件演进两个维度来分析其发展轨迹、创新机遇与技术挑战，并介绍 openEuler 和 OpenHarmony 的实践。

关键词

操作系统，openEuler，OpenHarmony

1 引言

按照《计算机科学技术百科全书（第三版）》[1]的定义，操作系统是“管理硬件资源、控制程序运行、改善人机界面和为应用软件提供支持的一种系统软件”。操作系统自诞生以来，其内涵与外延一直在不断扩大，早期的操作系统只包含操作系统内核与原始的 Shell（如命令行终端），现代操作系统的功能不断扩大，包含了扩展操作系统内核管理与抽象硬件资源功能的操作系统服务以及为应用创建执行环境与管理应用执行的应用框架（如图 1）。

操作系统在整个计算系统中发挥“承上启下”的关键作用。承上就是为应用提供运行时和开发环境服务，并在一些场景下还作为应用生态与云服务的入口；启下则是高效、安全地管理硬件，最大程度地发挥硬件潜力，并使能硬件生态。

本文将分别从产业与应用演进和硬件演进两个维度来观察和思考其发展轨迹，并从应用场景驱动与硬件驱动两个维度来展望操作系统发展方向。最后，本文将介绍 openEuler 和 OpenHarmony 的创新实践。

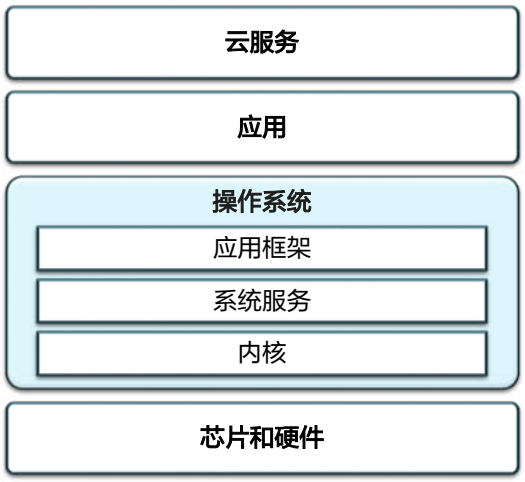


图 1 操作系统在整个计算系统中的定位

2 产业演进中的操作系统

2.1 操作系统伴随产业浪潮诞生与发展

回顾操作系统的发展史（见图 2），在业界取得成功的操作系统通常都是伴随着产业浪潮诞生与发展的，并与产业互相促进。

在早期的计算机中软件硬件深度耦合，从而使得应用程序开发的专业性高、开发效率低，对更方便地使用计算机并进行应用开发的需求直接推动了编程语言和操作系统的诞生，也使得软硬件持续解耦。

1956 年诞生的 GM-NAA I/O [2] 系统可以被称为操作系统的雏形，主要提供了输入和输出的管理系统。直到 20 世纪 60 年代，才形成了真正意义上的操作系统。在此之前，大型机的软件与硬件深度耦合，软件往往需要等待硬件研制后数年才得以研发成功上市，从而影响整体系统的上市节奏。1964 年，IBM 发布的 OS/360 [3] 首次推动了软件与硬件解耦；并且，这一操作系统的复杂研制过程还催生了“软件工程”学科的开创性书籍《人月神话：软件项目管理之道》[4]，该书对今天我们认识软件的复杂性仍然具有很强的现实指导意义。1969 年，贝尔实验室的 Ken Thompson 和 Dennis Ritchie 在参加多任务信息与计算系统（Multiplexed Information and Computing Service，Multics）[5] 的研究后，认为这一系统过于复杂，从而在简化 Multics 系统的设计理念的基础上研制了 Unix 操作系统（原本叫 Unics，其中的 Uni- 与 Multics 中 Multiplexed 对应）[6]；基于此操作系统的理念，诞生了后续包括 Linux 在内的一系列操作系统。

20 世纪 80 年代面世的一系列操作系统，推动了小型机到 PC 机和嵌入式的演进与发展。举例而言，QNX [7] 诞生于 1982 年，广泛应用于嵌入式领域，当前在智能车领域被广泛应用；VxWorks [8] 诞生于 1982 年，主要在嵌入式和工业场景广泛使用；Windows 的诞生则推动了 PC 的发展，并在随后的 30 多年时间里始终在个人电脑和 PC 服务器市场上占据主流地位；此外，前文提及的 1991 年诞

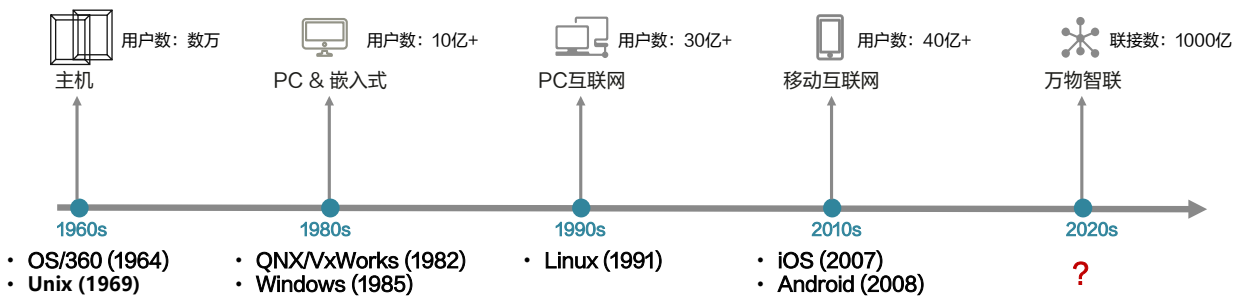


图 2 操作系统与产业应用的共生发展历史

生的 Linux 系统，推动了 PC 服务器和云计算的发展。

iOS 和 Android 等移动操作系统的诞生则推动了手机等移动设备从功能机走向智能机，并伴随着移动互联网的发展，成为移动互联网时代的王者。

面向未来，随着信息技术（Information Technology，IT）、通信技术（Communication Technology，CT）和运营技术（Operational Technology，OT）的逐步融合，以及人工智能（Artificial Intelligence，AI）与 5G 等技术的发展，操作系统将迎来万物智联的新发展机遇与挑战。

2.2 操作系统的成功要素也伴随着产业不断演进

操作系统所承载的价值和成功要素也在不断演进，大致经历了硬件附属、作为独立的软件产品、作为生态与云服务入口以及赋能和赋智千行百业等阶段（见图 3）。

早期的操作系统（IBM OS/360 和早期的 Unix 等）作为硬件的附属品，其价值也依附于硬件的价值产生。例如，IBM OS/360 作为 IBM 大型机的附属而体现其价值。彼时，硬件设备性能约束大，在这个阶段的操作系统关键的成功要素是性能。

20 世纪 80 年代以来，操作系统逐步成为一个独立的软件产品。例如，Windows 就是这一阶段作为独立的软件产品销售的典型代表。由于软件的可复制性非常强，销售软件授权（License）可以获得巨大价值，这使得微软在很长一段时间内都是市值最高的 IT 公司之一。这一阶段，操作系统获得成功的关键要素是通用。Windows 对设备的兼容性和应用的通用性，推动了 PC 时代的发展。

2000 年以后，操作系统的商业模式逐步演进为作为生态与云服务的入口。例如苹果 iOS + iCloud + App Store 与安卓 + GMS + 应用市场形式，通过提供一个生态入口，加上与云的连接，形成一个非常强大的生态，它的成功因素主要是生态的粘性。

展望未来，随着数字世界进入每个人、每个家庭、每个组织，从而构建万物互联的智能世界，操作系统支撑这一战略的关键是要能够赋能、赋智各行各业，而其核心成功要素是柔性和智能。

需要强调的是，上述四种模式并不是替代关系，而是逐步演进的，旧的模式在相当长的时期里会一直存在。面向未来，我们将看到多种形态的共存。例如，操作系统将持续支撑设备类产品的竞争力，操作系统作为生态与云服务入口的方式方兴未艾，同时业界也在积极探索如何赋能、赋智千行百业。

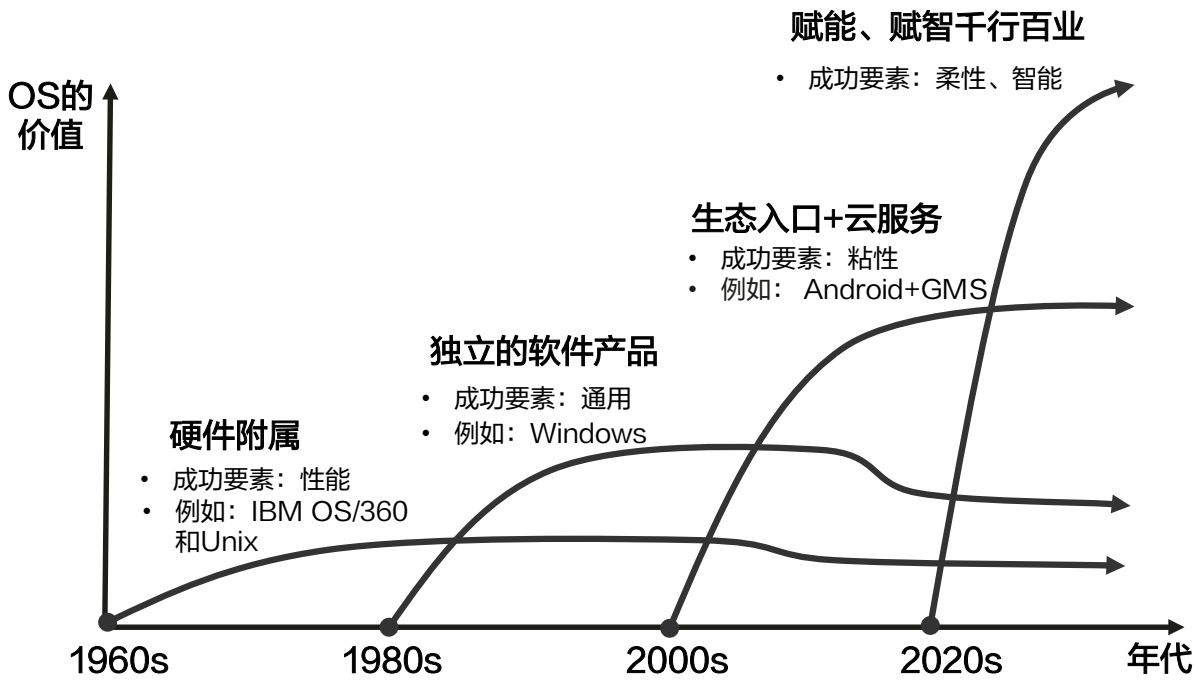


图 3 操作系统的价值与成功要素演进

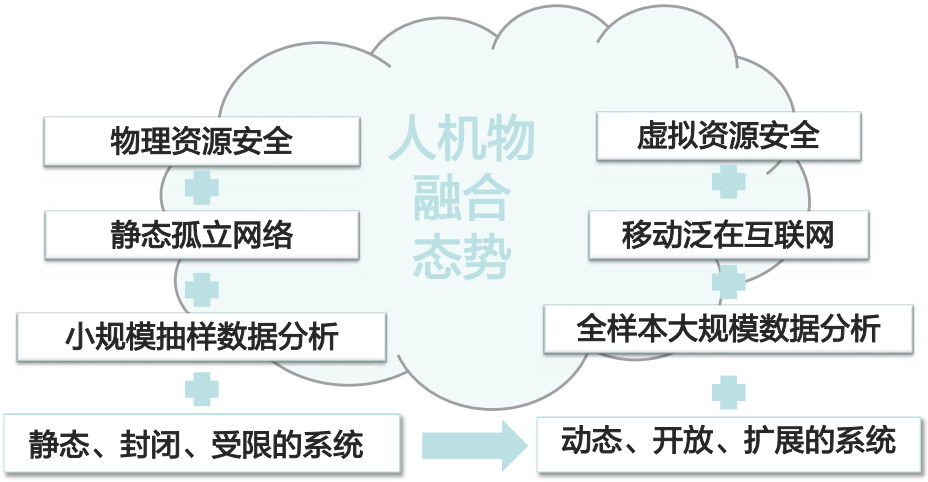


图 4 人机物融合与云网边端协同新使命 [9]

面向未来赋能、赋智千行百业，终端、IT 与 CT 等技术逐步融合，从某种程度来讲，操作系统将承担起人机物融合以及云网边端协同的新使命（见图 4）。封闭场景下的操作系统所面对的往往是物理资源安全的、静态孤立的网络域，适用小规模抽样数据分析，整体是一个静态、封闭、受限的系统。而面向千行百业，操作系统从一个静态、封闭、受限的系统演进到一个动态、开放、扩展的系统，适用全样本大规模的数据分析，需要更加关注虚拟资源的安全 [9]。

3 业务场景驱动的操作系统创新方向展望

3.1 “昆虫纲悖论”呼唤元 OS 架构创新

万物智联会带来许多新的变化，其中一个显著变化就是产品和应用场景会呈现爆炸式地形态进化、杂交和不断演进。我们传统认知上的一种设备，跟其他设备进行杂交、融合，就可能会诞生出新的硬件形态，并增加原有硬件的附加值（见图 5）。

“昆虫纲悖论”是东京大学的坂村健教授对个性化与通用性之间矛盾提出的一个形象比喻。地球上的哺乳动物大约有 5 千多种，好比传统的服务器、PC 和手机等设备；昆虫大概有 100 多万种，好比万物智联时代的多样化设备。一方面，数量更大的万物智联设备带来的累计价值，可能会超过 PC 和手机市场；但另一方面，由于缺乏可大批量复制的软硬件和应用，没有批量也就难以有低成本，因而很难产生很大的市场。

面向万物智联时代的硬件形态不断创新，会带来操作系统的“昆虫纲悖论”挑战，亟需应对多样化场景下的资源大小、功能、性能、安全、生态等多方面的差异化。操作系统必须实现组件式解耦与按需合成，既能够保证架构的一致

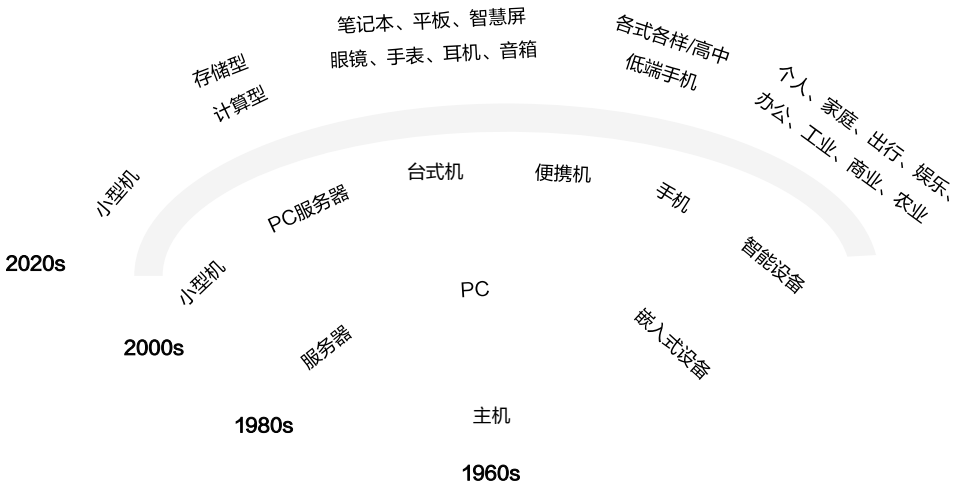


图 5 硬件形态爆炸式地进化、杂交和演变

性，便于统一维护，同时可以通过弹性组合的方式去解决个性问题。我们将满足这样设计意图的操作系统架构称为“元 OS 架构”。

元 OS 架构的目标是“One OS Kit for All”，即我们要打造的不再只是一个操作系统，而是一个可以被灵活组装的操作系统能力集合（OS Kit），基于一个高度弹性的架构，将这些操作系统能力组合成为满足各行各业场景需要的场景化操作系统，进而缓解“昆虫纲悖论”挑战。

3.2 新形态计算呼唤操作系统诞生新的计算抽象

回顾虚拟化和容器这两个当前主流的计算抽象的历史（见图 6），前者诞生于上世纪 70 年代，早在 IBM OS/370 就已提供了虚拟化能力，而创立于 1998 年的 VMware 则开启了 PC 服务器虚拟化的时代序幕；后者在上世纪 70 年代也出现了雏形，比如，早在 1979 年 Unix V7 就提供了早期的隔离环境，而现在业界主流的 Linux 容器（Linux Container，LXC）诞生于 2008 年，Docker 则诞生于 2013 年。虚拟化和容器在一定程度上使能了云计算革命。

然而，随着函数计算、边缘计算的兴起，虚拟机和容器这样的计算抽象存在一定的局限性。以函数计算为例，对运行时环境的新诉求逐步突显，包括（但不仅限于）：

- 75% 的函数计算的最大执行时间仅为 10 秒，希望运行时环境具备极致轻量化和超短生命周期能力；
- 81% 的函数计算为低频调用（平均每分钟少于 1 次），要求更高效地减少驻留内存，提升资源利用率；
- 传统的分支预测在面对短函数时错误率非常高，呼吁打造更有效的新型调度机制；
- 场景更加开放，对安全隔离的诉求更高等 [10]。

这些对操作系统的新诉求与传统操作系统的差异非常大，虚拟机和容器的抽象显得过于厚重，安全隔离性也存在可提升的空间。以上新形态计算诉求正在呼唤操作系统领域诞生新的运行时抽象。

3.3 系统的复杂性和敏捷性之间的矛盾逐步催生“学习型系统”

系统的复杂性和敏捷性之间的矛盾催生“Learned Systems”，即 AI 与系统相结合。在封闭系统下，过去的操作系统是一个结果确定的、整体等于部分之和、稳定封闭的操作系统，因此适用基于经验的调优；然而，面向一个开放、成长的复杂系统，它的整体可能会大于部分总和，而且会不断演化，因此更适合用 Learning 的方式提升系统效率 [9]（见图 7）。然而，用机器学习或者用 AI 来解决系统问题，需要思考如何把机器学习的方式和传统的基于经验的或者基于系统设计的方式，进行有效结合，才能发挥学习型系统的优势。

图 7 还原论与系统论的差别 [9]

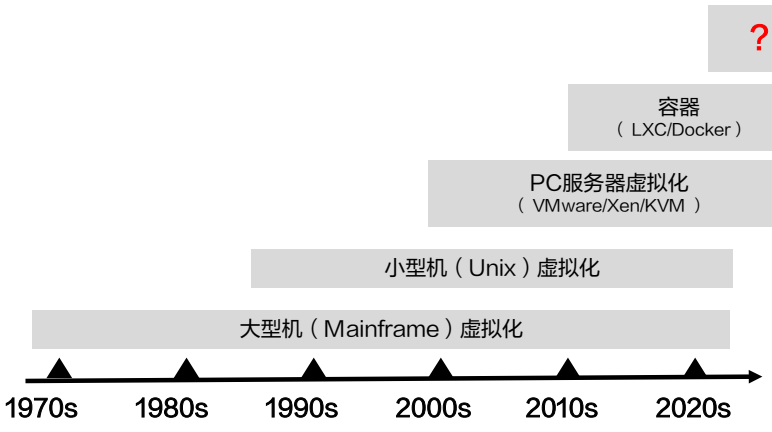


图 6 操作系统运行时抽象的演进

操作系统的参数数量庞大、业务负载与系统状态瞬息万变，如何实现自动调优和运维是业界面临的普遍挑战，也是学术界的研究热点。近年来，学术界在该领域也不断取得新的研究进展，包括（但不仅限于）：基于学习的操作系统 [11]、基于学习的可扩展索引 [12]、基于学习的缓存 [13] 与基于学习的并发控制 [14] 等。

4 硬件驱动的操作系统创新方向展望

4.1 异构众核 SoC、XPU 协同和新计算架构演进呼唤新的计算范式

从算力的角度看，操作系统需要协同和推动异构众核片上系统（System on a Chip，SoC）、XPU 协同和新计算架构三大方面的计算范式演进（见图 8）。

- **异构众核 SoC：**CPU 从单核、多核演进到了当前的众核，在服务器场景下已经可见数百核，甚至千核的规模；同时，我们观察到大小核 / 异构核也从过去的手机等终端设备，进入到 PC 场景（例如苹果 M1），并且有演进到服务器的趋势。围绕异构众核 SoC，操作系统需要研究诸多挑战性的课题，包括但不仅限于：合成大规模的可扩展同步原语 [15]、提升同步原语的可靠性和可扩展性以便能够可靠地释放并发算力 [16]、非统一内存访问（Non-uniform Memory Access，NUMA）+ 非对称多处理（Asymmetric Multiprocessing，AMP）架构感知和瞬态线程迁移等。
- **领域专属架构（Domain-Specific Architecture，DSA）下的 XPU 算力协同：**操作系统需要突破 XPU（例如 CPU、GPU、NPU、DPU 等）间的高效协同，从过去的单一操作系统演进为 SoC 上的一组协同的操作系统，从而最大程度地发挥 SoC 的整体能效。
- **新计算架构探索：**DSA 的弊端逐步显现，例如，厂家要看护多种硬件架构、功能重叠浪费、软件栈难以共享、XPU 间协同调度效率难以大幅提升等。牧村定律 [17] 认为半导体技术每十年完成一次定制化和通用化的回归。业界也期待 DSA 之后能够诞生新的计算架构。而在新的计算架构下，传统操作系统的基础架构和基础能力也将发生根本性的变革。

4.2 新存储介质和互联技术需要操作系统提供新的数据管理范式

传统的缓存、内存、存储的速度与成本是泾渭分明的，它们以 CPU 为中心，形成了梯次缓存的多层存储结构。而存储级内存（Storage Class Memory，SCM）等非易失

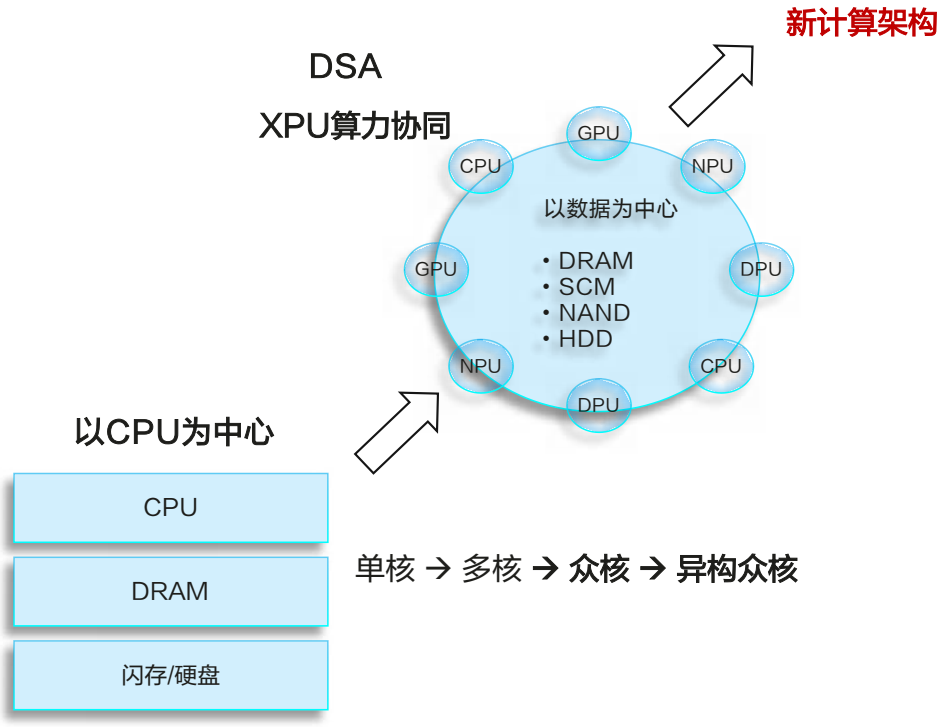


图 8 算力演进示意图

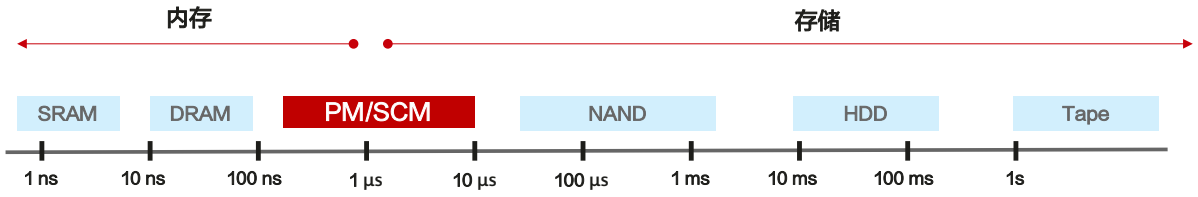


图 9 新介质正在使得内存与存储融合

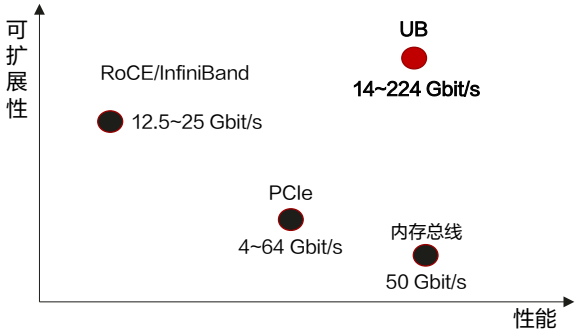


图 10 互联技术的性能与可扩展性

内存的出现，使得存储介质时延从毫秒级进入到微秒级（见图 9）；同时，结合 UB 等高性能、高可扩展性的新型互联技术出现 [18]（见图 10），传统的数据管理范式正在面临挑战。

存储介质时延从毫秒级进入到微秒级，软件栈的时延在整个 I/O 中的占比呈数量级地提升，因此超低时延的存储软件栈 [19]、高并发下数据一致性 [20] 等成为近年的研究热点。

在新介质与新互联技术的共同推动下，传统操作系统的内存和存储两层存储管理架构将可能逐步演进到以数据为中心的单级存储（Single-Level Store，SLS）架构，即内存和存储在逻辑层面上实现了融合管理，从而大幅减少数据搬移；进一步，如何使数据离计算节点更近，实现以业务为中心的效果？这也呼吁近数计算（Near Data Computing，NDC）系统的出现（见图 11）。

4.3 机密计算架构呼唤操作系统领域研究新的信任范式

“机密计算”（Confidential Computing）是指在可信硬件支持下的隔离环境中运行安全计算任务，从而实现对安全计算任务的代码和数据进行保护的一项安全技术。随着数据安全与隐私监管的不断加强（例如，欧盟的 GDPR、以及中国的《数据安全法》等），机密计算受到业界越来越多的关注。从 2002 年 ARM 提出 TrustZone 架构起，出现了各种机密计算技术方案和抽象演进（见图 12）[21]。

TrustZone 是 ARM 在 2002 年提出的隔离机制，其隔离对象包含了 CPU、内存、总线结构和外围设备，提供了独立的物理机抽象 [22]。目前，TrustZone 在移动端 ARM 手机上被广泛部署和使用，提供生物信息识别（例如，指纹识别、人脸识别、虹膜识别）以及安卓系统密钥管理等安全特性。通常，在 ARM TrustZone 会运行一个小而安全的操作系统，例如，Huawei iTrustee [23]、OP-TEE [24]、Trustonic [25] 等。

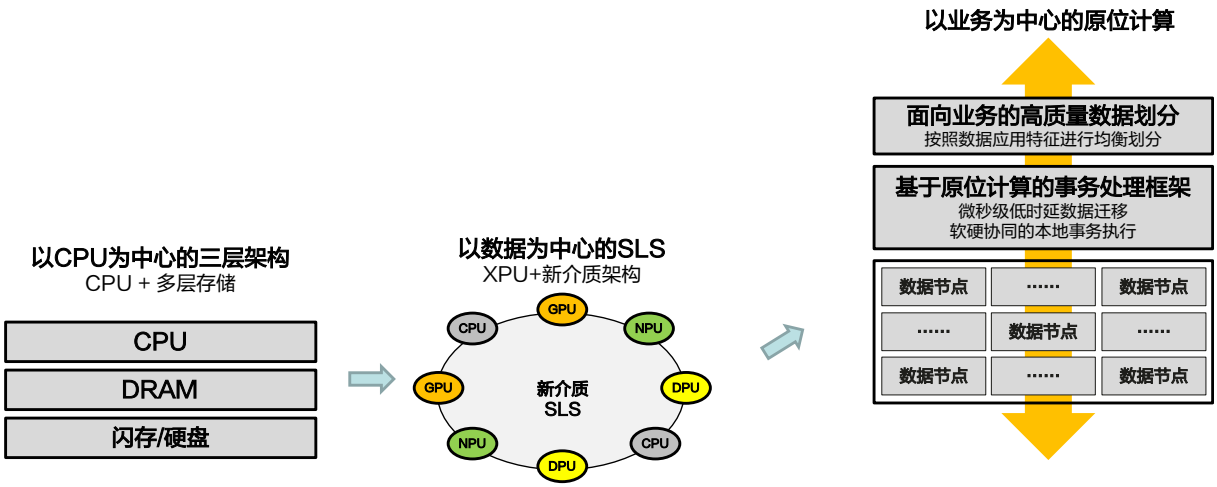


图 11 操作系统计算架构与范式的演进趋势

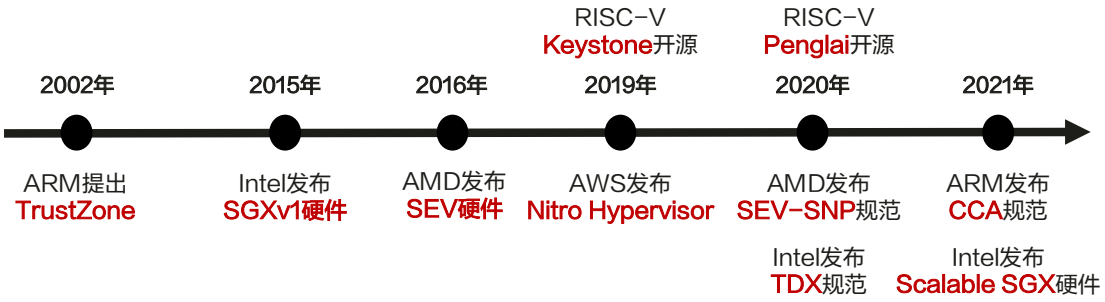


图 12 机密计算主要技术及其发展历史

2021 年的 ARM Vision Day 上，ARM 发布了新一代 ARM V9 架构，其中提出了 ARM 机密计算架构（Confidential Compute Architecture，CCA），在原有的 Non-Secure World 和 Secure World（之前的可信执行环境，TEE）基础上，新引入一个安全区叫 Realm。CCA 新架构的本质是改变了信任范式，从此前终端领域的“信任设备，不信任应用”，演进到云计算领域的“客户信任自己的应用，但不必信任基础设施提供商”。操作系统需要支持 Realm Manager 以及其上的可信任操作系统，才能有效支撑 ARM CCA 的这一信任范式改变 [26]。

2015 年，Intel 正式宣布软件防护扩展（Software Guard Extensions，SGX）技术并发布第一代 SGX。SGX 在用户态进程中划分出一块安全“飞地”（Enclave），Enclave 内的代码和数据对外界完全不可见。程序员需要显式地对代码进行静态划分：安全部分（Enclave）与非安全部分（App），并规定好二者交互的接口。SGX 的安全威胁模型不信任操作系统，因此 SGX 在硬件设计中定义了大量功能，用于检验底层不可信特权软件的行为，主要包括安全页表管理、安全线程管理、安全内存的换页管理、以及安全内存的完整性校验等。由于 Enclave 抽象是应用的一部分，而应用与操作系统的接口与交互极其复杂并且经常演进，导致 SGX 的硬件设计比较复杂，硬件需要承载很多软件的功能，并且性能开销也较大。2022 年 1 月 Intel 宣布在除服务器之外的新桌面 / 笔记本 / 嵌入式处理器中放弃所有的 SGX 特性 [27]。

信任域扩展（Trust Domain Extensions，TDX）是 Intel 继 SGX 之后推出的新安全计算抽象。在设计上，TDX 引入了单独的可信的 Hypervisor/VMM，和原有的云平台 VMM 共享同一物理主机。可信的 Hypervisor 称为 TDX Module，是一小块安全模块，负责可信域（Trusted Domain，TD）内的可信虚拟机和外部不可信 VMM 之间的交互检查。

从 ARM CCA 和 Intel TDX 等演进趋势看，操作系统领域需要研究相应的信任范式演进，通过创新操作系统的架构设计（例如，微内核架构）来减少机密计算抽象内的可信计算基（Trusted Computing Base，TCB）的大小，同时也需要使用合理的方式来保障机密计算抽象中 TCB 的安全性与可信性，如软硬件协同安全加固、形式化验证、安全性更高的语言（例如 Rust）等。

5 openEuler 和 OpenHarmony 的实践

5.1 openEuler 的实践

2019 年 12 月，华为创建欧拉开源项目（openEuler），通过开源的方式，把多年来积累的操作系统能力开放出来，携手产业伙伴共同发展操作系统产业；2021 年 11 月，在操作系统产业峰会上，华为携手社区伙伴，将欧拉正式捐赠给开放原子开源基金会，实现了欧拉开源操作系统从企业主导到产业主导的重要转变 [28]。

从产业应用角度看，openEuler 打造了一套灵活的架构支持服务器、云计算、边缘计算、嵌入式等 ICT 和 OT 场景，成为面向数字基础设施统一的开源操作系统；同时，从硬件角度看，openEuler 支持多样性算力的协同释放，打造了面向 SCM 等新存储介质的存储软件栈，同时还打造了自动化、智能化的性能调优引擎。

在各参与方的携手努力下，截至 2022 年 3 月，openEuler 及其发行版已经实现了在十多个行业的应用，实现对 5000 多种软件的认证支持，并已兼容 60 多种整机和 200 多种主流板卡，支持鲲鹏、x86、飞腾、龙芯、申威、RISC-V 等多种处理器架构 [29]。

5.2 OpenHarmony 的实践

华为于 2020 年起陆续将其智能终端操作系统（即“鸿蒙操作系统”）的基础能力捐献给开放原子开源基金会，由开放原子开源基金会整合其他参与者的贡献，形成 OpenHarmony 开源项目，同时华为持续参与 OpenHarmony 开源项目的共建。

OpenHarmony 是面向万物互联时代的智能终端统一的操作系统，为不同设备的智能化、互联与协同提供了统一的语言，为消费者带来简捷、流畅、连续、安全可靠的全场景交互体验。具有三个独有特征：

- 一套操作系统可以满足大小设备需求，实现统一 OS 弹性部署；OpenHarmony 通过组件化和弹性化等设计方法，做到硬件资源的可大可小（支持从百 KB 级到 GB 级的内存大小），并覆盖了 ARM、RISC-V 和 x86 等多种处理器架构。
- 搭载该系统的设备在系统层面融为一体、形成超级终端，让设备的硬件能力可以弹性扩展，实现设备之间硬件互助、资源共享；OpenHarmony 的分布式能力使得终端设备间能够快速发现与互联，并实现高效数据传输，以及跨设备任务协同。
- OpenHarmony 提供了跨终端的应用开发框架和 API 一致性，面向开发者，实现一次开发、多端部署。

OpenHarmony 在技术架构上整体遵从分层设计，从下向上依次为：内核层、系统服务层、框架层和应用层。系统功能按照“系统 > 子系统 > 组件”逐级展开，在多设备部署场景下，支持根据实际需求裁剪某些非必要的组件。

5.3 openEuler 与 OpenHarmony 的协同创新

openEuler 侧重面向数字基础设施，OpenHarmony 则侧重智能终端。而面向未来，openEuler 和 OpenHarmony 协同创新，预期将有机会打造出一个覆盖云、管、边、端能力的协同操作系统。

- **openEuler 和 OpenHarmony 之间的能力共享：**包括（但不仅限于）内核、驱动框架、协作总线、端云协同运行时环境等（见图 13）。
- **openEuler 和 OpenHarmony 组合实现一机多域、优势互补：**在一些复杂场景下——例如，办公 PC 既要打造流畅、低功耗的用户体验，又要利用现有的生产力工具和应用生态——可以通过打通 openEuler 和 OpenHarmony，实现“一机多域”混合部署（见图 14）。
- **openEuler 和 OpenHarmony 协同突破云网边端的跨域融合：**基于近远场融合软总线等技术，实现近远场设备的自发现、自连接、弹性规模和服务水平协议（Service Level Agreement，SLA），打造体验感知的链路管理，并实现云网边端算力互助（见图 15）。

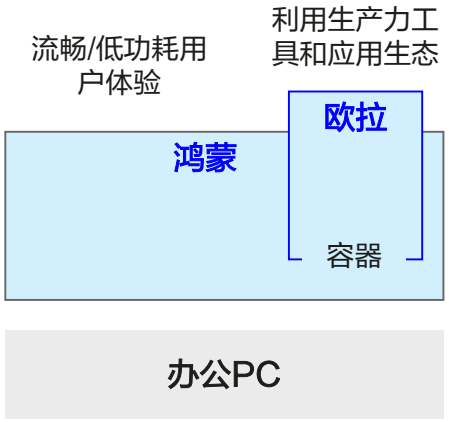


图 14 openEuler 和 OpenHarmony 多域组合

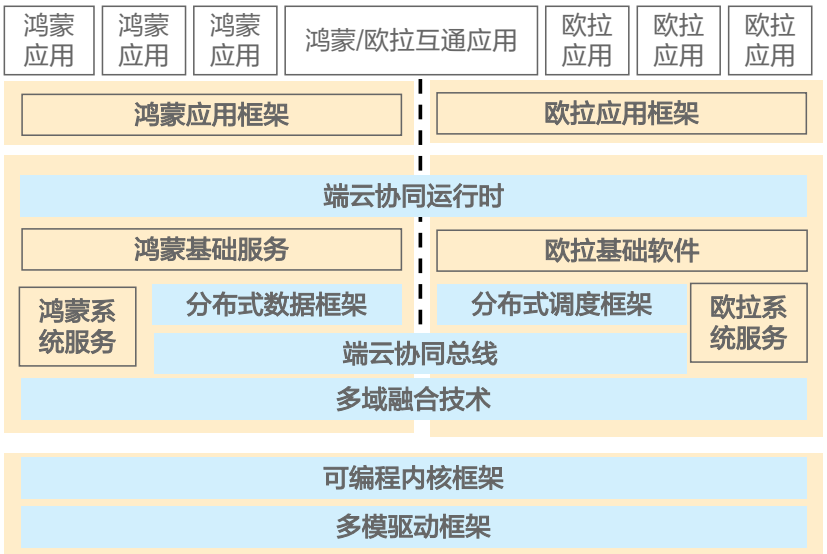


图 13 openEuler 和 OpenHarmony 能力共享

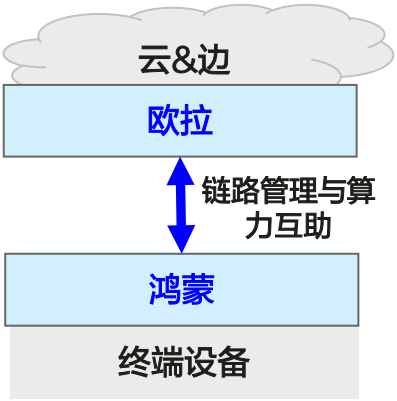


图 15 openEuler 和 OpenHarmony 云网边端跨域融合

6 结语

操作系统承上启下，向上服务应用，向下管理、挖潜硬件能力。本文分别从产业应用场景演进和硬件演进两个视角来观察和思考操作系统的发展轨迹。

从产业应用演进看，万物智联时代产品形态多样化，要求操作系统具备弹性架构；函数计算、边缘计算等新计算形态兴起，呼唤操作系统诞生新的极致轻量、短生命周期、安全隔离的计算抽象；而系统的复杂性和敏捷性之间的矛盾，又将推动“学习型系统”的发展。

从硬件演进看，异构众核 SoC、DSA 架构下的 XPU 协同、以及新计算架构演进，呼唤操作系统领域研究新的计算范式；新存储介质和互联技术呼唤研究新的数据管理范式；同时，ARM V9 发布的 CCA 架构正在推动操作系统信任范式从终端设备的“信任设备、不信任应用”到云计算场景的“客户信任自己的应用，但不必信任基础设施提供商”转变，催生机密计算操作系统。

openEuler 侧重面向数字基础设施，OpenHarmony 则侧重智能终端，而 openEuler 和 OpenHarmony 协同创新，正在逐步打造一个覆盖云、管、边、端能力的协同操作系统体系。

参考文献

[1] 张效祥等 . 计算机科学技术百科全书（第三版）. 清华大学出版社，2018.

[2] https://en.wikipedia.org/wiki/GM-NAA_I/O

[3] Frederick P. Brooks Jr., "The IBM operating system/360," *Software Pioneers*, Springer, Berlin, Heidelberg, 2002, 170-178.

[4] Frederick P. Brooks Jr., *The Mythical Man-Month: Essays on Software Engineering*, Pearson Education, 1995.

[5] V. A. Vyssotsky, F. J. Corbató, and R. M. Graham, "Structure of the Multics supervisor," *Proceedings of the November 30--December 1*, 1965, Fall Joint Computer Conference, part I, 1965.

[6] Dennis M. Ritchie, "The evolution of the Unix time-sharing system," *Symposium on Language Design and Programming Methodology*, Springer, Berlin, Heidelberg, 1979.

[7] David C. Sastry and Mettin Demirci, "The QNX operating system," *Computer*, 28.11 (1995): 75-77.

[8] https://www.windriver.com/products/vxworks

[9] 王怀民 . 分布计算 2.0：基于网络的联接计算 . 2020 CCF 中国软件大会特邀报告，2020.

[10] Mohammad Shahrad, Jonathan Balkind, and David Wentzlaff, "Architectural implications of function-as-a-service computing," *Proceedings of the 52nd Annual IEEE/ACM International Symposium on Microarchitecture*, 2019.

[11] Yiying Zhang and Yutong Huang, "Learned operating systems," *ACM SIGOPS Operating Systems Review*, 53.1 (2019): 40-45.

[12] Chuzhe Tang *et al.*, "XIndex: A scalable learned index for multicore data storage," *Proceedings of the 25th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, 2020.

[13] Xingda Wei, Rong Chen, and Haibo Chen, "Fast RDMA-based ordered key-value store using remote learned cache," *14th USENIX Symposium on Operating Systems Design and Implementation*, Banff, Alberta, Canada, November 2020.

[14] Jiachen Wang, Ding Ding, Huan Wang, Conrad Christensen, Zhaoguo Wang, Haibo Chen, and Jinyang Li, "Polyjuice: High-performance transactions via learned concurrency control," in *Proceedings of the 15th USENIX Symposium on Operating Systems Design and Implementation*, July 2021.

[15] Rafael Lourenco de Lima Chehab *et al.*, "CLoF: A compositional lock framework for multi-level NUMA systems," *Proceedings of the ACM SIGOPS 28th Symposium on Operating Systems Principles*, 2021.

[16] Jonas Oberhauser *et al.*, "VSync: push-button verification and optimization for synchronization primitives on weak memory models," *Proceedings of the 26th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, 2021.

[17] Tsugio Makimoto, "Implications of Makimoto's Wave," *Computer*, 46.12 (2013): 32-37.

[18] Tan Kun *et al.*, "UB: Unified and scalable memory-semantic interconnection for next-generation computing systems," *Communications of HUAWEI RESEARCH*, vol. 1.

[19] Mingkai Dong and Haibo Chen, "Soft updates made simple and fast on non-volatile memory," *2017 USENIX Annual Technical Conference (USENIX ATC 17)*, 2017.

[20] Jifei Yi *et al.*, "HTMFS: Strong consistency comes for free with hardware transactional memory in persistent memory file systems," *Proceedings of the 20th USENIX Conference on File and Storage Technologies*, 2022.

[21] Fahmida Y. Rashid, "The rise of confidential computing: Big tech companies are adopting a new security model to protect data while it's in use-[news]," *IEEE Spectrum*, 57.6 (2020): 8-9.

[22] Sandro Pinto and Nuno Santos, "Demystifying Arm TrustZone: A comprehensive survey," ACM Computing Surveys (CSUR), 51.6 (2019): 1-36.

[23] https://developer.huawei.com/consumer/cn/forum/topic/0202513076169910329?fid=0102501639476130680&postId=0302513076169910363

[24] https://www.op-tee.org/

[25] https://www.trustonic.com/

[26] Dominic P. Mulligan *et al.*, "Confidential computing—a brave new world," *2021 IEEE International Symposium on Secure and Private Execution Environment Design (SEED)*, 2021.

[27] https://community.intel.com/t5/Blogs/Products-and-Solutions/Security/Rising-to-the-Challenge-Data-Security-with-Intel-Confidential/post/1353141

[28] https://openeuler.org/

[29] https://www.openeuler.org/zh/compatibility/

GaussDB：云原生分布式数据库

Andy Li，任阳
高斯部

摘要

华为 GaussDB 是由 2012 实验室中央软件院高斯部开发的企业级分布式数据库内核。该内核历经十数年的开发，现在已经被广泛应用于国内各地的企业客户，包括一些大规模金融机构。中国前七大银行中有五家选择了 GaussDB，并广泛应用在其核心业务场景如核心银行、互联网金融服务、渠道业务、客户关系管理（Customer Relationship Management，CRM）/企业资源计划（Enterprise Resource Planning，ERP）等。截止到 2021 年年底，GaussDB 在中国有 1500 多家企业客户。本文介绍了 GaussDB 的架构和它的一些主打特性。GaussDB 在高可用、高性能、混合负载、安全以及自治五大方面构筑竞争力。

关键词

分布式数据库，云化，高性能，高可用，混合负载，安全，自治，openGauss，GaussDB

1 引言

随着新业务和新基础设施的出现，应用对数据库提出了新的技术要求。随着移动互联网的普及，2C 服务大量出现。金融服务、移动设备上的电子商务应用、社交应用以及物联网（Internet of Things，IoT）设备正在产生各种类型的海量数据。这一切的变化，要求改变传统业务基于柜台和集中处理的服务模式。在此背景下，数据库技术从集中式数据库走向分布式数据库，再发展到基于云的分布式数据库。为更好地满足金融、企业服务以及相关新型应用需求并高效利用云计算基础设施，华为发布了自主研发的企业级云分布式数据库 GaussDB。与传统的分布式应用架构和分布式中间件架构数据库不同，GaussDB 分布式数据库从一开始就为分布式而设计，拥有结构化查询语言（Structured Query Language，SQL）引擎、执行引擎和存储引擎等组件。GaussDB 在以下五个方面构筑自己的竞争力：高可用、高性能、混合工作负载、安全和自治。GaussDB 采用基于 shared-nothing 大规模并行处理（Massively Parallel Processing，MPP）架构，支持美国国家标准学会（American National Standards Institute，ANSI）SQL 2008 标准。

本文的其余部分组织如下。第 2 节介绍了 GaussDB 的架构概况以及竞争力的五个主要方向。第 3 节中我们进行总结并对未来可能的工作进行了一些讨论。

2 架构

在这一节中，我们首先介绍 GaussDB 的架构。

GaussDB 基于 shared-nothing 架构，单套数据库可以线性扩展到数百台物理机，可以并行处理各种工作负载。数据库的数据按照一定条件分布在并存储在各个数据节点（Data Node，DN）中。这些节点满足本地“原子性，一致性，隔离性和持久性”（Atomicity, Consistency, Isolation, Durability，ACID）属性，系统通过使用两阶段提交（Two Phase Commit，2PC）和全局事务管理器（Global Transaction Manager，GTM）来保证跨节点的一致性。GaussDB 同时支持行存储及列存储格式。基于列存储格式，GaussDB 实现了向量执行引擎以高效利用处理器最新的单指令多数据（Single Instruction Multiple Data，SIMD）指令，从而实现指令级别的细粒度并行。系统生成的执行计划以及执行器都针对分布式系统进行设计，可在数百台服务器上对执行计划进行并行处理，节点间按需交换数据且并行执行查询。GaussDB 采用基于时间戳的 GTM-Lite 技术，最大限度降低跨节点事务的性能衰减。系统中相关的逻辑实例有运维管理器（Operation Manager，OM）、集群管理器（Cluster Manager，CM）、GTM、协调节点（Coordinator Node，CN）和 DN。应用程序通过 CN 发送 SQL 语句，经 CN 产生执行计划，下推到相关 DN 执行查询，DN 将结果汇总到 CN 上，最后由 CN 返回给应用程序。

表 1 系统实例描述

实例	功能
CN	接收 SQL 语句并产生执行计划，协调 DN 节点完成执行计划
DN	存储实际数据。DN 相互配合一起完成 SQL 查询。
GTM	产生全局时间戳并提供全局快照。
CM	由 CM Agent 以及 CM Server 组成，负责集群仲裁和提供高可用能力。
Gauss Data Source (GDS)	提供数据并行加载能力。

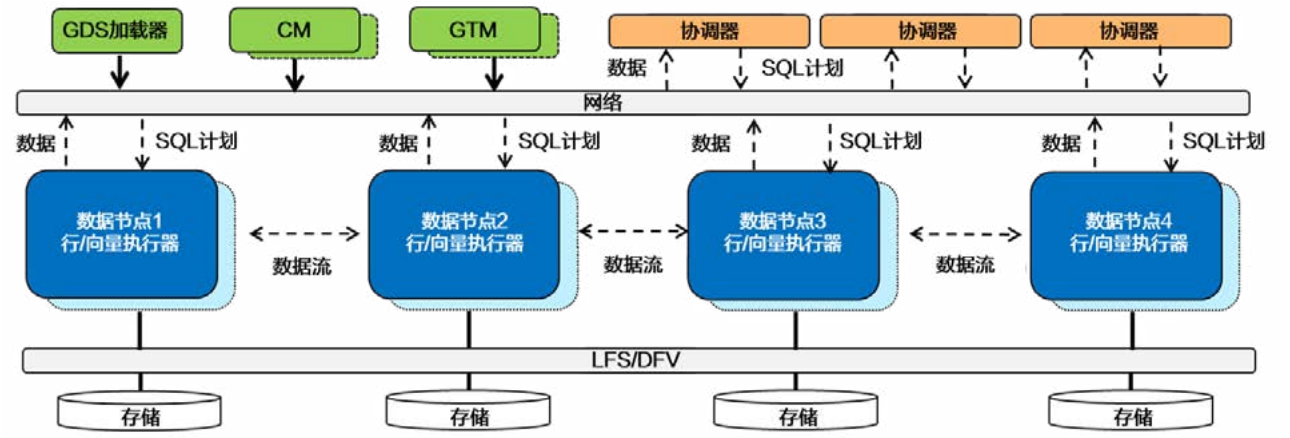


图 1 GaussDB 高级逻辑系统架构

2.1 高性能

GaussDB 的高性能建立在充分利用硬件资源基础上。针对鲲鹏处理器，GaussDB 重构了数据库内核中的关键数据结构从而避免了全局共享数据结构的跨片访问，大大减少了 CPU 伪共享（False Sharing）的问题。同时 GaussDB 使用鲲鹏处理器的原子“取有效地址”（Load Effective Address，LEA）指令重构了数据库中的自旋锁（Spin Lock），将自旋锁的性能提高了 4 倍。基于存储级内存（Storage Class Memory，SCM）和远程直接数据存取（Remote Direct Memory Access，RDMA）技术，GaussDB 团队研究了以内存为中心的架构和下一代分布式数据库通信网络。所有这些使 GaussDB 在单节点 2-socket 鲲鹏服务器上提供 150 万 tpmC 处理能力，在单节点 4-socket 鲲鹏服务器上提供超过 230 万 tpmC 处理，线性伸缩比达到 1.5。

此外，GaussDB 通过以下四个方面增强系统性能。

- Shared-Nothing 架构，GaussDB 将数据按照一定规则分布在各个 DN 上。在执行查询时，DN 在本地处理自己的数据，并根据需要在节点之间打散数据。
- 对称多处理（Symmetric Multiprocessing，SMP）技术，为了充分利用现代 CPU 多核能力，GaussDB 实现了查询的 SMP 技术。在一个节点内，数据被分成多个部分并由不同的线程并行处理。在一些计算密集型的查询中，系统可以充分利用 CPU 来加速查询处理。
- 指令级并行（Instruction-Level Parallelism，ILP）技术，基于现代 CPU 的 SIMD 能力，GaussDB 实现了向量执行引擎。向量执行引擎以批处理模式（每次 1000 个 tuple）处理数据，使得系统在 TPC-H 和 TPC-DS 基准测试中获得高分。
- 动态编译技术，为了不断提高处理性能，我们采用了动态编译策略。使用低级虚拟机（Low Level Virtual Machine，LLVM）技术精简查询涉及的 CPU 指令数，从而提升查询性能。

2.2 高可用

GaussDB 的高可用能力包括五个部分：防错、容错、纠错、快速定位和服务可用性。从初始设计开始，GaussDB 的架构理念是防止系统单点故障（Single Point of Failure，SPOF），系统的相关硬件交换机、网卡和电源等硬件组件通过冗余达到该目标。对于本地盘，系统使用 RAID5 阵列提供数据保护，防止单个磁盘数据损坏时导致业务中断，对于 EVS 云盘，系统使用云盘自身的多副本机制提供数据可用性。除了硬件冗余外，系统对于实例角色的架构设计也采用高可用架构，系统中 CN 采用多副本对等架构、DN 采用主备副本机制，CM、GTM 等也都采用温备份机制。各个角色的主备裁决均由 CM 进行，CM 自身的主备裁决依赖于分布式键值存储系统（Editable Text Configuration Daemon，ETCD）组件。CM 作为集群的大脑，决定实例的主备角色。其自身有多个温备份，当系统主 CM 的主实例发生故障时，经过 ETCD 裁决，选择一个 CM 备进行升主操作，从而接管集群仲裁能力。GTM 和 DN 等角色的主实例和温备实例定期向 CM 主实例发送心跳。根据心跳，CM 确定各个角色的主实例，并在主实例故障时进行切换。除此之外，GaussDB 提供多种日志分析工具、核心转储分析工具、分布式跟踪工具等，在系统出现故障时提供高效问题定位手段。同时，GaussDB 还提供运维操作不中断服务的能力，在线升级和在线扩容功能可以保证在升级和扩容操作时业务工作流程不中断。

图 2　GaussDB 高可用能力全景图

在实际金融行业的生产系统中，根据不同业务的高可用要求，GaussDB 提供多层次的高可用能力：单个可用分区（Availability Zone，AZ） 高可用、跨 AZ 高可用以及跨 Region 高可用能力。在 AZ 中，基于无单点故障的设计，GaussDB 在提供 RPO = 0，RTO < 10s 的高可用能力。在保证 AZ 级高可用能力基础上，GaussDB 在 100+ 物理节点上能够提供 1.2 亿 tpmC 的超高性能。GaussDB 提供跨 AZ 集群部署，提供强一致性以及对应的高可用能力（RPO = 0 和 RTO < 30s）。除此之外，当服务需要提供超过 1000 公里的跨 Region 高可用能力时，GaussDB 可以提供 RPO 小于 10s、RTO 分钟级的能力。对于更大的地理区域的高可用需求，例如针对半径超过 2000 公里的区域，GaussDB 提供基于全球时钟一致性的 global database 能力，并为整个系统提供 5 个 9 的可用性。基于此可以看出，GaussDB 在兼顾高性能基础上，在满足不同应用的多层级高可用能力要求。

2.3 混合工作负载

同时支持联机事务处理（Online Transactional Processing，OLTP） 和联机分析处理（Online Analytical Processing，OLAP）服务能力是企业级数据库的标准能力。当前大量使用的 Oracle 数据库以及 SQL Server 数据库等都具备对应的能力。同时在实际的企业级服务中，也很难说一个业务是纯 OLTP 或者纯 OLAP 负载。混合工作负载能力是企业级数据库的必备条件。通过全并行架构，包括节点间并行、节点内并行和指令并行，GaussDB 大大提高了数据库的复杂查询能力。利用企业级的分布式优化器能力，GaussDB 实现了诸如视图扩展、参数化路径和 Lazy 聚合等功能。在相同的硬件环境和测试集下，GaussDB 分布式数据库在复杂查询方面的耗时比竞品少 82%。基于强大的复杂查询能力，轻量级的全局快照 GTM-Lite 技术大大提升了系统的交易处理性能，且系统性能随着节点数的增加而线性增加。在实际使用中，最大的 GaussDB 单集群的节点数接近 1000 个，集群消耗的 vCPU 核数接近 10 万个。

混合工作负载的性能易受系统资源的可获取性影响。GaussDB 依赖于全局工作负载管理器控制系统并发运行的查询数量。工作负载管理器优化了系统的吞吐量，同时避免了因查询争抢系统资源而导致的处理性能下降。工作负载管理器由三个主要部分组成：资源池、工作负载组和控制器。资源池用于为系统中运行的查询分配共享的系统资源，如内存和磁盘 I/O，并用于设置各种执行阈值，确定允许查询的执行方式。所有查询都在一个资源池中运行，工作负载组用于将到达的查询分配到资源池中。控制器评估查询，并根据查询的资源需求（即成本）和系统的可用资源（即容量）动态地做出执行决定。如果一个查询的估计成本不高于系统的资源可用容量，则执行该查询。否则，该查询将进入等待队列。资源记录和反馈机制被用于跟踪系统的可用容量。当系统的容量变得足够大时，查询被从等待队列中弹出，并由执行引擎执行。

2.4 安全能力

在云端，GaussDB 分布式数据库在数据传输、查询、计算、运行维护和闲置期间对客户数据提供全面的安全保护。为了解决查询过程中的数据安全问题，GaussDB 提供了全加密的查询功能，数据在密态被处理完成查询请求，只有当数据返回客户端时才被解密，从而确保传输和查询过程中的安全。GaussDB 使用跟踪链来记录所有的数据库变化，以识别运行和维护期间数据的历史变化。在有多方参与的情况下，被篡改的数据会被系统自动发现并纠正。

由于开放的环境和模糊的网络边界，云数据库正面临着比以往更多样化、更严重的威胁。尽管大多数云数据库系统都采用了各种保护机制，如数据加密、身份验证和审计

等，但这些保护机制都有一个共同的假设：数据库用户信任云基础设施和特权用户（如管理员）。然而这个假设具有一定的脆弱性，从而使用户的敏感数据面临较大风险。基于这种状况，数据库系统一般采用了相关的端到端防御机制，如全同态加密和属性保护加密。微软的 Azure database 允许数据所有者在列粒度上进行加密，并利用可信执行环境（Trusted Execution Environment，TEE）安全可靠地对密文数据进行复杂操作。因此，即使是有特权的用户或基础设施供应商也无法获得用户的敏感数据。但是这种技术理念对数据库的设计和实现带来对应的挑战性，GaussDB 提出了一种名为 FE-in-GaussDB 的新安全机制，在私有云环境和公有云环境中对数据提供安全保护能力。FE-in-GaussDB 结合了软件和硬件的各自优势和安全功能来实现处理灵活性。软件模式包括数据加密和索引方案，用于对密文数据进行有效的等于查询和范围比较操作，在富执行环境（Rich Execution Environment，REE）——即 Normal World——中运行。硬件模式利用芯片提供的硬件 TEE（即 Secure World），安全地处理密文数据的解密和复杂计算，如字符串搜索操作和聚合功能。FE-in-GaussDB 支持全场景的 SQL 查询处理，其优势总结为如下几点：

- 一种新的数据库安全能力模式，将现有的数据软件加密方案与利用硬件 TEE 进行的保密计算进行结合。
- 全生命周期密态处理，使得 FE-in-GaussDB 对数据库用户和应用透明。
- 基于数据库的执行模式达到安全和性能之间的平衡。

为实现 FE-in-GaussDB 安全机制，GaussDB 对优化器、执行器等进行了对应的修改，包括：

- 修改优化器，以确定请求是否与密码字段有关，以及是否应在 TEE 内处理。
- 建立两层的密码索引。上层索引使用标签来代表不同的处理范围。下层索引利用 TEE 来保持顺序存储。
- 数据定义语言（Data Definition Language，DDL）操作和具有等值查询类的数据操纵语言（Data Manipulation Language，DML）语句都直接在数据库中使用软件模式处理，不需要访问 TEE。
- 软件模式和硬件模式都使用相同的客户端加密驱动和客户端解析器模块。

2.5 自治能力

GaussDB 建立了一个数据库的自治框架，并将其整合到 openGauss 中，该框架主要包含以下五个部分：

- 学习型优化器。GaussDB 提出了学习型查询重写、学习型代价 / 基数估计、学习型计划生成器（包括连接顺序选择和物理运算符选择）。学习型查询重写使用一种基于蒙特卡洛树搜索（Monte Carlo Tree Search，MCTS）的方法，通过考虑重写的好处和重写的顺序，将 SQL 查询重写成一个等价的但更高效的查询。学习型代价 / 基数估计使用树状长短期记忆（Tree Long Short-Term Memory，Tree-LSTM）同时估计成本和基数。学习型计划生成器利用深度强化学习来选择一个好的连接顺序和适当的物理运算符。
- 学习型数据库自监测自诊断。GaussDB 实现了基于学习的自我监测、自我诊断、自我配置和自我优化技术来监测、诊断、配置和优化数据库。自我监控数据库的指标，并使用这些指标来优化其他组件运行行为。自我诊断使用 Tree-LSTM 检测异常情况，并对异常情况进行根因分析。自我配置使用深度强化学习方法来调整配置。自我优化使用编码器 - 解码器模型来进行视图推荐，同时使用深度强化学习模型进行推荐索引。
- 模型验证。为验证一个学习模型对工作负载是否有效，GaussDB 团队提出了一个基于图嵌入的性能预测模型。对于学习型优化器或学习型顾问中的模型，系统在部署之前对其性能进行预测。如果性能变好，就部署这个模型；否则就放弃它。
- 模型管理。由于学习型优化器和学习型顾问中的大多数模型是通过结合多种强化学习或深度学习算法实现的，为实现统一的资源调度、模型管理，GaussDB 团队提供了一个机器学习（Machine Learning，ML）平台。它封装了 ML 的复杂性，提供面向开发者的训练、预测、模型管理的能力。
- 训练数据管理。GaussDB 收集运行时的数据库指标（如资源消耗、锁信息）、历史 SQL 查询（如查询延迟）和系统日志（如异常情况）等数据，作为系统训练模型的原始输入，并对这些数据进行以下组织和管理：
  - 智能地将相关的列组织到同一个表中，以减少连接开销；
  - 智能地选择训练数据来训练一个学习模型。

为提高系统的运维能力，GaussDB 利用深度强化学习和图神经网络（Graph Neural Network，GNN）等人工智能能力完成了 Tuner、系统诊断、物化视图（Materialized View，MV）推荐、索引推荐等功能，大大提高了系统效率。以客户实际场景中使用的索引推荐功能为例，使用该功能后，在保证客户实际生产系统性能提升基础上，索引冗余度减少了 83%，占用的磁盘空间减少了 70%。

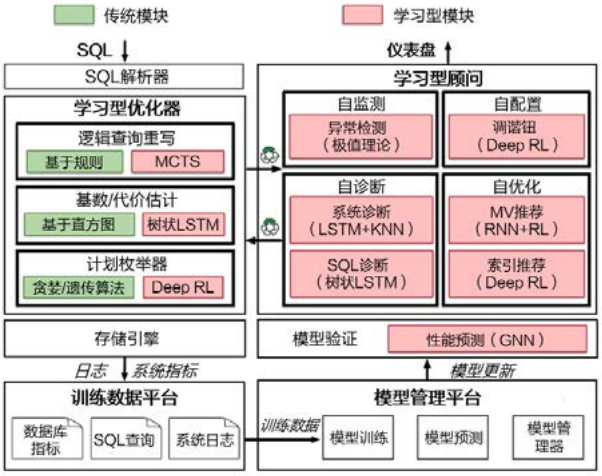


图 3　GaussDB 自治数据库的逻辑架构

3 结语

在不断的工程实践基础上，我们做了大量的数据库创新研究。近年来，通过与清华大学等的学术合作，我们在数据管理国际会议（Special Interest Group on Management of Data，SIGMOD）、大规模数据库国际会议（International Conference on Very Large Data Bases，VLDB）和国际数据工程会议（International Conference on Data Engineering，ICDE）上发表了 50 多篇学术论文。论文主题涵盖人工智能原生数据库、交易处理、内存数据库（In-Memory Database，IMDB）和分布式数据库系统等场景。接下来，我们将进一步扩大与全球学术界的合作，继续探索工程实践和数据库创新，以适配新需求场景、新硬件架构和技术。未来，GaussDB 将继续在软硬件深度协同、扩展性、serverless 技术、多模型融合等方面进行探索性研究。同时将继续推动 openGauss 社区的发展，将其打造成主流数据库社区。

MindSpore：人工智能开源生态系统实践

于璠，时北极，王紫东，金学峰，陈雷，苏腾，李锐锋，周斌，丁诚，谭焜

摘要

MindSpore 是一种全新的深度学习计算框架，旨在实现易开发、高效执行、全场景覆盖三大目标。为了实现易开发的目标，MindSpore 采用基于源码转换的自动微分机制，该机制可以用控制流表示复杂的组合。函数借助于中间表达可以构造出能够在不同设备上解析和执行的计算图。在执行前，计算图上应用了图算融合等多种软硬件协同优化技术，从而实现高效执行的目标，并同时提升端、边、云等不同场景下的性能和效率。除此以外，MindSpore 还支持简洁的动态图和静态图之间的模式切换。为了在大型数据集上有效训练大模型，MindSpore 可以灵活的支持数据并行、模型并行和混合并行训练，并具有“自动并行”能力，它通过在庞大的策略空间中进行高效搜索来找到一种快速的并行策略。除了卷积神经网络（CNN），MindSpore 还支持图神经网络（GNN）等多种 AI 模式，并已形成相比于其他开源框架的差异化竞争优势。

鉴于在华为昇腾系列芯片上实现的卓越性能，MindSpore 的开源生态系统正蓬勃发展。来自高校、企业以及社区的关注度日益增长，在学术发展、商业推广以及社区运营实践方面积累了丰富的经验。基于 MindSpore 生态链的相关技术正在多个企业以及 AI 计算中心落地。

在本文中，我们将介绍 MindSpore 的主要架构与核心技术。此外，将进一步介绍 MindSpore 在人工智能开源生态系统的实践探索。

关键词

深度学习框架，自动微分，自动并行，图算融合，动静图统一，全场景，大模型，生态系统实践

1 引言

深度学习研究和应用在近几十年得到了爆炸式的发展，在图像识别 [1]、语音识别与合成 [2]、游戏 [3]、语言建模与分析 [4] 等方面取得了巨大的成功。深度学习框架 [5-10] 的不断发展使得在大型数据集上训练神经网络模型时，可以方便地使用大量的计算资源。

目前有两种主流的深度学习框架。一种是在执行之前构造一个静态图，定义所有操作和网络结构，典型代表是 TensorFlow [5]。这种方法以牺牲易用性为代价，来提高训练期间的性能。另一种是立即执行的动态图计算，典型代表是 PyTorch [6]。通过比较可以发现，动态图更灵活、更易调试，但会牺牲性能。因此，现有深度学习框架难以同时满足易开发、高效执行的要求。

本文介绍华为自研的一种新的深度学习计算框架—MindSpore，该框架旨在实现三个目标：易开发、高效执行和全场景覆盖。MindSpore 由如下主要组件组成：MindExpression、MindCompiler、MindRE、MindData 和 MindArmour。表 1 总结了 MindSpore 各特性的技术贡献。

1. MindExpression 为用户提供了 Python 编程范式，MindCompiler 提供基于中间表达的即时编译优化能力，他们具有以下三个突出特点。
   - 自动微分：采用基于源码转换的自动微分机制，在训练或推理阶段，将一段 Python 代码转换为数据流图。因此，用户可以方便地使用 Python 原生控制逻辑来构建复杂的神经网络模型。
   - 自动并行：由于大规模模型和数据集的不断增加，跨分布式设备并行化深度神经网络（Deep Neural Network，DNN）训练已经成为一种常见做法。然而，当前框架（如 TensorFlow、Caffe 和 MXNet）用于并行化训练的策略仍然简单直接，而且通常是次优的。MindSpore 并行化训练任务，透明且高效。“透明”是指用户只需更改一行配置，提交一个版本的 Python 代码，就可以在多个设备上运行这一版本的代码进行训练。“高效”是指该算法以最小的代价选择并行策略，降低了计算和通信开销。
   - 动态图：MindSpore 支持动态图，无须引入额外的自动微分机制（如算子重载微分机制），从而大大增加了对动态图和静态图的兼容性。
2. MindData 负责数据处理，并提供工具来帮助开发者调试和优化模型。通过自动数据加速技术实现了高性能的流水线来进行数据处理。各种自动增强方案使用户不必再寻找合适的数据增强策略。训练看板将多种数据集成在一个页面，方便用户查看训练过程。分析器可以打开执行黑匣子，收集执行时间和内存使用的相关数据，从而有针对性地进行性能优化。
3. MindArmour 负责提供工具，以帮助开发者防御对抗性攻击，实现隐私保护的机器学习。在形式方面，MindArmour 提供了以下功能：生成对抗代码、评估模型在特定对抗环境中的性能、开发出更健壮的模型，此外 MindArmour 还支持丰富的隐私保护能力。
4. MindRE 负责 AI 网络的执行，抽象和适配各底层硬件的操作接口，触发网络执行。支持端、云多种硬件环境的运行时系统

MindSpore 支持目前所有主流深度学习框架（如 TensorFlow、PyTorch、MXNet）中的模型，支持端、边、云全场景全栈协同开发，可以适用所有的 AI 应用场景，极大地降低了开发门槛，显著减少模型开发时间。它对本地 AI 计算的支持，更是解决了业界最为关注的隐私安全保护问题。MindSpore 一经面世就受到国内外的广泛关注，开源仅一年多就已成为中国最活跃的开源社区。目前，MindSpore 已累计下载超过 65 万次，吸引 17 万开发者，上线应用 2000 多个。并且在 8 大行业达到规模应用，在逾 100 家高校开设课程，超 1500 个核心开发者参与贡献，拥有 300 余篇基于 MindSpore 的科研创新顶会论文投稿。MindSpore 的开源生态正蓬勃发展。

表 1 MindSpore 的特性对实现三个目标的贡献

特性 / 目标	MindExpression & MindCompiler				MindRE	MindData			
	SCT AD	动态图	自动并行	图管理器	运行时系统	训练看板	分析器	自动增强	自动数据加速
易开发	✓	✓	✓	○	○	✓	○	✓	○
高效执行	✓	○	✓	✓	○	○	✓	○	✓
全场景覆盖	○	○	○	✓	✓	○	○	○	○

注：✓ 指主要贡献；○指次要贡献。

本文的其余内容如下：第 2 章介绍了 MindSpore 的架构；第 3 章介绍 MindSpore 的核心模块，具体阐述自动微分、自动并行、动态图、图算融合等创新技术的设计细节以及 MindSpore 在图神经网络领域的延伸；第 4 章主要分享 MindSpore 在开源生态方面的实践经验以及典型项目的落地成果；最后，对我们的工作进行了总结，并展望未来的重点工作方向。

2 MindSpore 概述

2.1 MindSpore 架构

MindSpore 的核心架构如图 1 所示。

MindExpression 提供 Python 接口，用于定义用户级应用软件编程接口（Application Programming Interface，API），构建和训练神经网络。由于 MindSpore 采用基于源码转换策略的自动微分机制，用户可以采用 Python 高效率的编程。

MindCompiler 是高性能执行和自动微分的核心。如果用户选择以 PyNative 模式运行，则逐个下发算子运行。如果以图模式运行，则 MindCompiler 会使用流水线生成计算图，通过解析 Python 代码，生成抽象语法树，然后将其转换为 A-Normal-Form（ANF）图。ANF 是图形化的，而不是语法化的，因此从算法上操作起来要容易得多。如果用户需要训练神经网络，流水线自动生成反向计算节点，并添加到 ANF 图中。流水线在构造整图之后会进行许多优化（例如内存复用、算子融合、常数消除等）。如果在分布式环境中训练模型，流水线将应用自动并行提供的优化策略。后端的虚拟机通过会话管理计算图，调用后端来运行图，并控制图的生命周期。

MindData 中的数据处理在训练过程中完成数据流水线，包括数据加载、数据论证、数据转换等。indData 也提供简单的 API，支持 CV/NLP/GNN 等全场景的数据处理能力。

MindInsight 提供训练看板、溯源、分析器和调试器。分析器可以打开执行黑匣子，以收集执行时间和内存使用的相关数据。调试器是图执行模式下的调试工具，用户可以在训练过程中查看图的内部结构和节点的输入 / 输出。MindInsight 对训练过程生成的日志文件进行分析，开发者可以轻松地可视化训练过程。

MindArmour 帮助用户开发更健壮的模型，保护用户在训练和推理数据中的隐私。MindArmour 中的对抗性攻击防御有三个主要模块：攻击、防御和评估。攻击模块是一个在黑盒和白盒攻击场景下生成对抗样本的工具。防御模块从攻击模块接收对抗样本，并在训练过程中用这些样本来提

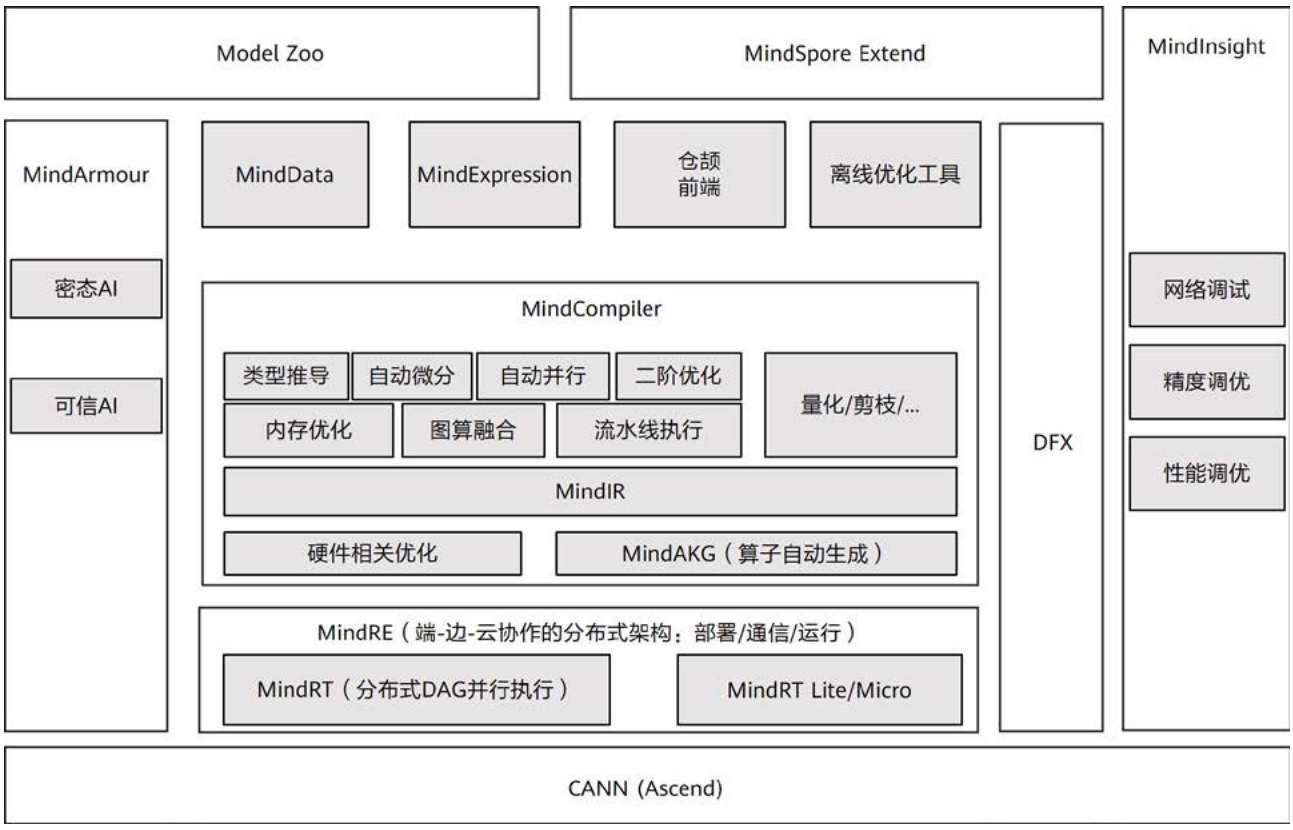


图 1 MindSpore 架构

高模型的鲁棒性。评估模块提供了多种评估指标，允许开发者更容易可视化其模型的鲁棒性。MindArmour 为了实现隐私保护机器学习，实现了一系列差分隐私感知优化器，这些优化器在训练过程中自动将噪声添加到生成的梯度中。

2.2 编程范式

MindSpore 为用户提供 Python 编程范式。借助基于源码转换的自动微分，用户可以使用原生 Python 控制语法和其他一些高级 API，如元组（Tuple）、列表（List）和 Lambda 表达。

为避免用户混淆，MindSpore 引入了尽可能少的接口和概念。在单机平台上训练简单的神经网络时，用户只需要了解以下五个组件：

- 张量（Tensor）：是一个包含相同数据类型元素的多维矩阵，张量与 NumPy 对象可以相互转换。与其他一些训练框架不同，MindSpore 中没有标量（Scalar）变量（Variable）的概念，而是根据张量的自动微分属性来判断是否需要对其求导。
- 数据集（Dataset）：是一个单独的异步流水线，该流水线为在训练中将张量无延迟地馈入网络的剩余部分做准备。
- 算子（Operator）：是构成神经网络的基本计算单元。MindSpore 支持大多数常用的神经网络算子（如卷积、批标准化、激活等）和数学算子（如加法、乘法等）。此外，还支持自定义算子，允许用户添加新的算子来适配特定的硬件平台，或者合并现有算子来生成复合算子。
- 单元（Cell）：是张量和算子的集合，是所有神经网络单元的基本类。一个单元也可以包含其他单元，允许它们嵌套在同一个树状结构中。用户通过在单元中定义 construct 函数来表达神经网络的计算逻辑，该函数中的计算将在每个调用中执行。
- 模型（Model）：是 MindSpore 中的一个高级 API，便于用户能够高效地使用推理和训练功能。如果用户熟悉低级 API，并希望对计算过程建立细粒度的控制，则不一定需要使用模型。

从用户的角度来看，用 MindSpore 编写程序的核心是构建一个对应于神经网络的单元。这个过程首先从输入张量开始；然后采用框架提供的不同算子构造一个单元；最后，用户可以使用模型封装这个单元来训练神经网络，也可以直接将输入数据传递给单元来执行推理任务。

3 MindSpore 核心模块与创新技术

3.1 MindExpression 和 MindCompiler

3.1.1 基于源码转换的自动微分

不同于 TensorFlow 和 PyTorch 分别基于静态和动态计算图的转换方式，MindSpore 开发了基于源码转换的自动微分新策略。该技术从函数式编程框架演化而来，对中间表达（程序在编译过程中的表达形式）以即时（Just-In-Time，JIT）编译的形式进行自动微分变换（如图 2 所示）。一方面，它支持流程控制的自动微分，因此可以像 PyTorch 一样方便的构建模型。另一方面，MindSpore 可以对神经网络进行静态编译优化，从而获得像 TensorFlow 一样良好的性能。该策略同时避免了动态图模式难以进行性能优化的困难，以及静态图模式组建或调试网络复杂的缺陷。

MindSpore 的自动微分可以理解为对程序本身进行符号微分，因为 MindSpore IR 是函数式的中间表达，它与基本代数中的复合函数有直观的对应关系，只要已知基础函数的求导公式，就能推导出由任意基础函数组成的复合函数的求导公式。MindSpore IR 中每个原语操作可以对应为基础代数中的基础函数，这些基础函数可以构建更复杂的流程控制。

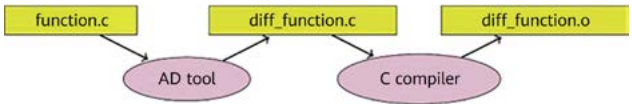


图 2 基于源码转换的自动微分

3.1.2 自动并行

随着深度学习的发展，为了实现更高的准确率和更丰富的应用场景，训练数据集和深度神经网络模型的规模日益增大。在大型数据集上训练大型模型，不仅需要深度学习框架支持数据并行和模型并行，还需要支持混合并行。诸如 TensorFlow [5]、Caffe [7] 以及 MXNet [8] 等框架均通过手动切分深度神经网络模型来实现模型并行，该方式切分难度大，需要有丰富经验的专家来操作。实现混合并行（数据并行和模型并行同时进行）又极大增加了开发的复杂度。

MindSpore 支持在模型训练过程中数据并行与模型并行间的转换。通过在并行化策略搜索中引入张量重排布（见图 3），使得输出张量的设备布局在输入到后续算子之前能够被转换。在此基础上，定义相应的反向算子来实现对通信算子的求导，从而使得自动微分过程可以一次性地微分整个

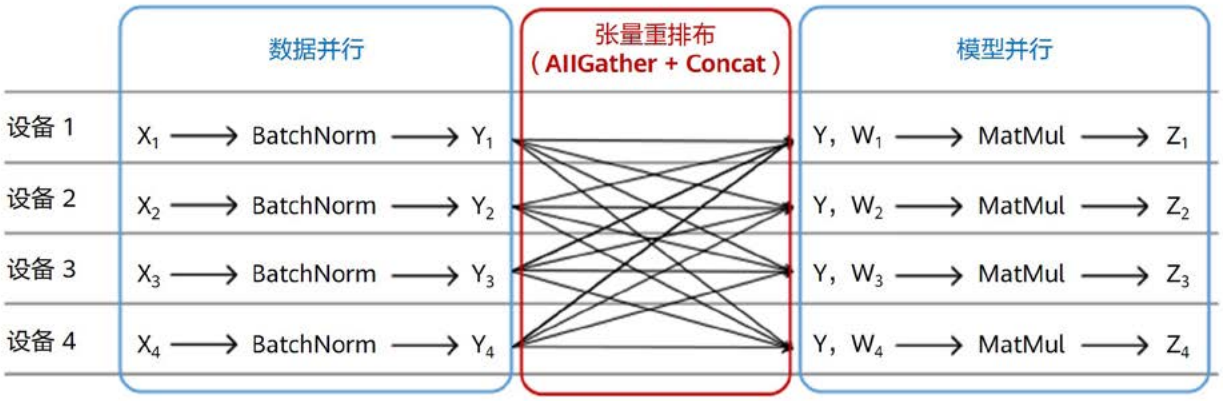


图 3 数据并行性向模型并行性转换

前向图。另外，MindSpore 在同时考虑计算和通信开销的情况下，建立代价模型来选择更优策略。一种是支持多图操作的算法，将原始图转换成线性图；一种是策略分离机制，在保证返回解的精度的同时，有效地缩小搜索空间。

MindSpore 的并行实现非常灵活，它支持用户指定的高级策略配置，称之为半自动并行。为避免在训练新模型时多次配置策略，耗时耗力，MindSpore 的自动并行机制不需要人工干预即可找到一个有效的策略。

3.1.3 动态图

静态图由于编译器能获得其全局信息，在大多数情况下都表现出更好的运行性能。而动态图可以保证更好的易用性，使用户能够更加方便地构建和修改模型。为了同时支持静态图和动态图，多数框架需要维护基于 Tape 和基于图的两种自动微分机制。该方式维护成本较高并且模式间的切换也相当复杂。

MindSpore 采用了基于源码转换的自动微分机制，同时支持静态图和动态图。两种模式间的切换只需要一行代码，高效易用。在静态图模式中，框架将神经网络模型编译成一张整图，然后下发执行。该模式利用图优化等技术提高运行性能，同时有助于规模部署和跨平台运行。而动态图模式则将神经网络中的各个算子逐一下发执行，方便用户编写和调试神经网络模型。在该模式下，MindSpore 还支持 staging 功能的混合运行机制。在指定函数前添加 ms_function 装饰器，该函数将被编译成计算图从而以静态图模式运行，显著提高计算效率。

3.1.4 二阶优化

优化器作为反向传播算法的核心部件，是深度学习中非常重要的一部分。常见的优化算法可分为一阶和二阶优化算法，其中梯度下降法是机器学习中最经典和最常用的一阶优化算法。在其基础上的诸多改进算法，例如 Momentum [11]、AdaGrad [12] 和 Adam [13] 等，可以利用随机梯度的历史信息来自适应地更新步长，使得调参更为容易且方便使用。

由于神经网络的损失函数是高度非凸函数，而且曲面的曲率不平衡，因此使用更多的信息来指导参数更新会得到更好的收敛速度。二阶优化算法利用目标函数的二阶导数进行曲率校正来加速一阶梯度下降。其收敛速度更快，能高度逼近最优值，几何上的下降路径也更符合真实的最优下降路径。与一阶优化算法相比，二阶优化算法基于二阶信息矩阵的逆矩阵进行权重更新。常见的二阶优化算法有牛顿法，自然梯度法等，其对应的二阶信息矩阵分别为 Hessian 矩阵和 Fisher 矩阵。

二阶优化算法虽然收敛速度快，但是对其求逆的时间复杂度极高。在深度学习中，模型参数通常至少在数百万的量级，此时逆矩阵无法计算。因此降低信息矩阵求逆的计算复杂度成为二阶优化算法有效性的关键。

MindSpore 针对该问题，提出了自研算法 THOR（Trace-based Hardware-driven layer-oriented Natural Gradient Descent Computation）[14]，THOR 基于自然梯度法对二阶算法进行如下三点改进：（1）调整矩阵更新频率。Fisher 矩阵在前期变化剧烈，后期逐步稳定并最终收敛到稳态分布，其演化可视为马尔可夫过程。因此在训练过程中逐步增大 Fisher 矩阵的更新频率，可以在不影响收敛速度的情况下，减少训练时间。（2）分层更新。将 Fisher 矩阵按层解耦，可以发现不同层的 Fisher 矩阵趋于稳态的速度不同。基于此，使用二阶信息矩阵的迹作为判断条件，可以更加细粒度的调整每一层的更新频率。（3）硬件感知矩阵切分。THOR 假设 Fisher 矩阵各层之间解耦并且每个网络层中的输入和输出块之间也是独立的。根据该假设对二阶信息矩阵做进一步的切分，再次提高了计算效率。

3.2 图算融合

随着 AI 的发展，深度神经网络模型结构快速演化，如 GPT-3 [15] 等大模型逐渐成为新的趋势。该类模型对运行性能和效率的要求也越来越高。基于人工优化的方式已无法满足日益增长的需求。为此，业界 AI 框架进行了大量的探索，诸如 XLA（Accelerated Linear Algebra）[16] 和 TVM（Tensor Virtual Machine）[17] 等编译优化技术已取得了许多优秀的成果。

已有实践结果 [18] 表明基于自动算子编译的图层和算子层联动融合优化是提高模型运行性能和效率的重要手段。MindSpore 从这点出发提出了图算融合技术，助力模型在框架运行中获得极致性能。图算融合，目的是联合图层与算子两个阶段，让图层阶段计算图结构的变换更有利于高效算子的生成，同时让算子生成的规律反向指导图层变换的进行。图算融合主要从访存融合、并行融合以及跨边界算子计算优化三个维度着手联动起来实现计算图极致的性能表现。

访存融合

计算图由算子构成，算子间通过数据或控制关系联系起来，从而保证算子的计算完整性和计算独立性，也增加了相对应的数据访存需求。一方面，计算完整性与独立性，让针对性的算子优化得以实现，但同时也舍弃了更大范围的算子优化可能。另一方面，如果算子的输入和输出数据的数据量比较大，则访存在执行总耗时上占比高，数据访存将成为性能上的瓶颈。访存融合，通过访存关系吸纳整合基础算子，择劣处理复合算子，通过对前后算子进行融合来实现更优的访存融合优化。该方法可以有效地减少访存数，提高计算密度，是一种重要的优化手段。

为了让算子边界所能表达的计算逻辑更多，算子边界的数量更少，图算融合先尽可能多地去融合算子，同时以合理有效的方式对融合后的巨大边界进行拆解，重新划分出新的边界。合理地设计新边界，可以通过必要的访存损耗，赢取更多的优化机会。MindSpore 中的 AKG（Auto Kernel Generator）基于 Polyhedral 技术实现了优质的自动调度，其中包括自动向量化、自动切分、Thread/Block 映射、依赖分析和数据搬移等，并针对后端进行包括 Tensor Core 使能、双缓冲区、内存展开和同步指令插入在内的各项优化。给定具体的算子描述，AKG 将通过系列分析生成优质的可执行算子。AKG 为图算融合提供了自动算子生成的能力，也为算子性能的优质性提供了保障。

更优的算子需要图层变换的协作配合。AKG 一方面保证了算子的性能，另一方面 AKG 生成算子的经验和对后端优化的思考也对图层重新划分算子边界起到导向性作用。图算融合针对算子边界形成的访存和计算优化问题，对计算图的算子进行先聚合、再拆分，以这种破而后立的方式，让计算优化可以跨越原始算子边界进行。降低了单一计算逻辑带来的优化局限性的影响，并且不一味的扩大聚合范围，依据评估结果生成收益更高的必要边界，让访存成为提升性能的手段。同时 AKG 自动生成高效可执行算子的能力，让图算融合突破了传统一对一的手工算子融合的限制，让融合的形式更灵活有效。图算融合的访存融合使计算图中算子的计算密度及计算有效性得到有效提升，同时减少了算子边界间低效的访存操作，进而改善全图执行性能。

并行融合

为了让计算图能够正确的执行，通常需要为各个计算任务分配执行顺序，也因此被隐式地添加上了执行上的依赖关系。如果大量细碎的计算任务被加载到设备中执行，则将可能让设备在大多数情况下处于空闲状态，硬件没有得到完全使能，造成计算资源的浪费。为此图算融合提出了并行融合：分析计算图中算子间的依赖关系，将存在并行性质且对硬件资源使用率低的算子融合在一起。在生成该并行算子时，对各部分独立分析调度，并安排相互独立的计算资源。并行融合在保留生成算子优质性的同时，将计算资源使用率小的算子捆绑成一个大算子，提高设备的使用率，减少因空置造成的计算资源浪费，提升整体执行性能。

跨边界算子计算优化

粗放式聚合起来的计算逻辑状态并不是最优的，在进一步拆解前需要进行计算逻辑上的优化。聚合前存在的一些优化壁垒，例如：在未聚合时，由于算子边界的存在，算子只能完整引入。因此在构建深度学习网络时，为了其中部分输出的结果而引用了复合算子，会将其中不需要的冗余操作也一并带入到网络中来；不同复合算子的操作逻辑间含有相似的操作，在未进行聚合前，这些操作逻辑都是不可分割合并的。这些问题，随着计算逻辑的裸露和聚合，相互间的关联变得可见，优化便变得可能。在聚合起来的计算块中，通过公共子表达式消除、死代码消除等手段可以提高计算逻辑效率，减少无效操作。

3.3 MindData

MindData 的数据处理是一个单独的异步流水线，它为张量馈入模型做好准备。在流水线内，数据被组织成一系列具有不同列的行，所有列都用列名标识，并且可以独立访问。流水线总是从一个源数据集算子开始，该算子从磁盘（如 MindDataset）读取数据，并包含选择 shuffling 和 sharding 策略的标记。为了访问流水线中的数据，可使用迭代器（Python 访问）或设备队列（直接发送到加速器设备）。

数据处理的体系结构本质上是流水线且并行的，管道的运行默认是异步的，但用户也可以在图中插入同步点，以便流水线算子能够实时反馈循环。流水线将支持动态调整，

以充分利用所有可用的资源，包括用于图像处理的硬件加速器或用于缓存的可用内存，用户只需配置默认参数以获得良好的性能，而无须进行手动调优。

为了让用户实现快速迁移，MindData 的数据处理支持用户已有的 Python 数据增强作为自定义算子接入，同时现有的 Python 数据集类可以作为参数传递给自定义数据集接口接入。不断涌现的新网络对数据处理的灵活性提出了新要求。数据处理支持使用用户自定义的函数（或调度）来更改参数，例如 mini-batch 大小；还支持用户对整个 mini-batch 进行自定义转换，以进行 mini-batch 级的图像大小或多行操作（如图像混合）。为了获得更多样的增强，每个样本的增强可以从一个数据增强操作列表中随机选择。另外，可以通过外部搜索在候选数据增强操作列表中进行选择，实践结果表明，只随机从大量数据增强策略中选择合适的，而不采用耗费大量搜索时间的自动数据增强方法同样能获得很好的结果。在训练中收集度量的反馈（如 loss）也可以传回至数据集，以在数据处理中执行动态调整。

MindRecord 数据集格式将用户的训练数据按不同类型、不同页面进行存储，建立轻量化、高效的索引，并提供一套接口，方便用户将训练数据转换成 MindRecord 格式，然后使用 MindDataset 完成训练数据的读取。同时，可以快速从数据集读取重要的元数据（即数据集大小或数据布局）以提高性能或简化用户访问。MindRecord 能够支持小块数据的高效顺序 I/O，还可以根据用例需要支持高效的随机行访问和下沉过滤。随着新用例的出现，进一步优化的功能会下沉到该数据集。

3.4 MindInsight

MindInsight 是 MindSpore 的可视化调试调优工具，可以可视化地呈现训练过程、优化模型性能、调试精度问题、解释推理结果。还可以通过其提供的命令行方便地搜索超参，迁移模型。在 MindInsight 的帮助下，开发者可以更好地观察和理解训练过程，这样可以提升模型优化效率并优化开发者体验，从而更轻松地获得满意的模型精度和性能。

MindInsight 以模型训练中生成的日志文件作为输入。通过文件解析、信息提取、数据缓存、图表绘制等一系列过程，将二进制训练信息转换成易于理解的图表，并展示在网页上。MindInsight 通过训练看板支持训练过程可视化。训练看板包括训练标量信息、参数分布图、计算图、数据图、数据抽样等模块。训练看板是 MindInsight 在训练过程呈现方式上的创新。通过在一个页面中集成多种类型的数据，用户只需要打开训练看板就能概览训练情况。图 4 展示了训练看板的一个示例。

MindInsight 还支持溯源可视化。通过将多个训练的溯源信息整合到表格和图形中，用户可以轻松选择最优的数据处理流水线和超参设置。溯源可视化包括模型溯源可视化和数据溯源可视化。模型溯源函数可以记录模型训练的关键参数，如损失函数、优化器、迭代次数、精度等。MindInsight 中显示多次训练的超参和评估指标，帮助用户选择最优超参。

为了满足工程师对神经网络性能优化的要求，MindInsight 设计并实现了性能分析工具。它可以打开 MindSpore 神经网络的执行过程，收集各算子的时间、内

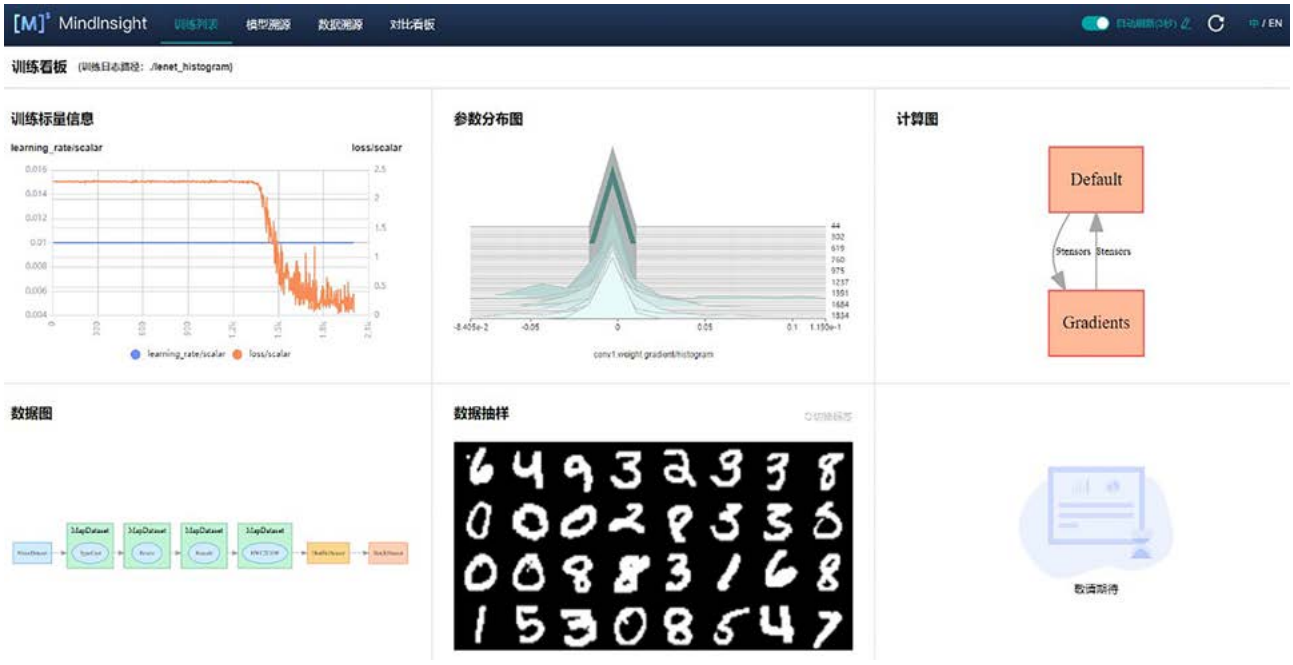


图 4 MindInsight 训练看板

存等数据。MindInsight 会进一步对性能数据进行整理、分析，呈现多维度、多层次的性能分析结果。性能数据展示维度包括：迭代轨迹、算子性能、时间轴、MindData Profiling。这为神经网络性能优化提供了方向。

神经网络训练中经常出现如无穷大等数值误差情况，用户希望分析无法收敛的原因。MindInsight 调试器是图模式下训练调试的工具，该工具克服了由于计算被封装为黑盒在图执行时难以定位错误的困难。用户可以在训练过程中查看图的内部结构以及节点的输入 / 输出，例如查看一个张量的值，查看图中的节点对应的 Python 代码等。此外，用户还可以选择一组节点设置条件断点，实时监控节点的计算结果。

3.5 MindArmour

对抗性攻击 [19，20] 对机器学习模型安全的威胁日益严重。攻击者可以通过向原始样本添加人类不易感知的小扰动来欺骗机器学习模型 [21，22]。为了防御对抗性攻击，MindArmour 模块提供了攻击（对抗样本生成）、防御（对抗样本检测和对抗性训练）、评估（模型鲁棒性评估和可视化）等功能。

给定模型和输入数据，攻击模块提供简单的 API，能够在黑盒和白盒攻击场景下生成相应的对抗样本。这些生成的对抗样本被输入防御模块，以提高机器学习模型的泛化能力和鲁棒性。防御模块还实现了多种检测算法，能够根据恶意内容或攻击行为来区分对抗样本和正常样本。评估模块提供了多种评估指标，开发者能够轻松地评估和可视化模型的鲁棒性。

隐私保护也是人工智能应用的一个重要课题。MindArmour 考虑了机器学习中的隐私保护问题，并提供了相应的隐私保护功能。针对已训练模型可能会泄露训练数据集中的敏感信息问题 [23，24]，MindArmour 实现了一系列差分隐私优化器，自动将噪声加入反向计算生成的梯度中，从而为已训练模型提供差分隐私保障。特别地，优化器根据训练过程自适应地加入噪声，能够在相同的隐私预算下实现更好的模型可用性。同时提供了监测模块，能够对训练过程中的隐私预算消耗进行动态监测。用户可以像使用普通优化器一样使用这些差分隐私优化器。

3.6 端云协同

MindSpore 旨在构建一个从端侧到云侧全场景覆盖的人工智能框架，支持“端云”协同能力，包括模型优化、端侧训练和推理、联邦学习等过程，如图 5 所示。

模型生成与优化工具包

移动和边缘设备通常资源有限，如电源和内存。为了帮助用户利用有限的资源部署模型，MindSpore 支持一系列优化技术，如模型自适应生成、量化策略等，如图 5 左侧所示。模型自适应生成是指应用神经架构搜索技术 [25] 来生成在不同设备下时延、精度、模型大小均满足需求的模型。量化策略是指通过以更少位数的数据类型来近似表示 32 位有限范围浮点数据类型的过程，MindSpore 支持训练后量化和量化感知训练。

端侧训练和联邦学习

虽然在大型数据集上训练的深度学习模型在一定程度上是通用的，但是在某些场景中，这些模型仍然不适用于用户自己的数据或个性化任务。

MindSpore 提供端侧训练方案，允许用户训练自己的个性化模型，或对设备上现有的模型进行微调，同时避免了数据隐私、带宽限制和网络连接等问题。端侧将提供多种训练策略，如初始化训练策略、迁移学习、增量学习等。

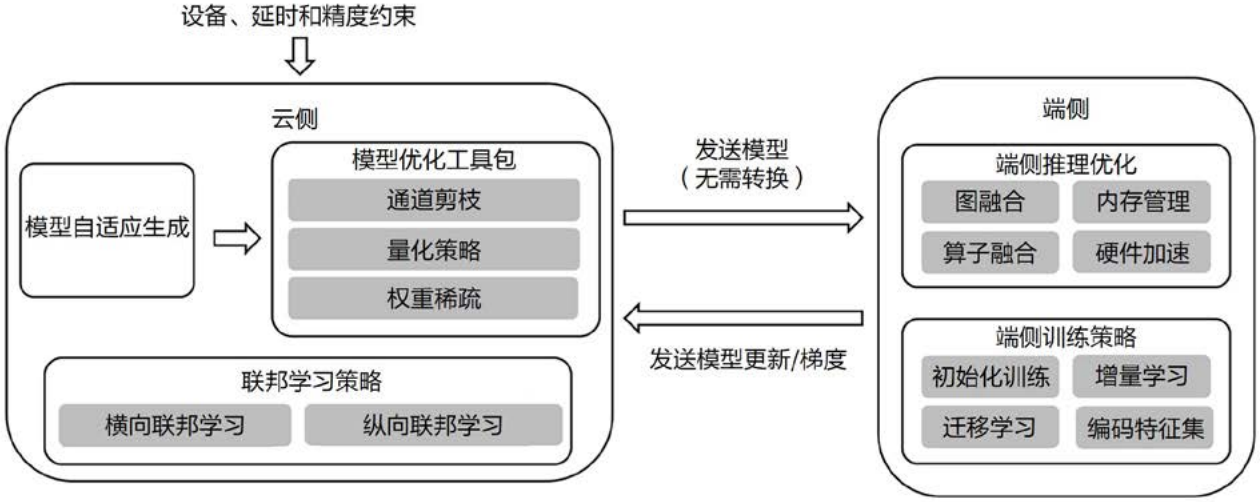


图 5 MindSpore 端云协同架构

MindSpore 还支持联邦学习，通过向云侧发送模型更新 / 梯度来共享不同的数据。基于联邦学习，模型可以学习更多的通用知识。

移动和边缘设备部署

MindSpore 提供轻量级的计算引擎，支持模型在设备上高效执行。在将预先训练好的模型部署到设备侧时，通常需要进行模型转换。然而，这个过程可能导致性能降低和精度损失。在 MindSpore 中，端侧推理模式能够兼容云上训练好的模型，因此，在设备上部署已经训练好的模型时，无须进行转换，这样避免了潜在的性能损失。此外，MindSpore 还内置了针对设备的各种自动优化，例如图和算子融合、精细复杂的内存管理、硬件加速等，如图 5 右侧所示。

3.7 MindSpore Serving

MindSpore Serving 是一个轻量级、高性能的推理服务模块，旨在帮助 MindSpore 开发者在生产环境中高效部署在线推理服务。当用户使用 MindSpore 完成模型训练后，导出 MindSpore 模型，即可使用 MindSpore Serving 创建该模型的推理服务。

MindSpore Serving 提供如下功能：加载模型文件生成推理引擎，提供推理功能；预测请求和处理结果的消息交互，支持 gRPC 和 RESTful 两种请求方式；预测接口调用，执行预测，返回预测结果；模型的生命周期管理；服务的生命周期管理；多模型多版本的管理。

3.8 图神经网络框架

图神经网络（GNN）在工业界和学术界都引发了极大兴趣。应用场景覆盖了药物识别与发现，推荐系统，交通流量预测，芯片设计等等领域。与卷积神经网络一样，GNN 模型也在快速迭代中，然而现有的 GNN 框架（如 DGL [26]、PyG [27]、PGL [28]、GraphLearn [29] 等）均无法提供一个简洁易用，更贴近 GNN 算法的编程模型，并且训练速度也难以令人满意。因此，MindSpore 联合香港中文大学提出了新的 GNN 框架，尝试从更易用，执行更快的角度寻求突破。该框架的整体架构如图 6 所示，其核心是以节点为中心的编程模型和深度图学习编译优化算法。

GNN 相较于计算机视觉任务的一个显著区别是用户需要在给定的图结构（由边，点组成）上做信息的传递和聚合。现有框架针对此类操作无法支持类似 Python 语法的直观表达方式。用户必须依赖 GNN 系统提供的框架，如 message-passing 来实现这样的聚合方式。这种以张量为中心的编程模型，相比于直接的 Python 实现有着较大的差距，给用户造成了陡峭的学习曲线。

MindSpore 的 GNN 框架实现了以节点为中心的编程模型。这种表达方式最大程度的贴合了用户的思考和使用习惯，使得编写 GNN 模型像编写普通 Python 程序一样简单。以节点为中心的编程模型极大降低了用户开发新模型的门槛，方便用户快速对模型进行实现和迭代。该方法和基于 message-passing 的方式具有同等表达能力，可以支持当前已有的任意 GNN 模型。基于此编程模型，MindSpore 迅速复现了如今最流行的 GNN 框架 DGL [26] 的卷积库和模型库。目前已支持了 10+ 经典模型，覆盖了从同构模型，异构模型，推荐模型，知识图谱，到生命科学等多个领域，充分证明了该编程模型的易用性和强大的表达能力。

支持以节点为中心的编程模型面临着以下挑战：第一，将编程模型与现有的 MindSpore 代码进行融合，做到两种不同编程模型代码的无缝衔接。第二，支持 MindSpore 现有的特性如自动并行，动态图静态图的自由切换，计算图编译，甚至版本演化出的新特性。第三，减少对 MindSpore 现有框架的侵入。为了解决这三个挑战，MindSpore 的

图 6 MindSpore 图神经网络架构

GNN 框架创新地提出了利用源到源转换的方法，将用户代码转换为 MindSpore 原生支持的代码。用这种方法不同编程模型都以 Tensor 进行数据交换，可以无缝衔接。并且源到源转换的结果是正常的 MindSpore 代码，保证了对 MindSpore 框架所有特性的兼容性，又不需要对现有框架做任何修改。GNN 框架还提供精确到行的代码转换对照工具，从而方便用户对生成的代码进行检查。

GNN 任务具有内存密集的特点，核心开销在于大量的张量在显存 – 缓存 – 核心层级间的搬运。减少内存搬运的传统做法是算子融合。现有 GNN 框架中的算子融合优化方案是通过人工挑选常用的 GNN 算子组合，用手写算子组合前向和反向的方法来进行逐点优化。一方面，这种方式通用性较差，新的 GNN 算子组合方式需要专家进行手动编写和优化；另一方面，它只利用了部分算子融合的机会，性能一般。

对于 GNN 任务，无论是同构图还是异构图，通过 Scatter/Gather 转化为 MindSpore 系统计算后，都存在 Gather–Injective（包括 +/–/*/÷，激活函数）–Scatter（GIS）执行模式。为了进一步提升性能同时增强可扩展性，MindSpore GNN 基于框架已有的图算融合和自动算子编译技术，提出了新的算子融合优化模式 GIS。GIS 可以将一个或多个 Gather 算子，Injective 算子，Scatter 算子融合成一个算子。算子融合之后，在 AKG 算子编译层面，通过 polyhedral 技术进行自动的算子优化和生成。对于大规模数据来说，融合后的计算图中会有一些算子具有超大输出结果，由于融合算子往往计算量较小，瓶颈在内存读取，所以可以通过重计算提高运行速度，从而可以减少显存使用。通过减少张量在显存内的实例化，MindSpore 的 GNN 框架比现有最流行框架 DGL 具有平均 3 到 4 倍的性能提升。此外，这个融合模式除了能够覆盖现有框架中已有的算子融合优化，还能够覆盖大量新的算子组合。

4 MindSpore 生态

MindSpore 自开源以来已经发布 12 个版本，在短短一年多的时间已经快速成长为国内最活跃的开源社区。除了技术创新的快速迭代，MindSpore 的生态布局也在迅速铺展开来。MindSpore 生态通过高校、比赛、科研、企业四方面的深度融合，相互促进，以 AI 产业聚集人才，人才又激发 AI 科研创新，创新驱动 AI 产业发展，这样形成 MindSpore 在 AI 产业的“产学研赛”融合的正循环。

如图 7 所示，高校是人才培养的根据地，是 AI 人才发展的未来，因此以高校教学作为 MindSpore 生态的起点。目前中国教育部正全面深化新工科建设，培养引领 AI 科技创新和产业变革的人才，打造世界科学中心和人才高地。因此加深与高校合作，通过高校开始孵化打通“产学研赛”，构建 AI 产业驱动的知识体系建设，结合“产学研赛”的人才体系建设，为 AI 产业持续创新提供源源不断的人才。除了通过与学术界的紧密联系在技术前沿进一步打造自己的影响力，MindSpore 积极推进产业结合，已经在医疗、交通、金融、互联网、运营商、制造、能源等行业广泛应用落地，并帮助传统行业客户解决 AI 模型研发效率低、AI 应用落地门槛高等问题。MindSpore 剑指 AI 基础设施，并向上发展应用生态，向下夯实硬件能力，积极发挥发挥国内 AI 框架的全球影响力。

下面本文就 MindSpore 在学术、商业生态以及社区运营方面的实践展开介绍。

4.1 学术生态

MindSpore 通过新工科、智能基座、沃土计划三个计

图 7 MindSpore 生态的“产学研赛”融合

划作为进入高校教学起点的抓手，对高校老师提供教学开发、师资培训等计划，赋能老师更好地给学生授予 AI 的知识，用 MindSpore 来实践。学生学习完 MindSpore 之后，低年级的本科生可以参与到社区的开发者活动比赛，国家教育部认可的互联网 + 比赛或者华为 ICT 大赛中 MindSpore 创新实践赛道当中，做到学以致用。针对偏向于学术研究的学生，MindSpore 跟中国人工智能基金 CAAI 合作，驱动高校使用 MindSpore 进行联合创新，并给高校提供科研难题的技术、资金、硬件设备、算力的支持和支撑。

高校秋季开课迎来 MindSpore 大规模教学落地

在高校教学方面，MindSpore 在成都、武汉、上海等 8 个城市完成 10 场师资培训，在包括清华北大在内的国内 101 所 TOP 高校完成 416 门课程开课，覆盖 AI 领域老师 1000+、学生 3.5 万 +，覆盖人工智能、计算机科学等重点学科，覆盖《机器学习》《深度学习》《计算机视觉》《自然语言处理》等典型课程；秋季学期待开课课程 159 门，涉及老师 200+、学生 1 万 +；组织 10+ 场 MSG 校园行覆盖 5000+ 师生（见图 8）。实现了生态在教学方面的突破，让更多的学生学会 MindSpore，让更多的老师会教 MindSpore，AI 从高校启航。

互联网 + 大赛 MindSpore 参赛踊跃，华为 ICT 启动

MindSpore 在高校的推广主要以育人为入口，学生学习完 MindSpore 最好的方式就是锻炼，实操。“实践是检验真理的唯一标准”，MindSpore 在各项大型 AI 领域比赛中设置相关赛题，吸引上万学生学习、使用 MindSpore，参与到开源社区中来（见图 9）：

1. 互联网 + 大赛产业赛道，有 170 个团队报名 MindSpore 相关赛题，其中 13 个团队获得省赛金奖，成功进入全国前 300 强、入围全国赛，4 支队伍进入全国前 50 强，并将在线下参与全国金奖角逐；
2. 绿盟杯比赛报名队伍 3000+，当前正在进行黄金赛段激烈比拼，将选出最优秀的 8 支队伍代表 MindSpore 在 11 月 19~20 西安 CCF 大会现场争夺年度总冠军；
3. 开源之夏顺利进行，MindSpore 相关项目申请团队 200+，当前已陆续顺利结项，相关代码合入社区，培养核心贡献者 30+；
4. 华为 ICT 大赛、CCF-BDCI 大赛、信通院 AI+ 遥感和 AI+ 制造等赛事 MindSpore 相关赛道同步进开启中，大量团队参与火热。

40+ 科研合作项目落地 170+ 论文

MindSpore 坚持 AI 生态与技术两手抓策略，通过科研合作和论文创新在原创 AI 生态领域进行了深入布局。在科研合作领域，与 20+ 国内外顶尖高校和科研院所展开了科研项目合作来拓展 MindSpore 竞争力。围绕 AI+ 科学计

图 8 MindSpore 高校教学

算、万亿参数超大模型、AI 安全等领域构建差异化竞争力，发布了一系列创新性研究成果如遥感模型、终端电磁仿真模拟及多模态大模型等。

在论文创新方向，MindSpore 与中国人工智能学会共同发起了《中国人工智能学会 – 华为 MindSpore 学术奖励基金》。面向高校及科研院所的 AI 科研人员搭建学术交流平台，提供经费、算力、技术支持等服务。推动 MindSpore 在学术领域的应用，并支持基于 MindSpore 框架的国际国内高水平会议和期刊的学术论文发表。通过激励原创性科学研究开展，构建中国人工智能科学研究的全球影响力。

目前，已和 45+ 高校老师建立起合作关系，合作领域覆盖 CV、NLP、科学计算、生物、无人驾驶等热门 AI 方向。基于 MindSpore 落地 170+ 顶会论文，助力 AI 研究人员基于 MindSpore 发表一系列 Top 级影响力成果。

4.2 商业生态

今年以来，人工智能计算中心快速上线，鹏城、武汉等地先后上线运营，西安、成都、南京等地也即将上线运营，为各地高校和企业提供普惠算力。依托当地计算中心算力，国内外高校及科研机构基于 MindSpore 联合创新，打造中国 AI 开发生态，赋能产业实践。

首个全开源 2000 亿参数中文预训练语言模型 – 鹏程 . 盘古

在华为生态大会，鹏城实验室超大模型「鹏程 . 盘古」重磅亮相。这是业界首个全开源 2000 亿参数中文预训练语言模型。鹏程 . 盘古基于「鹏城云脑 Ⅱ」和全场景 AI 计算框架 MindSpore 的自动混合并行模式，实现在 2048 卡集群上的大规模分布式训练，是国产全栈式 AI 基础设施第一次支持 2000 亿级超大规模语言模型训练。探索并验证了国产 E 级智算平台在软硬件协同优化、大规模分布式并行训练等核心关键技术上的可行性。

大规模的分布式训练是大模型开发的难点。MindSpore 为超大模型的分布式训练提供了解决方案，提供了多维混合并行方式：数据并行、算子级模型并行、Pipeline 模型并行、优化器模型并行和重计算。在图编译阶段，有机融合了多维度的并行。

鹏程 . 盘古支持丰富的应用场景，在知识问答、知识检索、知识推理、阅读理解等文本生成领域表现突出。目前，鹏程 . 盘古性能全球领先。16 个下游任务中性能指标优于业界 SOTA 模型，其中零样本学习任务 11 个任务领先、单样本学习任务 12 个任务领先、小样本学习任务 13 个任务领先。

联合武汉大学打造全球首个遥感专用框架，共同推进遥感产业化应用

遥感对地观测技术在国土资源规划、自然环境监测等领域广泛应用。武大遥感专业在全球处于领先地位，但由于领域性质特征，遥感影像解译比通用图像识别问题更为复杂，目前遥感测图任务大多依赖人工解译，急需结合 AI 来解决这项难题。同时，遥感影像处理的深度学习技术，亟需大规模的遥感影像样本库，以及具有遥感特性的深度学习框架和模型来进行支持，这也是产业界的迫切需求。武汉大学与昇腾 +MindSpore 协同打造全球首个遥感特性的深度学习框架 LuojiaNet 和样本库 LuojiaSet（见图 10）。

全场景 AI 框架 MindSpore 支持超大分辨率图片处理及超大图像输入切分到多机多卡并行运行，解决图像的切分图片的边界计算问题。遥感影像专用框架 LuojiaNet 是基于 MindSpore 构建，为国内遥感科研创新和产业化提供专用框架。可以处理 30K*30K 超大分辨率，最大支持 256 通道波普的遥感影像，抽象遥感知识图谱等领域特性。围绕遥感样本库、专用框架，模型等，武汉将打造相关遥感产业，如光谷信息、立德空间等。

图 9 MindSpore 互联网 + 大赛

联合中科院打造多模态通用人工智能平台，开发业界首个三模态大模型

自 GPT/BERT 模型提出后，预训练模型迎来了爆发式发展，其具有在无监督情况下自动学习不同任务、并快速迁移到不同领域的优势。多模态预训练模型被广泛认为是从限定领域的弱人工智能迈向通用人工智能的路径探索。现阶段，音视频数据呈高速增长，占比超过 80%。纯文本的预训练模型只涵盖较少部分，更丰富的语音、图像、视频等数据并未被充分利用与学习。人类的信息获取、环境感知、知识学习与表达，都是通过多模态信息方式来执行的。

中科院自动化所提出了视觉 – 文本 – 语音三模态预训练模型，基于 MindSpore 框架成功打造出了全球首个三模态预训练大模型“紫东 . 太初”（见图 11）。该模型同时具备跨模态理解与跨模态生成能力，实现了以图生音、以音生图的技术和表现，图文音语义的统一表达。三模态大模型采用百 TB 级三模态数据，百卡级昇腾集群分布式训练。MindSpore 为多模态大模型训练提供了数据并行、算子级模型并行、Pipeline 模型并行、优化器模型并行、异构并行、重计算、高效内存复用等多维度、全种类的分布式并行策略。大幅缩短模型在分布式训练上的代码开发、调试和训练的周期，极大减少了系统性能优化的代价。

紫东太初大模型将文本 + 视觉 + 语音各个模型高效协同，实现超强性能，在图文跨模态理解与生成性能上都能领先目前业界的 SOTA 模型，高效完成跨模态检测、视觉问答、语义描述等下游任务。视频理解与描述的性能更是实现了全球第一。在 2021 年的两个国际大赛中，ACM Multimedia（国际多媒体大会）和 ICCV（国际计算机视觉大会）紫东

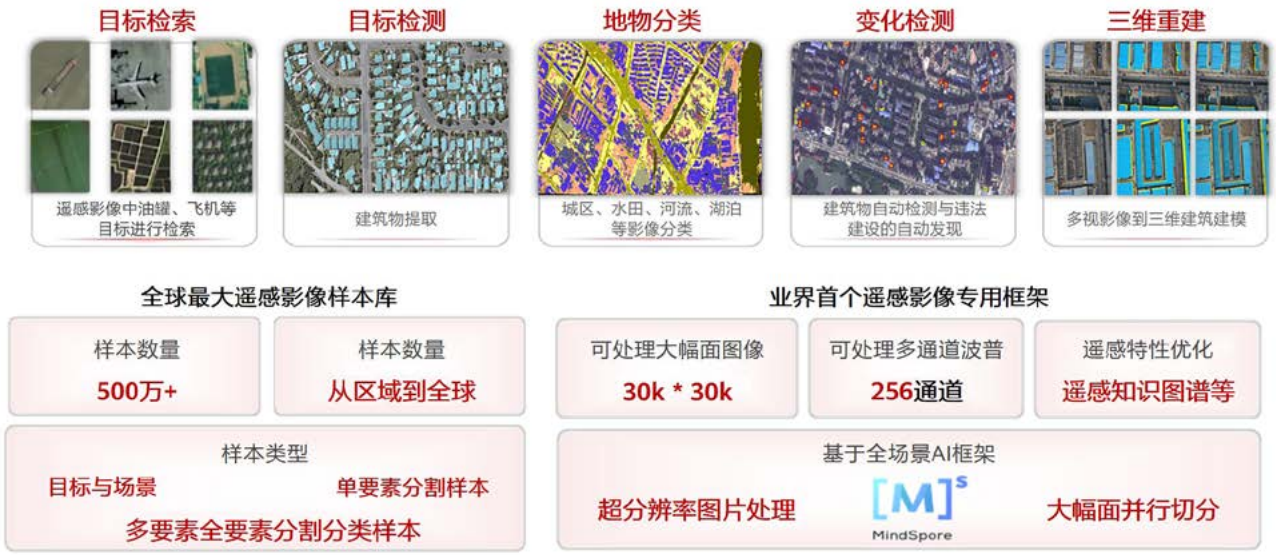


图 10 业界首个遥感影像样本库与专用框架

图 11 业界首个三模态大模型紫东 . 太初

太初都获得了第一名的成绩。紫东太初大模型的优异性能引起企业关注，正在与上汽集团、魏桥创业、爱奇艺等企业伙伴探讨如智能驾驶、工业质检、影视创作等领域的应用落地。

西部最强算力，联合西安高校打造业界首个中文语音基础模型和智能雷达遥感应用

2021 年 9 月，西安未来人工智能计算中心启动上线仪式，MindSpore 与西安电子科技大学、西北工业大学、陕西师范大学达成民生应用创新的合作，重点打造智能遥感、中文及多语种语音基础模型，使能智能应用。

西电智能遥感主要是面向雷达遥感影像，具有不受天气干扰的优势。面对如地震、暴雨、台风等灾害天气也不受影响，可实现全天候、全天时对地观测分析。此次合作的智能雷达影像预训练模型主要针对地物要素提取和配准与变化检测两大场景（见图 12）。可应用在海洋监测、山地变化、城市发展、设施扩建，植被变化，特别是地震或洪灾的灾后评估，为快速救灾提供指导。

语音是最自然的交互方式，但目前的语音模型高度依赖标注数据，且语音数据的标注远高于图像和文本的标注成本。因此，MindSpore 联合西工大研发大规模语音预训练模型来解决大量标注数据高度依赖问题，实现少量标注数据即可构建稳健的语音处理系统。提升不同领域、方言、语种的语音综合处理能力，服务语音识别、语音合成、口语理解、语音翻译、声纹识别等众多语音 AI 应用。目标打造 4 大核心基础模型：（1）全球首个中文语音预训练大模型，填补中文语音模型缺失的空白；（2）全球首个中文方言语音预

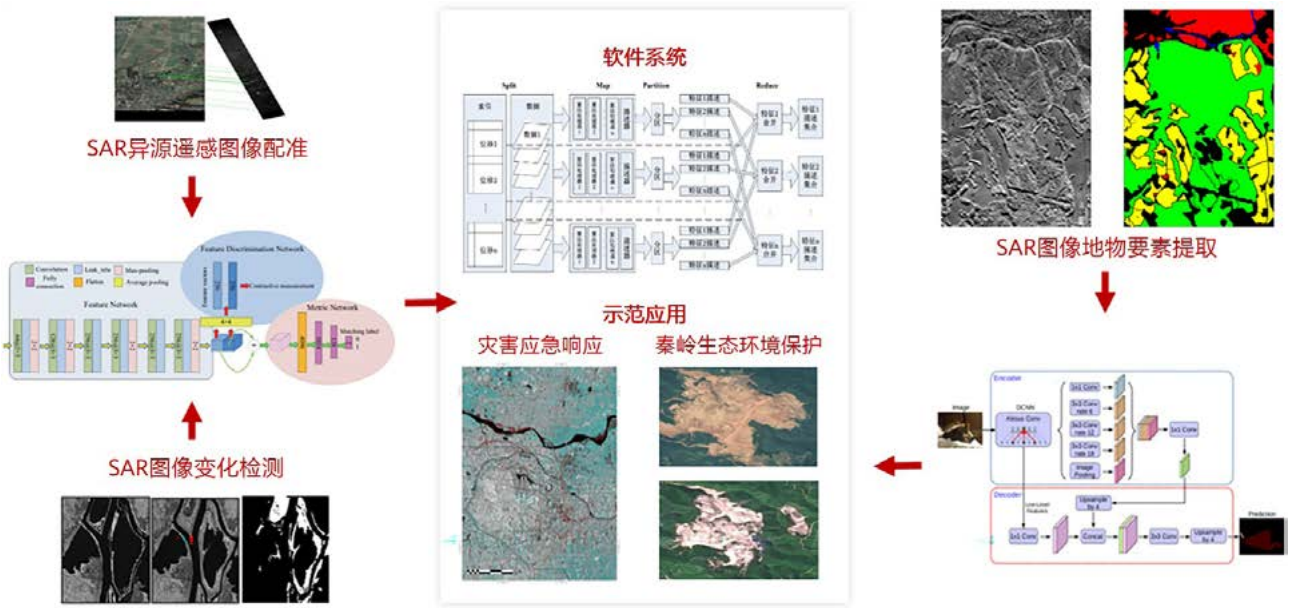


图 12 智能雷达影像预训练模型

图 13 MindSpore 智造解决方案

训练模型，提升口音方言的综合处理能力；（3）多语种语音预训练模型，提升多语种语音处理的能力。（4）泛音频大模型，感知声音，理解声音。

携手企业拥抱开源、坚持开放，使能行业智能化升级

目前 MindSpore 在医疗、交通、金融、制造、能源等 8 大行业应用树立行业标杆，与众多行业伙伴携手打造如智慧医疗、智慧金融、智能制造等解决方案。超过 60+ 行业伙伴通过 MindSpore 认证，年底预计超过上百家。

在智能制造领域，基于 MindSpore 的智造解决方案（见图 13）已快速推广复制。在宝德 PC 产线、长江计算产线、合肥美的冰柜和海尔洗衣机等产线部署上线。长江计算装配车间实现了提质增效的智能蜕变，检测准确率提高 10%，约 2 小时即可完成产线算法更换与迭代，提升了企业生产效率和经营效益。在智慧零售方面，微晟科技上线智能 AI 防损系统提升了零售效率。

MindSpore 还在武汉举办了生态伙伴研讨会议。敦锋科技、光谷信息、立得空间、斗鱼网络、工控研究院、精英路通、微晟技术、纳思科技、极目智能、微创光电等 10 家公司共同探讨未来发展的思路并建立了新的合作商机。

智能影像标注神器 – Pair，让医生从繁重的手动标注中解脱

Pair 由深圳大学医学超声影像计算实验室的倪东教授和杨鑫博士等带队研发的“全家桶式”的医学图像标注软件，具备专业便捷、通用易用且智能化的特点（见图 14）。

精细的分割标注是医学影像标注中最为耗时的任务。不同的影像模态、解剖结构和病灶需要不同专业与资质的医生进行繁重的逐点分割标注。为解决上述关键问题，Pair 团队研发了重磅的智能功能 AutoSeg，旨在实现通用的解剖结构精细分割标注，大幅节省手动标注的耗时，进而推动 AI 影像研究的快速实现与普及。

Pair 软件的自动分割标注功能是基于 MindSpore 框架实现的。通过 MindSpore Lite 的推理引擎完成分割模型的推理部署，相比于团队原采用的 LibTorch 库来说，模型推理速度整体提升 30% 左右。MindSpore Lite 中高性能算子实现充分发挥普通 PC 中的 CPU 的算力，有效平衡了模型算力需求和模型精度要求，使得推理耗时缩短至 1 秒内，极大地提升了 AutoSeg 的用户体验，非常好的满足了医生在普通 PC 上标注的需要。实践结果表明，使用 Pair 的 AutoSeg 自动分割标注功能，可节省约 70% 的分割标注耗时。

MindSponge 下一代人工智能分子动力学模拟软件框架

分子模拟作为一个重要的理论研究手段，可以在微观分子世界与宏观可观测量之间搭建桥梁，从而为人们在分子水平上理解物质的结构和动力学性质提供工具。其在化学化工、生物医药、能源、材料等多个领域都有广泛的应用。虽然近几十年来国际学术界和工业界已发展出很多具有特色的分子模拟软件，但是在分子模拟领域国内一直没有成熟的自主知识产权软件。而另一方面，现有的分子模拟软件在实际应用中也尚有大量科学与技术问题没有很好解决，可靠性和效率都亟待提高。

基于上述两方面原因，由北大和深圳湾实验室高毅勤课题组与华为 MindSpore 团队联合开发的新一代人工智能分子动力学模拟程序 MindSponge 应运而生。该框架是国内完全自主研发的分子模拟软件库，具有高性能、模块化等特性，支持科学家进行高效且便捷地搭建分子动力学模拟中所需要的相关计算模块。基于 MindSpore 自动并行、图算融合等特性，它可以可高效地完成传统分子模拟过程（见图 15）。除此之外，利用 MindSpore 自动微分的特性，MindSponge 天然地支持将神经网络等 AI 方法与传统分子模拟进行结合，并且能运用 MindSpore 框架自身的高性能计算特性。

MindSponge 力求成为在算力时代引领技术变革的下一代分子模拟软件平台。它不仅是国内首个开源发布的通用分子模拟软件框架，更布局以大数据和深度学习为代表的人工智能技术，实现人工智能和分子模拟的交叉融合与创新突破。

图 14 智能影像标注 – Pair

4.3 社区运营

近年以来，开源社区的发展越来越迅猛。根据中国开源代码托管平台 Gitee 数据显示，2020 年开源项目增长 192%，开源用户增长在世界名列前茅。在开源用户迅猛发展的同时，也面临着开源用户对于开源、开源社区文化了解薄弱等问题，给开源社区治理带来严峻挑战。在此背景下，作为以热点 AI 技术为主体的 MindSpore 社区，也在积极探索有效的社区治理方案，希望给社区用户和开发者带来极致的社区体验。

社区治理的其中一个重要环节就是以 Issue（问题处理）和 PR（代码合并）为核心的社区开发活动。如何做好 Issue 的快速闭环和代码的高效高质量合入，以避免或因参与门槛高，或因响应速度慢，导致开发者“乘兴而来，败兴而归”，是一个社区发展的关键。MindSpore 社区运营团队调研了业界优秀社区的工程实践，联合上海交通大学曹健老师团队在 MindSpore 社区成立开发者体验 SIG 组。致力于使用 AI 技术赋能社区机器人辅助社区治理运营（见图 16），降低开发者参与门槛，提升运营人员治理效率，让社区更有温度！

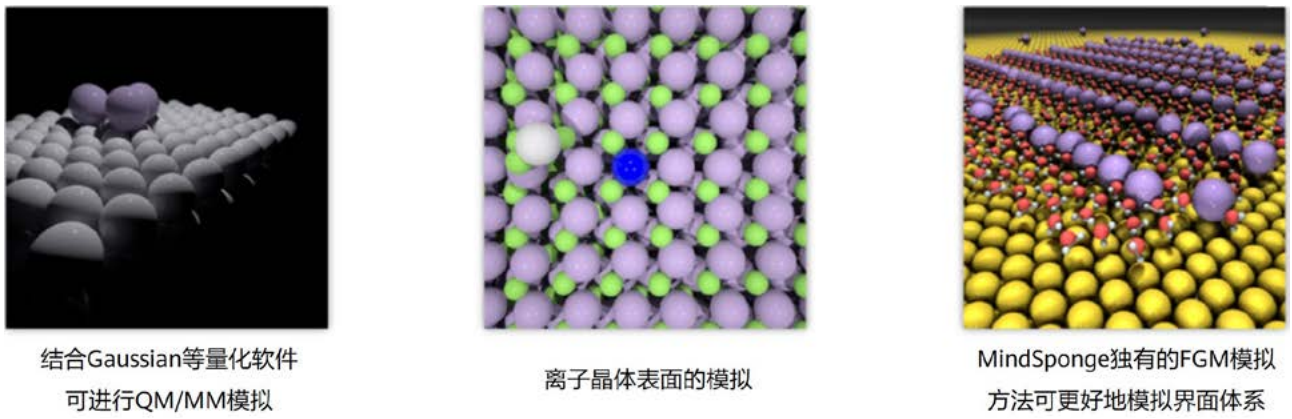


图 15 MindSponge 的典型应用场景

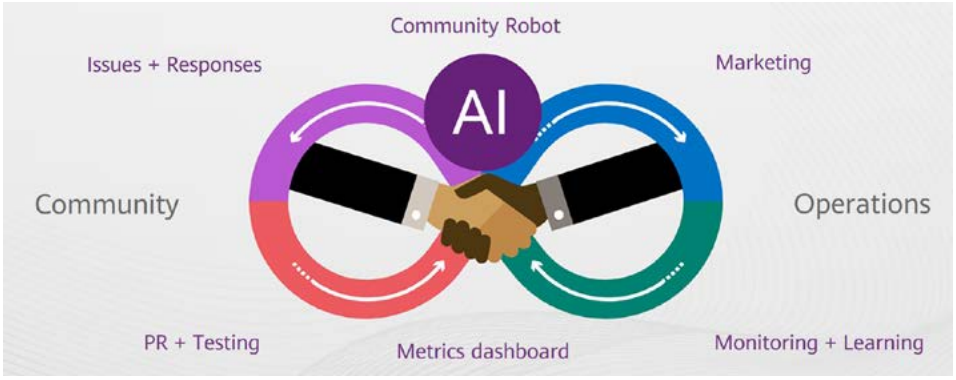


图 16 MindSpore 社区机器人辅助治理

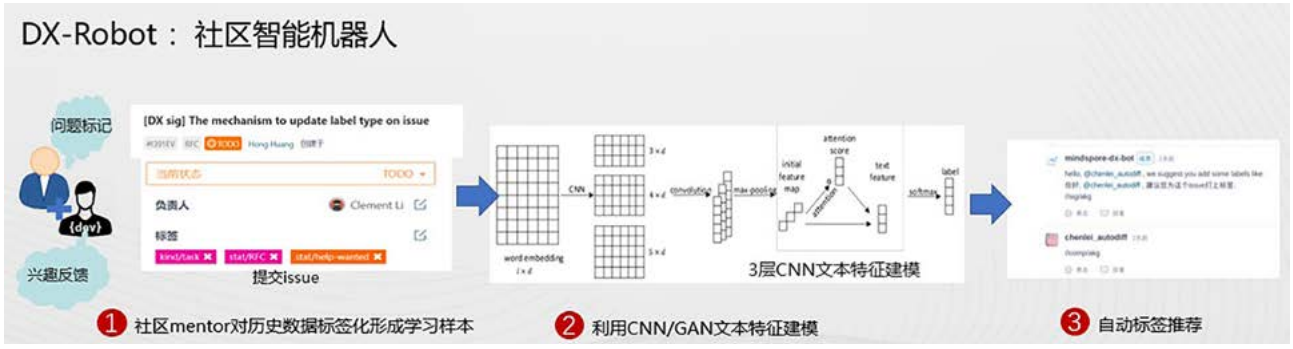


图 17 MindSpore 社区 Issue 指派流程

以图 17 所示 Issue 指派为例，通过社区 Mentor 对历史数据标签化形成学习样本；然后利用 CNN/GAN 文本特征建模；最终通过社区智能机器人进行自动的标签或人员推荐，加速了问题的流转。最终将社区 Issue 首次响应时间由原来的 140 小时降低到 20 小时，该服务已在 MindSpore 社区上线，日均调用 80+ 次，社区问题的首次响应得到明显改善。下一步社区将继续探索通过开发者特征建模，构建开发者画像和关系图谱，识别社区活跃 / 不活跃开发者，进一步提升社区开发者留存率。

5 总结与展望

AI 框架是深度学习基础设施建设的核心。以 Torch、Theano 和 Caffe 为代表的第一代 AI 框架奠定了基于 Python 构建深度学习模型、自动微分以及基于计算图进行模型表示与执行的基本设计思路。在第二代 AI 框架中，TensorFlow 通过分布式训练、多样的部署能力在工业界广泛使用；PyTorch 则以动态图机制带来的灵活性吸引了大量的研究者和算法工程师。MindSpore 作为一个新兴的框架，在充分吸收现有框架特性的基础上，快速迭代更新，通过一系列自主创新技术已形成差异化的竞争优势。

结合当前 AI 发展的趋势以及驱动力和挑战，我们希望 MindSpore 将在如下几个方面引领 AI 框架的演进：

1. 极致体验 AI

   从深度学习框架演进到通用张量可微计算框架，MindSpore 将进一步的兼容 Pandas 以及 NumPy/SciPy 等大数据分析和科学计算工具，提供更为简洁易用的科学计算表达。将灵活的模型表达和高效的模型执行更好的结合起来，提供更通用的 AI 编译器，为支持更多的应用类型提供可能性。MindSpore 还将进一步的实现可视化智能调优，从而实现全流程极简。除此以外，MindSpore 将通过性能 / 精度自适应、异构并行、模型及代码等多项创新技术，以及统一支持自研以及第三方芯片驱动注册，来持续发挥其全场景协同的优势。

2. 超大规模 AI

   基于类脑科学以及目前特征学习的发展，超大规模的神经网络训练将对目前的业务性能带来革命性的升级，模型和数据的规模和复杂度持续提升。MindSpore 将在数据 / 模型并行的基础上，进一步发展混合并行技术，并通过支持弹性训练和高阶优化技术，结合软硬件协同优化来应对海量数据和超大模型的双重挑战。

3. AI 融合计算

   AI 与科学计算结合已经在业界进行如下三个方向的探索：一是 AI 建模替代传统的计算模型，目前已经有很大进展，如分子动力学的 DeePMD。二是 AI 求解科学计算方程，目前已经有对应的 AI 求解方法，比如 PINNs、PINN-Net 等，但是依然存在较大挑战，特别是在精度及收敛性方面。三是使用框架来加速方程的求解，也就是把框架看成面向张量计算的分布式框架，科学计算的模型和方法都不变，但是与深度学习使用同一个框架来求解。MindSpore 将通过高阶混合微分、多模数据融合以及多尺度混合计算方法实现典型科学计算问题的数学求解器，将框架的负载从单一的深度学习模型向通用的张量可微计算演进。

4. AI 安全可信

   作为 AI 业务的承载，AI 框架需要具备使能 AI 责任（如安全、隐私、公平、透明、可解释等）的能力。目前 AI 框架需要解决如下几个挑战：对于 AI 责任的各个方面，缺乏通用的分析方法、度量体系，以及场景感知的自动化度量方法；AI 模型鲁棒性、隐私保护技术、密态 AI 在实际场景下对模型性能影响较大；AI 的可解释性还缺乏理论和算法支撑，难以给出人类友好的推理结果解释。MindSpore 会逐步内置并优化对抗性训练、差分隐私、密态 AI、联邦学习以及可解释 AI 等能力，逐步引领框架从消费级 AI 演进到企业级 AI。

技术是生态的根，MindSpore 在深耕 AI 技术演进的同时，其开源生态之树必将愈加枝繁叶茂。我们期望 MindSpore 能够通过 AI 技术生态加速人工智能产业生态的建立，加快人工智能与产业融合，为智能经济的发展和产业数字化转型提供稳固的底层支撑。

参考文献

[1] A. Krizhevsky, L. Sutskever and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," In Advances in neural information processing systems, pp. 1097-1105, 2012.

[2] G. E. Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, *et al.*, "Deep neural networks for acoustic modeling in speech recognition," IEEE Signal processing magazine, pp. 82-97, Nov. 2012.

[3] M. Volodymyr, K. Koray, S. David, A. R.Andrei and V. Joel, "Human-level control through deep reinforcement learning," Nature, vol. 518, pp. 529-533, 2015.

[4] Y. Bengio, R. Ducharme, P. Vincent and C. Jauvin, "A neural probabilistic language model," Journal of machine learning research, vol. 3, pp. 1137-1155, Feb. 2015.

[5] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, *et al.*, "Tensorflow: large-scale machine learning on heterogeneous distributed systems," 2016, arXiv:1603.04467. [Online].

[6] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga and A. Lerer, "Automatic differentiation in pytorch," In Advances in neural information processing systems, pp. 1-4, 2017.

[7] Y. Q. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama and T. Darrell, "Caffe: convolutional architecture for fast feature embedding," In Proceedings of the 22nd ACM international conference on Multimedia, pp. 675–678, ACM, 2014.

[8] T. Q. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Y. Zhang and Z. Zhang, "MXNet: a flexible and efficient machine learning library for heterogeneous distributed systems," 2015, arXiv:1512.01274. [Online].

[9] G. Neubig, C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, *et al.*, "Dynet: the dynamic neural network toolkit," 2017. arXiv:1701.03980. [Online].

[10] S. Tokui, K. Oono, S. Hido and J. Clayton, "Chainer: a next-generation open source framework for deep learning," In Advances in neural information processing systems, pp. 1-6, 2015.

[11] N. Qian, "On the momentum term in gradient descent learning algorithms. Neural networks," the official journal of the International Neural Network Society, vol. 12, pp. 145–151, 1999.

[12] J. Duchi, E. Hazan, and Y. Singer, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization," Journal of Machine Learning Research, vol. 12, pp. 2121–2159, 2011.

[13] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," International Conference on Learning Representations, pp. 1–13, 2015.

[14] M. Y. Chen, K. X. Gao, X. L. Liu, Z. D. Wang, *et al.*, "THOR, Trace-based hardware-driven layer-oriented natural gradient descent computation," Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 7046-7054, 2021.

[15] T. B.Brown, B. Mann, *et al.*, "Language models are few-shot learners," 2020. arXiv:2005.14165. [Online].

[16] C. Leary and T. Wang. XLA: TensorFlow, compiled, 2017.

[17] T. Q. Chen and T. Moreau, "TVM: an automated end-to-end optimizing compiler for deep learning," the Proceedings of the 13th USENIX Symposium on Operating Systems Design and Implementation, pp. 579-594, Oct, 2018.

[18] M. Li, Y. Liu, *et al.*, "The deep learning compiler: a comprehensive survey," 2020. arXiv:2002.03794.

[19] I. J. Goodfellow, J. Shlens and C. Szegedy, "Explaining and harnessing adversarial examples,"2014. arXiv:1412.6572. [Online].

[20] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," IEEE Access, vol. 6, pp. 14410–14430, 2018.

[21] A. Kurakin, I. Goodfellow and S. Bengio, "Adversarial examples in the physical world," 2016. arXiv:1607.02533. [Online].

[22] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," In 2017 IEEE symposium on security and privacy, pp. 39–57, 2017.

[23] M. Fredrikson, S. Jha and T. Ristenpart, "Model inversion attacks that exploit confidence information and basic countermeasures," In Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1322–1333, 2015.

[24] R. Shokri, M. Stronati, C. Song and V. Shmatikov, "Membership inference attacks against machine learning models," In 2017 IEEE Symposium on Security and Privacy, pp. 3–18, 2017.

[25] T. Elsken, J. H. Metzen and F. Hutter, "Neural architecture search: a survey," 2018. arXiv:1808.05377. [Online].

[26] M. Wang, D. Zheng, *et al.*, "Deep Graph Library: a graph-centric, highly-performant package for graph neural networks," 2020. arXiv:1909.01315. [Online].

[27] M. Fey and J. E. Lenssen, "Fast graph representation learning with PyTorch Geometric," CoRR, abs/1903.02428, 2019.

[28] J. Z. Huang, H. F. Wang, Y. B. Sun, M. Fan, Z. J. Huang, C. Y. Yuan, Y. W. Li, "HGAMN: heterogeneous graph attention matching network for multilingual POI retrieval at Baidu Maps," Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 3032-3040, 2021.

[29] R. Zhu, K. Zhao, *et al.*, "AliGraph: a comprehensive graph neural network platform," 2019. arXiv:1902.08730. [Online].

华为毕昇编译器的创新与实践

高耀清 [1]，华保健 [2]

[1] 编译器与编程语言实验室

[2] 中国科学技术大学软件学院

摘要

计算产业的场景诉求、硬件架构和微架构的不断改进和创新，如软硬件协同 / 软件定义硬件、ISA 领域扩展等新的研究趋势和进展，不断对编译器设计和实现提出新的研究挑战。为应对这些研究挑战，毕昇编译器通过对业务诉求、工业及学术界研究热点和发展方向的研究，积极吸收借鉴国内外先进编译器架构设计成功经验的同时，通过对经典“三段式架构”的逐“端”技术创新突破，结合业务痛点有针对性的构建关键差异化竞争力，快速迭代演进，实现编译器使能鲲鹏和昇腾极致性能，支持典型场景性能优势和技术生态构建。同时面向未来，毕昇编译器积极拓展并布局新的创新研究方向。本文在总结分析编译器技术演进的基础上，系统介绍华为毕昇编译器的技术创新与实践，分析核心关键技术特性，并指出进一步演进和创新的技术方向。

1 引言

编译器是一种重要的基础软件，它将面向编程人员的高级语言翻译成硬件能够执行的低级机器语言。在翻译的过程中，编译器对程序进行复杂的分析及优化，改进程序的执行性能。同时，编译器还需要兼顾高层编程语言实现的正确性和处理底层目标体系结构的复杂性。操作系统、数据库、网络协议栈、人工智能及机器学习框架等高层软件，都需要底层编译器进行有效支撑，由于编译器在整个软件栈中的重要作用，它被称为软件开发中的皇冠。

编译器是计算机科学中理论和实践结合最紧密的学科，围绕编译器的理论研究和工程实践，理论界和工业界已经研究了丰富的编译器设计理论并进行了大量卓有成效的工程实践，这些理论和实践包括程序设计语言类型、文法 [1-3]、类型系统、基于格和不动点的数据流分析 [4-6]、图论及应用 [7, 8]，约束求解等。由于编译器设计中的许多问题，例如后端的寄存器分配 [9, 10] 或指令调度 [11, 12] 等，都是 NP 难问题，对这些问题的探索仍是编译器研究的热点问题。

编译器工程也已经取得了令人瞩目的成果和进展，除了开源社区中应用非常广泛的开源编译器外，很多大学、研究机构和公司等都维护或发布了面向不同语言和不同目标体系结构的开源或商业版本的编译器，这些编译器工程实践对于推进学术研究或支撑商业成功都起到了非常关键的作用。

随着应用新场景诉求的变化，硬件架构的演进和基础理论及技术的发展，编译技术的理论研究创新和实践探索都取得了非常大的进展。这些新的编译器设计和实现创新和进展有效支撑了大数据、云计算、异构计算、深度学习等新的应用场景。但是，目前尚没有最新文献，对编译器设计的新成果、新进展进行系统性介绍和总结，以及对新方向进行系统性展望。

针对这一问题和挑战，本文的主要目标是基于华为毕昇编译器 [13]，对编译器的技术创新和实践进行系统总结和介绍，并探讨未来发展方向。本文主要内容包括：1）首先介绍编译器的发展史、 编译器架构、编译器业界现状等内容，并通过深入分析当前最流行的 LLVM 开源编译器的社区最新进展等，对编译器重要概念进行总结。2）重点讨论华为毕昇编译器的架构，尤其是其使用的先进编译优化技术如：循环优化、自动向量化等，并讨论了基于 AI 的编译器自动调优等内容，这些特性对支持毕昇编译器的成功起到了关键作用。3）通过对业界趋势的分析，探讨了华为毕昇编译器未来规划可能发展的几个重要方向、以及将带来的重要价值。

2 编译器简介

2.1 编译器基本概念

编译器（Compiler）是一种计算机程序，它将一种编程语言（源语言）写成的源代码转换成另一种等价的编程语言（目标语言）编写的程序。它的主要目的是将用便于人编写、阅读、维护的高级编程语言所写的源程序，翻译为计算机能解读、运行的低阶机器语言的程序，也就是可执行文件。源语言一般为高级语言（High-level Language），如 C、C++、Java、Rust、Go 等，而目标语言则是汇编语言或目标机器的目标代码（Object Code），有时也称为机器代码（Machine Code），如 x86、ARM、RISC-V 等 [14]。

图 1 编译器场景

编译器结构可分解为两个主要部分：前端（Front-end）和后端（Back-end）。前端和后端有明确的分工：前端专注于理解源语言程序，把源程序分解为多个词法单元，并检查这些词法单元是否满足语言规定的语法结构，检查该源程序是否符合程序语义的规定，生成一种中间表示（Intermediate Representation，IR）。

后端专注于将中间表示映射到目标机器上，其要完成的任务包括为源程序的语法结构选择合适的机器指令，以及为源语言的变量分配适当的目标机器的资源，等等。

现代的优化编译器一般还会引入中端（Middle-end），它是一个和具体语言以及具体目标机器相对无关的中间阶段。引入中端的主要作用和优势有两个：第一，引入中端可以使得编译器的前端和后端解耦，即多个语言的前端都可以生成公共的中端代码，而中端代码进一步又可以生成不同目标机器的代码；第二，在中端上，可设计并实现与语言以及目标机器无关的程序优化算法，对程序的性能、规模或其它指标进行通用优化。

基于编译器复杂性和应用场景的广泛性，编译器的设计要同时满足以下目标：向上面向应用开发者编程产能，向下支持多样化硬件极致性能，解决 3P 的问题，具体包括：

1：开发者编程产能（Productivity）：屏蔽硬件架构信息，易于程序员高效编程；

2：使能硬件极致性能（Performance）：实施静态动态编译优化，使能硬件极致性能，并有利于软硬协同设计；

3：软件兼容可移植性（Portability）：支持开放式场景友商生态迁移，解决硬件跨代源码 / 二进制级兼容以及编译器升级软件行为和性能兼容。

同时，编译器还需要安全可靠（Safety and Reliability），能够通过动静态程序分析做到检错纠错，保证程序代码的安全可靠。

由于编译器理论和实践的重要性，编译器一直在计算机科学中处于核心地位。以计算机科学的最高奖 ––– 图灵奖为例：第一个图灵奖就颁发给了编程语言和编译器专家 Alan Perlis 教授；此后，平均约每三到五年就有编程语言编译器相关领域的研究获奖。

年度	作者	获奖方向	备注
2020	Jeffrey Ullman，Alfred Aho	语言编译器基础理论和算法	
2017	Hennessy, John L, Patterson, David	硬件结构，编译器	
2013	Lamport, Leslie	并发编程	
2008	Liskov, Barbara	编程语言，系统设计	
2006	Allen, Frances ("Fran") Elizabeth *	编译优化，自动并行	IBM编译团队
2005	Naur, Peter *	编程语言设计	
2003	Kay, Alan	面向对象语言Smalltalk	
2001	Dahl, Ole-Johan "Nygaard, Kristen *	面向对象语言Simula	
1999	Brooks, Frederick ("Fred")	硬件架构，软件工程	
1987	Cocke, John *	RISC架构，编译优化	
1986	Hopcroft, John ETarjan, Robert (Bob) Endre	形式语言理论	
1984	Wirth, Niklaus E	编程语言，编译系统，体系结构	
1980	Hoare, C. Antony ("Tony") R.	编程语言，并发理论	
1979	Iverson, Kenneth E. ("Ken") *	编程语言APL	
1977	Backus, John *	编程语言Fortran，编译器	IBM编译团队
1972	Dijkstra, Edsger Wybe *	程序设计	
1966	Perlis, Alan J *	程序设计，编译结构	

图 2 图灵奖中编译器相关研究获奖情况

2.2 编译器发展史

自 20 世纪 50 年代中期以来，编译器设计就一直是计算机科学重要研究领域。Fortran 编译器是第一个被广泛使用的高级语言编译器，它由 IBM 开发，是一个多遍系统，其设计和实现引入了现代编译器中的很多重要概念。

如独立的词法分析器、语法分析器，以及包括寄存器分配在内的很多程序优化算法等。

在 20 世纪六七十年代，研究者研究并构建了许多有影响力的编译器。这其中包括经典的优化编译器 FORTRANH [15, 16]、Bliss−11 和 Bliss−32 编译器 [17]，以及可移植的 BCPL 编译器 [18]，等等。这些编译器可以为各种复杂指令集计算机（Complex Instruction Set Computer，CISC）体系结构生成高质量的目标代码。

在 20 世纪 80 年代，精简指令集计算机（Reduced Instruction Set Computer，RISC）体系结构的问世对编译器设计产生了深远影响。该体系结构要求编译器设计者不仅需要跟踪新的程序设计语言特征，还要研究并设计新的编译算法，以便能最大限度地发挥新硬件的计算能力。这些趋势导致了新一代编译器 [19–21] 更加专注于强有力的中端代码优化技术、后端代码优化以及代码生成技术的研究。这一阶段编译器的典型架构如下图所示，现代 RISC 体系结构上的编译器仍然遵循该架构模型。

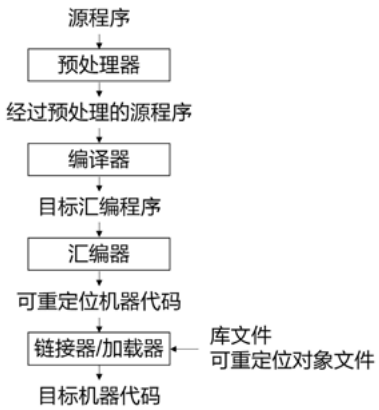


图 3 编译器典型架构

从 20 世纪 90 年代开始，编译器研究人员开始专注于处理微处理器系统结构中提出的挑战，包括处理器中多功能部件、内存延迟和代码并行化，等等。事实证明，20 世纪 80 年代提出的针对 RISC 架构的编译器的结构和组织，仍然具有足够的灵活性来应对这些挑战，因此研究人员可以通过构建新的遍（Pass），插入到编译器的优化器和代码生成器中，来应对体系结构提出的挑战 [22]。

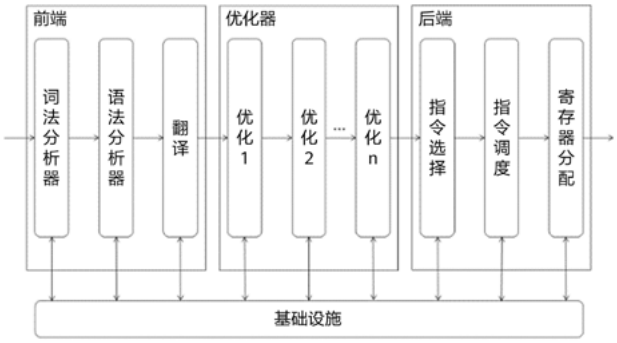


图 4 典型优化编译器架构

单靠各组织机构独立开发全自研封闭编译器成本越来越高昂，这个时期，Richard Matthew Stallman 决定开发对未来影响深远的 GCC 开源编译器。作为 GNU 系统的官方编译器（包括 GNU/Linux 家族），它也是编译与创建其他操作系统的主要编译器，包括 BSD 家族、Mac OS X、NeXTSTEP 与 BeOS 等。

采用三段式架构的 GCC 是跨平台软件的编译器首选。有别于局限于特定系统与运行环境的编译器，GCC 在所有平台上都使用同一个前端处理程序，产生一样的中间表示，在各个其他平台上编译，并正确无误的输出程序 [23]。

2.3 LLVM 开源编译器介绍

由于体系架构，编程语言种类等不断增加，GCC 在快速适配新的需求，构建竞争力上也开始显得不够灵活。LLVM 开源编译器在编译器三段式架构基础上，采用模块化设计和库实现，其生态兼容、友好的软件许可证，社区活跃度、学习成本等因素被学术和工业界追捧，已经具备明显的主导趋势。

LLVM 是一种涵盖多种编程语言和目标处理器的编译器，以 C++ 编写而成，包括了前端、后端、优化器、众多的库函数以及若干的模块，对开发者保持开放，并兼容已有脚本。LLVM 已作为实现各种静态和运行时编译语言的通用基础结构。目前有 140 多个公司参与 LLVM 项目，涵盖不同产业和硬件架构，近 10 年 LLVM 社区持续保持高热度。

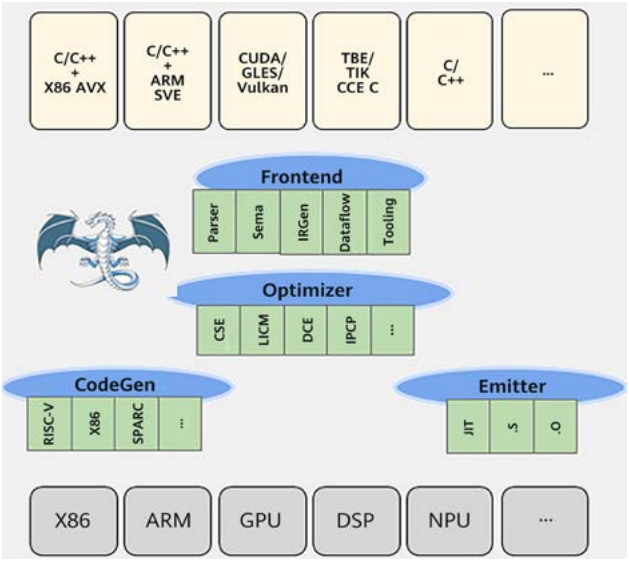


图 5 LLVM 架构图

其中主要贡献者主要来自于 Apple、Google 和芯片大厂，用于支持多种标准语言和自研语言如 Swift、Rust、GO、及使能不同架构的芯片如 X86、ARM、RISC-V 等性能，同时也衍生出一些创新的技术如 MLIR 等 [21, 24]。

目前行业大厂纷纷加入 LLVM 编译器的布局，除了参与社区贡献外，基于 LLVM 架构，通过差异化竞争力的编译技术，构建自己品牌的闭源编译器：Apple、Google、高通、IBM、AMD、ARM、Intel、Facebook、微软，华为等都对 LLVM 进行支持；涉及的场景包括终端、5G、计算 /HPC/ 服务器等广泛领域。

公司	编译器	应用场景
Apple	Swift/Objective-C	终端
Google	Android 编译器工具链	终端
高通	Android 编译器 + Hexagon编译器	终端+5G
IBM	XL C/C++编译器	高性能计算/通用服务器
AMD	AOCC Fortran/C/C++编译器	高性能计算/通用服务器
ARM	AHC编译器/Mali GPU编译器	终端/高性能计算/通用服务器
Intel	DPC++编译器	高性能计算/通用服务器
其它	Facebook，微软，英伟达，Xilinx，Synopsys... ...	

图 6 大厂 LLVM 布局情况

3 华为毕昇编译器

3.1 毕昇编译器介绍

毕昇编译器是华为公司推出的高性能多样化算力编译器，它基于开源的 LLVM 编译框架 [25]，针对应用场景，编程语言和硬件架构，构建差异化竞争力的通用编译优化技术，架构相关和应用场景相关的深度协同优化等。图 7 简要展示了毕昇编译器的总体架构。不同编程语言编写程序作为编译器的输入，被转换为内部统一的中间表示，编译器组织了一系列优化遍，如循环优化、内存布局优化和自动向量化等，对中间表示进行优化变换，挖掘程序中的并行性和局部性，充分利用内建加速指令等硬件特性。同时毕昇编译器还集成了 Autotuner 特性，对静态优化过程中无法确定的优化参数进行自动调优。最终，编译器输出针对鲲鹏平台的高效可执行程序。

图 7 毕昇编译器架构简图

3.2 毕昇编译器关键竞争力技术

毕昇编译器在高性能编译算法、定制加速指令集优化、AI 迭代调优（Autotuner）多个方向重点布局关键竞争力。毕昇编译器使用先进的高性能编译算法，挖掘程序中的并行性和局部性；使用架构相关指令优化，选择定制的指令集加速程序；对于静态优化时无法确定的调优选择，使用 AI 迭代调优自动地进行优化。

3.2.1 高性能编译优化算法

传统芯片追求通用性，微结构复杂，导致软件难以发挥出硬件性能。为了挖掘晶体管潜力，针对特定场景，在性能功耗优先的前提下，按领域特征针对性优化，通过软件的复杂度换取性能功耗比提升，以此来实现进一步的性能收益是业界目前主要的手段。

2017 年，计算机架构领域两位重量级人物 David Patterson 与 John Hennessy 在斯坦福大学发表演讲时举的例子，通过使用多种架构设计及编译优化算法技术实现了程序性能的大幅提升 [26]，图 8 展示了矩阵乘计算在不同实现和不同优化技术下的性能加速比。

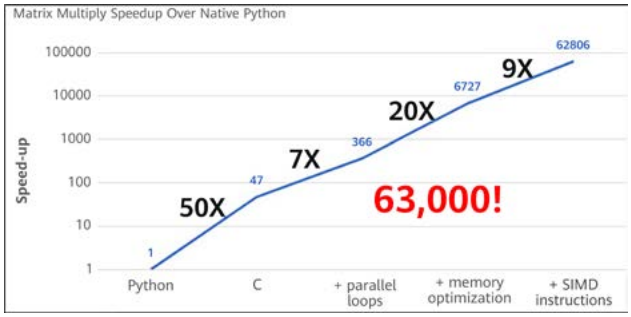


图 8 矩阵乘计算的性能加速比

为了获取更好的性能，毕昇编译器同样从多个方向发力，实现应用对硬件的充分利用，目前重点发力的方向有：

1. 针对特定领域、不同层次更有效的并行
   - 循环并行
   - 指令集并行（Instruction-Level Parallelism，ILP）、内存层次并行（Memory-Level Parallelism，MLP）
   - 超标量、多线程、多核、单指令流多数据流（Single Instruction Multiple Data，SIMD）、单线程多数据流（Single Thread Multiple Data，STMD）
   - 超长指令字（Very Long Instruction Word，VLIW）vs 投机、乱序
2. 空间 / 时间局部性下更有效内存利用
   - 用户控制 vs 高速 Cache
   - Cache 访问、内存搬移、函数分块等
3. 消除不必要数据精度
   - 低精度 FP 取代 IEEE 标准 FP
   - 32~64 位整数到 8~16 位整数
   - 混合精度计算模式

基于以上思路，毕昇针对性增强编译优化算法。

3.2.1.1 循环并行优化

循环内的语句通常被多次执行，针对循环进行优化能够得到显著的收益。因此循环优化是编译器重要的性能提升手段，具有广泛和多样化的优化算法。如《计算机体系结构将迎来一个新的黄金时代》[26] 中提到，通过循环并发，用例获得了大幅度的性能提升。编译器可以通过不同的算法（如：提高缓存利用率、降低寄存器压力、减少动态指令数和复用不同迭代加载或计算值暴露其它优化的机会：向量化、指令调度等）来提升循环的性能。

毕昇采用了多种循环优化手段，其中重要的算法如下：

1. Loop Unrolling and Jamming

该优化技术通过展开嵌套循环的外层循环，融合内层循环的循环体提升 Cache 的利用率，如下例所示：

```
for (int i = 0; i < n; i++) {
  for (int j = 0; j < m; j++) {
    a[j] += b[i][j] + b[i-1][j];
  }
}
```

编译优化后：

```
for (int i = 0; i < n-n%2; i+= 2) {
  for (int j = 0; i < m; j++) {
    a[j] += b[i][j] + b[i-1][j];
    a[j] += b[i+1][j] + b[i][j];
  }
}
```

通过以上转换，Cache 命中率能得到有效改善。

2. Loop Fusion/Distribution

Loop Fusion 将两个循环的循环体合并，寻找创建一个循环的机会。这样的优化好处在于，通过合并寻找可重用值、暴露指令调度机会；

Loop Distribution 是 Loop Fusion 的逆过程，将一个循环的循环体划分为两个单独的循环体，这样优化的好处在于暴露向量化优化的机会，同时找到其它优化空间。

下面的例子通过优化将前两个语句保存在一起，以重用加载的值，同时分离出不可向量化的第三个语句。

```
for (int i = m; i < n; i++) {
  a[i] = b[i-1] + b[i];
  c[i] = b[i-1] - b[i];
  d[i-1] = d[i-2] + 1;
}
```

编译优化后：

```
for (int i = m; i < n; i++) {
  a[i] = b[i-1] + b[i];
  c[i] = b[i-1] - b[i];
}

for (int i = m; i < n; i++) {
  d[i-1] = d[i-2] + 1;
}
```

3. Loop Unrolling

循环展开技术通过复制循环的循环体，减少循环迭代次数，从而减少分支跳转和循环边界检查。另外，展开后的循环体内可能还会新增可重用值，暴露更多指令调度机会。

```
for (int i = 0; i < n; i++) {
  a[i] = b[i-1] + b[i];
}
```

编译优化后：

```
for (int i = m; i < n; i++) {
  a[i] = b[i-1] + b[i];
  c[i] = b[i-1] - b[i];
}

for (int i = m; i < n; i++) {
  d[i-1] = d[i-2] + 1;
}
```

4. Loop Unswitching

Loop Unswitching 通过外提循环内判断条件不发生变化的条件分支语句，减少分支跳转，提供更多局部优化的时机。

```
for (int i = 0; i < n; i++) {
  a[i] = 1;
  if (b[j] > 10)
    a[i] += 1;
}
```

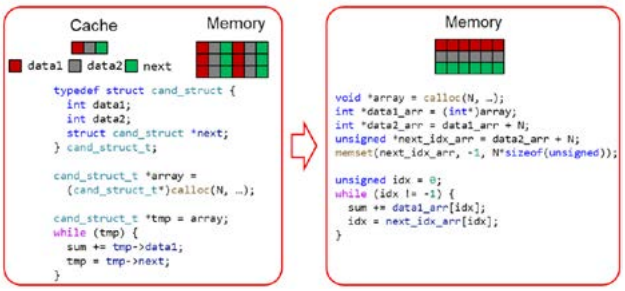
编译优化后：

```
if (b[j] > 10)
  for (int i = 0; i < n; i++)
    a[i] = 1;
else
  for (int i = 0; i < n; i++)
    a[i] = 2;
```

3.2.1.2 内存优化

1. 结构体内存布局优化

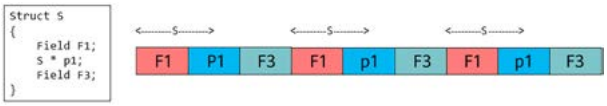
毕昇编译器针对结构体内存优化布局做了强化，结构体内存布局优化基于全程序（Whole-program）优化，用以提高 Cache 利用率。优化的主要手段是将结构体数组转换为数组结构体。结构体可以是显式的，也可以通过检查循环中的数组使用情况来推断它们。



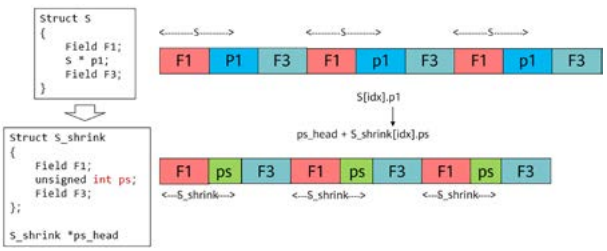
2. 结构体指针压缩优化

结构体指针压缩优化技术通过降低内存使用空间，提升 D-cache 的命中率。

下文以一个简单的结构体优化示例来具体解释该优化技术。如下图所示，指针成员 P1 占据 8 字节。



通过将域成员指针压缩，指针外提，减少了每个结构体 Node 的内存体积，从而提升 Cache Line 中可以存储的结构体数据，进而提升 Cache 命中率。



通过全局结构体指针 ps_head 和 ps 变换为相对基址 + 偏移的组合访问原来的域指针成员。

3. 预取优化

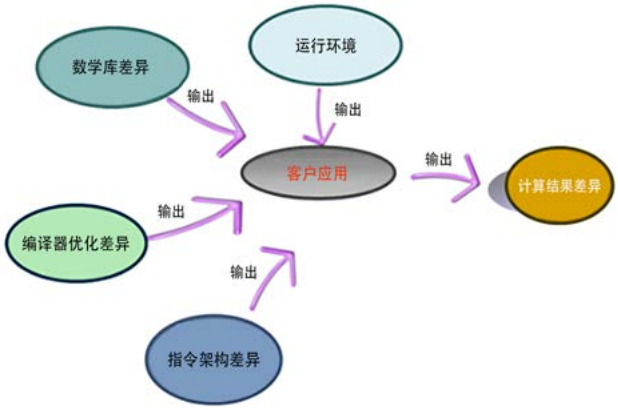
软件预取是一种通过插入预取指令提前从内存中读取所需数据的优化技术。插入预取效果取决于“提前时间”，数据太早到达会浪费宝贵的 Cache 空间，而数据太晚到达则仍需要等待访存过程。因此预取的效果取决于预取的“提前量”。在选择软件预取“提前量”的过程中需要综合考虑 Cache Line 大小、访存延时、循环大小等。毕昇编译器针对鲲鹏微架构特征调整软件预取参数，选择精确的预取时机，从而提升 D-cache 的命中率。

```
for (int i = 0; i < N; i++) {
   sum += a[i] * b[i];
}
```

```
for (int i = 0; i < N; i++) {
   prefetch(&a[i+k]);
   prefetch(&b[i+k]);
   sum += a[i] * b[i];
}
```

3.2.1.3 浮点精度调优

HPC 应用（如气象 WRF，Grapes 等）迁移至鲲鹏服务器后，往往存在计算结果与原有数据结果比对不一致的场景。如下图所示，计算结果的精度差异主要来自四个方面：数学库、编译器优化、架构指令差异、运行环境。



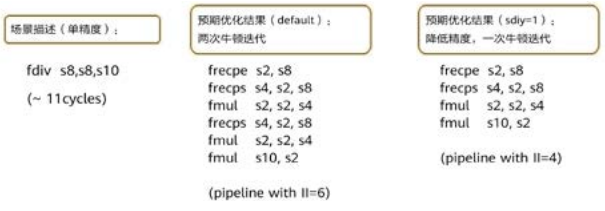
精度差异原因

针对四种可能导致精度差异的主要因素，对应的解决思路如下：

- 编译器优化差异：支持 fp-model 等精度控制选项，编译器优化行为调整及精度优化
- 指令架构差异：结合转码工具分析指令差异，使用软浮点模拟等手段消除差异
- 数学库差异：改写数学库函数，或使用不同精度数学库
- 运行环境差异：集群和多核环境差异（如 MPI 及 OpenMP 等）
- 计算结果差异：优化及改进不同精度模式下的浮点编译算法

场景一：

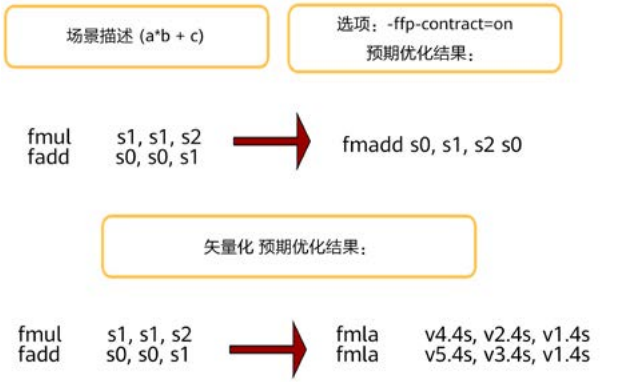
- 乘法指令替换除法，可以使能流水线化
- 调节牛顿迭代，按需调整结果精度



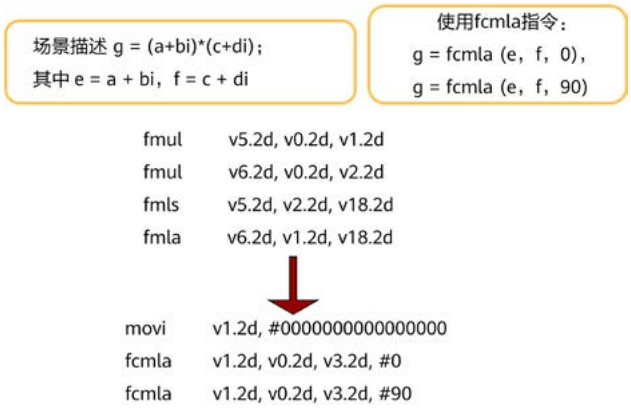
场景二：

- 乘加指令合并，通过控制（复数）乘加指令生成数量，进行精度优化

乘加指令合并



复数乘加指令合并

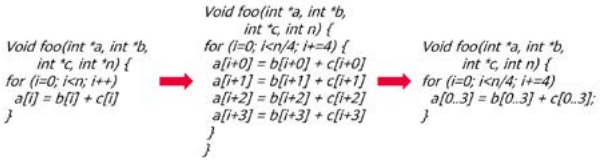


3.2.2 加速指令集优化

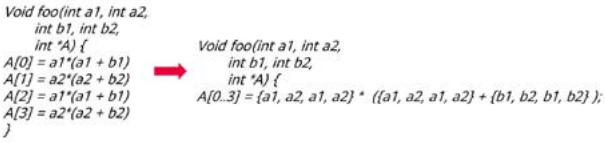
定制加速指令是自研芯片一个重要的竞争力构建手段，编译器在指令生成和指令选择的优化过程中发挥了重要作用，本小节介绍其中向量化定优化场景。

在并行计算中，自动向量化是自动并行化的一个特殊场景，其中计算机程序从一次处理一对操作数的标量实现转换为一次操作中同时处理多对操作数的向量实现。例如，基于 AArch64 的鲲鹏 920 处理器，具有 32 个 128 bits 的向量寄存器，可以一次操作 4 路 32 位或者 2 路 64 位的数据。毕昇编译器重点优化了循环向量化及超字级并行向量化（Superword-Level Parallelism，SLP），充分保持程序局部性，高效提升计算密集型场景的性能。

下图中展示了循环向量化原理。

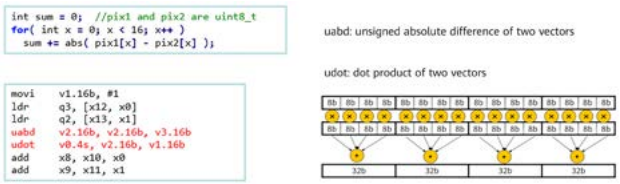


下图中展示了 SLP 向量化原理。

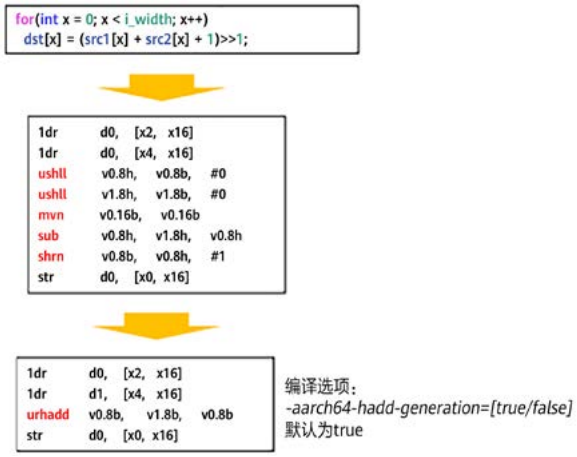


基于以上原理，毕昇编译器通过自动向量化，自动生成鲲鹏向量指令。以下两个示例具体展示了自动向量化优化的两个典型应用场景。

场景一：



场景二：

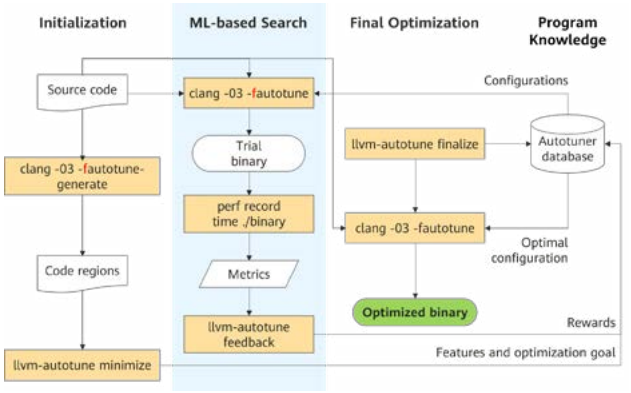


3.2.3 自动调优 Autotuner

自动调优是一种自动化的迭代过程，通过操作编译选项来优化给定程序，以实现最佳性能。在过往的研究中，基于机器学习的编译器自动调优技术被证明比手工调优有更好的性能。

毕昇编译器的 AI 自动调优是由两个组件配合完成：毕昇编译器和 Autotuner 命令行工具。

- 毕昇编译器是带有自动调优特性的编译器，配合 Autotuner 可以更细粒度地控制优化。
- Autotuner 是一个命令行工具，需要与毕昇编译器一起使用。它管理搜索空间的生成和参数操作，并驱动整个调优过程。



毕昇自动调优 Autotuner 基本架构

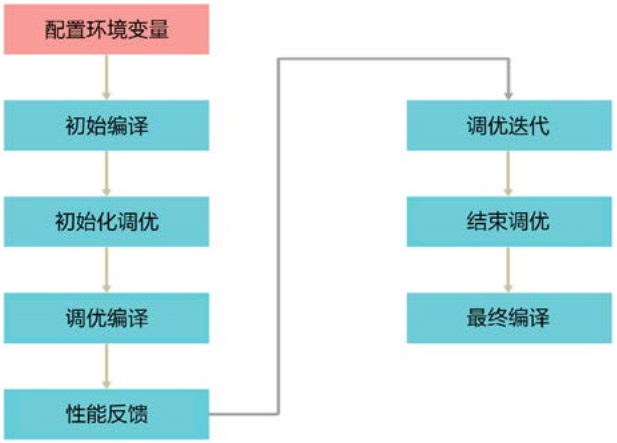
毕昇自动调优 Autotuner 引入基于 ML 的自动搜索技术（ML-based Search），其关键技术点有：

1）知识库（Autotuner Database）：根据静动态分析信息，建立知识库，支持决策系统进行优化；

2）优化决策系统（Optimal Configuration）：根据热点和性能评估信息、知识库信息，综合考虑确定优化措施；

3）热点标记和性能评价：热点标记，瓶颈检测，性能评估；

4）查找驱动（Feedback）：将优化决策系统反馈的优化建议，反馈给编译器执行优化措施。



毕昇自动调优 Autotuner 流程图

自动调优的关键流程如下：

1) 使用环境变量 AUTOTUNE_DATADIR 指定与调优相关数据的存放位置；

2) 添加毕昇编译器选项 -fautotune-generate，完成初始编译，生成调优机会；

3) 运行 llvm-autotune 命令，初始化调优任务。生成最初的编译配置文件供下一次编译使用；

4) 添加毕昇编译器选项 -fautotune，读取当前 AUTOTUNE_DATADIR 中的配置并编译；

5) 运行程序，并根据自身需求获取性能数字，使用 llvm-autotune feedback 反馈；

6) 根据用户设定的迭代次数，重复调优编译和性能反馈步骤进行调优迭代；

7) 经过多次迭代后，可选择终止调优，并保存最优的配置文件；

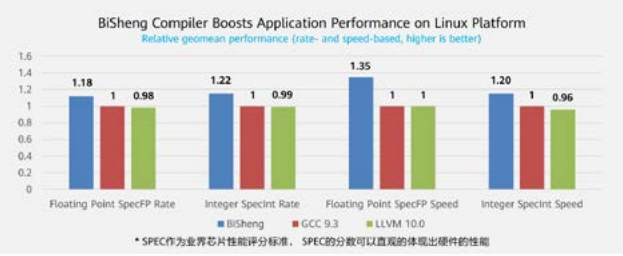
8) 使用上一步得到的最优配置文件，进行最后编译。

3.3 毕昇编译器优化效果

通过与鲲鹏芯片协同，毕昇编译器充分发挥芯片性能，提升鲲鹏硬件平台上业务的性能体验。本小节展示了毕昇编译器在不同 Benchmark 和不同应用场景下的实验结果。

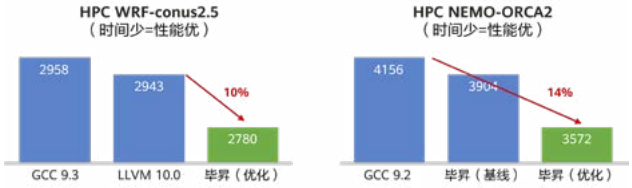
3.3.1 业界 CPU Benchmark 跑分

下图展示了在鲲鹏上平台上，测试程序使用毕昇编译器，GCC 9.3 和 LLVM 10.0 三个编译器优化后的相对性能。实验结果清晰地显示毕昇编译器的优化效果优于其余两个编译器，且将 SPEC 2017 性能平均提升 20% 以上。



3.3.2 HPC 典型应用性能提升

HPC 场景典型应用优化，结合鲲鹏芯片特性，通过毕昇编译器优化技术提升 HPC 应用的性能体验。



4 毕昇编译器未来创新演进方向

毕昇编译器将通过基础理论创新，向下软硬协同架构创新，向上业务场景精准调优，进一步全方位的技术创新。重要研究的方向有：

1. 程序持续优化（Continuous Program Optimization，CPO）：编译优化已不再局限在开发阶段，学术界和工业界均在探索高性能运行态优化技术，未来全流程持续优化将会是性能优化的重要布局方向。比如：IBM CPO 布局低成本的程序运行时性能监控技术，通过运行态程序行为变化分析，挖掘静态编译无法精确预测和优化机会点 [27]，还有米兰理工的 libVC 动态实时编译框架，从开发态的离线编译优化，拓展到运行态的在线编译优化，打通软件全生命周期性能持续优化能力 [28]，这些布局技术的角度虽然有所不同，最终通过构建持续优化，实现程序实时全生命周期的优化。

2. AI 辅助优化（AI for Compiler）：人工智能在编译优化的应用目前已被大公司广泛研究与应用，未来借助 AI 辅助的编译优化会是编译优化重要发力方向。比如：Intel 和 UC Berkley 联合研究 - 深度强化学习辅助循环自动向量化，使用深度强化学习（Deep Reinforcement Learning，DRL）来对循环向量化进行优化决策，辅助循环向量化自动化。通过稳定的预测模型可直接嵌入到编译的自动向量化阶段，得到较为优化的向量化结果 [29]；同样，Google 也提出了 AI 辅助 Inling 优化策略，采用机器学习模型替代 Inliner 的启发式算法，决策是否做 Inling，实现更精细化的优化。

3. SoC 级编译器：SoC 级架构芯片已成为业界芯片公司发展的主要趋势，未来 SoC 级编译器可能会成为新的研究方向，比如：苹果的 M1 芯片，PC 机处理器做成了手机 SoC，N 合一芯片，Intel Alder Lake，单一、高度可扩展的 SoC 架构，其中涉及统一内存架构（UMA 架构）、混合设计 SoC 架构（能效核和性能核）等新技术，需要布局探索配套的编译技术。

4. 超级优化器（Super Optimizer）：学术界一直在尝试拓展编译器的基础理论研究，近几年超级优化器有从学术界到产业界加速应用落地的趋势，超级优化器将优化问题转换为空间搜索问题，通过探索新理论算法对程序特征序列（指令集（ISA）/ 中间表示（IR））的有限搜索空间进行详尽搜索来尝试找到能够提供等效输出的最优程序。比如。斯坦福的 STOKE [30]、MTK 的 Souper [31] 等技术，已应用到 LLVM 编译器中。

5 总结

毕昇编译器针对不同的芯片，不同的场景，不同的应用特点，使用不同的编译优化手段构建关键竞争力，在编译优化的过程中权衡考虑代价与收益，综合考虑性能收益，代码体积，编译时间，可调试性等多方面因素，通过软件与硬件的协同优化，充分发挥硬件极致算力。

参考文献

[1] Melvin E. Conway, "Design of a separable transition-diagram compiler," *Commun. ACM*, 6(7), pp. 396–408, 1963.

[2] Philip M. Lewis II and Richard Edwin Stearns, "Syntax-directed transduction," *J. ACM*, 15(3), pp. 465–488, 1968.

[3] Donald E. Knuth, "On the translation of languages from left to right," *Inf. Control*, 8(6), pp. 607–639, 1965.

[4] Allen, F. E., "Control flow analysis," *SIGPLAN Notice,* 5(7), pp. 1–19, 1970.

[5] Gary A. Kildall, "A unified approach to global program optimization," *POPL* 1973, pp. 194–206.

[6] Barry K. Rosen, "High-level data flow analysis," *Commun. ACM*, 20(10), pp. 712–724, 1977.

[7] G. J. Chaitin, "Register allocation and spilling via graph coloring," *United States Patent 4,571,678*, February 1986.

[8] G. J. Chaitin, "Register allocation and spilling via graph coloring," in *Proceedings of the ACM SIGPLAN '82 Symposium on Compiler Construction*, *SIGPLAN Not.*, 17 (6), pp. 98–105, 1982.

[9] R. Lo, F. Chow, R. Kennedy, S. Liu, and P. Tu, "Register promotion by sparse partial redundancy elimination of loads and stores," in *Proceedings of the ACM SIGPLAN '98 Conference on Programming Language Design and Implementation*, *SIGPLAN Not.*, 33 (5), pp. 26–37, 1998.

[10] A. V. S. Sastry and R. D. C. Ju, "A new algorithm for scalar register promotion based on SSA form," in *Proceedings of the ACM SIGPLAN' 98 Conference on Programming Language Design and Implementation*, *SIGPLAN Not.*, 33 (5), pp. 15–25, 2008.

[11] Keith D. Cooper and Philip J. Schielke "Non-local instruction scheduling with limited code growth," *LCTES 1998*, pp. 193–207.

[12] Jack L. Lo and Susan J. Eggers, "Improving balanced scheduling with compiler optimizations that increase instruction-level parallelism," *PLDI 1995*, pp. 151–162.

[13] Edward S. Lowry and C. W. Medlock, "Object code optimization," *Commun. ACM 12(1)*, pp. 13–22, 1969.

[14] Randolph G. Scarborough and Harwood G. Kolsky, "Improved optimization of FORTRAN object programs," *IBM J. Res. Dev. 24(6)*, pp. 660–676, 1980.

[15] R. G. G. Cattell, Joseph M. Newcomer, and Bruce W. Leverett, "Code generation in a machine-independent compiler," *SIGPLAN Symposium on Compiler Construction 1979*, pp. 65–75.

[16] Martin Richards, "The portability of the BCPL compiler," *Softw. Pract. Exp. 1(2)*, pp. 135–146, 1971.

[17] Marc A. Auslander and Martin Hopkins, "An overview of the PL.8 compiler," *SIGPLAN Symposium on Compiler Construction 1982*, pp. 22–31.

[18] John Cocke and Peter W. Markstein, "Measurement of programming improvement algorithms," *IFIP Congress 1980*, pp. 221–228.

[19] Mark Scott Johnson and Terrence C. Miller, "Effectiveness of a machine-level, global optimizer," *SIGPLAN Symposium on Compiler Construction 1986*, pp. 99–108.

[20] John Hennessy and David Patterson, "A new golden age for computer architecture," Stanford and UC Berkeley, 13 June 2018.

[21] Chris Lattner, "The golden age of compilers," *ASPLOS 2021*, April 19, 2021.

[22] Ameer Haj-Ali, Nesreen K. Ahmed, Ted Willke, Sophia Shao, Krste Asanovic, and Ion Stoica, "Neuro vectorizer end-to-end vectorization with deep reinforcement learning."

[23] Abhijat Vichare, "GCC Internals: A Conceptual View," http://kcchao.wikidot.com/gcc-internals, Dec 2007.

[24] Chris Lattner, "MLIR primer: a compiler infrastructure for the end of Moore's law," *CGO*, 2019.

[25] David Chisnall, "What LLVM can do for you," *LLVM Developers' Meeting,* April 13, 2012.

[26] David Patterson and John Hennessy, "A new golden age for computer architecture," *IBM Journal of Research and Development*, pp. 239–248, 2006.

[27] C. Cascaval, E. Duesterwald, P. F. Sweeney, and R. W. Wisniewski, "Performance and environment monitoring for continuous program optimization," *ACM*, 2019.

[28] S. Cherubin and G. Agosta, "LIBVERSIONINGCOMPILER: An easy-to-use library for dynamic generation and invocation of multiple code versions," *SoftwareX*, pp. 95–100, 2018.

[29] Nesreen Ahmed, Ted Willke, Sophia Shao, Krste Asanovic, and Ion Stoica, "NeuroVectorizer: end-to-end vectorization with deep reinforcement learning," *International Symposium on Code Generation and Optimization 2020 (CGO 2020)*, February 2020.

[30] Eric Schkufza, Rahul Sharma, and Alex Aiken, "Stochastic Superoptimization," *SIGPLAN*, 2013.

[31] Raimondas Sasnauskas, Yang Chen, Peter Collingbourne, Jeroen Ketema, Gratian Lup, Jubi Taneja, and John Regehr, "Souper: a synthesizing superoptimizer," *Computer Science*, 2017.

在线匹配领域的最新进展

唐志皓 [1]，张宇昊 [2]

[1] 上海财经大学理论计算机科学研究所

[2] 上海交通大学

摘要

自从 Karp、Vazirani 和 Vazirani [36] 的开创性研究以来，在线匹配成为在线算法的基本问题之一。2013 年，Mehta [45] 就这一主题撰写了一份全面的调查报告，提出很多有趣的开放式问题，这些问题对算法的发展起到了指导性的作用。九年后的今天，研究人员取得了突破性的进展，回答了这些开放式问题，并在新场景（比如共享出行平台）的驱动下提出了新的在线匹配模型。

本文将呈现在线匹配领域的最新进展。在第 1 节中，我们将讨论经典的单侧顶点到达模型的研究成果，具体结论，请参见 1.3 节。在第 2 节中，我们将介绍最近提出的超越单侧顶点到达的模型。

1 单侧顶点到达

以下是一个互联网广告实例：

假设我们运营一个搜索引擎。广告商们希望向搜索某些关键字的用户推送他们的广告。当用户搜索时，搜索引擎需要立即选择对搜索词感兴趣的一个广告商，并向用户投放其广告。

这个问题可以归为在线二分图匹配问题。假设有一个潜在二分图，其顶点分别对应广告商和搜索用户。广告商顶点被称为离线顶点，这一侧的顶点对于搜索引擎来说是预知的。搜索用户顶点被称为在线顶点，这一侧的顶点按顺序到达。当每个在线顶点到达时，其与离线顶点的关联边就显现出来了。然后，算法需要立即决定匹配哪条边。在本例中，在线顶点和离线顶点之间的边表示用户搜索的关键字正是广告商所感兴趣的。

为方便表达，我们会介绍一些专有名词的定义，但本文并不会提供完整的分析和证明，而只是罗列一些优秀的算法以及他们背后的思想。

1990 年，Karp 等人 [36] 率先提出了无权在线二分图匹配，目标是最大化算法选择的匹配数。

在线二分图匹配（无权）

我们使用$G=(L\cup R,E)$来表示潜在二分图，其中L是离线顶点的集合，R是在线顶点的集合，每个顶点最多可以匹配一次。L中的离线顶点是预先给定的，R中的在线顶点是按顺序到达的。每个在线顶点到达时，我们就可以得到其与离线顶点的关联边，并且决定是否立即进行不可撤销的边匹配。我们的目标是最大化匹配到的边的数量。

竞争比：一个算法的竞争比Γ定义为所有可能的输入图和图中在线顶点到达顺序中，算法给出的（期望）匹配数与图中最大匹配数比值的最小值。严格来说，给定一个潜在二分图$G=(L\cup R,E)$和在线顶点的到达顺序$\sigma(R)$以及算法在该图和该到达顺序下所获得的匹配，我们可以得到算法的解 $\mathrm{ALG}\,(G,\sigma(R))$至少等于图 G 中最大匹配数 $\mathrm{OPT}\,(G)$（又称离线最优解）的Γ倍。

竞争比（最差图和最差顺序）

- $\Gamma=\min\limits_{G,\sigma(R)}\dfrac{\mathrm{ALG}(G,\sigma(R))}{\mathrm{OPT}(G)}$
- $\Gamma=\min\limits_{G,\sigma(R)}\dfrac{\mathbb{E}[\mathrm{ALG}(G,\sigma(R))]}{\mathrm{OPT}(G)}$（用于随机算法）

1/2 竞争比瓶颈：贪心算法简单地将一个顶点与一个任意的未匹配邻点进行匹配。贪心算法总能获得一个极大匹配，所以，该算法的竞争比为 1/2，即产生的匹配数至少是最大匹配数的 1/2。我们可以证明该算法是最优的确定性算法。

值得注意的是，这个发现可以推广到稍后介绍的每一个加权版本和到达版本。因此，最基本的研究问题自然是：

如何设计出竞争比严格大于 1/2 的随机在线算法?

Karp、Vazirani 和 Vazirani 提出了著名的 Ranking 算法，该算法实现了最优竞争比$1-1/e$。

Ranking 算法

Ranking 算法首先均匀地选择一个随机的离线顶点排列，然后在每个在线顶点到达时，根据这个排列将其与第一个未匹配的邻点进行匹配（如果该邻点存在）。

定理 1.1 [36]：针对无权在线二分图匹配问题，Ranking 算法实现了$1-1/e$的竞争比。

定理 1.2 [36]：针对无权在线二分图匹配问题，任何随机算法都无法实现大于$1-1/e$的竞争比。

1.1 加权模型

经典的无权模型有多个加权版本，我们将在下文详述。

点权版本：在点权版本中，每个离线顶点u与一个权重$w(u)$相关联，目标是最大化匹配到的离线顶点的总权重。Aggarwal 等人 [1] 泛化了 Ranking 算法，并同样获得了最优竞争比$1-1/e$。

边权：在边权版本中，不同的边可以具有不同的权重，目标是最大化被选定匹配的总权重。研究发现，即使在只有一个离线顶点的情况下，该模型版本也不存在任何非平凡的研究成果。一个常见假设是可抛弃假设 [16, 39]。它允许抛弃之前已匹配的边，以便匹配新的、权重更大的边。2020 年，Fahrbach 等人 [15] 的算法达到了 0.5086 的竞争比，这一结果首次突破了长期存在的 1/2 竞争比瓶颈。之后，[4, 22] 又将该竞争比进一步提升到了 0.536.

AdWords：在 AdWords 问题中，每个离线顶点u与一个预算B_u相关联，它的每个边与一个出价b_{uv}相关联。那么，将该顶点匹配到一组在线顶点S的收益等于其预算和其关联边的总权重（出价）之间的最小值，即$r_u=\min\left\{B_u,\sum_{v\in S}b_{uv}\right\}$。因此，该问题对应的模型版本也被称为加和预算版本。AdWords 问题存在一个广为人知的低出价假设，即假设每个边的出价b_{uv}与相应的B_u相比非常低。在此假设下，[47] 提出的 MSVV 算法实现了最优竞争比$1-1/e$。然而，如果允许一般出价，那么直到 2020 年才有算法首次突破了 1/2 竞争比瓶颈。该算法由 Huang、Zhang 和 Zhang [31] 提出，其竞争比达到了 0.5016。

次模收益：次模收益最大化模型是最泛化的模型。在此模型中，每个离线顶点u都与一个估价函数f_u相关联。当在线顶点到达时，算法可以将在线顶点分配给离线顶点，该

分配是不可撤销的。最后，每个离线顶点u获得的收益为$f_u(V_u)$，其中V_u是分配给它的在线顶点的集合。该算法的目标是最大化离线顶点的总收益，即$\sum_{u\in L} f(V_u)$。不难发现，上面所有模型都是这个泛化模型的特例（如果边权模型采用可抛弃假设）。Lehmann 等人 [42] 证明了贪心算法在这个模型中可以实现 1/2 竞争比，即使在随机算法中，这也是最优的了。

不可竞争性结果：点权、边权（可抛弃假设）、AdWords（一般出价）和次模收益模型都泛化了无权模型，因此他们与无权模型的最差情况是一样的，这里的无权模型指的是由 Karp 等人 [36] 实现的竞争比为$1-1/e$的版本。值得注意的是，即使计算能力没有限制，他们也会受限于这个上限。至于具有低出价假设的 AdWords 问题，虽然它不是直接基于无权问题泛化而来，但 Mehta 等人 [47] 证明了在此问题上任何随机算法的竞争比都无法突破$1-1/e$。如果将算法限制在多项式时间算法上，除非 NP = RP，否则在次模收益问题上任何随机算法的竞争比都无法超过 1/2。

1.2 随机分布到达模型

除了提出经典模型的加权版本外，研究人员还试图通过放宽最差情况下的到达顺序限制来绕过$1-1/e$这个障碍。他们采用均匀随机到达模型和 i.i.d. 到达模型，并重新定义竞争比。这两个模型都具有前面小节中提到的加权版本。值得注意的是，均匀随机到达模型的难度严格大于 i.i.d. 到达模型。也就是说，如果采用相同的加权版本和算法，那么均匀随机到达模型的竞争比最多与 i.i.d. 模型持平。 下面我们将给出这两种到达模型的竞争比定义，然后再证明这一观察结果。

- **均匀随机到达模型**

在均匀随机到达模型中，在线顶点集R有一个均匀随机排列$\pi(R)$，我们将根据算法在该随机排列中的期望性能来评估算法。这里，竞争比定义为算法的期望解与所有可能潜在图中的最优离线解的最小比。

均匀随机到达模型的竞争比（最差图和均匀随机到达假设）

$$\Gamma = \min_{G} \frac{\mathbb{E}[\mathrm{ALG}(G,\pi(R))]}{\mathrm{OPT}(G)}$$

研究成果：如果将均匀随机到达模型应用于无权在线二分图匹配问题，Mahdian 和 Yan 证明了 Ranking 算法的竞争比可以达到 0.696 [43]（Karande 等人 [35] 证明竞争比可以达到 0.656）。该模型于 2011 年提出，而这两个研究成果最早超越了贪心算法的$1-1/e$。 Karande 等人通过一个例子说明 Ranking 算法的竞争比上限为 0.726。Mahdian 和 Yan 则指出，即使在已知 i.i.d. 模型中，任何（随机）算法的竞争比上限都无法超过 0.83。 因为 i.i.d. 模型的难度严格小于均匀随机到达模型，所以均匀随机到达模型也存在一个上限。

在点权版本中，直到 2018 年才由 Huang 等人 [28] 打破了$1-1/e$这个界限。他们证明了加权 Ranking 算法的竞争比为 0.653，而 Jin 和 Williamson [33] 又将这一结果提高到了 0.662。

在具有低出价假设的 AdWords 问题中，Goel 和 Mehta [23] 采用贪心算法也可以达到$1-1/e$。在之后的 2009 年，Devanur 和 Hayes [9] 提出了接近最优竞争比$(1-\epsilon)$的算法，而 Mirrokni 等人则证明 MSVV 算法的竞争比为 0.76。之前我们提到，在具有低出价假设的最差情况模型中，MSVV 的竞争比可达到$1-1/e$。如果允许在 AdWords 问题中采用一般出价，那么我们还不确定，是否可以对更差情况模型或泛化次模收益模型（以及均匀随机到达模型）进行改进。

在次模收益模型中，Korula 等人 [40] 通过采用均匀随机到达假设率先使竞争比突破了 0.5，而 Buchbinder 等人 [7] 又将这一竞争比提高到了 0.5096。

- **i.i.d. 到达模型**

在 i.i.d. 到达模型中，算法预先知道在线顶点将从几种可能类型的一个给定分布中采样，而顶点的类型刻画了它所关联的边以及边上的权重。之后，所有到达的在线顶点都将从这个给定分布中独立采样。注意，此模型中的图G本身是随机化的；因此，OPT (G)相应也是随机的。竞争比定义为 ALG (G) 的期望值（基于算法和图的随机性）与 OPT (G)的期望值（基于G的随机性）的最小比。

i.i.d. 到达模型的竞争比（最差图和最差分布）

$$\Gamma = \min_{D} \frac{\mathbb{E}[\mathrm{ALG}(G)]}{\mathbb{E}[\mathrm{OPT}(G)]}$$

为什么 i.i.d. 到达模型易于均匀随机到达模型？在 i.i.d. 到达模型中有两个特定的版本：已知 i.i.d. 和未知 i.i.d.。它们之间的区别在于是否将分布预先提供给算法。 我们可以证明 i.i.d. 模型易于均匀随机到达模型。也就是说，如果一个算法在均匀随机到达模型中的竞争比至少是Γ，那么即使不知道任何分布信息，它在 i.i.d. 模型中的竞争比也可以达到Γ。 i.i.d. 模型的概率空间中的事件可以被视为一个随机图加上在线顶点的一个均匀随机到达模型。 考虑到每个可能的图，均匀随机到达模型中具有Γ竞争比的算法的期望解优于 OPT 的Γ倍。因此，基于图的随机性，期望解至少等于期望 OPT 的Γ倍。

研究成果：在无权版本中，Feldman 等人 [17] 的算法在整数速率假设下首次突破了$1-1/e$竞争比。 在没有这一假设的情况下，Manshadi 等人 [44] 率先给出了他们的成果，后来 Jaillet 和 Lu [32] 将竞争比提升至 0.706（这在 2013 年之前是最高的）。而在 2021 年，Huang 和 Shu 给出了最先进的算法，其竞争比达到了 0.711。

对于点权版本，Huang 和 Shu [26] 将均匀随机到达

模型中的具有 0.662 竞争比的算法进行了改进，使竞争比达到了 0.7009。

在采用可抛弃假设的边权版本中，我们可以在未知 i.i.d. 和已知 i.i.d. 两种模型中都实现$1-1/e$，这是因为采用了针对次模收益问题的算法，而该算法在未知 i.i.d. 模型中的竞争比为$(1-1/e)$。

针对具有低出价假设的 AdWords 问题，Devanur 等人 [10] 给出了一个更好的接近最优的算法。与另外一个接近最优的算法 [11] 相比，虽然它们都达到了$1-\epsilon$的竞争比，但参数ϵ在未知 i.i.d. 模型中表现更优，而在已知 i.i.d. 模型中甚至更优。 如果允许一般出价，我们仅知道 Devanur 等人 [10] 证明了贪心算法在未知 i.i.d. 模型中实现了$1-1/e$的竞争比。

针对次模收益问题，Kapralov 等人 [34] 展示了未知 i.i.d. 模型中的一个具有$(1-1/e)$竞争比的算法，这一结果也是已知 i.i.d. 模型的最优结果。

不可竞争性结果：这些随机分布模型确实有助于在最差情况分析中绕过难度障碍，但我们能做到的最好结果是什么呢？ Manshadi 等人 [44] 证明，在无权已知 i.i.d. 到达模型中，任何随机算法的竞争比都无法突破 0.823。注意，已知 i.i.d. 是均匀随机、未知 i.i.d. 和已知 i.i.d. 达到模型中最容易的随机分布模型，因此 0.823 也可以作为这三个模型的泛化模型的上限，这些模型包括无权、点权、边权、AdWords（一般出价）以及次模收益模型。

在具有低出价假设的 AdWords 问题中，[10] 证明了任何随机算法的竞争比都无法达到$1-o(\sqrt{\gamma})$，其中γ表示出价与预算的最大比，其在低出价假设下足够小。

在计算能力有限的情况下，Kapralov 等人 [34] 证明了在未知 i.i.d. 到达的次模收益模型中，除非 NP = RP，任何随机算法的竞争比都无法突破$1-1/e$。

1.3 研究成果总结

我们在以下两个表格中总结了单侧顶点到达模型的不同版本的研究成果。

表 1 使用**粗体**突出显示 2013 年之后的最新结果，箭头表示对应的竞争比是基于一个更强的结果得出的。例如，在无权未知 i.i.d. 模型中，竞争比 0.696 旁边的左箭头表示该竞争比来自（更难的）无权均匀随机到达模型中的 0.696。

表 2 给出的是模型的不可竞争性结果。星号*表示对应的结果建立在有限计算能力的假设下，参数γ表示 AdWords 问题中的出价与预算的比。

表 1 单侧顶点到达模型的最新算法竞争比结果

	最差情况	均匀随机到达	未知 i.i.d.	已知 i.i.d.
无权	$1-1/e$([36])	0.696([43])	0.696(←)	**0.711**([26])
点权	$1-1/e$([1])	**0.662**([33])	0.662(←)	**0.701**([26])
AdWords（一般出价）	**0.5016**([31])	0.5096(↓)	$1-1/e$([10])	$1-1/e$(←)
次模收益	1/2	**0.5096**([40])	$1-1/e$([34])	$1-1/e$(←)
边权（可抛弃假设）	**0.536**([4, 22])	0.536(←)	$1-1/e$(↑)	$1-1/e$(↑)
AdWords（低出价）	$1-1/e$([47])	$1-\epsilon$([11])	$1-\epsilon$([10])	$1-\epsilon$([10])

表 2 单侧顶点到达模型的最新不可竞争性结果

	最差情况	均匀随机到达	未知 i.i.d.	已知 i.i.d.
无权	$1-1/e$([36])	0.823(→)	0.823(→)	0.823([44])
点权	$1-1/e$(↑)	0.823(→)	0.823(→)	0.823(↑)
边权（可抛弃假设）	$1-1/e$(↑)	0.823(→)	0.823(→)	0.823(↑)
AdWords（低出价）	$1-1/e$([47])	$1-o(\sqrt{\gamma})$(→)	$1-o(\sqrt{\gamma})$(→)	$1-o(\sqrt{\gamma})$([10])
AdWords（一般出价）	$1-1/e$(↑)	0.823(→)	0.823(→)	0.823(↑)
次模收益	1/2(*, [34])	$1-1/e$(*, →)	$1-1/e$(*, [34])	0.823(↑)

1.4 其他版本

在线匹配问题提出的主要动因是在线广告，上述模型不能准确地覆盖现实世界中的所有场景。所以，人们仍然试图定义不同的模型来应对现实应用中的各个方面。在本节中，我们将列出其中的一些方向。

在针对受限图实例的在线匹配问题研究过程中，人们又发现了有趣的研究课题。这里列举几个，他们都考虑了图中顶点的有界度数。Buchbinder 等人 [5] 证明，如果每个在线顶点的度数最多为d，那么可以通过确定性算法实现$1-(1-1/d)^d$的竞争比。Azar 等人 [3] 证明，即使在随机算法中，该竞争比也是最优的。如果进一步限制离线顶点的度数至少为k , Naor 和 Wajc [49] 证明了确定性算法可以获得$1-(1-1/d)^k$的竞争比。研究表明，在各顶点的度数均为d的正则图上，确定性算法可以获得$1-(1-1/d)^d$的竞争比。针对d − 正则图，Cohen 和 Wajc [8] 后来提出了一种随机算法，该算法实现了$1-O(\frac{\sqrt{\log d}}{\sqrt{d}})$的竞争比。

Mehta 和 Panigrahi [46] 提出了一种称为随机分布收益（Stochastic Rewards）的模型。在经典的在线二分图匹配模型中，我们决定匹配一条边后就会获得一个收益，然后匹配的在线和离线顶点就消失了。然而，在现实场景中，即使我们向在线用户推送了广告，用户的点击率也只是预计值。如果在线用户没有点击广告而离开了，我们仍然可以再次匹配未点击的离线广告。为此，随机分布收益模型将每个边与成功概率关联起来。在决定不可撤销地将他们匹配后，我们可以知道用户（在线顶点）是否点击了广告，目的是使得成功匹配的离线顶点数达到最大。Mehta 和 Panigrahi [46] 率先研究了等概率这个特殊情况下的模型，并提出了具有 0.534 竞争比的算法，其上限为 0.621。他们还研究了消失概率这个特殊情况，即所有成功概率都趋于零。他们指出，在等概率和消失概率情况下，竞争比都为 0.567。后来，Huang 和 Zhang [27] 将该竞争比提高到了 0.576。Mehta 等人 [48] 提出的算法仅在消失概率情况下就可以达到 0.534 竞争比，Huang 和 Zhang [27] 又将该竞争比提高到了 0.572。

Feng 等人 [20] 发起了两阶段随机分布在线二分图匹配模型的研究。在该模型中，一侧的顶点被分成两组，D_1和D_2。在第一阶段，将D_1提供给算法，算法为这组顶点做出匹配决策。在第二阶段，根据一个预知的分布来随机采样D_2，并且基于图的实现进行决策。他们在这个问题上获得了最优竞争比 3/4。

Feng 和 Niazadeh [19] 研究了经典的点权二分图匹配模型和 AdWords 模型的K阶段版本，即在线顶点以K个批次到达。他们为这两个模型设计了最优的部分匹配算法，其竞争比达到$(1-(1-1/K)^K)$。

2 超越单侧顶点到达的模型

2.1 泛化到达模型

除了泛化加权模型，在线匹配领域的另一个有趣的课题是通过到达模式来泛化单侧在线二分图匹配模型。单侧在线模型假设基础图是二分图，单侧顶点是预先给定的。那么我们会自然想到：

如果允许所有顶点都在线到达会怎么样呢?

泛化顶点到达：Wang 和 Wong [51] 在 2015 年率先提出了泛化顶点到达模型。假设$G=(V,E)$是基础图，V中的所有顶点以在线方式到达。在每个顶点到达时，可以得到它与已到达顶点之间的关联边。我们可以在v到达时将v与已到达但未匹配的邻点进行匹配，或者等以后再匹配。（注意，由于不要求一侧顶点必须是离线顶点，因此自然会想到在一般图上定义问题。）

Wang 和 Wong [51] 研究了模型中的在线分数松弛算法，并提出了一种具有 0.526 竞争比的基于对偶问题的算法。2019 年，Gamlath 等人 [21] 率先通过整数算法突破了 1/2 竞争比瓶颈，实现了一个小常数ϵ的突破。在不可竞争性研究方面，Buchbinder 等人 [6] 的最新研究表明，即使是（比任何随机积分算法都强的）分数松弛算法，也无法实现 0.591 的竞争比；该算法的竞争比最近提高到了 0.583。

接下来，我们考虑其中的在线算法。注意，我们只允许算法在u和v中的较晚到达时间显示边(u,v)并做出匹配决策。然而，为什么算法在得到更多边后不能做出更好的选择呢？在下面的模型中，我们允许算法在得到边之后并且在u和v之中最多一个离开后才匹配边。

完全在线（带截止时间的顶点）：Huang 等人 [29] 首先提出了完全在线匹配模型。假设$G=(V,E)$是基础图，V中的所有顶点以在线方式到达，每个边(u,v)都可以在u和v均到达之后显示。每个顶点还对应一个截止时间，我们可以在截止时间前的任何时间匹配顶点。如果两个顶点之间没有边，那么一个顶点可以晚于另一个顶点的截止时间到达。

Huang 等人 [29] 率先研究了完全在线模型中表现良好的 Ranking 算法，并证明了该算法的竞争比在二分图中为 0.567（Ranking 算法所能达到的最优竞争比），在一般图中可以达到 0.5211。之后，Huang 等人 [30] 给出了平衡 Ranking 算法，将竞争比提高到 0.569。此外，他们还提出一个分数松弛算法，实现了 0.592 的竞争比（最近提高到了 0.6）。在不可竞争性研究方面，Eckl 等人 [12] 证明了任何分数松弛算法的竞争比都无法超越 0.629。

实际上，我们可以进一步泛化到达模式。

边到达：Gamlath 等人 [21] 研究了最泛化和最难的到达模型，即边到达模型。假设$G=(V,E)$是基础图，E中的所有边以在线方式到达。当一个边到达时，需要立即决定是否匹配它（如果能匹配）。

不幸的是，Gamlath 等人 [21] 给出了一个不可竞争性结果，即在该模型中，没有分数松弛算法的竞争比能超过 0.5。因此，平凡的贪心算法已经是最优的了。

总结和比较 关于研究成果的总结，请参见表 3。按照不可竞争性可以将三种模型进行排序，即，完全在线≤泛化顶点到达≤边到达。注意，表中的结果已经显示了这三种模型之间的差异。完全在线模型中的一个分数松弛算法已经突破了泛化顶点到达模型中的不可竞争性瓶颈，因此，至少从分数松弛算法的角度来看，完全在线模型严格易于泛化顶点到达模型。此外，边到达模型的不可竞争性结果为 1/2，因此，该模型严格难于其他两种模型。

2.2 加权模型

接下来，我们将讨论上述不同到达模型的边权版本。我们首先注意到，如果不做任何额外的假设，就无法获得任何非平凡的理论研究成果。事实上，即使在只有一个离线顶点的经典单侧到达模型中，如果考虑边权，将无法实现任何具有常数竞争比的算法。

文献显示，两个最流行的模型都对在线顶点的均匀随机到达顺序或在线顶点的随机分布信息进行了假设。这两个模型对应的问题也被称为秘书匹配问题和预知匹配问题，分别由经典的秘书问题和预知不等式问题泛化而来。注意，在预知模型中，我们假设在线顶点是独立地从已知分布中采样，而分布不一定是相同的。为了与前几节的概念更一致，我们将该模型称为随机分布非 i.i.d. 模型。还需注意的是秘书模型和预知模型在一般情况下是不可比较的。

除了秘书模型和预知模型外，Ashlagi 等人 [2] 还提出一个在线时间窗匹配模型，该模型类似于文献 [29] 提出的完全在线匹配模型，适用于边权图。为了取得非平凡的理论研究成果，该模型做出了额外的假设。

我们在表 4 中总结了所有已知结果，并在后面进行了详细说明。

单侧顶点到达：研究表明，在经典单侧到达模型中，已经存在可以达到最优竞争比的算法。Kesselheim 等人 [37] 将秘书问题的$1/e$经典算法泛化为单侧在线二分图匹配模型。Feldman 等人 [18] 将预知不等式问题的 1/2 经典算法泛化为单侧在线二分图匹配模型。这两个竞争比都是最优的。

泛化顶点到达：Ezra 等人 [13, 14] 研究了泛化顶点到达模型中的秘书匹配和预知匹配，并实现了最优竞争比 5/12 和 1/2。有趣的是，虽然泛化顶点到达模型来自最差到达顺序的单侧到达模型，但是在均匀随机到达假设下的最

表 3 最差情况下不同到达模型的最新结果

	整数（随机）	分数	不可竞争性
单侧	$1-1/e$	$1-1/e$	$1-1/e$
完全在线（二分图）	0.569	0.6	0.613
完全在线（泛化）	0.5211	0.6	0.613
泛化顶点到达	1/2 + ϵ	0.526	0.584
边到达	1/2	1/2	1/2

表 4 不同到达模型的边权版本的最新结果

边权	最差情况	均匀随机到达	随机分布非 i.i.d. 到达
单侧（不抛弃）	-	$1/e$	1/2
泛化顶点到达	-	5/12	1/2
边到达	-	1/4	0.337
在线时间窗	1/4	0.279	-

优竞争比$1/e$并没有得以延续。Ezra 等人 [13] 的研究表明，当所有顶点以随机顺序在线到达时，匹配确实更容易。

边到达：在算法博弈论文献中，对于边到达模型中的秘书匹配和预知匹配有更广泛的研究。最新研究成果显示，Ezra 等人 [13, 14] 的算法达到了竞争比 1/4 和 0.337。在已知的不可竞争性结果中，一个是上限$1/e$，这个是针对由经典秘书问题泛化而来的秘书匹配问题；另一个是上限 3/7，这个来自 Pollner [50]。在此之前，也有对预知模型的研究，分别是 [24] 针对二分图的研究以及 [38] 在拟阵交集约束下的研究，后者将二分预知匹配作为一个特殊情况来对待。对于秘书模型，Kesselheim 等人 [37] 通过将超图中的边到达缩减为顶点到达，实现了$1/2e$竞争比。

边权在线时间窗匹配：与 Huang 等人 [29] 同期展开研究的还有 Ashlagi 等人 [2]，他们引入了在线时间窗匹配模型。该模型类似于完全在线匹配模型，但有个先进先出的额外假设，这意味着较早到达的顶点也将较早地离开。有了这个额外的假设，Ashlagi 等人 [2] 为边权图提供了具有 1/4 竞争比的算法，并且为均匀随机到达模型提供了具有 0.279 竞争比的算法。不可竞争性研究表明，这些算法的上限是 1/2。

2.3 其他版本

最近，Gravin 等人 [25] 研究了在二分图上采用边到达模型的无权随机匹配，并实现了具有 0.503 竞争比的算法，突破了最差情况下的 1/2 竞争比瓶颈。

Lee 和 Singla [41] 提出了在线匹配中的批量边到达模型，即在每个时间步长中批量显示图中的边。如果图在两个阶段中呈现，他们设计的算法可以达到 2/3 的竞争比。如果图在s个阶段呈现，他们提出的随机算法可以达到$\frac{1}{2}+2^{-O(s)}$的竞争比。

参考文献

[1] G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *SODA*, pages 1253–1264, 2011.

[2] I. Ashlagi, M. Burq, C. Dutta, P. Jaillet, A. Saberi, and C. Sholley. Edge weighted online windowed matching. In *EC*, pages 729–742, 2019.

[3] Y. Azar, I. R. Cohen, and A. Roytman. Online lower bounds via duality. In *SODA*, pages 1038–1050. SIAM, 2017.

[4] G. Blanc and M. Charikar. Multiway online correlated selection. *CoRR*, abs/2106.05579, 2021.

[5] N. Buchbinder, K. Jain, and J. Naor. Online primal-dual algorithms for maximizing ad-auctions revenue. In *ESA*, volume 4698 of *Lecture Notes in Computer Science*, pages 253–264. Springer, 2007.

[6] N. Buchbinder, D. Segev, and Y. Tkach. Online algorithms for maximum cardinality matching with edge arrivals. *Algorithmica*, 81(5):1781–1799, 2019. doi: 10.1007/s00453-018-0505-7. URL https://doi.org/10.1007/s00453-018-0505-7.

[7] N. Buchbinder, M. Feldman, Y. Filmus, and M. Garg. Online submodular maximization: beating 1/2 made simple. *Math. Program.*, 183(1):149–169, 2020. doi: 10.1007/s10107-019-01459-z. URL https://doi.org/10.1007/s10107-019-01459-z.

[8] I. R. Cohen and D. Wajc. Randomized online matching in regular graphs. In A. Czumaj, editor, *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2018, New Orleans, LA, USA, January 7-10, 2018*, pages 960–979. SIAM, 2018. doi: 10.1137/1.9781611975031.62. URL https://doi.org/10.1137/1.9781611975031.62.

[9] N. R. Devanur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In J. Chuang, L. Fortnow, and P. Pu, editors, *Proceedings 10th ACM Conference on Electronic Commerce*

(EC-2009), Stanford, California, USA, July 6–10, 2009, pages 71–78. ACM, 2009. doi: 10.1145/1566374.1566384. URL https://doi.org/10.1145/1566374.1566384.

[10] N. R. Devanur, K. Jain, B. Sivan, and C. A. Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *J. ACM*, 66(1):7:1–7:41, 2019. doi: 10.1145/3284177. URL https://doi.org/10.1145/3284177.

[11] N. R. Devenur and T. P. Hayes. The adwords problem: online keyword matching with budgeted bidders under random permutations. In *ACM Conference on Electronic Commerce*, pages 71–78, 2009.

[12] A. Eckl, A. Kirschbaum, M. Leichter, and K. Schewior. A stronger impossibility for fully online matching. *Operations Research Letters,* to appear.

[13] T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. Secretary matching with general arrivals. *CoRR*, abs/2011.01559, 2020.

[14] T. Ezra, M. Feldman, N. Gravin, and Z. G. Tang. Online stochastic max-weight matching: Prophet inequality for vertex and edge arrival models. In *EC*, pages 769–787. ACM, 2020.

[15] M. Fahrbach, Z. Huang, R. Tao, and M. Zadimoghaddam. Edge-weighted online bipartite matching. In *FOCS*, pages 412–423. IEEE, 2020.

[16] J. Feldman, N. Korula, V. S. Mirrokni, S. Muthukrishnan, and M. Pál. Online ad assignment with free disposal. In *WINE*, pages 374–385, 2009.

[17] J. Feldman, A. Mehta, V. S. Mirrokni, and S. Muthukrishnan. Online stochastic matching: Beating 1-1/e. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 117–126. IEEE Computer Society, 2009. doi: 10.1109/FOCS.2009.72. URL https://doi.org/10.1109/FOCS.2009.72.

[18] M. Feldman, N. Gravin, and B. Lucier. Combinatorial auctions via posted prices. In *SODA*, pages 123–135. SIAM, 2015.

[19] Y. Feng and R. Niazadeh. Batching and optimal multi-stage bipartite allocations (extended abstract). In *ITCS*, volume 185 of *LIPIcs*, pages 88:1–88:1. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[20] Y. Feng, R. Niazadeh, and A. Saberi. Two-stage stochastic matching with application to ride hailing. In *SODA*, pages 2862–2877. SIAM, 2021.

[21] B. Gamlath, M. Kapralov, A. Maggiori, O. Svensson, and D. Wajc. Online matching with general arrivals. In *FOCS*, pages 26–37. IEEE, 2019.

[22] R. Gao, Z. He, Z. Huang, Z. Nie, B. Yuan, and Y. Zhong. Improved online correlated selection. *CoRR*, abs/2106.04224, 2021.

[23] G. Goel and A. Mehta. Online budgeted matching in random input models with applications to adwords. In *SODA*, pages 982–991, 2008.

[24] N. Gravin and H. Wang. Prophet inequality for bipartite matching: Merits of being simple and non adaptive. In *EC*, pages 93–109. ACM, 2019.

[25] N. Gravin, Z. G. Tang, and K. Wang. Online stochastic matching with edge arrivals. In *ICALP*, volume 198 of *LIPIcs*, pages 74:1–74:20. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

[26] Z. Huang and X. Shu. Online stochastic matching, Poisson arrivals, and the natural linear program. In *STOC*, pages 682–693. ACM, 2021.

[27] Z. Huang and Q. Zhang. Online primal dual meets online matching with stochastic rewards: configuration LP to the rescue. In *STOC*, pages 1153–1164. ACM, 2020.

[28] Z. Huang, Z. G. Tang, X. Wu, and Y. Zhang. Online vertex-weighted bipartite matching: Beating 1-1/e with random arrivals. *ACM Transactions on Algorithms*, 15(3):1–15, 2019.

[29] Z. Huang, N. Kang, Z. G. Tang, X. Wu, Y. Zhang, and X. Zhu. Fully online matching. *J. ACM*, 67(3):17:1–17:25, 2020.

[30] Z. Huang, Z. G. Tang, X. Wu, and Y. Zhang. Fully online matching II: beating ranking and water-filling. In *FOCS*, pages 1380–1391. IEEE, 2020.

[31] Z. Huang, Q. Zhang, and Y. Zhang. Adwords in a panorama. In *FOCS*, pages 1416–1426. IEEE, 2020.

[32] P. Jaillet and X. Lu. Online stochastic matching: New algorithms with better bounds. *Math. Oper. Res.*, 39(3):624–646, 2014. doi: 10.1287/moor.2013.0621. URL https://doi.org/10.1287/moor.2013.0621.

[33] B. Jin and D. P. Williamson. Improved analysis of RANKING for online vertex-weighted bipartite matching. *CoRR*, abs/2007.12823, 2020.

[34] M. Kapralov, I. Post, and J. Vondrák. Online submodular welfare maximization: Greedy is optimal. In S. Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1216–1225. SIAM, 2013. doi: 10.1137/1.9781611973105.88. URL https://doi.org/10.1137/1.9781611973105.88.

[35] C. Karande, A. Mehta, and P. Tripathi. Online bipartite matching with unknown distributions. In *STOC*, pages 587–596, 2011.

[36] R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *STOC*, pages 352–358, 1990.

[37] T. Kesselheim, K. Radke, A. Tönnis, and B. Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *ESA*, volume 8125 of *Lecture Notes in Computer Science*, pages 589–600. Springer, 2013.

[38] R. Kleinberg and S. M. Weinberg. Matroid prophet inequalities and applications to multi-dimensional mechanism design. *Games Econ. Behav.*, 113:97–115, 2019.

[39] N. Korula, V. S. Mirrokni, and M. Zadimoghaddam. Bicriteria online matching: Maximizing weight and cardinality. In *WINE*, volume 8289 of *Lecture Notes in Computer Science*, pages 305–318. Springer, 2013.

[40] N. Korula, V. S. Mirrokni, and M. Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM J. Comput.*, 47(3):1056–1086, 2018. doi: 10.1137/ 15M1051142. URL https://doi.org/10.1137/15M1051142.

[41] E. Lee and S. Singla. Maximum matching in the online batch-arrival model. *ACM Trans. Algorithms*, 16(4):49:1–49:31, 2020.

[42] B. Lehmann, D. Lehmann, and N. Nisan. Combinatorial auctions with decreasing marginal utilities. *Games Econ. Behav.*, 55(2):270–296, 2006. doi: 10.1016/j.geb.2005.02.006. URL https://doi.org/10.1016/j.geb.2005.02.006.

[43] M. Mahdian and Q. Yan. Online bipartite matching with random arrivals: an approach based on strongly factor-revealing LPs. In *STOC*, pages 597–606, 2011.

[44] V. H. Manshadi, S. O. Gharan, and A. Saberi. Online stochastic matching: Online actions based on offline statistics. *Math. Oper. Res.*, 37(4):559–573, 2012. doi: 10.1287/moor.1120. 0551. URL https://doi.org/10.1287/moor.1120.0551.

[45] A. Mehta. Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.*, 8(4): 265–368, 2013.

[46] A. Mehta and D. Panigrahi. Online matching with stochastic rewards. In *FOCS*, pages 728– 737. IEEE, 2012.

[47] A. Mehta, A. Saberi, U. V. Vazirani, and V. V. Vazirani. Adwords and generalized online matching. *J. ACM*, 54(5):22, 2007.

[48] A. Mehta, B. Waggoner, and M. Zadimoghaddam. Online stochastic matching with un- equal probabilities. In P. Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM- SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1388–1404. SIAM,

2015. doi: 10.1137/1.9781611973730.92. URL https://doi.org/10.1137/1.9781611973730.92.

[49] J. S. Naor and D. Wajc. Near-optimum online ad allocation for targeted advertising. *ACM Trans. Economics and Comput.*, 6(3-4):16:1–16:20, 2018. doi: 10.1145/3105447. URL https://doi.org/10.1145/3105447.

[50] T. Pollner. Two problems in combinatorial optimization under uncertainty, 2020.

[51] Y. Wang and S. C. Wong. Two-sided online bipartite matching and vertex cover: Beating the greedy algorithm. In *ICALP*, pages 1070–1081, 2015.

面向组合优化的学习增强算法设计

黄棱潇，王彧弋，阎翔
理论计算机科学实验室

摘要

组合优化是计算机科学的一大重要领域，广泛应用于各类异构场景。对于组合优化问题的求解，传统研究主要采用理论算法和启发式算法，而近来出现了一些机器学习方面的探索。本文先介绍组合优化的三种范式，即理论算法、启发式算法和机器学习算法，考察如何结合运用这些范式。然后总结机器学习算法的应用面临哪些限制与挑战，并提出克服相关困难的潜在方法和解决之道。本文从理论和应用场景两方面展开论述，旨在研究和设计基于机器学习的统一框架，以增强求解组合优化问题的传统理论算法或启发式算法。

1 问题设定

组合优化是计算机科学中最重要的课题之一，在电子设计自动化（Electronic Design Automation，EDA）、云计算、自动驾驶、编译器、数据库、供应链、金融等诸多行业均有广泛应用。通常认为，组合优化是从有限的对象集中找到一个最优对象。组合优化问题一般很难求解，原因在于很多此类问题往往都是 NP 问题，使用穷举搜索方法不易解。在运筹学、算法理论和计算复杂度理论等领域，都存在大量有关组合优化的研究。典型的组合优化包括旅行商、资源管理、路由、作业调度、装箱和表达式简化等问题，如图 1 所示。

求解组合优化问题的传统方法主要包括理论算法和启发式算法，如贪心算法、混合整数线性规划（Mixed-Integer Linear Programming，MILP）算法和局部搜索算法。传统方法求得的解可能并非最优解，但通常对解的质量或执行时间提供可证明的保证。组合优化问题重要性高，备受大公司重视，包括谷歌 OR-Tools、IBM CPLEX、阿里巴巴 MindOpt、Gurobi 和微软 Z3 在内的许多软件都采用传统方法求解。

近年来，机器学习发展迅猛，研究人员开始探索如何将机器学习模型应用于组合优化，从而提高解的质量。著名的例子如谷歌的 AlphaGo，把启发式算法（蒙特卡洛树搜索）和机器学习模型（深度强化学习）有效结合了起来。其他例子也包括谷歌的 AlphaStar 和 AlphaFold，以及 Facebook 的 ELF OpenGo 和 ReLA。阿里巴巴还引入机器学习模型来改进装箱策略 [34]。

然而，利用机器学习增强理论和启发式算法的研究仍处于起步阶段。而且，由于组合优化的独特性质，机器学习方法的应用还面临一些限制与挑战，涉及敏感性、扩展性和适应性等方面。本文旨在考察理论 / 启发式算法和机器学习算法的各种组合方式，希望通过统一的增强算法框架设计来克服上述挑战。

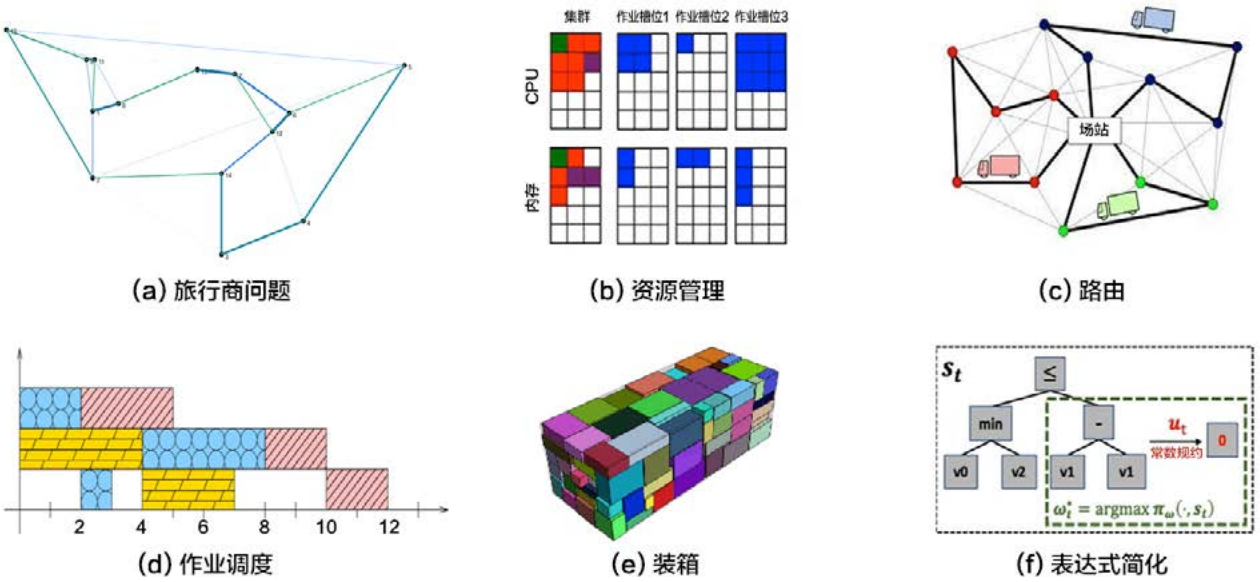
2 背景

组合优化问题的形式通常如下：给定基集 $\mathcal{E}=\{1,2,\ldots,n\}$、可行域 $\mathcal{X}$ 和目标函数 $f:2^{\mathcal{E}}\to R$，需搜索最小化（或最大化）的解，以找到最优解 $x^*\in\mathcal{X}$，使得 $f(x^*)\le f(x)$（或 $f(x^*)\ge f(x)$）对所有 $x\in\mathcal{X}$ 都成立。

本节首先介绍一些重要的组合优化问题；然后考察组合优化算法设计的三种范式，即理论算法、启发式算法和机器学习算法，并分析其优劣；接着介绍这些范式的已有组合；最后再列举经典的理论和启发式方法，并评述这些方法与机器学习的现有组合应用。

2.1 组合优化问题

组合优化问题有很多类型，我们将其大致分为三类——图的组合优化问题、调度与管理问题、逻辑问题，并针对每类问题列举了华为的主要应用示例。



(a) 旅行商问题　(b) 资源管理　(c) 路由

(d) 作业调度　(e) 装箱　(f) 表达式简化

图 1 经典的组合优化问题

图的组合优化问题

- **顶点覆盖**。给定图$G=(V,E)$，有顶点集V和边集E。顶点覆盖问题在于找到具有最小基数$|S|$的集合$S \subseteq V$，使得S包含每条边$e \in E$的至少一个顶点。部分应用如下。
  - 无线：无线基站选址
- **最大割**。给定图$G=(V,E)$，有顶点V和边E。图G的切割是指将G的顶点划分为两个不相交的子集$G_1=(V_1,E_1)$和$G_2=(V_2,E_2)$，且$V_1 \cup V_2 = \emptyset$。切割的大小定义为被切断的边的数量。若某种切割不小于任意其他切割，这种切割就是图G的最大割。部分应用如下。
  - EDA：图分区
- **旅行商问题**。给定有限个城市的集合N和距离矩阵$(c_{ij})(i,j \in N)$，求解$\min_\pi \sum_{i \in N} c_{i\pi(i)}$。其中，$\pi$是指城市$N$的所有循环排列，$\pi(i)$具体表示旅行商从城市$i$沿排列$\pi$抵达的下一个城市。部分应用如下。
  - EDA：印刷电路板钻孔，芯片插装问题
  - 供应链：仓库订单拣货问题
  - 计算机网络布线
- **聚类**。一个图由顶点（也称节点）和连接每对顶点的边构成。图聚类任务是将图的顶点划分为簇，同时考虑图的边结构，使得每个簇内的边较多，而簇之间的边较少。部分应用如下。
  - 云计算：数据分析
  - HMS（Huawei Mobile Services）广告：社区发现
  - 数据库：快速索引

调度与管理

- **完工时间**。给定n个任务的列表，每个任务i在m台机器上并行执行，执行时间为t_i。完工时间问题需要构造分配方案$\pi:[n] \to [m]$，使得完工时间$\max_{j \in [m]} \sum_{i \in [n]:\pi(i)=j} t_i$最短。部分应用如下。
  - 编译器：流水编排
  - 分布式系统：调度
- **装箱**。给定有限列表$L=(a_1,a_2,...,a_n)$，列表值为$(0,1]$范围内的实数。又给定一系列具有单位容量的箱子$BIN_1,BIN_2,...,BIN_m$，箱子从左到右排列。装箱问题是指将上述列表值分配或打包到箱子中，使得任意一个箱子里的数值总和不超过 1，且使用的箱子（即非空的箱子）数量最少。部分应用如下。
  - 供应链：装箱
- **路由**。在 Q 路由中，每个节点x使用一张包含值$Q_x(y,d)$的表进行路由决策。节点x有邻居y和终点d，表中的每个值表示数据包经由邻居y发送到节点d所需的估计时间，该时间不包括数据包在节点x的队列中所花费的时间。当节点x进行路由决策时，会选择$Q_x(y,d)$最小的邻居y，即$Q_y(z,d)=\min_{\tilde{z} \in N(y)} Q_y(z,d)$。部分应用如下。
  - 数字通信：路由器
  - 供应链：车辆调度

逻辑问题

- **约束满足问题（Constraint Satisfaction Problem，CSP）**。该问题表示为一组变量和一组约束。变量即问题的未知数，每个变量都有一个有限的值域。解是指从各变量的值域选取合适的值，构成一个值元组，从而对这组变量进行完整赋值，使赋值满足约束条件。约束则表示某些变量之间的关系，并限定变量的取值。因此，约束满足问题的解就是满足相应约束的赋值。由求解器提供的可用约束集被称为建模语言。由于用户常希望获得由优化条件描述的特定解，因此一种名为约束编程的建模语言被开发了出来，用于求解约束满足问题 [108]。部分应用如下。
  - 编译器：调度，组合等价性检查等
  - 供应链：规划
- **布尔可满足性（Boolean Satisfiability，SAT）**。考虑一个布尔（命题）逻辑表达式，由布尔变量、括号，以及运算符 AND（合取）、OR（析取）和 NOT（取反）组成。字面量是指布尔变量或其否定形式。子句是字面量的析取。布尔表达式是有限个子句的合取。布尔可满足性问题在于找到所有变量的布尔赋值，使给定的表达式为真，或判定这样的赋值不存在。部分应用如下。
  - EDA：模型检验，自动测试模式生成等
  - 编译器：调度，组合等价性检查等
  - 供应链：规划
- **表达式简化**。简化涉及两方面的问题：1）获得等价但更简单的对象；2）计算等价对象的唯一表示。设T为一类语言对象，S为有效过程。第一个问题可以表述为找到函数S，该函数将T映射到T，且对于T中的所有对象t均满足$S(t) \sim t, S(t) \leq t$。第二个问题在于找到函数S，该函数将T映射到T，且对于T中的所有对象s,t均满足$S(t) \sim t, s \sim t \Rightarrow S(s)=S(t)$。部分应用如下。
  - 机器学习编译器：计算表达式简化
  - EDA：功能验证

2.2 组合优化算法设计的三种范式

本节介绍组合优化算法设计的三种范式，包括理论算法、启发式算法和机器学习算法。举例来说，旅行商问题 [40] 是已知的 NP 完全问题，也就是说，目前还没有精确的多项式时间算法能求解这类问题。如果能找到这种精确算法，就意味着$P = NP$。

旅行商问题在规划、物流、微芯片制造等领域有大量的实际应用，因而成为很多理论研究的焦点。一个主要的理论方向是为旅行商问题设计**精确算法**。最直接的办法是尝试顶点的所有排列，并从中找出权重最小的一个。使用这种方法，运行时间处于$O(n!)$多项式因数内，换句话说，即使只有20个顶点，算法的运行也要花费很长的时间。而 Held-Karp 算法 [50] 是一种动态规划算法，可以在$O(n^2 2^n)$时间内求解旅行商问题。若使用其他算法求解，如各种分支限界算法 [75, 101]，则可以获得更好的运行时间性能。不过，尚不清楚是否存在一种精确算法，能够在$O(1.9999^n)$时间内求解旅行商问题 [116]。

另一个主要的理论方法是，设计一种能够在多项式时间内运行的**近似算法**，用来求得近似最优解。一般来说，如果不对边权重加以限制，就无法在$poly(n)$的任何因数内获得旅行商问题的近似解，除非$P = NP$ [100]。但在实践中，许多旅行商问题的应用处于度量空间中，也就是说，边权重满足三角不等式。在这种情况下，Christofides [25] 提出的算法可以近似到 1.5 以内，这被认为是度量旅行商问题的最佳上界。它的当前最佳下界则是 123/122 [65]。而对于欧几里德旅行商问题，则存在一个多项式时间近似方案 (Polynomial-Time Approximation Scheme，PTAS) [5, 89]。

精确算法可以输出最优解，因而具备性能上的优势，但同时也存在时间效率上的代价。另一方面，近似算法虽然具有运行时间上的优势，但最差性能或实际性能都可能表现得差强人意。现有的近似算法虽然提供最佳近似保证，但在部分情况下很难实现。因此，需要从算法实现的角度去研究这些算法的关键思想。

有时也会使用**启发式**方法来求解问题，当问题实例的规模较大时尤其如此。启发式算法旨在找到一个次优解，在实际应用中可以提供良好的性能，尽管这种性能并不具备理论上的保证。不足之处在于，尚缺乏一般性的框架来设计好的启发式算法，使算法既可以泛化应用于任意问题，同时又提供良好的性能。尽管如此，还是有各种启发式搜索技术被开发了出来，利用这些技术，可以在一些高难度的组合优化问题中得到较好的解。元启发式就是其中一种颇具前景的技术，这种技术尝试在高阶框架中融入基本的启发式方法，从而高效地探索给定组合问题的可行解集。部分例子包括模拟退火 [69]、禁忌搜索 [42, 43, 44]、进化算法（如遗传算法）[46]、蚁群优化 [33, 32]、迭代局部搜索 [82]、可变邻域搜索 [92]、散射搜索 [45]、路径重连 [45] 和贪心随机自适应搜索过程 [39] 等。

传统的理论或启发式方法由人工设计，具有可证明的保证。**机器学习算法**则不同，通常由机器自动学习到一个模型，虽然可以实现更优的性能，但模型的训练需要使用大规模的数据集。近年来，机器学习科学家一直在研究如何应用机器学习方法来求解组合优化问题，相关调研请参见 [114, 87, 19]。譬如，针对旅行商问题，就有多项研究提出采用不同的机器学习模型来求解，这些模型包括指针网络 [81, 84]、注意力机制 [96]、Transformer 架构 [71, 30, 117]，structure2vec [68]、图神经网络（Graph Neural Network，GNN）[102, 99] 等。然而，当前的机器学习算法仍存在若干局限性。例如，由于组合优化问题非常复杂，多数模型都存在扩展性问题。此外，多数现有工作只涉及机器学习模型的训练，也即基于输入实例来直接输出解。至于机器学习算法如何与前文提到的现有理论方法或启发式方法相结合，相关的研究尚不多见。

2.3 组合优化的范式组合

接下来，我们介绍各个范式的已有组合。本节先介绍理论算法和启发式算法的几种著名组合，这些组合方式是机器学习兴起之前的主流应用。后续章节则探讨传统的理论算法或启发式算法如何与机器学习算法相结合。

传统的理论分析总是基于最坏情况来分析算法的性能。但在实践中，最坏情况可能不会发生。以线性规划问题为例，其目标是对取决于线性约束的线性函数进行最大化。单纯形法 [28]、椭球法 [67]、内点法 [64] 等算法可用于求解该问题。从理论上可知，最坏情况下，单纯形法的运行需要指数时间 [70, 94]，而内点法和椭球法只需要多项式时间 [67, 64]。但事实上，单纯形法的一些适度优化版本仍然是实践中最常用的算法。单纯形法的经验性能优异，最常见的情况是，其运行时间与变量的数量呈线性关系。Spielman 和 Teng [113] 提出的“平滑分析理论”给出了最令人满意的解释。该理论指出，在“几乎所有”实例上均可证明单纯形法在多项式时间内运行。

3 理论算法与机器学习的组合

传统算法的环境信息极少。虽然有些研究基于属性良好的环境，但假设通常过于强烈，与实际情况不符。同时，当前急速发展的人工智能（具体即指机器学习）可以很好地学习和预测世界。可以预期，借助机器学习及其对世界的充分洞察，可以设计出性能更好的算法。这就引出了“预测式算法”[91] 的概念。作为近年来的热门话题，预期式算法在计数草图 [53]、调度 [103] 和在线匹配 [31] 等方面

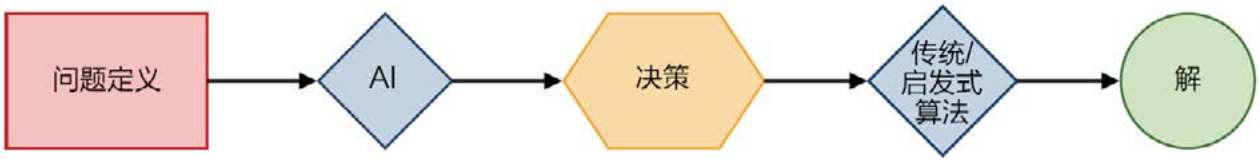


图 2 机器学习模型利用价值信息来增强理论算法或启发式算法

均有应用。其总体思路是，使用机器学习方法提供基于输入的预测，进而又基于预测结果来制定更有效的算法，相应结构请参见图 2。在算法性能的理论分析中，应该将性能看作预测准确率的函数，理想情况下，预测准确率越高，性能就越好。第 3.1 节会详细介绍有关预测式算法的最新工作。

关于组合优化问题的求解，还有一个类似的研究方向，被称为“学习后优化（Learning to Optimize）”，它更深入地研究了学习过程本身如何影响算法设计。求解组合优化问题之前，通常会先建立一个模型或确定一个目标函数 f，再基于模型的输入来设计算法。此处的输入因模型而异，可以是已具备所有模型参数的给定实例（如最短路径、旅行商、最小生成树等问题的加权图），也可以是 f 的隐式值神谕（如覆盖问题的覆盖编号神谕）。然而，实践中可能缺乏足够的信息，无法获得准确的输入，比如，只能通过包含随机性或噪声的历史数据来学习或估计交通道路的通过时间。因此，基于带噪声的已学习实例来设计“鲁棒的”优化算法是一种有益的方法。第 3.2 节将考察有关学习后优化的最新工作。

基于机器学习方法来配置理论算法，另一种思路是利用机器学习指导超参数的选择 [20, 51]。复杂的优化算法通常需要在优化过程中选择一组超参数，比如（随机）梯度下降的学习率或步长。细致选取这些参数可以显著提高优化算法的性能，因此，可以使用机器学习方法来为给定实例学习到好的参数。由于该方向的理论分析很少，本文不作详细考察。

3.1 预测式算法

3.1.1 预测式在线算法

在线算法的作用是处理未来输入数据的不确定性，其质量通常用竞争比来衡量。算法的竞争比定义为最坏情况下算法成本与离线最优值的比值。在我们的设定中，竞争比是指预测器误差 η 的函数 $c(\eta)$。若 $c(\eta) \leq \gamma$ 对所有 η 都成立，则认为算法是 γ-鲁棒的，若 $c(0) = \beta$，则认为算法是 β-一致的。

- **一致性**。若预测准确率增加，算法的性能应随之提升，也即算法中应使用预测器。
- **鲁棒性**。若出现坏的预测结果，算法的性能应随之优雅降级。具体来说，即便使用了不好的预测器，算法性能也应该有适当的界限。

滑雪租赁。滑雪租赁 [104] 是首个采用预测式算法的在线问题。该问题中，滑雪者打算滑雪的天数未知，每天可以按单价租赁滑雪板，也可以按更高的价格 b 购买滑雪板，购买之后就不必再租赁。问题的不确定性在于滑雪的天数，这可以用预测器来估计。对于滑雪租赁问题，最著名的确定性算法是盈亏平衡算法——即前 $b-1$ 天租用，第 b 天购买。容易得出，盈亏平衡算法的竞争比为 2，该值优于任何其他确定性算法。另外，[63] 设计了一种随机算法，得到的最优竞争比为 $\frac{e}{e-1} \approx 1.58$。

设 x 为算法中未知的实际滑雪天数，y 为预测天数，则有预测误差 $\eta = |y - x|$。设 $\lambda \in (0,1)$ 为超参数。针对带预测器的滑雪租赁问题，[104] 先得到一个 $(1+1/\lambda)$-鲁棒且 $(1+\lambda)$-一致的确定性在线算法。该算法的思想是谨慎相信预测，也即，若 $y \geq b$，则在第 $\lceil \lambda b \rceil$ 天购买，否则在第 $\lceil b/\lambda \rceil$ 天购买。接着，研究人员又得到一个 $\left(\frac{1}{1-e^{-(\lambda-1/b)}}\right)$-鲁棒且 $\left(\frac{\lambda}{1-e^{-\lambda}}\right)$- 一致的随机算法（$b$ 为购买成本），通过该算法优化了前一个算法的界限。

（非透视）作业调度。预测式在线算法考虑的第二个问题是非透视作业调度。该问题中，一组作业全部就绪，需要由一台机器来调度完成，在调度过程中，任何一项作业都可以被抢占并稍后继续。算法的目标是最小化各项作业完成的总时长。问题的不确定性在于，调度器在作业实际完成之前并不知道作业的运行时长。需要注意，即使如此，预测器仍然可以基于作业的特征、资源需求及其过往行为进行建模，从而预测作业的运行时长。非透视作业调度由 Motwani 等人 [93] 提出，是长期以来在线算法的一个基本问题。该问题不仅在现实系统中得到显著应用，它的许多变种和扩展也在相关文献中被广泛研究 [16, 15, 55, 54]。Motwani 等人 [93] 的研究显示，循环算法的最优竞争比为2。[104]获得一个 $(2/(1-\lambda))$-鲁棒且 $(1/\lambda)$-一致的随机算法。[56] 则提出另一种误差度量，因为 ℓ_1 误差度量不满足类利普希茨性质——达成完成总时长目标所需要的自然属性。这种新的误差度量可以更好地捕捉目标的敏感特性，研究人员因此得出的算法具有最高达 $(1+\varepsilon)opt + O_\varepsilon(1) \cdot v(p, \hat{p})$ 的竞争比，其中 $v(\cdot,\cdot)$ 即为新的误差度量。

在线缓存（分页）。在线缓存（或在线分页）问题考虑了一个两级内存系统：大小为k的快速内存和（近乎）无限大的慢速内存。缓存算法需要处理一系列对存储元素的请求。如果被请求的元素在快速内存中，则发生缓存命中，算法可以无成本地满足请求。如果被请求的元素不在快速内存中，则发生缓存未命中，算法从慢速内存中获取元素，并将其放入快速内存中，从而满足请求。如果快速内存已满，则必须逐出其中的一个元素。逐出策略是该问题的核心。算法的目标是找到一个逐出策略，使缓存未命中数量最少。

已知的几种算法包括$O(k)$−竞争确定性算法和$O(\log k)$−竞争随机算法[1]，竞争比下界分别为$\Omega(k)$和$\Omega(\log k)$。

在线缓存问题经过扩展后，已经允许神谕不完美地预测被请求元素的下一次到达。[83]证明，如果遵循神谕的建议，即使平均误差相当低，也可能导致极低的性能。该研究还展示了如何修改标记算法，以便将神谕的预测纳入考量，并证明了整体ℓ_1预测误差以η为界时，这种组合方法可以实现$O(1+\min(\sqrt{\eta/opt},\log k))$的竞争比，其中$opt$是最优离线算法的成本。该竞争比随神谕误差的降低而减小，且始终以$O(\log k)$为上界，该上界可以在无神谕输入的情况下实现。[106]提供了一种改进的算法，竞争比为$O(1+\min((\eta/opt)/k,1)\log k)$，下界为$\Omega(\log\min((\eta/opt)/(k\log k),k))$。最近，[115]得到一个确定性算法以及一个随机算法，竞争比分别为$2\min(\min(1+2\eta/opt,2+4\eta/((k-1)opt)),k)$和$(1+\varepsilon)\min(\min(1+2\eta/opt,2+4\eta/((k-1)opt)),H_k)$，其中后者的$H_k$是指第$k$个谐波，且$\varepsilon$平衡和额外成本都具备可加性。[115]还指出，任何学习增强的确定性在线缓存算法的竞争比必须至少为$1+\Omega(\min(\eta/(k\cdot opt),k))$。

值最大化秘书问题。在秘书问题中，秘书集合$\{1,\cdots,n\}$以均匀随机顺序到达，每个秘书有值$v_i\geq 0$（$i\in 1,...,n$）。每当一个秘书到达时，就必须不可撤销地决定是否聘用该秘书。如果决定聘用，则自动拒绝后续的所有候选人。算法的目标是选择值最大的秘书。秘书问题有两个版本。经典秘书问题的目标是最大化选到最佳秘书的概率。另一版本存在少许差异，目标是最大化所选秘书的期望值，这就是所谓的值最大化秘书问题。对于秘书问题的这两个版本，（最优）解[77, 37]都是首先观察n/e个秘书。然后，以这部分秘书的最佳值为参照，从其余的秘书中选取第一个高于该最佳值的秘书。这样一来，对于两个问题版本，都会得到一个$1/e$−近似的结果。

该问题中，机器学习其实是对所有秘书中的最大值$OPT=\max_i v_i$作出预测$p*$，而不是去判定哪个秘书的值最大。[4]指出，对于任意$\lambda\geq 0$和$c>1$，（值最大化）秘书问题都有一个确定性算法，算法的期望竞争比渐近为$g_{c,\lambda}(\eta)$，表示为：

$$g_{c,\lambda}(\eta)=\begin{cases}\max\{\frac{1}{ce},[f(c)(\max\{1-\frac{\lambda+\eta}{opt},0\})]\} & \text{if } 0\leq\eta<\lambda\\ g_{c,\lambda}(\eta)=\frac{1}{ce} & \text{if } \eta\geq\lambda\end{cases}$$

函数$f(c)$是依据Lambert W函数的两个分支W_0和W_{-1}得出，写作$f(c)=\exp\{W_0(-1/(ce))\}-\exp\{W_{-1}(-1/(ce))\}$。

在线二分匹配。在线二分匹配问题中，有二分图$G=(L\cup R,E)$（其中，$|L|=n$且$|R|=m$），其节点集L以均匀随机顺序在线到达[73, 66]。一个节点到达后，会显示与各邻居之间的边权重R。算法必须不可撤销地决定是否把到达的在线节点匹配给R中的某个（当前尚未匹配的）邻居。针对此设定，Kesselheim等人[66]给出了一个紧的$1/e$−竞争确定性算法，对经典秘书算法中的相同保证进行了显著泛化[77, 37]。此设定中的预测结果即为值向量$p*=(p*_1,\ldots,p*_m)$，在某个固定最优（离线）二分匹配中，该向量可以预测节点$r\in R$相邻各边的权重。也就是说，预测值$p*$表示存在一个固定的最优二分匹配，该匹配中的每个节点$r\in R$都与一条权重为$p*_r$的边相邻。于是，预测误差就是取$r\in R$所有节点的最大预测误差，并在所有最优匹配中最小化。这样就实现了经典秘书问题中预测能力的泛化。这种类型的预测与顶点加权在线二分匹配问题高度对应[2]。[4]的研究显示，对于任意$\lambda\geq 0$和$c>d\geq 1$，在线二分匹配问题存在一种确定性算法，算法的期望竞争比渐近为$g_{c,d,\lambda}(\eta)$，表示为：

$$g_{c,d,\lambda}(\eta)=\begin{cases}\max\{\frac{1}{c}\ln(\frac{c}{d}),[\frac{d-1}{2c}(\max\{1-\frac{(\lambda+\eta)|\psi|}{opt},0\})]\} & \text{if } 0\leq\eta<\lambda\\ \frac{1}{c}\ln(\frac{c}{d}) & \text{if } \eta\geq\lambda\end{cases}$$

式中，$|\psi|$为实例中最优（离线）匹配ψ的基数。

装箱。装箱是经典的优化问题，也是一个NP难问题。在装箱问题的在线变种版本中，事先并不清楚物品集，物品是以序列的形式逐一呈现。一个新物品到达时，在线算法将其放入一个已打开的现有箱子中（以不超出箱子容量为前提），或放入一个新的箱子。假设每个物品的大小是$[1,k]$中的一个整数，k表示箱子的容量——这也是[27, 41, 110]等许多有效装箱算法所依赖的自然假设。此外，在不限制物品大小的前提下，[88]的研究发现，如果在线算法输出的大小建议与输入的大小呈次线性关系，那么算法的竞争比不优于1.17（即使该建议为零误差）。这一负面结果意味着，为了利用基于频率的预测，需要对物品大小进行限制。考虑输入序列σ。对于任意$x\in[1,k]$，令$n_{x,\sigma}$表示σ中大小为x的物品数量。定义$f_{x,\sigma}$，表示σ中大小x的出现频率，令该频率等于$n_{x,\sigma}/n$，于是有$f_{x,\sigma}\in[0,1]$。算法使用这些频率作为预测值。具体而言，对于每个大小x（$x\in[1,k]$）在σ中的出现频率，都有一个预测值，表示为$f'_{x,\sigma}$。预测会有误差，所以通常情况下，$f'_{x,\sigma}\neq f_{x,\sigma}$。为量化预测误差，令$f_\sigma$和$f'_\sigma$分别表示$\sigma$中大小$x$的出现频率及其预测值，也即k维空间中的点。根据前文有关预测式在线算法的研究（如[104]），误差η即为f_σ与f'_σ之间距离的L1范数。

度量任务系统。给定包含一系列状态的度量空间M，其中的状态可以解释为动作、投资策略或者某种生产机器的配置。算法从预定义的初始状态x_0开始。在每个时间点$t=1,2,\ldots$，有成本函数$\ell_t: M \mapsto R^+ \cup \{0,+\infty\}$，算法的任务是决定要停留在状态$x_{t-1}$并支付成本$\ell_t(x_{t-1})$，还是要移动到（可能成本更低的）另一状态$x_t$并支付成本$dist(x_{t-1},x_t)+\ell_t(x_t)$，其中$dist(x_{t-1},x_t)$表示在$x_{t-1}$和$x_t$两个状态间迁移的成本。算法的目标是最小化随时间推移所产生的总成本。在每个时间点t，预测器都会针对算法在该时间点应处的状态生成预测值p_t。对于某个离线算法 OFF，预测误差为：

$$\eta = \sum_{t=1}^{T} \eta_t;\ \eta_t = dist(p_t, o_t)$$

式中，o_t表示 OFF 在时间点t所处的状态，T表示输入序列的长度。

[3] 证明了两种一般化的结果。其一，令A为α−竞争的确定性在线算法，用于求解度量任务系统（Metrical Task System，MTS）问题P。存在一种求解问题P的预测式确定性算法，与任意离线算法 OFF 相比，该算法实现了$9\cdot\min\{\alpha, 1+4\eta/OFF\}$的竞争比，其中$\eta$表示相对于 OFF 的预测误差。其二，令$A$为$\alpha$−竞争的随机在线算法，用于求解 MTS 问题$P$，问题的度量空间直径为$D$。对于任意$\varepsilon \le 1/4$，存在一种求解问题$P$的预测式随机算法，实现低于$(1+\varepsilon)\cdot\min\{\alpha, 1+4\eta/\mathrm{OFF}\}\cdot\mathrm{OFF}+O(D/\varepsilon)$的成本，其中$\eta$表示相对于 OFF 的预测误差。于是，若 OFF（近似）最优，且$\eta \le$ OFF，则竞争比接近$1+\varepsilon$。

3.1.2 预测式流算法

数据流模型是一种基础模型，用于在内存有限的情况下快速处理海量数据集。[60] 研究了一种大流量对象神谕，能够预测某个数据元的出现频率是否高于给定阈值。研究表明，这种神谕可以应用于数据流中的广泛问题。借助该神谕，研究人员获得了此类问题的第一个最优界限，有时界限还突破了未使用该神谕时的已知下界。研究中还将该神谕用于流差异中离散元数量的计算、频率矩的估计、级联聚合的估计，以及几何数据流的矩估计。针对离散元问题，该技术能够突破已知下界，从而得到一种新的空间最优算法。对于$0<p<2$时第p个频率矩的估计，该研究获得了空间和更新时间均为最优的首批算法。对于$p>2$时第p个频率矩的估计，则实现了二次方量级的内存节省，突破了无神谕时的已知下界。

设数据流模型中有基础频率向量$x \in \mathbb{Z}^n$，初始化为0^n，该值随着流的过程而演变。该流由形如(i,w)的更新组成，即$x_i \leftarrow x_i + w$。设流的过程中有M，且令$||x||_\infty = \max_{i\in\{1,\ldots,n\}}|x_i| \le M$。在流的结尾处，要求为函数$f: \mathbb{Z}^n \to R$近似出函数$f(x)$。由于流的规模很大，多数算法采用近似和随机的处理方式。通常做法是，算法输出一个估计值Z，值的范围为$P\{(1-\varepsilon)f(x) \le Z \le (1+\varepsilon)f(x)\} \ge 2/3$，式中的概率取决于算法使用的随机种子，而与输入流的随机性无关，即输入流可能为最坏情况。以恒定次数独立重复运行算法，并获取中位数估计，可以将成功概率$2/3$替换为大于$1/2$的任意常数。算法的空间复杂度以比特为单位，目标在于减少存储x所需的空间，使得实际所用空间显著少于平凡n比特空间。

考虑在数据流模型下对以下公共函数$f(x)$进行估计。

- 离散元：$f(x)=||x||_0$，式中$||x||_0$为x的非零坐标数量，定义为$||x||_0 = |\{i: x_i \ne 0\}|$。此数量用于拒绝服务（Denial of Service，DoS）攻击的检测和数据库查询的优化 [61]。
- F_p 矩：$f(x)=||x||_p^p$，式中x的ℓ_p范数定义为$||x||_p = (\sum_i |x_i|^p)^{1/p}$。当$0<p<2$时，这些范数通常比欧几里德范数具有更好的鲁棒性 [57]。当$p>2$时，这些范数用于估计斜率和峰度 [58]。
- (k,p)级联范数：该问题中，x是一个$n\times d$矩阵，可以接收对其单个条目的更新，且有$f(x)=||x||_{k,p} := (\sum_i(\sum_j |x_{ij}|^p)^{k/p})^1/k$。该矩阵有助于对数据库执行多个查询 [26, 59]。

假设有一个大流量对象神谕，基于输入$i\in[n]$，判断流结尾处的x_i是不是大流量对象，并输出判断的结果。大流量对象的定义因数据流问题的类型而异。神谕则分两类，第一类指示$|x_i| \ge T$是否成立，第二类指示$|x_i|^p \ge \frac{1}{T}||x||_p^p$是否成立，其中$T$表示与问题关联的预设阈值。我们将使用第一类神谕来处理离散元问题，而使用第二类神谕来处理所有其他问题。

在x的坐标存在增删的情况下，要估计$L_0=||x||_0$，目前的最佳算法由 [61] 提出。[61] 的研究表明，基于训练好的神谕，有一种算法可以得到$(1\pm\varepsilon)$−近似的L_0，算法使用的空间为$O(\varepsilon^{-2}(\log n)log(1/\varepsilon))$，成功概率至少为$2/3$，更新与上报时间为$O(1)$。

[61] 还得出以下结果。

在大流量对象神谕的完美假设下，可以在$1\pm2\varepsilon$因数内估计$||x||_p^p$，空间复杂度为$O(\varepsilon^{-4}n^{1/2-1/p}\log(n)\log(M))$比特，成功概率至少为$3/5$。考虑神谕给出错误预测的概率为$\delta>0$。那么，在$1\pm2\varepsilon$因数内估计$||x||_p^p$时所需的空间，当$\delta=O(1/\sqrt{n})$时为$O(\varepsilon^{-4}(n)^{1/2-1/p}\log(n)\log(M))$比特，否则为$O(\varepsilon^{-4}(n\delta)^{1-2/p}\log(n)\log(M))$比特。这两种保证都具有至少$3/5$的成功概率。

令$0 < p < 2$且$0 < \varepsilon < 1/2$。如果使用完美大流量对象神谕，存在一个随机算法，输出$(1 \pm 3\varepsilon)||x||_p^p$的概率至少为0.7，使用的空间为$O(\varepsilon^{-2} \max \log(nmM), \log^3(1/\varepsilon) \log\log(1/\varepsilon))$比特，期望的摊销更新时间为$O(1)$，上报时间为$O(1/\varepsilon^2)$。如果使用带噪声的神谕，则假设对于固定的$s$，神谕识别的大流量对象$x_i$不会超过$O(s)$个（即$|xi| > \frac{1}{s}||x||_p^p$）。假设大流量对象神谕出错的概率为$\delta = O\left(\frac{\varepsilon^2}{\log^2(1/\varepsilon)\log\log(1/\varepsilon))}\right)$，那么也存在具有相同保证度的算法。

对于级联范数，设$\varepsilon > 0$和$k \geq p \geq 2$为常数。存在一种随机算法，以坐标增删流的$n \times d$矩阵x为输入，可以按至少$2/3$的概率将$(1+\varepsilon)$-近似的结果输出到$||x||_{k,p}$，使用的空间为$\tilde{O}(n^{1-\frac{1}{k}-\frac{p}{2k}} d^{\frac{1}{2}-\frac{1}{p}})$。

此外，在大流量对象神谕的假设下，存在一个矩形高效的单通流算法，当$p > 2$时，以$1-\delta$的概率将$(1 \pm \varepsilon)$-近似的结果输出到$||x||_p^p$。处理流中的每个矩形时，使用的空间为$O^*(\Delta^{d(1/2-1/p)})$比特，时间为$O^*(\Delta^{d(1/2-1/p)})$。

3.2 学习后优化

本节回顾“学习后优化（Learning to Optimize）”的当前研究进展，考察了“基于样本的优化”和“面向噪声的优化”两个主要研究方向。以下介绍这两个方向涉及的数学模型，以及不同设定下的不可能结果和算法结果。

3.2.1 基于样本的优化

本节将历史数据看作从某个基础分布中提取的样本，目的是研究如何基于样本数据来优化目标函数。考虑以下形式的（约束）组合优化问题：

$$\max_{S \in \mathcal{M}} f(S)$$

式中，$f: 2^N \to \mathbb{R}$为目标函数，$\mathcal{M} \subseteq 2^N$为约束集合。假设存在值神谕$O_f$，可以返回子集$S \subseteq [N]$的值$f(S)$。为评估用于优化的样本的能力，Balkanski 等人 [11] 首次提出以下模型，称之为**基于样本的优化（Optimization from Samples，OPS）**。

定义 3.1：OPS 模型

给定函数$f: 2^N \to \mathbb{R}$，设$\mathcal{S} := \{S_i, f(S_i)\}_{i=1}^t$为样本的集合，其中每个$S_i$都是从某个基础分布$\mathcal{D}$中抽取的独立同分布样本。给定参数$\alpha \in (0,1]$，若要使函数族$\mathcal{F}: 2^N \to \mathbb{R}$对于分布$\mathcal{D}$和约束$\mathcal{M}$是$\alpha$-可近似的，则需存在算法$A$，在给定误差参数$\delta \in (0,1)$并以集合$\mathcal{S}$为输入的情况下，输出结果$S \in \mathcal{M}$并满足：

$$\Pr_{S_1,\ldots S_t \sim \mathcal{D}^t}\left[\mathbb{E}_A[f(S)] \geq \alpha \cdot \max_{T \in \mathcal{M}} f(T)\right] \geq 1-\delta$$

式中，α被称为近似比，t表示样本复杂度。还需注意，此处并不要求A为多项式算法，因为我们关注的是 OPS 模型下样本的功效。

显然，分布$\mathcal{D}$显著影响$\mathcal{F}$的可近似性。譬如，若$\mathcal{D}$总是返回空集或固定的子集，则无法期望从样本中学习到足够的信息用于优化。因此，要考虑分布的“合理性”，并考察在查询模型O_f下可近似的函数到了 OPS 模型下是否仍然可近似，比如次模函数。为此，Balcan 等人 [8] 提出的一个名为**PMAC-可学习性（Probably Mostly Approximately Correct Learnability）**的函数族进入了人们的视野。大体来说，若对于任意子集$S \sim \mathcal{D}$，可以通过样本集合$\mathcal{S}$高概率地学习到$f(S)$，则说函数f对于某个分布$\mathcal{D}$是 PMAC-可学习的。PMAC 包含大量的函数族，其中，OPS 模型下研究得最多的是覆盖函数和影响函数。下面先介绍覆盖函数的定义。

定义 3.2：覆盖最大化

给定二分图$G = (L, R, E)$，L和R分别表示左、右节点集，E表示L和R之间的边集。覆盖函数$f: 2^L \to \mathbb{R}_{\geq 0}$被定义为子集$S \subseteq L$的邻居数量，即$f(S) = |N(S)|$。覆盖最大化问题在于选择一个大小为$k \geq 1$的子集$S \subseteq L$，使得$f(S)$最大化，即$\max_{S \subseteq L:|S|=k} f(S)$。

值得注意的是，覆盖最大化问题是 PMAC-可学习的 [6]，在查询模型下可采用$(1-1/e)$-近似贪心算法求解 [98]。接下来介绍影响函数的定义。可以认为，影响函数是覆盖函数泛化后的一般有向图，它在社交网络中应用甚广。

定义 3.3：影响最大化

给定有向图$G = (V, E, p)$，V为节点集，E为边集，p_e为E中每条边$e \in E$的概率向量，每个节点有已激活或未激活两种状态。给定种子集S_0，令其在时间点$t = 0$处于已激活状态，接着按下述过程激活其他节点：首先，在时间点$t = 1, 2, \ldots$，令$S_t = S_{t-1}$。然后，对于任意$v \notin S_{t-1}$，令$N^{in}(v)$表示节点集合u，且有$(u, v) \in E$。每个节点$u \in N^{in}(v) \cap (S_{t-1} \setminus S_{t-2})$以概率$p_{uv}$独立地激活$v$，最后将所有已激活节点$v$添加到$S_t$。当不再有新的节点被激活时，就得到一个已激活子集的序列$S_0 \subseteq S_1 \subseteq S_2 \subseteq \cdots$。给定$S_0$，定义$\phi(S_0)$为最终已激活节点的集合，并将影响函数$f: 2^V \to \mathbb{R}$定义为$f(S) = \mathbb{E}[|\phi(S)|]$——即种子集$S_0$的期望已激活数量。影响最大化问题在于选择一个大小至多为k的种子集S，使$f(S)$最大化。

需要注意，覆盖函数和影响函数都属于次模函数。[1]

OPS 模型可以自然地分成两个步骤：

1. 基于具备 PMAC−可学习性保证的样本，构造函数$\tilde{f}$，作为对 f 的估计。
2. 在查询模型 $\tilde{f}$ 下优化原有问题。

不过，接下来我们会看到，虽然这两个步骤都不难实现，但一旦组合起来可能就无法得到合理的（近似）算法。下面介绍目前在 OPS 模型下的不可能结果和算法结果，模型包含覆盖最大化和影响最大化这两个问题。

不可能结果。Balkanski 等人 [11] 首先研究了 OPS 模型下的覆盖最大化问题，提出以下定理。

定理 3.4：覆盖最大化问题的不可能结果 [11]

对于任意分布 $\mathcal{D}$，覆盖最大化算法的近似度不优于 $n^{-1/4}$的紧界限，无法实现所观察样本数少于指数数量的目标。

这一结果有些出乎意料，因为覆盖最大化问题已被证明是 PMAC−可学习的 [6]。得出该结果的主要思路是构造一类函数，使这些函数从分布 $\mathcal{D}$ 的多数子集学习时可以获得较好的学习效果，但不足以学习到（近似）最优解，如此一来，样本就无法提供足够的信息用于优化。

不可能结果也存在于其他问题设定中。Balkanski 等人 [12] 构造了一类次模最小化问题 $f: 2^N \to [0,1]$，并证明任何基于多项式样本的算法都无法提供 $(1/2 - O(1))$ 加性近似。此结果可以扩展到凸最小化领域 [13]。总体而言，我们注意到，即使函数是 PMAC−可学习的，在查询模型和 OPS 模型下的可近似性仍可能存在巨大的差别。为克服上述不可能结果，进一步研究的进展如下。

算法结果。一般可以通过三种尝试来获得算法结果。第一种是假设 f 满足额外属性。例如，Balkanski 等人 [10] 考察了曲率为 $c \in [0,1]$ 的单调次模函数。[2] 通过研究，他们提供了一个紧的 $(1-c)/(1+c+c^2)$−近似算法。该算法符合我们对 OPS 模型下线性函数可求解的直觉。不过，覆盖函数和影响函数的曲率都是 1。对于影响最大化问题，Balkanski 等人 [9] 假设社交网络 G 由随机块模型生成，并提出一种常数近似算法。总体而言，这种尝试并没有改变 OPS 模型，因此，需要为不同的优化问题设计特定的属性。

第二种尝试是弱化 OPS 模型的目标。Rosenfeld 等人 [107] 并未对整个函数进行优化，而是提出了一个基于样本的分布性优化（Distributional Optimization from Samples，DOPS）模型。模型的目标是，对于从同一分布中抽取的一个较小样本集合和另一个未知的较大样本集合，可以基于这个较小集合，从较大集合中找到优化的样本。该研究证明，在 DOPS 模型下，可近似性与 PMAC−可学习性是等价的。这种尝试为采样模型下的 PMAC−可学习性提供了解释。然而，我们可能仍希望对合理分布 $\mathcal{D}$ 找到全局最优解。

第三种尝试则假设我们不仅拥有样本 S_i 的值 $f(S_i)$，还获得了额外的结构化信息。Chen 等人 [23] 首先针对覆盖最大化问题，对这一方法开展了研究，提出以下基于结构化样本的优化（Optimization from Structured Samples，OPSS）模型。模型中，每个样本由 $(S_i, N_G(S_i))$ 表示，$N_G(S_i)$为 S_i的邻居集合。该研究证明，如果$\mathcal{D}$满足某项温和的假设（比如$\mathcal{D}$是某种均匀分布），那么可以为覆盖最大化问题求得 $O(1)$−近似的解。此外，Chen 等人 [24] 还将这一结果扩展到影响最大化问题，问题中的各样本表示为 $(S_{i,0}, S_{i,1}, \ldots, S_{i,n-1})$，形式上由完整的传递路径组成。基于$\mathcal{D}$是乘积分布的假设，该研究提供了一个常数近似算法。总结而言，额外的结构化信息可以弥补可学习性和可近似性之间的差距。另外，结构化信息因问题而异，因此考察适用于其他问题的结构化信息将有所裨益。

3.2.2 面向噪声的优化

如前文所述，OPS 模型下的主要困难似乎在于，可能无法基于样本估计来得到最优解的值。那么进一步的问题是，如果能对所有子集进行估计，是否就可以实现优化呢？这就引出另一研究方向——“面向噪声的优化（Optimization with Noises）”，也即可以学习到函数 $\tilde{f}$，作为基础目标 f 的估计，使得对于任意$S \subseteq 2^N$，有 $\tilde{f}(S) \in (1 \pm \varepsilon) f(S)$。此处，$\varepsilon \in (0,1)$表示噪声参数，$\varepsilon$值越小越好。要获得 $\tilde{f}$ 这个估计值，可以基于历史样本，通过神经网络或构建草图的方式来学习。出乎意料的是，即使得到了$\tilde{f}$，优化问题仍可能很难求解。下面讨论误差神谕（Erroneous Oracle）和噪声神谕（Noisy Oracle）两种设定下的情况。

误差神谕。Hassidim 等人 [48] 提出一个误差神谕，对任意$S \in 2^N$，有：

$$\tilde{f}(S) = \varepsilon_S \cdot f(S)$$

式中，$\varepsilon_S \in [1-\varepsilon, 1+\varepsilon]$。此处，$\varepsilon_S$ 可以由对手选择。该研究证明了以下定理，表明使用误差神谕很难求解次模最大化问题。

[1] 若对于任意$S \subseteq T \subseteq N$和$u \notin T$，有$f(S \cup u) - f(S) \geq f(T \cup u) - f(T)$，则认为 f 是次模函数。

[2] 曲率用于测量次模函数趋于线性的程度。曲率越接近 0，次模函数就越趋于线性。

定理 3.5：使用误差神谕时的不可能结果 [48]

有误差为ε的神谕。存在基数约束，使得对误差神谕的查询少于指数次。在此约束下，要使单调次模函数最大化，任何随机算法都无法达到 $n^{-0.5+\delta}$ 的近似度。

该研究的主要思路是构造函数 f_1 和 f_2，两个函数对于几乎所有子集 $S \in 2^N$ 都互相近似，比如 $f_1(S) \in (1 \pm \varepsilon) f_2(S)$。因此，对手可以选择噪声参数 ε_S，使得对于这些子集无法将 f_1 和 f_2 区分开来。然而，f_1 和 f_2 的最优解值存在很大差异。Hassidim 等人 [48] 证明，在此设定中，多项式查询数量不足以区分 f_1 和 f_2。

为消除这种不可能的结果，Hassidim 等人 [48] 提出一种噪声神谕，它的 $\varepsilon_S \sim D$ 是从独立同分布中抽取的样本，而非由对手选择。该研究利用噪声神谕，获得了 $(1 - 1/e - \varepsilon)$ 的近似度。但是，算法所要求的查询数量必须达到 $n^{O(1/\varepsilon)}$，复杂度仍有待降低。

本节从“学习后优化”的两个方向——“基于样本的优化”和“面向噪声的优化”，回顾了相关研究进展。这些研究颇具启发性，但仍然存在若干问题有待解决，譬如，需要有更合理的模型以消除不可能结果，现有算法结果的样本复杂度或查询复杂度也有待降低。

4 启发式算法与机器学习的组合

接下来，我们介绍启发式算法与机器学习的组合。与理论算法相比，在启发式算法中运用机器学习时，可以认为启发式算法是负责高阶结构的控制，同时调用机器学习模型来辅助低阶决策。图 3 总结了这一学习增强框架。算法通常为解决某类一般性问题而设计，换句话说，可以从这类问题中获取任意可求解的实例作为输入，进而输出所需的解。而且，理论研究通常基于所有问题实例的最坏情况来分析算法的属性，如收敛速度和近似比。然而，对于实践中的每个特定实例，算法的性能通常对算法参数和初始解的选择非常敏感（如果没有预先指定算法的初始解）。因此，很自然的思路是，通过学习为算法找到针对具体实例的最佳控制策略，而无需改造算法本身。

本节将这种方法命名为“学习增强算法（Learning Augmented Algorithm）”，并回顾相关研究进展。我们先介绍两个具有代表性的研究方向——“控制策略的学习”和“热启动的学习”，再探讨被称为“适应性学习增强”的新兴方向。

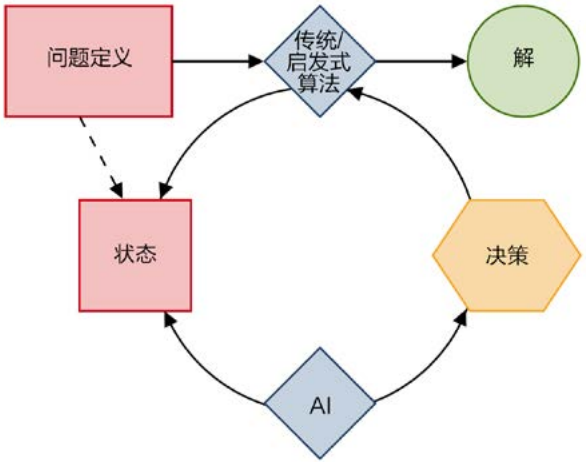


图 3 组合优化算法会重复查询同一机器学习模型以做出决策。机器学习模型以算法的当前状态为输入，算法可包含问题定义。

4.1 控制策略的学习

算法的控制策略主要指超参数，例如迭代算法的更新率，或启发式算法的搜索策略。本节介绍分支限界算法的一个应用示例，并总结与控制策略学习相关的其他类似学习增强算法。

分支限界法简称 BnB（Branch and Bound），主要是指求解离散优化问题的一种范式。考虑基于可行集 S 来最小化实值函数 f 的问题，其中 S 是离散值。BnB 算法通常从单个根状态开始，并维护一个已激活节点的队列 $\mathcal{L}$，该队列容纳 S 的子集，同时用作目标的全局上下界。只要队列 $\mathcal{L}$ 不为空，BnB 算法就使用某种节点选择策略，每一步从 $\mathcal{L}$ 中弹出一个节点 $S_i \in S$，通过必要的松弛、并利用高效求解器，计算出受子集 S_i 限制的目标界限。BnB 算法根据局部界限和全局界限来区分三种可能的情况：

1. 局部下界大于当前全局上界，这意味着，不可能通过进一步生成子集分支的方式来获得更好的解，因而修剪掉该子树（子集）。
2. 局部下界与局部上界匹配，则找到受限于子树（子集）内的最优解。
3. 否则，生成 S_i 的所有分支，并将这些分支推回 $\mathcal{L}$。

在情况 2 和 3 中，会相应更新全局上下界。当 $\mathcal{L}$ 为空或全局上下界的差距足够小时，算法终止。

BnB 算法最重要的控制策略是节点选择策略，准确地说就是，如何从队列中选择下一个子集来计算界限并决定是否生成分支。通常来说，一个好的策略可以尽快找到最优解。这就是说，策略必须先选择适当的分支子集，以便尽快找到好的解，并修剪掉多数不必要的子树（子集）。有几种策略在传统的应用中很流行，包括众所周知的先进

先出和后进先出策略，以及依赖另一优先级队列的最佳优先策略——按局部下界对节点排序就是其中一例。

另一种直接的思路是，利用训练数据来处理待求解的问题，从而学习到一个好的策略。尽管问题和队列都有定制化的表示，但自然的学习过程仍在于学习每个队列中最可能的节点，从而找到问题实例的最优解。

Bai 等人 [7] 应用了此范式来求解两个输入图之间的最大公共子图（Maximum Common Subgraph，MCS）。该研究把待分支的子集设计为输入图之间可能的节点对，利用强化学习，将所找到的公共子图最大化，从而学习到这些节点对。Liu 等人 [79] 也考虑了类似的方法，将学习目标设置为最小化分支生成的次数。相比于基于端到端学习的 MCS 问题求解器，这两种学习增强 BnB 算法都实现了更好的性能。

He 等人 [49] 应用了此范式来解决 MILP 问题。待分支的子集天然就是变量以及这些变量所属的区域。除了学习每个状态的恰当分支变量，该研究还学习到一个剪枝策略，用于预测最优解是否属于某个子树（子集）。剪枝策略进一步促进算法加速，由于已经预测出不包含最优解的多数子树，因而可以忽略这些子树的局部上下界计算。与面向 MILP 问题的 SOTA 求解器相比，这两种控制策略的运用都使 BnB 算法在速度和最优间隙方面获得了更好的性能。还有一些其他的研究路径，针对 MILP 问题设计出不同的机器学习技术，学习到分支限界树上每个节点所做的分支决策 [68, 80, 52]。类似地，针对 MILP 实例，Kruber 等人 [74] 使用机器学习预先确定 Dantzig-Wolfe 分解的应用是否会有效。Bonami 等人 [21] 使用机器学习来确定对问题的线性化处理是否有助于更快地求解。机器学习还可以用于选择 MILP 求解器和启发式构建框架的攻击性参数 [62, 85]。此外，在二次规划 (Quadratic Programming，QP) 领域，Baltean-Lugojan 等人 [14] 利用了机器学习来选择合适的切割平面。

缺点。学习到的控制策略通常由神经网络表示，执行这些策略也会占用计算资源、需要时间预算。在提到的 BnB 示例中，算法每次决定一个分支节点时都会调用策略。这意味着，学习到的控制策略必须大幅提高原有算法的性能，才能覆盖额外的成本。

4.2 热启动的学习

虽然许多启发式算法会将初始解构造为算法的第一个迭代步骤，但与控制策略相比，初始解的好坏似乎并不那么重要。类似情况也出现在许多算法的理论分析中。除专门用于识别初始解的算法以外，随机初始化或任意较差初始化算法的收敛性和最优性通常都得到了证明。

然而，针对特定问题实例，热启动能够预测恰当的初始解，这种能力肯定有助于提高算法的性能。例如，如果迭代算法的初始解足够接近最优解，那么算法获得最优解所需的时间自然就更少。这正是新近一些研究工作的出发点。这些工作基本上就是针对部分问题实例，基于训练数据学习到预测模型，进而获得问题的最优解。而其他工作中的问题实例则应用传统算法，并使用热启动作为预测解。

例如，Bemporad 和 Naikr 的论文 [18]，以及 Masti 和 Bemporad 的论文 [86]，都基于这一思路来学习 BnB 算法的热启动，从而求解 MILP 问题。Duan 等人 [35] 将预测解作为热启动，应用于求解纳什均衡的传统算法，在计算效率和近似最优性方面都取得了相似的改进效果。

缺点。“热启动学习”所采用的方法，存在与“控制策略学习”共通的缺点，即在求解每个问题实例之前，都要进行额外的神经网络推理。而且，针对特定算法学习最优热启动的研究甚少。从理论上讲，尚不清楚是否初始解越接近最优解，相应的热启动就越好。

相反方向的研究。一种有趣的观点是，可以反过来把“热启动的学习”看作是传统算法对机器学习的增强。例如，Li 等人 [76] 设计了一个学习框架来求解组合优化问题，主要涉及三个等效问题——最大团（Maximum Clique，MC）、最大独立集（Maximum Independent Set，MIS）和最小顶点覆盖（Minimum Vertex Cover，MVC）。这个学习框架为每个输入实例输出一组解，并运用局部搜索算法加以改进，最终找到最优解。然而，在 MC、MIS、MVC 的传统求解器中，先使用简单的方法生成初始解，然后运用局部搜索算法，是很常见的做法。换言之，这种学习方式其实就是为局部搜索学习到一组好的热启动。

4.3 适应性学习增强

前两种学习增强方案可以看作是“静态增强”，因为控制策略或热启动的学习模型在增强算法的运行过程中是固定的。最近一项工作 [118] 设计的学习过程则不同，它以每个问题实例为输入，在算法运行过程中进行适应性地学习增强。本节简要介绍这项工作所研究的问题、增强后的启发式算法，以及对应的学习框架。

旅行商问题定义为一个完整的无向图，图中每条边都有一个成本，表示这条边的两个端点之间的距离。该问题在于找到一条哈密顿线路，使线路中各条边的成本总和最小。LKH （Lin-Kernighan-Helsgaun）算法是一种基于局部搜索的著名启发式算法，用于求解旅行商问题。LKH 算法从初始解开始搜索，迭代运行一种名为k-opt 的特殊局部搜索技术。每个k-opt过程会考虑当前解的k条边，测试这些边的 $2k$ 个端点之间的一系列其他 k 条连接，并使用最优选择（最小总成本）来替换这 k 条边。LKH 算法的

关键是根据α值（及其扩展）来选择这k条边。此处忽略详细的计算，人们通常认为，具有较小α值的边更有可能构成旅行商问题的最优解。

Zheng 等人 [118] 设计了一个强化学习过程来取代 2-opt 过程。该过程基于α值确定每条边的初始 Q 值。之后每一步，系统的状态为节点p_i，采取的动作是根据 Q 值选择该节点的一条邻边(p_i, p_{i+1})，而后系统迁移到下一节点p_{i+2}，该节点连接到原有解中的p_{i+1}（这个操作实际是将上一个解中的两条边(p_{i-1}, p_i)和(p_{i+1}, p_{i+2})替换为另两条边(p_i, p_{i+1})和(p_{i+2}, p_{i-1})，而不影响解的可行性）。只要将每个状态动作对的奖励设置为解的改进程度，就可以通过标准强化学习来更新 Q 值。

该方法结合了强化学习技术和传统启发式过程，与 LKH 算法相比，在求解大型旅行商问题实例方面取得了更好的性能。Q 值作为α值的增强，在这个过程中发挥了重要作用。Q 值具有适应性更新的优雅设计，这种设计一方面为学习增强提供了新的潜力，另一方面也依赖于大量有关问题结构和启发式算法本身的领域信息。

5 基于学习的机制设计

机制设计可以看作算法设计的一个特殊过程，该过程认为参与者都是理性的。准确地说，可以定义一个算法，将特定问题的任意实例作为输入，进而返回满足某种预期属性的输出。而机制设计好之后，参与者为机制提供输入实例，并从机制的输出中受益。因此，在设计机制时，必须同时考虑机制的目标和参与者谎报的激励。

机制设计的自然思路是给参与者设计一种博弈，使用参与者的动作作为输入，使得博弈的纳什均衡产生预期的输出。幸运的是，根据著名的显示原理，我们可以专注于真实机制的设计，真实机制会如实报告每个参与者的均衡策略。这是因为，当参与者运用均衡策略时，如果机制优化了一个目标，就会有一个等效的真实机制来采纳参与者的真实报告并模拟最优机制。

不过总的来说，机制仍然很难设计，尤其在各参与者报告多项输入的场景中，这种困难体现得更为明显。传统的研究工作通常会为设计好的机制提供解析形式的实现，并从理论上证明其（近似）最优性。这在一定程度上限制了面向大规模问题的机制设计研究。有所不同的是，一个新兴的研究方向使用了深度神经网络来表示复杂的机制，显示出克服这种困难的潜力。本节将这个研究方向称为“基于学习的机制设计”，并总结了新近的相关研究。我们首先介绍最具代表性的机制设计问题——拍卖设计，也称“与金钱有关的机制设计”，多数研究都与此相关。然后，在“与金钱无关的机制设计”方面，我们再介绍一些基于学习的应用。最后，我们探讨了一种拍卖设计与机器学习的新组合。

5.1 拍卖设计中的机器学习

本节的拍卖设计问题围绕以下设定展开讨论。

- 一个卖家，拥有一个m件物品的集合。
- 一个n名竞拍者的集合，竞拍者具有独立私人估值$v_i : 2^m \to \mathbb{R}_{\geq 0}$，其中$i = 1, 2, \ldots, n$。
- 每个竞拍者i的估值遵循已知分布$v_i \sim F_i$。
- 一个拍卖机制，由分配规则$g(\cdot)$和支付规则$p(\cdot)$表示。当竞拍者i出价b_i、其他竞拍者出价b_{-i}时：
  - $g_i(b_i, b{-}i)$表示物品分配给竞拍者i的概率（向量）；
  - $p_i(b_i, b{-}i)$表示竞拍者i需要支付的金额；
  - 于是，竞拍者i的收益定义为 $u_i(v_i, b_i, b_{-i}) = v_i^T \cdot g_i(b_i, b_{-i}) - p_i(b_i, b{-}i)$。
- 设计一种机制，使得在以下约束下，期望收入$\mathbb{E}_{v_1 \sim F_1, \ldots, v_n \sim F_n}[\sum_i p_i(v_i, v_{-i})]$达到最大化。
  - 激励相容约束：$\forall i, v_i, b_i, v_{-i}, u_i(v_i, v_i, v_{-i}) \geq u_i(v_i, b_i, v_{-i})$
  - 个体理性约束：$\forall i, v_i, v_{-i}, u_i(v_i, v_i, v_{-i}) \geq 0$

Myerson [95] 提供了$m = 1$时的收入最大化拍卖机制，但即使$m = 2$，也不存在完整的解析解结果。之后的许多工作针对多件物品的设定来设计最优拍卖，设定中假设竞拍者数量固定（通常为两个），或估值分布的支撑集是有限的（仅包含两个取值）。其他工作还研究了与最优收入具有恒定近似比的机制设计。

Duetting 等人 [36] 首次引入深度神经网络表示拍卖机制。这种表示方式把拍卖设计问题转移到深度学习任务中，不仅近似地还原出已知的最优机制，而且在相应设定下，获得了比近似最优机制更多的期望收入。准确地说，神经网络将分配规则$g(\cdot)$和支付规则$p(\cdot)$参数化为$g^\theta(\cdot)$和$p^\theta(\cdot)$。具体表达式写作$p_i^\theta = \alpha_i(v_i^T \cdot g_i^\theta)$，式中$\alpha_i \in [0, 1]$，该表达式直接满足个体理性约束。激励相容约束则被转移到一组损失函数$rgt_i^\theta = \mathbb{E}_{v_1 \sim F_1, \ldots, v_n \sim F_n}[\max_{b_i} u_i^\theta(v_i, b_i, v_{-i}) - u_i^\theta(v_i, v_i, v_{-i})]$中，函数表示每个参与者对不谎报的后悔程度。若对所有i，有$rgt_i^\theta = 0$，则满足激励相容约束。于是，训练过程会从已知分布中抽样竞拍者估值，合并负值期望收入$-\mathbb{E}_{v_1 \sim F_1, \ldots, v_n \sim F_n}[\sum_i p_i^\theta(v_i, v_{-i})]$与$rgt_i^\theta$，并使用随机梯度下降法更新神经网络的参数$\theta$。

Feng 等人 [38] 将上述方法应用到带预算约束的设定中。Sakurai 等人 [109] 进一步引入防假名约束，利用类似激励相容约束的损失函数，防止竞拍者操纵虚假身份参与拍卖而从中获益。Shen 等人 [112] 也采用类似的思路，使用了神经网络参数化机制，但不同之处在于，该研究使

用另一个网络替代了与激励相容约束相关的损失函数，新的网络表示竞拍者在给定机制下采取的最优动作。换言之，研究中并未采用显示原理，转而从一个更大的参数化空间中搜索最优拍卖机制。Rahme 等人 [105] 进一步将机制网络与竞拍者网络之间的互动建模为两人博弈。另外，Cai 等人 [22] 对学习辅助式拍卖设计进行了动态扩展，研究了重复拍卖的情形，使得机制与竞拍者都可以随时间演进。然而，为了获得令人信服的性能，他们将拍卖机制的设计空间限制为带保留价的二价拍卖，并使用强化学习来找到每个时间周期中的最优保留价。最近，Liu 等人 [78] 还将基于学习的拍卖机制应用于广告拍卖的现实场景中，拍卖中假定投标人的估值分布具有特定结构。

其他相关工作。前文介绍的工作主要考察了基于学习的机制设计方法。另有大量论文研究了需要多少样本才能通过理论分析学习到拍卖机制。这个问题更为基础，本文不作相关工作的评述。

5.2 金钱无关机制设计中的机器学习

上述拍卖机制是用分配规则和支付规则表示的，而很多其他的机制设计问题并不涉及支付规则。这类问题被称为与金钱无关的机制设计。由于这类机制设计不具备统一的设定，我们只介绍一项最近的工作，它研究了如何通过机器学习来增强机制设计。

Golowich 等人 [47] 研究了设施选址问题，即政府要求参与者报告自己的位置，然后从中找到一个对所有参与者都有利的设施建造位置。该研究采取类似 [38] 的思路，将“选址规则”表示为神经网络，以参与者的报告为输入，进而输出设施的选址。经过训练，神经网络会最小化设施位置与每个参与者之间的总（加权）距离。每个参与者谎报的激励（谎报的目的是使设施位置尽可能靠近自己）转换为训练中使用的损失函数。

5.3 面向学习式参与者的拍卖设计

可以认为，拍卖设计是卖家设计博弈规则，买家参与博弈。前述工作主要使用机器学习来增强卖家，同时假设买家是理性的，也就是说买家能够找到自己的最优出价。确切地说，在收入最大化拍卖机制中，收入的定义是基于每个买家的出价都遵循纳什均衡的假设，而显示原理又将纳什均衡进一步转换为真实的出价。然而，在实践中，买家可能无法找到这个均衡。因此，拍卖机制应用于现实世界之前，拍卖设计必须考虑非理性的买家，尤其是学习式参与者。

典型情况是，卖家反复拍卖同一类型的物品，而独立买家（可能具有完全相同的估值分布）使用无悔学习算法来找到合理出价，从而实现长期收益的最大化（即最大限度地降低后悔程度）。Nekipelov 等人 [97] 首次开展了相关研究，针对二价拍卖，基于买家都是无悔学习者的假设，考察如何估计买家的出价。最近，Deng 等人 [29] 又考察了一价拍卖。一价拍卖尚无纳什均衡的理论表征。该研究指出，存在一种无悔学习算法，可以让买家获得纳什均衡。

6 限制与挑战

引入机器学习方法求解组合优化问题，当前仍面临一些限制与挑战。本节先通过组合优化与模式识别的比较，引出组合优化中应用机器学习的若干困难。之后提出，在当前研究中，增强算法还有哪些属性问题有待解决。

6.1 组合优化与模式识别的比较

下文通过与模式识别比较，总结机器学习引入到组合优化中所面临的困难。

1. **难以构造高质量的训练数据集。**如今，要训练出性能良好的机器学习模型，通常需要大规模的训练数据集，因此获得高质量的数据集至关重要。在模式识别中，数据点通常由特征和标签组成。有几种成本可接受的方法可用来获取训练数据，比如雇佣人工来打标签。但是，组合优化实例的标签通常就是最优解。计算最优解的成本高昂，对于大规模实例尤其如此，而且不太可能像模式识别中那样由人工来打“标签”。因此，构建包含不同规模实例的高质量数据集是一大挑战。

2. **对扰动和噪声的敏感性。**在模式识别中，鲁棒性是对机器学习模型的一项关键要求，也就是说，机器学习模型不应对小的扰动或噪声过于敏感。为提高所学到的机器学习模型的鲁棒性，可以在维护标签时轻微扰动训练数据。但是，组合优化问题却具有很高的敏感性，即使微小的扰动或噪声都可能对问题的最优解产生影响。以旅行商问题为例，添加单条边就会改变最优解，比如图 4 中，添加边 (A, D) 导致最优解中一半的边发生了改变。SAT 问题也存在同样的敏感性，在问题中添加子句可能使可满足状态改变为不可满足。此外，在 CSP 问题中，添加一条约束或从问题的约束关系中删除一个元组，原本容易满足的问题就可能变得难以满足。组合优化的这种敏感性也导致难以构建高质量的训练数据集，因为无法像模式识别那样通过小的扰动来构建不同的实例。因此，我们希望学习到的机器学习模型能够识别这种敏感性，但这一要求不同于模式识别，带来很大的挑战。

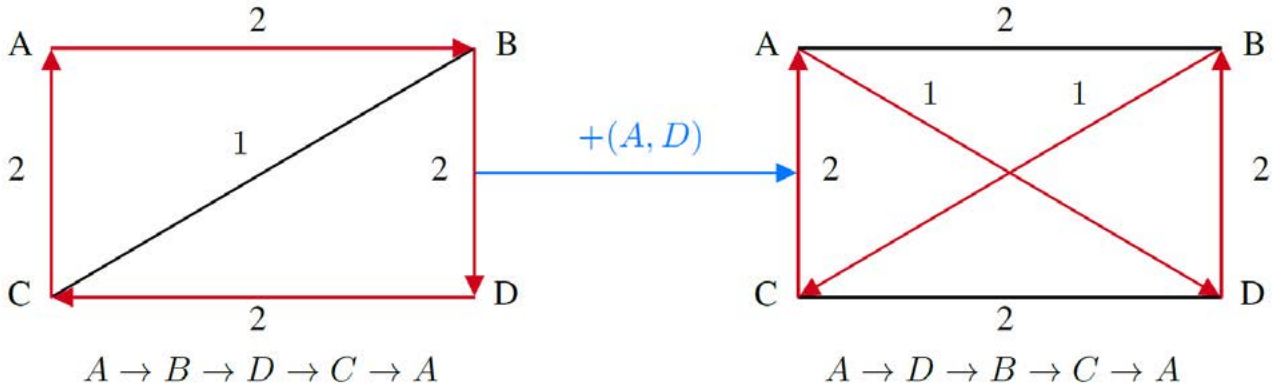


图 4 旅行商问题的敏感性示例

6.2 增强算法的期望属性

将机器学习纳入组合优化，尽管相关研究已取得一些进展，但在增强算法的设计上仍存在许多挑战。挑战体现在多个方面，包括组合优化的具体性质、对理论或启发式方法缺乏理解，以及现实的需求。

1. **扩展性**。构建大规模训练数据的难度很大。因此，为组合优化问题训练机器学习模型时，目前通常使用小规模的数据集，比如最多 1000 个变量。然而，研究人员仍希望设计出增强的算法，使算法在输入规模较大时也具备良好的性能。这使机器学习模型的扩展性成为关注的焦点，在新近的几次调研中 [114, 87, 19]，扩展性已被列为未来的研究方向。已有部分工作 [111, 68, 90] 尝试扩展到规模大得多的实例。不过，现有的很多增强框架并不具备良好的扩展性，当实例规模增加时，解的质量会迅速下降 [114]。

2. **适应性**。与经典的组合优化问题不同，实际的应用场景通常涉及额外的目标或约束。为减少问题变化时重新设计算法的成本，学习到的增强框架应能适应问题集内的数据扰动，或适配同一问题族中的其他问题。已有一些初步工作 [96] 对机器学习模型的适应性开展了研究。

3. **理论保证**。传统的理论算法通常具有执行时间、近似比和泛化界限等方面的可证明保证。在引入机器学习算法时，应尽量不影响已有的理论保证，这样可以提高所学模型的可解释性，但要做到这一点挑战很大。

   执行时间与模型性能一样，都与算法的实现密切相关。增强算法从设计上应减少算法运行时间，同时不影响输出解的质量。已有初步研究 [17, 72] 用更少的时间实现了相近的求解性能，提升了机器学习模型的效率。然而，这些研究提供的框架只能处理 100 个节点，远未达到最先进的水平。

   近似比是对输出解质量的理论保证。实现近似保证的一种方法是将近似算法与机器学习方法结合起来，例如在顶点覆盖问题和旅行商问题 [68] 中的组合应用。然而，现有的组合通常只考虑基本的理论方法或启发式方法，比如贪心算法或分支限界算法。因此有必要进一步考察其他近似算法，虽然这些算法可能更复杂，但它们具有更好的近似比。机器学习方法如果与更好的近似算法结合运用，可以提供更佳的近似保证。

   泛化能力反映所学模型能否在其他一般性实例上表现良好，体现模型的实用价值。传统理论算法考虑的是最坏情况界限，因而自然地满足泛化要求。然而，机器学习学到的模型通常依赖特定的训练数据集，其泛化能力就成为机器学习的重要课题。因此，有必要进一步研究，在引入机器学习方法的同时，如何保持模型的泛化能力。

   样本复杂度与上述所有因素相关。样本的数量通常决定训练花费的时长以及所学模型的质量，模型质量又可能影响增强算法的近似比。此外，如果训练数据和测试数据是从相同分布中提取的，那么泛化界限通常与从训练数据集估得的基础分布好坏程度有关，而这很大程度上取决于样本复杂度。

参考文献

[1] Dimitris Achlioptas, Marek Chrobak, and John Noga. Competitive analysis of randomized paging algorithms. *Theoretical Computer Science*, 234(1-2):203–218, 2000.

[2] Gagan Aggarwal, Gagan Goel, Chinmay Karande, and Aranyak Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1253–1264. SIAM, 2011.

[3] Antonios Antoniadis, Christian Coester, Marek Elias, Adam Polak, and Bertrand Simon. Online metric algorithms with untrusted predictions. In *International Conference on Machine Learning*, pages 345–355. PMLR, 2020.

[4] Antonios Antoniadis, Themis Gouleakis, Pieter Kleer, and Pavel Kolev. Secretary and online matching problems with machine learned advice. In *34th Conference on Neural Information Processing Systems*. Curran Associates, Inc., 2020.

[5] Sanjeev Arora. Polynomial time approximation schemes for Euclidean travelling salesman and other geometric problems. *Journal of the ACM (JACM)*, 45(5):753–782, 1998.

[6] Ashwinkumar Badanidiyuru, Shahar Dobzinski, Hu Fu, Robert Kleinberg, Noam Nisan, and Tim Roughgarden. Sketching valuation functions. In Yuval Rabani, editor, *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2012, Kyoto, Japan, January 17-19, 2012*, pages 1025–1035.

[7] Yunsheng Bai, Derek Xu, Alex Wang, Ken Gu, Xueqing Wu, Agustin Marinovic, Christopher Ro, Yizhou Sun, and Wei Wang. Glsearch: Maximum common subgraph detection via learning to search. *CoRR*, abs/2002.03129, 2020.

[8] Maria-Florina Balcan and Nicholas J. A. Harvey. Learning submodular functions. In Peter A. Flach, Tijl De Bie, and Nello Cristianini, editors, *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012, Bristol, UK, September 24-28, 2012. Proceedings, Part II*, volume 7524 of *Lecture Notes in Computer Science*, pages 846–849. Springer, 2012.

[9] Eric Balkanski, Nicole Immorlica, and Yaron Singer. The importance of communities for learning to influence. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5862–5871, 2017.

[10] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The power of optimization from samples. In Daniel D. Lee, Masashi Sugiyama, Ulrike von Luxburg, Isabelle Guyon, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain*, pages 4017–4025, 2016.

[11] Eric Balkanski, Aviad Rubinstein, and Yaron Singer. The limitations of optimization from samples. In Hamed Hatami, Pierre McKenzie, and Valerie King, editors, *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017,* pages 1016–1027. ACM, 2017.

[12] Eric Balkanski and Yaron Singer. Minimizing a submodular function from samples. In Isabelle Guyon, Ulrike von Luxburg, Samy Bengio, Hanna M. Wallach, Rob Fergus, S. V. N. Vishwanathan, and Roman Garnett, editors, *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 814–822, 2017.

[13] Eric Balkanski and Yaron Singer. The sample complexity of optimizing a convex function. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017,* volume 65 of *Proceedings of Machine Learning Research*, pages 275–301. PMLR, 2017.

[14] Radu Baltean-Lugojan, Pierre Bonami, Ruth Misener,

and Andrea Tramontani. Selecting cutting planes for quadratic semidefinite outer-approximation via trained neural networks. *Technical Report, Imperial College, London*, 2018.

[15] Nikhil Bansal, Kedar Dhamdhere, Jochen Könemann, and Amitabh Sinha. Non-clairvoyant scheduling for minimizing mean slowdown. *Algorithmica*, 40(4):305–318, 2004.

[16] Luca Becchetti and Stefano Leonardi. Non-clairvoyant scheduling to minimize the average flow time on single and parallel machines. In *Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, pages 94–103, 2001.

[17] Irwan Bello, Hieu Pham, Quoc V. Le, Mohammad Norouzi, and Samy Bengio. Neural combinatorial optimization with reinforcement learning. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

[18] Alberto Bemporad and Vihangkumar V. Naik. A numerically robust mixed-integer quadratic programming solver for embedded hybrid model predictive control. *IFAC-PapersOnLine*, 51:412–417, 2018.

[19] Yoshua Bengio, Andrea Lodi, and Antoine Prouvost. Machine learning for combinatorial optimization: A methodological tour d'horizon. *European Journal of Operational Research*, 2020.

[20] Bernd Bischl, Pascal Kerschke, Lars Kotthoff, Marius Lindauer, Yuri Malitsky, Alexandre Fréchette, Holger Hoos, Frank Hutter, Kevin Leyton-Brown, Kevin Tierney, et al. Aslib: A benchmark library for algorithm selection. *Artificial Intelligence*, 237:41–58, 2016.

[21] Pierre Bonami, Andrea Lodi, and Giulia Zarpellon. Learning a classification of mixed-integer quadratic programming problems. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and Operations Research*, pages 595–604. Springer, 2018.

[22] Qingpeng Cai, Aris Filos-Ratsikas, Pingzhong Tang, and Yiwei Zhang. Reinforcement mechanism design for e-commerce. In *Proceedings of the 2018 World Wide Web Conference*, WWW '18, page 1339–1348, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.

[23] Wei Chen, Xiaoming Sun, Jialin Zhang, and Zhijie Zhang. Optimization from structured samples for coverage functions. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 1715–1724. PMLR, 2020.

[24] Wei Chen, Xiaoming Sun, Jialin Zhang, and Zhijie Zhang. Network inference and influence maximization from samples. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1707–1716. PMLR, 2021.

[25] Nicos Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

[26] Graham Cormode and S Muthukrishnan. Space efficient mining of multigraph streams. In *Proceedings of the Twenty-Fourth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 271–282, 2005.

[27] Janos Csirik, David S Johnson, Claire Kenyon, James B Orlin, Peter W Shor, and Richard R Weber. On the sum-of-squares algorithm for bin packing. *Journal of the ACM (JACM)*, 53(1):1–65, 2006.

[28] George B Dantzig. Maximization of a linear function of variables subject to linear inequalities. *Activity Analysis of Production and Allocation*, 13:339–347, 1951.

[29] Xiaotie Deng, Xinyan Hu, Tao Lin, and Weiqiang Zheng. Nash convergence of mean-based learning algorithms in first price auctions, 2021.

[30] Michel Deudon, Pierre Cournut, Alexandre Lacoste, Yossiri Adulyasak, and Louis-Martin Rousseau. Learning heuristics for the TSP by policy gradient. In *International Conference on the Integration of Constraint Programming, Artificial Intelligence, and*

Operations Research, pages 170–181. Springer, 2018.

[31] Nikhil R Devanur and Thomas P Hayes. The adwords problem: Online keyword matching with budgeted bidders under random permutations. In *Proceedings of the 10th ACM Conference on Electronic Commerce*, pages 71–78, 2009.

[32] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE Computational Intelligence Magazine*, 1(4):28–39, 2006.

[33] Marco Dorigo and Gianni Di Caro. Ant colony optimization: A new meta-heuristic. In *Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406)*, volume 2, pages 1470–1477. IEEE, 1999.

[34] Lu Duan, Haoyuan Hu, Yu Qian, Yu Gong, Xiaodong Zhang, Jiangwen Wei, and Yinghui Xu. A multi-task selected learning approach for solving 3D flexible bin packing problem. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1386–1394, 2019.

[35] Zhijian Duan, Dinghuai Zhang, Wenhan Huang, Yali Du, Yaodong Yang, Jun Wang, and Xiaotie Deng. PAC learnability of approximate Nash equilibrium in bimatrix games, 2021.

[36] Paul Duetting, Zhe Feng, Harikrishna Narasimhan, David Parkes, and Sai Srivatsa Ravindranath. Optimal auctions through deep learning. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 1706–1715. PMLR, 09–15 Jun 2019.

[37] Evgenii Borisovich Dynkin. The optimum choice of the instant for stopping a Markov process. *Soviet Mathematics*, 4:627–629, 1963.

[38] Zhe Feng, Harikrishna Narasimhan, and David C. Parkes. Deep learning for revenue-optimal auctions with budgets. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 354–362, Richland, SC, 2018. International Foundation for Autonomous Agents and Multiagent Systems.

[39] Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[40] Merrill M Flood. The travelling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[41] Alex S Fukunaga and Richard E Korf. Bin completion algorithms for multicontainer packing, knapsack, and covering problems. *Journal of Artificial Intelligence Research*, 28:393–429, 2007.

[42] Fred Glover. Tabu search: part i. *ORSA Journal on Computing*, 1(3):190–206, 1989.

[43] Fred Glover. Tabu search: part ii. *ORSA Journal on Computing*, 2(1):4–32, 1990.

[44] Fred Glover and Manuel Laguna. Tabu search. In *Handbook of Combinatorial Optimization*, pages 2093–2229. Springer, 1998.

[45] Fred Glover, Manuel Laguna, and Rafael Martí. Fundamentals of scatter search and path relinking. *Control and Cybernetics*, 29(3):653–684, 2000.

[46] David E Goldberg. *Genetic Algorithms*. Pearson Education India, 2006.

[47] Noah Golowich, Harikrishna Narasimhan, and David C. Parkes. Deep learning for multi-facility location mechanism design. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 261–267. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[48] Avinatan Hassidim and Yaron Singer. Submodular optimization under noise. In Satyen Kale and Ohad Shamir, editors, *Proceedings of the 30th Conference on Learning Theory, COLT 2017, Amsterdam, The Netherlands, 7-10 July 2017,* volume 65 of *Proceedings of Machine Learning Research*, pages 1069–1122. PMLR, 2017.

[49] He He, Hal Daume III, and Jason M Eisner. Learning to search in branch and bound algorithms. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[50] Michael Held and Richard M Karp. A dynamic programming approach to sequencing problems. *Journal of the Society for Industrial and Applied mathematics*, 10(1):196–210, 1962.

[51] Holger H Hoos. Automated algorithm configuration and parameter tuning. In *Autonomous Search*, pages 37–71. Springer, 2011.

[52] Andre Hottung, Shunji Tanaka, and Kevin Tierney. Deep learning assisted heuristic tree search for the container pre-marshalling problem. *Computers & Operations Research*, 113:104781, 2020.

[53] Chen-Yu Hsu, Piotr Indyk, Dina Katabi, and Ali Vakilian. Learning-based frequency estimation algorithms. In *International Conference on Learning Representations*, 2018.

[54] Sungjin Im, Janardhan Kulkarni, and Kamesh Munagala. Competitive algorithms from competitive equilibria: Non-clairvoyant scheduling under polyhedral constraints. *Journal of the ACM (JACM)*, 65(1):1–33, 2017.

[55] Sungjin Im, Janardhan Kulkarni, Kamesh Munagala, and Kirk Pruhs. Selfishmigrate: A scalable algorithm for non-clairvoyantly scheduling heterogeneous processors. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pages 531–540. IEEE, 2014.

[56] Sungjin Im, Ravi Kumar, Mahshid Montazer Qaem, and Manish Purohit. Non-clairvoyant scheduling with predictions. In *Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures*, pages 285–294, 2021.

[57] Piotr Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of the ACM (JACM)*, 53(3): 307–323, 2006.

[58] Piotr Indyk and David Woodruff. Optimal approximations of the frequency moments of data streams. In *Proceedings of the Thirty-Seventh Annual ACM Symposium on Theory of Computing*, pages 202–208, 2005.

[59] Thathachar S Jayram and David P Woodruff. The data stream space complexity of cascaded norms. In *2009 50th Annual IEEE Symposium on Foundations of Computer Science*, pages 765–774. IEEE, 2009.

[60] Tanqiu Jiang, Yi Li, Honghao Lin, Yisong Ruan, and David P Woodruff. Learning-augmented data stream algorithms. In *International Conference on Learning Representations*, 2019.

[61] Daniel M Kane, Jelani Nelson, and David P Woodruff. An optimal algorithm for the distinct elements problem. In *Proceedings of the Twenty-Ninth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, pages 41–52, 2010.

[62] Daniel Karapetyan, Abraham P Punnen, and Andrew J Parkes. Markov chain methods for the bipartite boolean quadratic programming problem. *European* Journal of *Operational Research*, 260(2):494–506, 2017.

[63] Anna R. Karlin, Mark S. Manasse, Lyle A. McGeoch, and Susan Owicki. Competitive randomized algorithms for nonuniform problems. *Algorithmica*, 11(6):542–571, 1994.

[64] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, pages 302–311, 1984.

[65] Marek Karpinski, Michael Lampis, and Richard Schmied. New inapproximability bounds for TSP. *Journal of Computer and System Sciences*, 81(8): 1665–1677, 2015.

[66] Thomas Kesselheim, Klaus Radke, Andreas Tönnis, and Berthold Vöcking. An optimal online algorithm for weighted bipartite matching and extensions to combinatorial auctions. In *European Symposium on Algorithms*, pages 589–600. Springer, 2013.

[67] Leonid Genrikhovich Khachiyan. A polynomial algorithm in linear programming. In *Doklady Akademii Nauk*. Russian Academy of Sciences, 1979.

[68] Elias Khalil, Hanjun Dai, Yuyu Zhang, Bistra Dilkina, and Le Song. Learning combinatorial optimization algorithms over graphs. In *Advances in Neural Information Processing Systems*, pages 6348–6358, 2017.

[69] Scott Kirkpatrick, C Daniel Gelatt, and Mario P Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[70] Victor Klee and George J Minty. How good is the simplex algorithm. *Inequalities*, 3(3):159–175, 1972.

[71] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *International Conference on Learning Representations*, 2018.

[72] Wouter Kool, Herke van Hoof, and Max Welling. Attention, learn to solve routing problems! In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

[73] Nitish Korula and Martin Pál. Algorithms for secretary problems on graphs and hypergraphs. In *International Colloquium on Automata, Languages, and Programming*, pages 508–520. Springer, 2009.

[74] Markus Kruber, Marco E Lübbecke, and Axel Parmentier. Learning when to use a decomposition. In *International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 202–210. Springer, 2017.

[75] Eugene L Lawler. The travelling salesman problem: A guided tour of combinatorial optimization. *Wiley-Interscience Series in Discrete Mathematics*, 1985.

[76] Zhuwen Li, Qifeng Chen, and Vladlen Koltun. Combinatorial optimization with graph convolutional networks and guided tree search. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[77] Denis V Lindley. Dynamic programming and decision theory. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 10(1): 39–51, 1961.

[78] Xiangyu Liu, Chuan Yu, Zhilin Zhang, Zhenzhe Zheng, Yu Rong, Hongtao Lv, Da Huo, Yiqing Wang, Dagui Chen, Jian Xu, Fan Wu, Guihai Chen, and Xiaoqiang Zhu. Neural auction: End-to-end learning of auction mechanisms for e-commerce advertising. KDD '21, page 3354–3364, New York, NY, USA, 2021. Association for Computing Machinery.

[79] Yanli Liu, Chu-Min Li, Hua Jiang, and Kun He. A learning based branch and bound for maximum common subgraph related problems. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34: 2392–2399, 04 2020.

[80] Andrea Lodi and Giulia Zarpellon. On learning and branching: A survey. *Top*, 25(2):207–236, 2017.

[81] Michele Lombardi and Michela Milano. Boosting combinatorial problem modeling with machine learning. *arXiv preprint arXiv:1807.05517*, 2018.

[82] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. Iterated local search. In *Handbook of Metaheuristics*, pages 320–353. Springer, 2003.

[83] Thodoris Lykouris and Sergei Vassilvtiskii. Competitive caching with machine learned advice. In *International Conference on Machine Learning*, pages 3296–3305. PMLR, 2018.

[84] Qiang Ma, Suwen Ge, Danyang He, Darshan Thaker, and Iddo Drori. Combinatorial optimization by graph pointer networks and hierarchical reinforcement learning. *arXiv preprint arXiv:1911.04936*, 2019.

[85] Franco Mascia, Manuel López-Ibáñez, Jérémie Dubois-Lacoste, and Thomas Stützle. Grammar-based generation of stochastic local search heuristics through automatic algorithm configuration tools. *Computers & Operations Research*, 51:190–199, 2014.

[86] Daniele Masti and Alberto Bemporad. Learning binary warm starts for multiparametric mixed-integer quadratic programming. pages 1494–1499, 06 2019.

[87] Nina Mazyavkina, Sergey Sviridov, Sergei Ivanov, and Evgeny Burnaev. Reinforcement learning for combinatorial optimization: A survey. *arXiv preprint arXiv:2003.03600*, 2020.

[88] Jesper W Mikkelsen. Randomization can be as helpful as a glimpse of the future in online computation. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2016.

[89] Joseph SB Mitchell. Guillotine subdivisions approximate polygonal subdivisions: A simple polynomial-time approximation scheme for geometric TSP, k-MST, and related problems. *SIAM Journal on Computing*, 28(4):1298–1309, 1999.

[90] Akash Mittal, Anuj Dhawan, Sahil Manchanda, Sourav Medya, Sayan Ranu, and Ambuj Singh. Learning heuristics over large graphs via deep reinforcement learning. *arXiv preprint arXiv:1903.03332*, 2019.

[91] Michael Mitzenmacher and Sergei Vassilvitskii. Algorithms with predictions. *arXiv preprint arXiv:2006.09123*, 2020.

[92] Nenad Mladenović and Pierre Hansen. Variable neighborhood search. *Computers & Operations Research*, 24(11):1097–1100, 1997.

[93] Rajeev Motwani, Steven Phillips, and Eric Torng. Nonclairvoyant scheduling. *Theoretical Computer Science*, 130(1):17–47, 1994.

[94] Katta G Murty. Computational complexity of parametric linear programming. *Mathematical Programming*, 19(1):213–219, 1980.

[95] Roger B Myerson. Optimal auction design. *Mathematics of Operations Research*, 6(1) 58–73, 1981.

[96] Mohammadreza Nazari, Afshin Oroojlooy, Lawrence Snyder, and Martin Takác. Reinforcement learning for solving the vehicle routing problem. In *Advances in Neural Information Processing Systems*, pages 9839–9849, 2018.

[97] Denis Nekipelov, Vasilis Syrgkanis, and Eva Tardos. Econometrics for learning agents. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, page 1–18, New York, NY, USA, 2015. Association for Computing Machinery.

[98] George L. Nemhauser, Laurence A. Wolsey, and Marshall L. Fisher. An analysis of approximations for maximizing submodular set functions - I. *Math. Program.*, 14(1):265–294, 1978.

[99] Alex Nowak, Soledad Villar, Afonso S Bandeira, and Joan Bruna. A note on learning algorithms for quadratic assignment with graph neural networks. In *Proceeding of the 34th International Conference on Machine Learning (ICML)*, volume 1050, page 22, 2017.

[100] Pekka Orponen and Heikki Mannila. On *Approximation Preserving Reductions: Complete Problems and Robust Measures*. Citeseer, 1987.

[101] Manfred Padberg and Giovanni Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric travelling salesman problems. *SIAM Review*, 33(1):60–100, 1991.

[102] Marcelo Prates, Pedro HC Avelar, Henrique Lemos, Luis C Lamb, and Moshe Y Vardi. Learning to solve NP-complete problems: A graph neural network for decision TSP. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4731–4738, 2019.

[103] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In *Advances in Neural Information Processing Systems*, pages 9661–9670, 2018.

[104] Manish Purohit, Zoya Svitkina, and Ravi Kumar. Improving online algorithms via ML predictions. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[105] Jad Rahme, Samy Jelassi, and S. Matthew Weinberg. Auction learning as a two-player game. In *International Conference on Learning Representations*, 2021.

[106] Dhruv Rohatgi. Near-optimal bounds for online caching with machine learned advice. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1834–1845. SIAM, 2020.

[107] Nir Rosenfeld, Eric Balkanski, Amir Globerson, and Yaron Singer. Learning to optimize combinatorial functions. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018, volume 80 of Proceedings of Machine Learning Research*, pages 4371–4380.

[108] Francesca Rossi, Peter van Beek, and Toby Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., USA, 2006.

[109] Yuko Sakurai, Satoshi Oyama, Mingyu Guo, and Makoto Yokoo. *Deep False-Name-Proof Auction Mechanisms*, pages 594–601. 10 2019.

[110] Ethan L Schreiber and Richard E Korf. Improved bin completion for optimal bin packing and number partitioning. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[111] Daniel Selsam, Matthew Lamm, Benedikt Bünz, Percy Liang, Leonardo de Moura, and David L Dill. Learning a SAT solver from single-bit supervision. *arXiv preprint arXiv:1802.03685*, 2018.

[112] Weiran Shen, Pingzhong Tang, and Song Zuo. *Automated Mechanism Design via Neural Networks*, page 215–223. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2019.

[113] Daniel A Spielman and Shang-Hua Teng. Smoothed analysis of algorithms Why the simplex algorithm usually takes polynomial time. *Journal of the ACM* (JACM), 51(3):385–463, 2004.

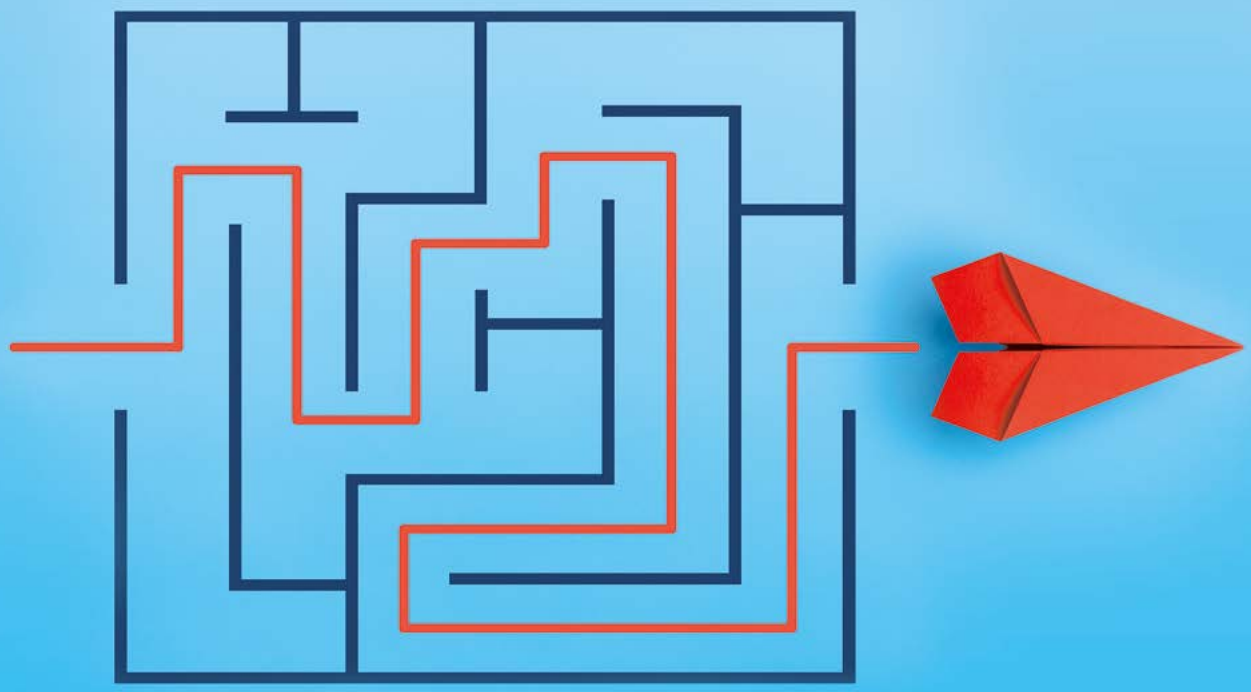
[114] Natalia Vesselinova, Rebecca Steinert, Daniel F Perez-Ramirez, and Magnus Boman. Learning combinatorial optimization on graphs: A survey with applications to networking. *arXiv preprint arXiv:2005.11081*, 2020.

[115] Alexander Wei. Better and simpler learning-augmented online caching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik, 2020.

[116] Gerhard J Woeginger. Exact algorithms for NP-hard problems: A survey. In *Combinatorial Optimization: Eureka, You Shrink!*, pages 185–207. Springer, 2003.

[117] Yaoxin Wu, Wen Song, Zhiguang Cao, Jie Zhang, and Andrew Lim. Learning improvement heuristics for solving the travelling salesman problem. *arXiv preprint arXiv:1912.05784*, 2019.

[118] Jiongzhi Zheng, Kun He, Jianrong Zhou, Yan Jin, and Chu-Min Li. Combining reinforcement learning with Lin-Kernighan-Helsgaun algorithm for the traveling salesman problem. *CoRR*, abs/2012.04461, 2020.



随机游走与适应度函数相结合求解布尔可满足性问题

蔡少伟，姜滔
中国科学院软件研究所计算机科学国家重点实验室

摘要

可满足性（Boolean Satisfiability，SAT）问题在计算机科学中相当重要。很多启发式算法可用于求解可满足性问题，但这些算法基于实验方式来评估，缺乏理论分析。本文将主要关注随机游走（Random Walk，RW）技术，并结合适应度函数，提升随机游走的性能。这种改进版的 RW 被称为 RWF。为了验证 RWF 的性能，我们基于三种 SAT 表达式，分析比较了 RW、(1+1) EA（一种典型的演化算法）和 RWF 的期望运行时间。结果表明，RWF 在部分实例上的性能优于 RW 和 (1+1) EA。

关键词

布尔可满足性，随机游走，运行时间分析

1 引言

布尔可满足性（Boolean Satisfiability，SAT）问题是指，确定是否存在使给定命题表达式为真的变量赋值的问题，是计算机科学中相当重要的一类问题。SAT 问题是第一个被证明为 NP 完全的问题，这意味着其他 NP 复杂度分类下的问题的求解难度不会超过 SAT 问题。除了理论上的重要地位，SAT 算法还在硬件验证、密码学、资源分配和规划等领域有着广泛的实际应用。

SAT 问题在理论上极具挑战性，尚无高效的求解算法。简单穷举算法可以穷尽n个变量的2^n个取值，得到$O^*(2^n)$的平凡边界，但迄今还没新的算法能突破这一界限。实际上，强指数时间假设（Strong Exponential Time Hypothesis，SETH）认为，当常数$c<2$，SAT 问题无法在$O^*(c^n)$时间内求解 [7]。大多数理论算法聚焦于k值固定的随机k −SAT 问题，这类问题中，每个子句中的随机字面量的数量固定为k。这些算法中，Schöning [11] 提出的局部搜索算法以及基于编码的 PPSZ [10] 算法是两种典型的快速算法。

SAT 问题有两种主要的求解方法，其中一种就是局部搜索，另一种是冲突驱动子句学习（Conflict−Driven Clause Learning，CDCL）[2]。CDCL 由回溯 DPLL 算法演进而来，拥有强大的推理能力。通常，局部搜索算法从变量的一组完整赋值开始，并在之后每一步中翻转一个变量的值，以此求解。局部搜索求解 SAT 问题的一个关键技术是随机游走（Random Walk，RW），也称为集中随机游走。具体来说，RW 每次从未满足的子句中随机选取一个变量，并将其翻转。这是 Schöning 局部搜索算法的核心思想，以此求解k −SAT 问题的时间复杂度为$O(poly(n)(2(1-1/k))^n)$。除了理论上的研究，RW 也在一些 SAT 问题局部搜索求解器中成功得到了应用，如 WalkSAT [4]、ProbSAT [1] 和 CCAnr [3]。不过，关于 SAT 局部搜索算法的理论分析仍十分有限 [5, 12]。

周育人等人 [12] 比较了一些 RW 启发式算法在 SAT 实例上的期望运行时间。[5] 的作者则介绍了一种基于汉明球的随机局部搜索算法。另一方面，有较多对 SAT 问题演化算法（Evolutionary Algorithm，EA）的分析研究。EA 可以看作是结合了特定操作（如突变）的局部搜索算法。周育人等人 [12] 精心设计了一些实例，对简单局部搜索算法和 (1+1) EA（一种简单的 EA 算法）的期望运行时间进行了比较。其中，EA 算法有适应度函数引导，而 RW 算法则没有该函数。

本文重点研究了 RW 这一广泛使用的随机局部搜索（Stochastic Local Search，SLS）方法。RW 方法在许多针对 SAT 问题的 SLS 求解器中有成功的应用，例如 WalkSAT [4]、ProbSAT [1] 和 CCAnr [3]。本文提出，将 EA 算法中使用的适应度函数与 RW 相结合，可得到改进版的 RW 算法，并将其命名为带适应度函数的随机游走（Random Walk with Fitness，RWF）算法。

本文用三种 SAT 表达式对 RW、(1+1) EA 和 RWF 进行了比较。其中，两种表达式来自以往的研究，第三种则由本研究提出。本文主要分析了使用马尔可夫链的实例最坏情况和平均情况的时间复杂度。

分析发现，三种算法的运行时间为指数时间或多项式时间。也就是说，三种算法在求解不同的 SAT 实例时各有优劣。

本文后续内容如下：第 2 节先介绍了 SAT 问题、吸收马尔可夫链以及两个向量范数的相关概念。第 3.1 节介绍了三种启发式算法，第 3.2 节分析比较了 RW 和 RWF 算法在两个 2−SAT 实例上的期望时间复杂度，第 3.3 节分析比较了三种启发式算法在一个 3−SAT 实例上的期望时间复杂度。

2 相关概念

2.1 布尔可满足性（SAT）问题

在布尔逻辑中，字面量是一个变量或它的否定形式，即x或$\bar{x}$。子句是字面量的析取，例如$x \vee y$是一个有两个字面量的子句。合取范式（Conjunctive Normal Form，CNF）表达式是子句的合取。一个k −SAT 实例就是一个 CNF 表达式，表达式的每个子句有且仅有k个字面量。SAT 问题就是指，给定一个布尔表达式，并为其变量赋值，能否使该表达式为真的问题。

2.2 吸收马尔可夫链

设$\{X_t\}$为一个具有有限状态空间S的离散马尔可夫链。T为非常返状态集，$A=S-T$为吸收集。$|\cdot|$表示集合的势，令$|T|=t$且$|A|=r$，则转移矩阵为

$$\begin{pmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{R} & \mathbf{T} \end{pmatrix}$$

其中，$\mathbf{I}$是一个 r 乘 r 矩阵，$\mathbf{R}$是一个 t 乘 r 非零矩阵，$\mathbf{T}$是一个 t 乘 t 非零矩阵。

定理 2.2.1 设吸收马尔可夫链从状态i开始，m_i为到达吸收状态的预期步数。令$\mathbf{m}=[m_i]_{i\in T}$，则有

$$\mathbf{m}=(\mathbf{I}_{t*t}-\mathbf{T})^{-1}\mathbf{1}$$

证明：参见 [8]。

2.3 两种向量范数

- **平均向量范数：** $\mathbf{X}=[x_i]_{i\in S}$，令$P(X_0=i)$为初始分布，则有

$$||\mathbf{m}||_1=\sum_{i\in S}P(X_0=i)m_i$$

- **最大向量范数：**

$$||\mathbf{m}||_\infty=\max_{i\in S}\{m_i\}$$

范数$||\mathbf{m}||_1$表示平均情况时间复杂度，$||\mathbf{m}||_\infty$表示最坏情况时间复杂度。

3 三种启发式算法的比较

3.1 算法介绍

EA 算法广泛应用于 SAT 问题，通常使用适应度函数来引导搜索过程。在最大可满足性（MaxSAT）问题中，标准适应度函数fit表示满足的子句数量。具体来说，给定一个由子句$c_1,\ldots,c_m$构成的 CNF 表达式，对于变量的任意赋值x，有

$$fit(x)=c_1(x)+c_2(x)+\cdots+c_m(x) \tag{1}$$

当$1\le i\le m$，若x满足c_i，则$c_i(x)=1$，否则$c_i(x)=0$。

本文将使用以上适应度函数。

RW 是 Papadimitiou [9] 提出的一种基本的不完备算法。该算法在第一步时随机选择一个未满足的子句，并在下一步随机翻转该子句中的一个变量（如算法 1 所示）。

算法 1 RW 算法

1: 随机选取一个初始比特串x；
2: **while** 表达式未满足 **do**
3: 随机选取一个未满足子句c；
4: 随机选取子句c中一个变量并翻转其值
5: **end while**

EA 算法的设计灵感源于对自然选择和遗传进化过程的建模。本文使用的是一个简单的 EA 算法，采用了突变和自然选择操作，种群数量为 1，记为 (1+1) EA [6]。(1+1) EA 是一种简单但高效的爬山 EA 算法，通常采用局部突变和全局突变两种突变方法。

- 局部突变：从个体$x=(x_1\cdots x_n)\in\{0,1\}^n$中随机选取一个比特$x_i(1\le i\le n)$，并将其翻转。
- 全局突变：以$1/n$的概率单独翻转个体$x=(x_1\cdots x_n)\in\{0,1\}^n$中的各个比特。全局突变中，比特翻转的总期望值为 1。

本文使用的是局部 (1+1) EA 算法。

算法 2 局部 (1+1) EA 算法

1: 随机选取一个初始比特串x；
2: **while** 表达式未满足 **do**
3: 从$x=(x_1,\ldots,x_n)$中随机选取一个比特x_i并将其翻转，得到解y；
4: 若 fitness(y) > fitness(x)，则 x:=y
5: **end while**

以下算法将 RW 方法和适应度函数相结合，是我们分析的重点。该算法每一步用一个适应值来引导搜索。

算法 3 带适应度函数的随机游走 (RWF) 算法

1: 随机选取一个初始比特串x；
2: **while** 表达式未满足 **do**
3: 随机选取一个未满足的子句c；
4: 随机选取c中的一个变量并翻转其值，得到解y；
5: 若 fitness(y) > fitness(x)，则 x:=y
6: **end while**

3.2 RW 与 RWF 算法对比

本节分析了 RW 和 RWF 两种算法求解两个不同 SAT 实例的时间复杂度（平均情况和最坏情况），以便从理论上认识这两种算法。

实例 3.2.1 设$x=(x_1\cdots x_n)\in\{0,1\}^n$，SAT 实例$\psi_1(x)$定义为

$$\psi_1(x)=(x_1\vee\overline{x}_2)\wedge\cdots(x_1\vee\overline{x}_n)\wedge(\overline{x}_1\vee x_2)\wedge\cdots(\overline{x}_1\vee x_n)$$

满足条件的赋值为$(0\cdots0)$和$(1\cdots1)$。

命题 3.2.1 在 SAT 实例ψ_1上，RW 的期望运行时间为$||\mathbf{m}||_1=\Theta(n^2)$。

证明：参见 [12]。

命题 3.2.2 在 SAT 实例ψ_1上，RWF 的期望运行时间为$||\mathbf{m}||_1=O(n)$。

证明：本文后续内容中，对于任意赋值x，用$|x|$来表示x中“1”的数量。

根据式（1），ψ_1的适应度函数为

$$fit_{\psi_1}(x)=\begin{cases}2(n-1)-|x|, & x=(0*\cdots*)\\ n-1+|x|-1, & x=(1*\cdots*)\end{cases}$$

该适应度函数只与x_1和$|x|$相关。具体来说，若$x=(0*\cdots*)$，函数随$|x|$单调下降，否则单调增加。图1为$n=20$时适应度函数$fit_{\psi_1}(x)$的图像。

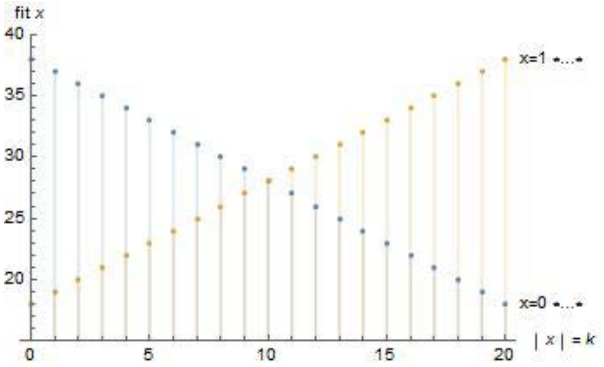


图1 $n=20$的适应度函数$fit_{\psi_1}(x)$

简单起见，设n为偶数。

将搜索空间S划分为$2n$个子空间：

$$S_{0,k}=\{x|x=(0*\cdots*),|x|=k\}\quad(k=0,\ldots,n-1)$$
$$S_{1,k}=\{x|x=(1*\cdots*),|x|=k\}\quad(k=1,\ldots,n)$$

令$X_t\in\{0,1\}^n\ (t=0,1,\ldots)$为随机变量，用于表示RWF算法求解SAT实例$\psi_1$时，在时间$t$所处的状态，则转移概率描述如下。

当$k=0$，有

$$P(X_{t+1}\in S_{0,k}|X_t\in S_{0,k})=1$$

当$1\le k\le\frac{n}{2}-1$，

对于$X_t\in S_{0,k}$，未满足子句的形式为$x_1\vee\overline{x}_i$。若算法使x_1为1，适应度函数下降，该突变不会被保留。

$$P(X_{t+1}\in S_{0,k}|X_t\in S_{0,k})=\frac{1}{2}$$
$$P(X_{t+1}\in S_{0,k-1}|X_t\in S_{0,k})=\frac{1}{2}$$

当$\frac{n}{2}\le k\le n-1$，

对于 $X_t\in S_{0,k}$，未满足子句的形式为$x_1\vee\overline{x}_i$。如果算法使x_1为1，适应度函数上升，该突变将被保留。

$$P(X_{t+1}\in S_{0,k-1}|X_t\in S_{0,k})=\frac{1}{2}$$
$$P(X_{t+1}\in S_{1,k+1}|X_t\in S_{0,k})=\frac{1}{2}$$

同理可得：

当$k=n$，有

$$P(X_{t+1}\in S_{1,k}|X_t\in S_{1,k})=1$$

当$\frac{n}{2}+1\le k\le n-1$，有

$$P(X_{t+1}\in S_{1,k}|X_t\in S_{1,k})=\frac{1}{2}$$
$$P(X_{t+1}\in S_{1,k+1}|X_t\in S_{1,k})=\frac{1}{2}$$

当$1\le k\le\frac{n}{2}$，有

$$P(X_{t+1}\in S_{0,k-1}|X_t\in S_{1,k})=\frac{1}{2}$$
$$P(X_{t+1}\in S_{1,k+1}|X_t\in S_{1,k})=\frac{1}{2}$$

引入具有状态空间$\{z_{0,0},z_{0,1},\ldots,z_{0,n-1};z_{1,1},z_{1,2},\ldots,z_{1,n}\}$的辅助齐次马尔可夫链$Z_t\quad(t=0,1,\ldots)$，则转移概率定义为

$$P(Z_{t+1}=z_{v,h}|Z_t=z_{u,k})=P(X_{t+1}\in S_{v,h}|X_t\in S_{u,k})$$

式中，$u,v\in0,1$且$h,k\in0,\ldots,n$。

此时，Z_t就是一个吸收马尔可夫链，其吸收状态为$z_{0,0}$和$z_{1,n}$。对于任意$x\in S_{u,k}(u\in0,1,k\in0,\ldots,n)$，平均首次到达时间$m_x$等于$m_{z_{u,k}}$。

根据吸收马尔可夫链理论，随机过程Z_t的平均首次到达时间表示为

$$\begin{cases}m_{0,k}=2k & 1\le k\le\frac{n}{2}-1\\ m_{0,k}=\frac{1}{2}(m_{0,k-1}+1)+\frac{1}{2}(m_{1,k+1}+1) & \frac{n}{2}\le k\le n-1\\ \\ m_{1,k}=2(n-k) & \frac{n}{2}+1\le k\le n-1\\ m_{1,k},=\frac{1}{2}(m_{0,k-1}+1)+\frac{1}{2}(m_{1,k+1}+1) & 1\le k\le\frac{n}{2}\end{cases}$$

接下来，需要证明

$$m_{0,k}\le 2k\ \text{式中}\ \frac{n}{2}\le k\le n-1$$

当$k=\frac{n}{2}$，有

$$\begin{aligned}m_{0,\frac{n}{2}}&=\frac{1}{2}(m_{0,\frac{n}{2}-1}+1)+\frac{1}{2}(m_{1,\frac{n}{2}+1}+1)\\&=\frac{1}{2}(2(\frac{n}{2}-1)+1)+\frac{1}{2}(2(\frac{n}{2}-1)+1)\\&=n-1\end{aligned}$$

可得$k=\frac{n}{2}$时，$m_{0,k}\leq 2k$成立。假设$k\geq\frac{n}{2}$时该结论也成立（即$m_{0,k}\leq 2k$），则有

$$\begin{aligned}m_{0,k+1}&=\frac{1}{2}(m_{0,k}+1)+\frac{1}{2}(m_{1,k+2}+1)\\&=\frac{1}{2}m_{0,k}+n-k-1\end{aligned}$$

进而可得$m_{0,k+1}\leq k+n-k-1\leq 2(k+1)\quad k\geq\frac{n}{2}$

对于$m_{1,k}$，可得出类似结论，概括如下：

$$\begin{cases}m_{0,k}\leq 2k & 0\leq k\leq n-1\\ m_{1,k}\leq 2(n-k) & 1\leq k\leq n\end{cases}$$

因此，$||\mathbf{m}||_1=O(n)$。

对于 SAT 实例ψ_1，RWF 的期望运行时间边界$O(n)$优于 RW 的$O(n^2)$。然而，下面另一个 SAT 实例ψ_2中，在一定条件下会出现相反的结果。

实例 3.2.2 SAT 实例ψ_2具有以下子句

$$x_i,\quad x_i\vee\overline{x}_j\quad \text{式中}\quad i,j\in\{1,\ldots,n\},\quad i\neq j$$

SAT 实例ψ_2唯一满足条件的赋值为$(1,\ldots,1)$。

命题 3.2.3 对于 SAT 实例ψ_2，RW 的期望运行时间为$||\mathbf{m}||_\infty=\Omega(n^2)$。

证明：参见 [12]。

命题 3.2.4 RWF 在 SAT 实例ψ_2上的期望运行时间为

$$m_i=\begin{cases}+\infty, & i<\frac{n-1}{2}\\ \Theta(n), & i\geq\frac{n-1}{2}\end{cases}$$

证明：首先，将搜索空间划分为$n+1$个子空间：

$$S_i=\{x||x|=i,x\in\{0,1\}^n\}\qquad i=0,1,\ldots,n$$

满足条件的集合为S_n。

对于$|x|=k$，SAT 实例ψ_2的适应度函数为

$$fit_{\psi_2}(x)=k^2+k(1-n)+n(n-1)$$

该函数是一个关于k的二阶多项式。如果 RWF 选取的初始解为$x(|x|<\frac{n-1}{2})$，根据 RWF 的搜索过程，可得到局部最优解$(0,\ldots,0)$，而非全局最优解$(1,\ldots,1)$。

当$k<\frac{n-1}{2}$时，始于S_k的 RWF 算法永远无法达到满足条件的赋值$(1,\ldots,1)$，因为适应度随着$|x|$的增加而降低。

当$k\geq\frac{n-1}{2}$，有

$$\begin{aligned}P(X_{t+1}\in S_k|X_t\in S_k)&=\frac{1}{2}\frac{k}{1+k}\\P(X_{t+1}\in S_{k+1}|X_t\in S_k)&=1-\frac{1}{2}\frac{k}{1+k}\end{aligned}$$

根据期望的定义：

$$n-i\leq m_i=\sum_{k=i}^{n-1}\frac{2k+2}{k+2}\leq 2(n-i)\quad \text{式中}\quad i\geq\frac{n-1}{2}$$

至此，证明结束。

先前的结果中，RWF 在 SAT 实例ψ_1上的期望运行时间优于 RW。而在 SAT 实例ψ_2上，RW 的期望运行时间优于 RWF，与先前的结果相反。

我们注意到，用 RW 算法求解 SAT 问题，求解过程的分析高度依赖于适应度函数。由于适应度函数表达式在书写上存在障碍，对于一般性 SAT 问题的求解，要分析 RWF 算法的期望运行时间非常困难。接下来，我们构造一个 SAT 实例$\psi_3(x)$，在不书写适应度函数表达式的情况下来分析 RWF 算法的期望运行时间。

3.3 RW、局部 (1+1) EA 和 RWF 算法对比

实例 3.3.1 对于$x=(x_1\cdots x_n)\in\{0,1\}^n$，SAT 实例$\psi_3(x)$由以下子句构成：

$$(x_1\vee\overline{x}_2),\cdots(x_1\vee\overline{x}_n),(\overline{x}_1\vee x_2)\cdots(\overline{x}_1\vee x_n),$$

$$(x_1\vee\overline{x}_2),\cdots(x_1\vee\overline{x}_n),(\overline{x}_1\vee x_2)\cdots(\overline{x}_1\vee x_n),$$

$$(x_1\vee\overline{x}_2\vee x_3),(x_1\vee x_2\vee\overline{x}_3),(x_1\vee\overline{x}_4\vee x_5),(x_1\vee x_4\vee\overline{x}_5)\cdots(x_1\vee\overline{x}_{n-1}\vee x_n),(x_1\vee x_{n-1}\vee\overline{x}_n),$$

$$(\overline{x}_1\vee\overline{x}_2\vee x_3),(\overline{x}_1\vee x_2\vee\overline{x}_3),(\overline{x}_1\vee\overline{x}_4\vee x_5),(\overline{x}_1\vee x_4\vee\overline{x}_5)...(\overline{x}_1\vee\overline{x}_{n-1}\vee x_n),(\overline{x}_1\vee x_{n-1}\vee\overline{x}_n)$$

满足条件的赋值为$(0\cdots0)$和$(1\cdots1)$。

需要注意，该实例包含两个相同子句，分别带有两个字面量。易得：

引理 3.3.1 对于任意赋值$x\in\{0,1\}^n$，带有两个字面量的未满足子句的数量是带有三个字面量的未满足子句的两倍。

引理 3.3.2 当x_1由 1 翻转为 0 或由 0 翻转为 1，带有三个字面量的未满足子句的数量不变。

命题 3.3.1 对于 SAT 实例$\psi_3(x)$，RW 的期望运行时间为$||\mathbf{m}||_\infty=O(2^n)$。

证明：设$S=\{0,1\}^n$为搜索空间，S^*为满足赋值的集合，$d(x)=\min_{y\in S^*}\{H(x,y)\}$为点$x\in S$到集合$S^*$的汉明距离，令$D_i=\{x\in S|d(x)=i\},i=0,1,\cdots,n$，则搜索空间$S$被划分为$n+1$个子空间 $S=\bigcup_{i=0}^{n}D_i$。

RW 将x转移到比特串$y\in D_{i-1}$的概率至少为$\frac{1}{3}$，将x转移到比特串$y\in D_{i+1}$的概率最大为$\frac{2}{3}$。构造一个辅助齐次链，带以下转移矩阵

$$\begin{pmatrix}1&0&0&\cdots&0&0&0\\\frac{1}{3}&0&\frac{2}{3}&\cdots&0&0&0\\\cdots&\cdots&\cdots&\cdots&\cdots&\cdots&\cdots\\0&0&0&\cdots&\frac{1}{3}&0&\frac{2}{3}\\0&0&0&\cdots&0&1&0\end{pmatrix}$$

根据该吸收马尔可夫链，可得

$$m_n=2(n-1)+6(2^{n-2}+n-3)$$

因此

$$||\mathbf{m}||_\infty=O(2^n)$$

由于k −SAT 的边界为$O((k-1)^n)$，该结果可能并非紧边界。

命题 3.3.2 对于 SAT 实例$\psi_3(x)$，(1+1) EA 算法的期望运行时间为$||\mathbf{m}||_\infty=\Omega(n\ln n)$。

证明：将搜索空间S分为$2n$个子空间：

$$S_{0,k}=\{x|x=(0*\cdots*),|x|=k\}\quad(k=0,\cdots,n-1)$$
$$S_{1,k}=\{x|x=(1*\cdots*),|x|=k\}\quad(k=1,\cdots,n)$$

令$X_t\in\{0,1\}^n\,(t=0,1,\ldots)$为随机变量，用于表示局部 (1+1) EA 算法求解 SAT 实例ψ_3时，在时间t所处的状态，可得转移概率，表示如下。

当$k=0$，有

$$P(X_{t+1}\in S_{0,k}|X_t\in S_{0,k})=1$$

当$1\le k\le\frac{n-1}{2}$，

对于$X_t\in S_{0,k}$，如果算法将x_i从 0 翻转为 1，适应度函数下降，该突变不会被保留。

$$P(X_{t+1}\in S_{0,k}|X_t\in S_{0,k})=1-\frac{k}{n}$$
$$P(X_{t+1}\in S_{0,k-1}|X_t\in S_{0,k})=\frac{k}{n}$$

当$\frac{n+1}{2} \leq k \leq n-1$，

对于$X_t \in S_{0,k}$，如果算法将x_1翻转为 1，适应度函数上升，该突变将被保留。

$$P(X_{t+1} \in S_{0,k-1} | X_t \in S_{0,k}) = \frac{k}{n}$$
$$P(X_{t+1} \in S_{0,k} | X_t \in S_{0,k}) = 1 - \frac{k+1}{n}$$
$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{0,k}) = \frac{1}{n}$$

同理，当$k=n$，有

$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = 1$$

当$\frac{n+1}{2} \leq k \leq n$，

对于$X_t \in S_{1,k}$，如果算法将x_1从 1 翻转为 0，适应度函数下降，该突变不会被保留。

$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = 1 - \frac{n-k}{n}$$
$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{1,k}) = \frac{n-k}{n}$$

当$1 \leq k \leq \frac{n-1}{2}$，

对于$X_t \in S_{1,k}$，如果算法将x_1翻转为 0，适应度函数上升，该突变将被保留。

$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{1,k}) = \frac{n-k}{n}$$
$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = 1 - \frac{n-k+1}{n}$$
$$P(X_{t+1} \in S_{0,k-1} | X_t \in S_{1,k}) = \frac{1}{n}$$

此处引入一个具有状态空间$\{z_{0,0}, z_{0,1}, \ldots, z_{0,n-1}; z_{1,1}, z_{1,2}, \ldots, z_{1,n}\}$的辅助齐次马尔可夫链$Z_t \quad (t=0,1,\ldots)$，则转移概率定义为

$$P(Z_{t+1} = z_{v,h} | Z_t = z_{u,k}) = P(X_{t+1} \in S_{v,h} | X_t \in S_{u,k})$$

式中，$u, v \in 0, 1$且$h, k \in 0, \ldots, n$。

此时，Z_t就是一个吸收马尔可夫链，其吸收状态为$z_{0,0}$和$z_{1,n}$。对于任意$x \in S_{u,k} (u \in 0,1, k \in 0, \ldots, n)$，平均首次到达时间$m_x$等于$m_{z_{u,k}}$。

根据吸收马尔可夫链理论，随机过程Z_t的平均首次到达时间表示为

$$\begin{cases} m_{0,k} = n(1 + \cdots + \frac{1}{k}) & 1 \leq k = \frac{n-1}{2} \\ m_{1,k} = n(1 + \cdots + \frac{1}{n-k}) & \frac{n+1}{2} = k \leq n-1 \end{cases}$$

即$||\mathbf{m}||_\infty = \Omega(n \ln n)$。

命题 3.3.3 对于 SAT 实例$\psi_3(x)$，RWF 的期望运行时间为$||\mathbf{m}||_1 = O(n)$。

证明：设$S = \{0,1\}^n$为搜索空间。

将搜索空间S分为$2n$个子空间：

$$S_{0,k} = \{x | x = (0 * \cdots *), |x| = k\} \quad (k = 0, \cdots, n-1)$$
$$S_{1,k} = \{x | x = (1 * \cdots *), |x| = k\} \quad (k = 1, \cdots, n)$$

若当前搜索点属于$S_{i,k}$，将$q_{i,k}$定义为 RWF 在第一步选择带有两个字面量的未满足子句的概率。

根据引理 3.3.1，有$q_{i,k} \geq \frac{2}{3}$。

令$X_t \in \{0,1\}^n \, (t = 0, 1, \ldots)$为随机变量，用于描述 RWF 算法求解 SAT 实例ψ_3时，在时间t所处的状态，则转移概率描述如下。

当$k=0$，有

$$P(X_{t+1} \in S_{0,k} | X_t \in S_{0,k}) = 1$$

当$1 \leq k \leq \frac{n-1}{2}$，

对于$X_t \in S_{0,k}$，如果算法将x_1翻转为 1，适应度函数下降，则该突变不会保留。

$$P(X_{t+1} \in S_{0,k} | X_t \in S_{0,k}) = \frac{1}{2} q_{0,k} + \frac{2}{3}(1 - q_{0,k})$$
$$P(X_{t+1} \in S_{0,k-1} | X_t \in S_{0,k}) = \frac{1}{2} q_{0,k} + \frac{1}{3}(1 - q_{0,k}) \geq \frac{1}{3}$$

当$\frac{n+1}{2} \leq k \leq n-1$，

对于$X_t \in S_{0,k}$，如果算法将x_1翻转为 1，适应度函数上升，该突变将被保留。

$$P(X_{t+1} \in S_{0,k-1} | X_t \in S_{0,k}) = \frac{1}{2} q_{0,k} + \frac{1}{3}(1 - q_{0,k})$$
$$P(X_{t+1} \in S_{0,k} | X_t \in S_{0,k}) = \frac{1}{3}(1 - q_{0,k})$$
$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{0,k}) = \frac{1}{2} q_{0,k} + \frac{1}{3}(1 - q_{0,k})$$

同理，当$k=n$，有

$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = 1$$

当$\frac{n+1}{2} \leq k \leq n$，

对于$X_t \in S_{1,k}$，如果算法将x_1翻转为 0，适应度函数下降，该突变不会被保留。

$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = \frac{1}{2} q_{1,k} + \frac{2}{3}(1 - q_{1,k})$$
$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{1,k}) = \frac{1}{2} q_{1,k} + \frac{1}{3}(1 - q_{1,k}) \geq \frac{1}{3}$$

当$1 \leq k \leq \frac{n-1}{2}$，

对于$X_t \in S_{1,k}$，如果算法将x_1翻转为 0，适应度函数上升，该突变将被保留。

$$P(X_{t+1} \in S_{1,k+1} | X_t \in S_{1,k}) = \frac{1}{2} q_{1,k} + \frac{1}{3}(1 - q_{1,k})$$
$$P(X_{t+1} \in S_{1,k} | X_t \in S_{1,k}) = \frac{1}{3}(1 - q_{1,k})$$
$$P(X_{t+1} \in S_{0,k-1} | X_t \in S_{1,k}) = \frac{1}{2} q_{1,k} + \frac{1}{3}(1 - q_{1,k})$$

此处引入具有状态空间 $\{z_{0,0}, z_{0,1}, \ldots, z_{0,n-1}; z_{1,1}, z_{1,2}, \ldots, z_{1,n}\}$的辅助齐次马尔可夫链$Z_t \quad (t = 0, 1, \ldots)$，则转移概率定义为

$$P(Z_{t+1} = z_{v,h} | Z_t = z_{u,k}) = P(X_{t+1} \in S_{v,h} | X_t \in S_{u,k})$$

式中，$u, v \in 0, 1$且$h, k \in 0, \ldots, n$。

此时，Z_t就是一个吸收马尔可夫链，其吸收状态为$z_{0,0}$和$z_{1,n}$。对任意$x \in S_{u,k} (u \in 0, 1, k \in 0, \ldots, n)$，平均首次到达时间$m_x$等于$m_{z_{u,k}}$。

根据吸收马尔可夫链理论，随机过程Z_t的平均首次到达时间表示为

$$\begin{cases} m_{0,k} \le 3k & 1 \le k \le \dfrac{n-1}{2} \\ m_{0,k} = \dfrac{1}{2}(m_{0,k-1} + 1) + \dfrac{1}{2}(m_{1,k+1} + 1) + \dfrac{1 - q_{0,k}}{2 + q_{0,k}} & \dfrac{n+1}{2} \le k \le n-1 \\ \\ m_{1,k} \le 3(n-k) & \dfrac{n+1}{2} \le k \le n-1 \\ m_{1,k} = \dfrac{1}{2}(m_{0,k-1} + 1) + \dfrac{1}{2}(m_{1,k+1} + 1) + \dfrac{1 - q_{1,k}}{2 + q_{1,k}} & 1 \le k \le \dfrac{n-1}{2} \end{cases}$$

接下来，需要证明

$$m_{0,k} \le 3k \text{ 式中} \frac{n+1}{2} \le k \le n-1$$

当$k = \dfrac{n+1}{2}$，有

$$\begin{aligned} m_{0,\frac{n+1}{2}} &= \frac{1}{2}(m_{0,\frac{n-1}{2}} + 1) + \frac{1}{2}(m_{1,\frac{n+3}{2}} + 1) + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &= \frac{1}{2}(3(\frac{n-1}{2}) + 1) + \frac{1}{2}(3(\frac{n-3}{2}) + 1) + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &= \frac{3n}{2} + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &\le 3\frac{n+1}{2} \end{aligned}$$

可得$k = \dfrac{n+1}{2}$时，$m_{0,k} \le 3k$成立。假设$k \ge \dfrac{n+1}{2}$时上述结论也成立（即$m_{0,k} \le 3k$），

$$\begin{aligned} m_{0,k+1} &= \frac{1}{2}(m_{0,k} + 1) + \frac{1}{2}(m_{1,k+2} + 1) + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &\le \frac{1}{2} m_{0,k} + \frac{3}{2} n - \frac{3}{2} k - 2 + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &\le \frac{3n}{2} - 2 + \frac{1 - q_{0,\frac{n+1}{2}}}{2 + q_{0,\frac{n+1}{2}}} \\ &\le 3(k+1) \end{aligned}$$

进而可得

$$m_{0,k} \leq 3k \text{ 式中 } 1 \leq k \leq n-1$$

对于$m_{1,k}$，可以得出类似结论，如下所示

$$\begin{cases} m_{0,k} \leq 3k & 0 \leq k \leq n-1 \\ m_{1,k} \leq 3(n-k) & 1 \leq k \leq n \end{cases}$$

因此，$||\mathbf{m}||_1 = O(n)$。

4 结语

本文基于两个 2-SAT 实例和一个 3-SAT 实例，对 RW、局部 (1+1) EA 和 RWF 算法进行了分析比较。结果显示，三种启发式算法在求解上述三个 SAT 实例时各有优劣，算法的期望运行时间范围为多项式时间或指数时间。本文深入分析了三种算法的运行时间表现。若不存在局部最优解，RWF 算法表现最佳；存在局部最优解时，最坏情况下预期运行时间可达无限长。

各算法在三个 SAT 实例上的预期运行时间对比如下表所示。

	局部 (1+1) EA	RW	RWF
ψ_1	$\Theta(n \ln n)$	$\Theta(n^2)$	$O(n)$
ψ_2	$+\infty$	$\Theta(n^2)$	$+\infty$
ψ_3	$\Omega(n \ln n)$	$O(2^n)$	$O(n)$

致谢

感谢刘兴武教授指出证明中的一个错误，并为本研究提供了宝贵的建议和意见。

参考文献

[1] Adrian Balint and Uwe Schöning. "Choosing probability distributions for stochastic local search and the role of make versus break ". In: *Proceedings of the 15th International Conference on Theory and Applications of Satisfiability Testing*. Berlin, Heidelberg, 2012, pp. 16–29.

[2] Armin Biere et al. "Conflict-driven clause learning SAT solvers". In: *Handbook of Satisfiability, Frontiers in Artificial Intelligence and Applications* (2009), pp. 131–153.

[3] Shaowei Cai, Chuan Luo, and Kaile Su. "CCAnr: a configuration checking based local search solver for non-random satisfiability". In: *SAT*. 2015.

[4] Amin Coja-Oghlan et al. "On smoothed CNF formulas and the WalkSAT algorithm". In: *Proceedings of the Twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*. 2009, pp. 451–460.

[5] Evgeny Dantsin, Edward A Hirsch, and Alexander Wolpert. "Algorithms for SAT based on search in Hamming balls". In: *Annual Symposium on Theoretical Aspects of Computer Science*. 2004, pp. 141–151.

[6] Stefan Droste, Thomas Jansen, and Ingo Wegener. "On the analysis of the (1 + 1) evolutionary algorithm". In: *Theoretical Computer Science* 276.1-2 (2002), pp. 51–81.

[7] Russell Impagliazzo and Ramamohan Paturi. "On the complexity of k-SAT". In: *J. Comput. Syst. Sci.* 62.2 (2001), pp. 367–375.

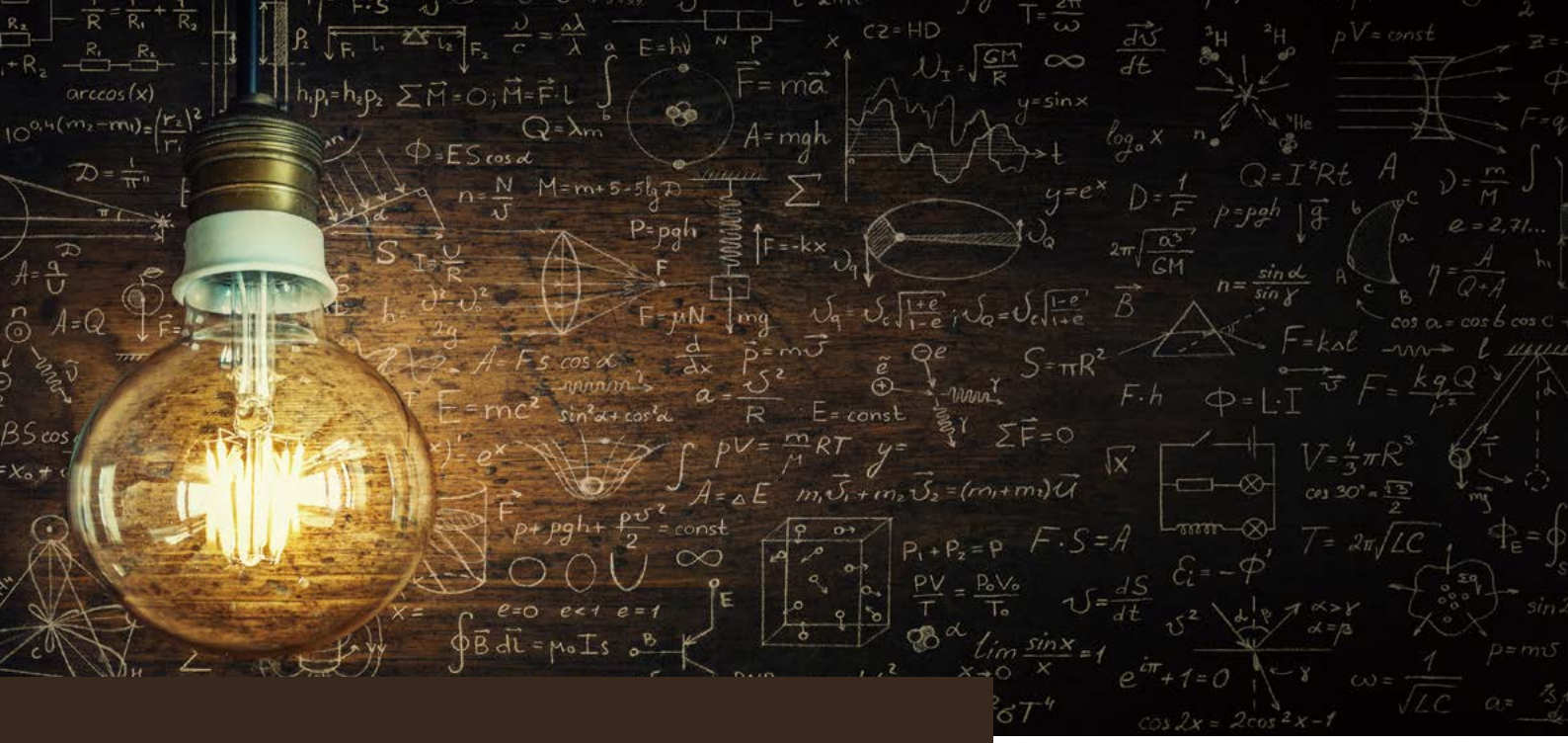
[8] Marius Iosifescu. *Finite Markov processes and their applications*. Courier Corporation, 2014.

[9] Christos H Papadimitriou. "On selecting a satisfying truth assignment". In: *[1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science*. IEEE Computer Society. 1991, pp. 163–169.

[10] Ramamohan Paturi et al. "An improved exponential-time algorithm for *k*-SAT". In: *J. ACM* 52.3 (2005), pp. 337–364.

[11] T Schoning. "A probabilistic algorithm for k-SAT and constraint satisfaction problems". In: *Proceedings of FOCS 1999*. 1999, pp. 410–414.

[12] Yuren Zhou, Jun He, and Qing Nie. "A comparative runtime analysis of heuristic algorithms for satisfiability problems". In: *Artificial intelligence* 173.2 (2009), pp. 240–257.



域理论与交互式计算

Glynn Winskel
爱丁堡研究所

摘要

20 世纪 60 年代末，Dana Scott 和 Christopher Strachey 提出了域理论和指称语义，首次为编程语言的语义及其分析工具提供了数学理论支持。域理论有着深远的影响。时至今日，与域理论相关的研究，尤其是关于其在概率编程、量子编程和可微分编程等领域的扩展研究依然活跃。本文将从域理论作为可计算函数的理论基础开始，分析经典域理论的局限性，并基于并行博弈和并行策略介绍近期域理论与交互式计算领域的最新进展。

1 引言

经过 50 多年的发展，域理论与计算机科学已密不可分。域理论虽然有其局限性，但仍然是计算形式化和计算分析不可或缺的手段。域理论用代数中的域（Domain）来刻画程序语言中的类型（Type），用（连续）函数来刻画程序本身。在许多情况下，域理论的模型和推理方法仍是我们能够想到的最简单的分析手段。即使从这个方面来说，域理论的重要性也不言而喻。域理论的发展也伴随着指称语义方法的发展。指称语义通过组合的方式刻画了编程语言和系统的行为。据我们所知，指称语义是管理编程语言和系统复杂性的唯一方法，其可靠性已在多年来的学术研究成果中得到验证。

计算机科学领域的实践和创新不断涌现，以至于大家有时候会忘记域理论曾经的历史地位。

然而时至今日，域理论和指称语义在很多领域中依然十分重要。例如，它们对函数式编程语言的发展演变过程中起到了关键作用，（编译中广泛使用的）Continuation 的概念也是从中发展而来。此外，抽象解释和指称语义一脉同源，是许多静态分析工具的理论基础。大家熟知的新晋系统编程语言 Rust 中重要的线性逻辑，也是从域理论中衍生而来的。正如本文所述，域理论和指称语义在交互式计算中相得益彰。时至今日，它们之间的联系依然紧密。

域理论最初的诞生，是由于人们希望以数学中的函数概念来刻画计算模型。但是随着计算的交互程度提高，域理论的局限性也慢慢显现，这点并不意外。函数式计算历史悠久，而交互式计算的相关理论则仍在发展当中。

最后，本文将介绍近期域理论在分布式博弈/并行博弈与交互式计算相结合方向上的研究。在这一方向中，域的角色被博弈取代，函数的角色则被策略取代。从计算机程序的角度来说，类型变成了博弈，而程序则变成了策略。

系统和环境的二分关系是所有交互的相关组成理论的核心。并行博弈和并行策略用局部的眼光，通过区分玩家行动（系统事件）和对手行动（环境事件），对这种二分关系进行细致的处理。

另一方面，由于函数式计算中的输入和输出的角色更加固定（甚至略显生硬），我们在用函数式计算刻画交互时，需要更加巧妙地处理这种二分关系，由此诞生了许多不同的函数式方法来刻画特定的交互行为，包括 Girard 提出的交互几何、Gödel 提出的辩证翻译，以及以透镜（Lens）为代表的函数式引用（Optics）[1] 及后者向容器 [2] 的扩展，这种扩展可以通过对透镜等函数式引的依赖类型的扩充实现。有些意外的是（至少笔者认为如此），这些函数式方法的域理论表达，都是来源于并行博弈中的一些相当简单的特定情况。

《程序设计语言的形式语义》一书对域理论及其历史进行了深入的探讨，该书中英文版本均已出版发行（英文书名为 The Formal Semantics of Programming Languages）[1]。如果想进一步了解域理论，可以参考 [2] 中的相关章节。本文的后半部分讨论了并行博弈和并行策略及它们与函数式计算的关系，相关主题在通讯文章 [3] 作了通俗易懂的阐述，[4] 则从技术角度进行了详细的说明。

撰写本文时，我尽量采用通俗易懂的表达。本文中斜体文字是技术细节及一些补充说明。如果想快速了解域理论的大体内容，可以考虑跳过斜体部分。

2 起源

域理论的历史可谓无人不知。在 20 世纪 60 年代中期，Christopher Strachey[3] 意识到，要理解他正在开发的复杂编程语言，需要一种新的技术——用数学模型提供编程语言的语义，这样才能验证编写的程序是否正确。

在牛津大学沃弗森学院的一次午餐上，物理学家 Roger Penrose 建议 Strachey 关注一下 λ 演算。逻辑学家常用 λ 演算来描述和推理可计算函数。与此同时，来自普林斯顿大学的逻辑学家 Dana Scott 却对 λ 演算的无类型特点持高度批评的态度。原因在于，λ 演算允许自应用（即一个函数使用了该函数自身）这一矛盾现象的存在，而自应用这种构造是传统集合论禁止的。不幸的是，λ 演算中用以描述递归定义的组合子恰恰依赖于自应用。Strachey 和 Scott 的相遇碰撞出各种火花。他们一见如故，开启了两人标志性的合作。

在 Scott 的劝说下，Strachey 将目光投向了安全类型的 λ 演算（Safe Typed λ−Calculus）。为了支持 Strachey 的想法，Scott 还提出了“可计算函数的逻辑”（Logic of Computable Functions，LCF）。作为逻辑学家，Scott 从一开始就关注了其背后的数学理论基础。他提出类型可以通过域（拓扑空间的简单形式）来刻画，其中的序关系可以由类型中的元素的近似程度来定义。虽然可计算函数可用于无限输入（例如，可计算函数的输入可以是自然数上的函数），但也只是对输入的有限逼近。因此，Scott 认为，可以将可计算函数看作是从输入域到输出域的连续函数。在域理论中，函数的连续性通常符合其在拓扑学中的一般理解，即函数能够保留（以近似程序定义的）序关系和链的上确界（Least Upper Bound）的不变性。

域的最简单形式是完全偏序，即在近似程度上定义的偏序 $(D, \sqsubseteq_D)$，其中最小元素记为 $\perp_D$。对于 D 中的链 $d_0 \sqsubseteq_D d_1 \sqsubseteq_D \cdots \sqsubseteq_D d_n \sqsubseteq_D \cdots$，定义它的上确界为 $\sqcup_n d_n$（可以认为是一种极限点）。当 $d \sqsubseteq_D d'$ 且 D 中任意链都有

[1] Optics 原意为光学元件，函数式计算研究人员常用光学体系的名称来命名函数式计算中的概念。

[2] 此处容器为函数式编程而非虚拟化领域的概念。

[3] 著名计算科学家，分时系统领域概念的提出者。

$\sqcup_n F(d_n) = F(\sqcup_n d_n)$*时，若*$F(d) \sqsubseteq_E f(d')$*，则从域*$(D, \sqsubseteq_D)$*到域*$(E, \sqsubseteq_E)$*的函数*$F$*为连续函数。根据 Kleene [1] 的早期理论，从域到其自身的连续函数*F*具有最小不动点*$\mathrm{fix(F)}$*，它可以由上确界*$\sqcup_n F^n(\perp) =: \mathrm{fix(F)}$*来构造。*

域和连续函数的一个显著特点在于，与一般的拓扑空间不同，在函数点态序（Pointwise Order，也称 Scott 序）下，从域D到域E的连续函数集本身构成了一个域，即函数空间$[D \to E]$。也就是说，域$D \times E$的笛卡尔积（其中每一个元素都是一个二元组），按坐标排序时就构成了一个域。在这种情况下，从D到E的递归函数可以由$[D \to E]$上的连续函数F的最小不动点与递归函数本身来表示。LCF 包括一个不等式逻辑，基于该逻辑可以利用 Scott 归纳等工具来对递归程序进行推理，例如构建递归程序的最小不动点等。

1969 年，Scott 写了一篇关于 LCF 的论文，这篇论文广为传播，很有影响力，但这篇论文直到 1993 年才正式发表 [5]。因为 Scott 突然意识到，将递归函数表示为最小不动点的方法，可以进一步推广到类型层面，以此提供一个非平凡域D，该域与其连续函数的域$[D \to D]$ ——（无类型）λ 演算的模型是同构关系。Scott 也因此不再对 λ 演算持批判态度 [6, 7]。Scott 的这一发现，引领了新一波研究热潮。

3 理论传播

在牛津之外，也有很多研究人员做出了不小的贡献。来自华威大学的 David Park 发现，在 Scott 的模型中，λ 演算的矛盾组合子（Paradoxical Combinator，也称为 Y 组合子）表示的是其最小不动点算子。Scott 和爱丁堡大学的年轻学者 Gordon Plotkin 各自独立发现了通用域（Universal Domain）。在通用域中，如果用域中的函数来描述类型，那么任何类型都可以用递归的方式定义。Plotkin 和 Mike Smyth（当时在华威大学从事博士后研究），在 Scott 的模型上进一步拓展，用范畴学的思想来处理递归定义域，这有助于理解各种递归类型，为函数式编程打下了数学的基础。

与此同时在斯坦福大学，Robin Milner、Malcolm Newey 和 Richard Weyrauch 对 LCF 的机械化证明有了进一步的研究 [8, 9]。Milner 发明了函数式语言 ML，作为一门元语言（Meta Language），ML 可以更好地支撑辅助证明。ML 结合了 LCF 以及 Peter Landin 发明的 ISWIM 语言[1]的函数式特点 [10]。通过严密的类型系统规则，ML 能够确保只有产生合法 LCF 证明的程序才能被定义为一条定理（Theorem）。Milner 开始研究并行计算的语义，并通过预言机（Oracle）来解决非确定性选择问题。同样来自斯坦福的 Zohar Manna 和他的学生 Jean Vuillemin 在递归定义程序推理方面的研究，将 Milner 的研究进一步发扬光大 [11]。

域理论在计算机科学和逻辑之间建立起了新的联系，编程语言和计算系统可以用数学方法进行分析。计算机科学家们相信，合理的语言结构可以被赋予数学定义，这增强了他们在研究编程语言时的信心。域理论所提供的指导原则对编程语言的设计产生了深远的影响，并为后来的函数式编程奠定了基础。

到了 20 世纪 70 年代中期，用域理论来解释编程语言的各种特性似乎只是时间问题。Strachey 和 Christopher Wadsworth 用 Continuation 为指令式语言中的跳转（Jump）指令赋予了语义 [12]。在 Egli 和 Milner 的早期思想的基础上，Plotkin 用他提出的幂域，将域理论扩展到对非确定性和并行程序的处理上 [13]，成功避免了 Milner 在使用预言机处理并行组合时遇到的模型无法满足结合性和交换性的问题。Nasser Saheb-Djahromi 进一步构建了概率幂域，为概率编程给出了指称语义。法国青年学者也开始在相关研究中发挥他们的聪明才智和数学专长。

4 局限性——早期迹象

域理论提供的能力似乎能够让我们持续地通过数学方法来刻画编程语言的语义。域理论是指称语义方法的基础，通过对编程语言语法的结构化归纳实现对程序语言的语义的组合定义。

然而早期研究的种种迹象也表明，域理论也存在限制。Gilles Kahn 发现，域理论非常适合数据流的处理，因为数据流进程可以简单地表示为输入流到输出流的连续函数，域理论用最小不动点处理递归的方法也可以用于处理网络中的循环 [14]。但如果数据流是非确定性的，就会出现问题。1981 年，Brock 和 Ackerman 指出，给非确定性数据流定义组合语义非常困难 [15, 16]。虽然有方法可以为非确定性数据流定义指称语义，但这些方法超出了传统域理论的范围。

在计算机理论研究中有一种传统，每当人们发明了一种新的理论（例如具备新的特性），往往会尝试在函数空间中测试这种理论（以弄清它们在函数空间中的表达）。在这种传统的指引下，Plotkin 基于 LCF 构造了一种新的编程语言，叫做程序可计算函数（Programming Computable Functions，PCF）[17]。他提出了**全抽象**（Full Abstraction）的概念，这一概念由 Milner 命名。全抽象指的是一种语言的指称语义和操作语义之间的一致性[2]。Plotkin 是通过研究 PCF 发现的这一概念。由于 Scott 定义的域中包含“parallel or”等额外语法（Parasitic Element），

[1] ISWIM 是 Peter Landin 开的一个玩笑，是“如果你明白我的意思”（If You See What I Mean）这句话的缩写。

[2] 所谓指称语义和操作语义之间的一致性，是指在指称语义下的等价性与在操作语义下的等价性是一致的。

而 PCF 中无法定义这些语法要素，因此在 PCF 中操作语义的等价性与指称语义的等价性存在分歧。

这也激发了将域理论与操作语义结合相关探索的热情，希望以此刻画 PCF和 λ 演算的顺序求值语义。对于（定义在域的积上的）连续函数$f : D_1 \times \cdots \times D_m \to E_1 \times \cdots \times E_n$要满足什么条件才是**顺序**函数的问题，Milner 和 Vuillemin 分别给出了定义（二人定义略有不同）：对于满足$f(x_1, \cdots, x_m) = (y_1, \cdots, y_n)$的特定输入$(x_1, \cdots, x_m)$，其输出$y_j$的增加必须是由于某个关键输入$x_i$的增加 。该定义有一个缺点，它依赖于将域解构为（各个子域的）积的方法。

在 1975 年，Kahn 和 Plotkin 定义了**顺序函数**的概念，这里的顺序函数被定义为**具体域**之间的映射，并且将函数的输入输出称为 Cell [18]，这种定义弥补了前面提到的不足。简单来说，具体域的每个元素由一组事件构成，每个事件可以向一个 Cell 中填充特定的值。随着事件发生，有更多的 Cell 变为可访问状态。每个可访问的 Cell 最多可填充一个值。根据 Kahn 和 Plotkin 的定义，具体域之间的顺序函数一定是连续的。该连续函数中的任意输入中，填充可访问的输出 Cell 需要填充关键可访问输入 Cell。需要注意的是，关键 Cell 的数量可能有多个。问题在于顺序函数所构成的空间本身并不是一个具体域，因此具体域并不适合用于给 PCF 定义指称语义。

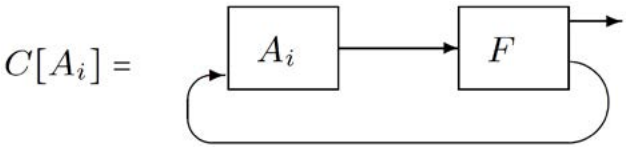
Gérard Berry 提出了两种解决问题的方案。首先他提出，将域间连续函数限制在**稳定**的函数上 [19]。（所谓稳定的函数，实质上是对顺序函数的一种近似）。从技术角度来说，稳定函数能够保持元素兼容子集的下确界。更直观地来说，在一个稳定函数中，输出的任何部分都取决于输入中一个唯一最小部分（即下文序关系中的“最小输入集”）。值得注意的是，如果将域用合理的方式公理化（得到 Berry 所称的 **dI 域**），就有了函数空间：dI 域之间所有稳定函数的集合，通过精化后的 Scott 序（**稳定序**）排序时，就形成了 dI 域。如果两个函数之间在 Scott 序中可以比较，且产生公共输出时的最小输入集相同，那这两个函数在稳定序的意义下也是可比较的。Berry 进一步研究了**双域**（Bidomain）。双域同时具有 Scott 序和稳定序。Girard 首先在多态模型中使用**定性域** [20]，然后又通过稳定函数关联的特殊类型 dI 域**相干空间**（Coherence Space）创造性地发现了线性逻辑（Linear Logic）[21]。Jean-Yves Girard 在 20 世纪 80 年代中期的这一系列研究成果，极大地推动了 Berry 的稳定域理论的发展。

稳定函数固然重要，但在顺序性面前，它只是一种近似。Berry 和他的学生 Pierre-Louis Curien 发现了一种为顺序性构建域的方法及相关理论，该方法可以为 PCF 提供指称语义，但偏离了函数抽象。他们提出，如果具体域使用**顺序算法**，则具体域具有函数空间的形式 [22]。顺序算法可以用局部决策来表示，一个局部决策决定是输出一个值还是检查某个 Cell 的值。Berry 和 Curien 开创性的研究成果是博弈语义的前身。François Lamarche 认为，顺序算法是一种策略，其决策对应博弈论中的“行动”[23]。然而，Berry 和 Curien 也发现顺序算法也可以视作是一种具有辅助函数的顺序函数，该辅助函数在有输入和可以访问的输出 Cell 时，可返回一个可访问的关键输入 Cell（由此引申出来了后来函数式编程中的透镜）。

从顺序算法开始，人们以一种更具交互性的视角来看待计算本身——计算不仅仅是从输入到输出的函数计算，而是算法主动查询输入、检查输入、给 Cell 赋值（最终获得输出）的过程。当时，越来越多的声音认为计算应以交互为基础。顺序算法和稳定函数的特点，让这种呼声更加高涨。

非确定性数据流的 Brock-Ackerman 异常：Kahn 证明，简单域理论在确定性数据流中的应用极具亮点，但却无法应用到非确定性数据流上。这也是计算需要更具交互性的一个早期信号。非确定性数据流的组合语义需要一种通用的关系，用来描述输入 - 输出组合实现的方式。Brock 和 Ackerman 在他们早期的研究中就提出了这一点，他们也是最早强调为非确定性数据流赋予组合语义具有挑战性的学者。不过我们的例子是基于 Rabinovich 和 Trakhtenbrot [24] 以及 Russell [25] 后来研究的简化版本。

用以下数据流为例：



这里包含两个简单的非确定性进程：A_1和A_2。虽然这两个进程的输入 - 输出关系相同，但在前面提到的同一个上下文$C[-]$中，两个进程的行为并不相同。如图所示，该上下文包括一个 Fork 进程F（该进程将每个输入复制到两个输出），自动机A_i的输出通过该进程与自己的输入相连。进程A_1的行为是非确定性的：要么输出一个符号然后停止，要么输出一个符号，然后等待一个输入的符号，再输出另一个符号。进程A_2的行为也是非确定性的，但与A_1略有不同：要么输出一个符号然后停止，要么等待一个输入符号，然后输出两个符号。对于这两个进程，输入 - 输出关系都可以归纳为：如果输入为空，则最终输出一个符号；如果输入非空，则最终输出一个或两个符号。$C[A_1]$可以输出两个符号，而$C[A_2]$只输出一个符号。需要指出的是，A_1有两种从空输入到单个输出符号的实现方法（两种不同行为），而A_2只有一种。因此，A_1在不同上下文中的行为更丰富，而这些行为上的区别无法单纯依靠输入 - 输出关系来体现。

*多年来，人们提出了各种方案，试图为非确定性数据流赋予组合语义，这些方案都依赖于广义的“关系”来区分不同的给定输入来生成输出的方式（或行为）。通常使用**稳定生成空间**（Stable Span），可以为（非确定性数据流）赋予组合语义。稳定生成空间是 Berry 所定义的稳定函数的一种扩充 [26]。因此，稳定生成空间作为并行策略的一种特殊形式重新进入我们的视野，具体请见第 8 节。*

5 交互

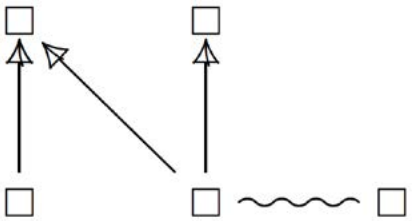
并行进程可以独立执行，但进程之间存在交互。长期以来，如何刻画并行进程一直是传统域理论的一个棘手难题。虽然确定性数据流等特例很容易通过域来表达，且 Plotkin 所定义的幂域能够借助递归定义的 **Resumption**[1] 域来通过动作的非确定性交错来刻画（进程的）并行组合 [13]。然而通过这些方式定义出来的并行程序指称语义可能看起来很复杂。事实上，在进行了一些域模型方面的早期尝试后，这些复杂的问题让 Robin Milner 放弃了域理论和指称语义，转而投向基于 Plotkin 结构化操作语义和互仿真等价性的交互系统演算（Calculus of Communicating Systems，CCS）模型 [27]。另一方面，Tony Hoare、Steve Brookes 和 Bill Roscoe 则提出了一套名为通信顺序进程（Communicating Sequential Process，CSP）[2] 的并行模型 [28]。该模型定义了失败集合（Failure Set）的概念并用它来刻画对外可见的行为。Hoare 和 Milner 都认为同步（可能带有值的交换）是通信的基本原语。多年来，并行性是一个相对独立的研究领域，且某种程度上与语法相关性更高 [3]。

20 世纪 60 年代初，Carl Adam Petri 等人开始开发一种全新的计算模型——**Petri 网**。Petri 网的基本概念是事件，事件仅仅改变局部的状态。Petri 网用**标记**来描述全局状态（标记可以视为所有局部状态的集合），标记可以告诉我们哪些局部条件是成立的。事件的发生会使其先决条件（有指向该事件箭头的条件）不成立，同时使后置条件（有从该事件引出箭头的条件）成立。

Petri 所提出的结构与 Kahn 和 Plotkin 在顺序性研究中发现的用来表示具体域的结构非常相似。事实上，通过**事件结构**（包括一系列具有因果和冲突关系的事件）这一中间概念，Mogens Nielsen、Gordon Plotkin 和笔者实现了在 Petri 网和域理论两个领域之间的概念转换。状态转移系统（的状态变迁过程）可以展开为树，Petri 网同理也可展开为事件结构 [29, 30]。值得一提的是，Petri 网中无混淆的理念，与 Kahn 和 Plotkin 将非确定性选择约束到 Cell 内的做法是不谋而合的。不久之后，人们意识到，Berry 的 dI 域其实是以包含关系作为序的，事件结构构造（Configuration）的域 [31, 32]。

*事件结构的最简单的形式为$(E, \leq, \#)$，包括一个代指因果依赖关系的偏序$\leq$、代指事件集合E上冲突的二元关系$\#$，这个二元关系是对称且非自反的。$e' \leq e$表示事件e在因果上依赖于e'的发生。$e \# e'$表示事件e或事件e'只能发生一个。上述关系在一起满足了两个公理：首先，一个事件在因果上只依赖于有限数量的事件；其次，如果两个事件依赖于冲突的事件，那么它们自身也必然是冲突的。事件结构的状态或事件结构的历史状态可以由“构造”这个概念来定义：事件结构的**构造**由事件子集构成，该子集相对于$\leq$是向下闭合的，且不存在冲突。如果两个事件相互之间既不冲突，也不存在因果依赖关系，则认为这两个事件因果独立——即**并行**关系。*

如果用方块表示事件，箭头表示直接因果依赖关系，波浪线表示直接冲突，如下图所示：



上图表示的是包含 5 个事件的事件结构。右下角的事件与另一个事件直接冲突（即波浪线）。除左下角的事件外，其他事件均与右下角事件有冲突；左下角的事件与右下角的事件为并行关系。

Petri 网主要用于对并行进程建模，域理论中对应的结构（即域本身）则是用来表示进程的类型。这种差异一度让人感到奇怪。后来随着域理论的拓展，二者实现了协调统一，类型和进程都可以表示事件结构，就如并行博弈和并行策略一样。

随着结构化模型以及模型在并行意义下的等价性越来越复杂，传统的域理论与并行计算变得越来越不合拍。例如，Petri 网承载的结构比信息域中的一个点可以支持的结构要多得多。

幸运的是，范畴论可以组织并行场景下的模型：单个模型（例如 Petri 网、事件结构或变迁系统）可以通过独有的映射方式来构成范畴。这些映射代表了关于事件的模拟形式，例如，可以关联并行组合的行为与其组成分支行为。不同范畴模型之间可以通过伴随关系联系起来，这有助于我们更加系统化地分析并行进程的等价关系 [33, 34]，同时将并行模型的语法与它们的操作语义解耦（在此之前二者往往是绑定的）。

例如，从E到E'的事件结构映射是关于事件的部分函数f，且需要保证事件及其构造的基本性质。它将E的事件结构构造x映射为E'中的构造$f\,x$，且保证x中两个不同的事件不会同时映射到$f\,x$中的同一个事件。虽然f不需要保持因果依赖关系，但这一因果依赖需要在局部反向保持：若$e, e' \in x$且$f(e) \leq f(e')$，则$e \leq e'$。因此，事件结构的映射自然会保留事件的并行关系。

这种分类法来源于现有模型，但它同时也提供了一类更泛化的模型，和域理论一样有着广泛的适用性，可以用域理论相同的方式进行调整——广义域理论的一种形式 [35，36]。进程的早期域模型认为，进程是一组它的计算路径的集合，

[1] 此处 Resumption 指的是可以恢复并执行后续计算的状态。
[2] 通信顺序进程即是近年来大火的 Go 语言所使用的并行理论模型。
[3] 以何种语法形式来进行并发在研究中受到更多的关注。

其中一个非常著名的就是 Hoare 在 CSP 理论中使用的“迹”（Trace）模型。在该模型中，进程表示的是一组该进程可执行的可见动作。广义域理论与之类似，但在广义域理论中，进程不是计算路径的**集合**，而是用计算路径的**预层**（Presheaf）来表示进程。本质上讲，预层类似于特征函数的一种推广，但是其中的真值由集合代替，这个集合包含了将会到达真值的路径。通过用预层来表示进程，进程具有不同计算路径（但它们在预层意义下有相同的形状），并可以跟踪进程是如何以确定性的方式分支的。并行预层模型将并行计算与更强的数学模型 联系起来，尤其是与物种数学（组合数学的一个分支）。不过，预层模型定义操作语义很有挑战性。有时，预层的指称语义可以用事件结构表示。从技术上讲，预层指称元素的范畴的形式与事件结构的构造相同。如果深入讨论事件结构的作用，就引出了基于事件结构的博弈语义，详见第 8 节。

6 来自法国的影响

Jean-Yves Girard 在法国逻辑学和计算学界有着很高的声望。他认为当时在逻辑的构建和分析中过度使用代数，且使用的代数方法过于简单，对此表示怀疑。他对 Alfred Tarski 关于一阶逻辑模型中对真值的定义以及指称语义持批判态度。不过，Girard 的研究却为指称语义做出了非常大的贡献[1]。Girard 强调通过操作语义来理解证明和计算机制，但这并不意味着他的研究抛弃了数学模型，或者说像传统证明论那样过度关注语法层面。相反，他所构建的模型兼具深度和独创性，关注到了理解证明和计算的新方法。

通过在受限相干空间中对稳定域理论的改造，Girard 在 20 世纪 80 年代中有了更重要的发现——**线性逻辑** [21]，将传统的逻辑解构为更为基础的资源敏感（Resource-conscious）型逻辑。这项研究让域理论的焦点，从支持单一积柯里化的函数空间，转向了更加广泛的张量积柯里化的函数空间。从范畴论的角度来说，重点从笛卡尔封闭范畴转向了幺半封闭范畴。如今，线性逻辑模型在计算语义中几乎无处不在[2]。Girard 的相干空间对应了一种非常特别的事件结构形式，这种形式的事件结构中，因果依赖是平凡恒等关系。

在研究线性逻辑的证明过程中，Girard 发现了**交互几何**（Geometry of Interaction，GoI）[38]。GoI 最初是用量子力学的数学结构来解释的，之后 Samson Abramsky 和 Radha Jagadeesan 发现 GoI 与传统域理论联系更加紧密，这里的交互可以使用最小不动点以 Kahn 网的形式实现 [39]。Martin Abadi、George Gonthier 和 Jean-Jacques Lévy 将 GoI 与 Lévy 的 λ 演算最优规约通过基于符号的计算联系了起来，影响了编程语言的实现 [40]。今天，GoI 常被视作是博弈语义的早期形式，具体请见第 8.3 节。

[1] 主要包括多态域模型 [37] 以及线性逻辑 [21] 等方面。

[2] 线性逻辑及其衍生的仿射逻辑是 Rust 语言的重要理论基础之一。

7 博弈语义

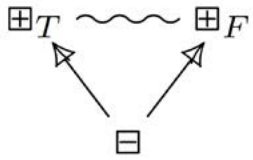
早期，人们对于解决 PCF 全抽象问题的关键还比较模糊。

直到 20 世纪 80 年代，在域的表示方法方面才取得了一些技术进展：具体数据结构 [18] 和事件结构 [32] 为函数如何进行计算给出了偏操作语义的描述；Scott 和笔者为信息系统给出了逻辑表示，极大简化了域的递归定义及其逻辑关系 [41-43]。Girard 的研究也常用到这类域模型的技术细节。

回顾这些历史研究就会发现，为 PCF 构建全抽象模型的早期尝试都是通过将额外的结构附加到域或域的表示中。例如，在双域和双结构中同时使用 Scott 序和稳定序 [44]，利用 Ehrhard 超相干 [45]，以及使用 O'Hearn 和 Riecke 所定义的表达能力更强的逻辑关系 [46]。这些尝试的努力被 Ralph Loader 终结，或者说至少被 Ralph Loader 泼了一盆冷水。Loader 著名的研究证明 PCF 无法有效实现全抽象，这与研究者们之前的共识是一致的。也就是说，PCF 全抽象域模型的表示是不可计算的 [47]。

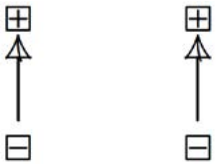
这又抛出了另一个中间问题——是否存在其他更加独立的抽象模型，且模型中的所有有限元素可以在用 PCF 中定义？如果这样的模型存在，那么可以通过商化得到一个域模型。这个问题被称为“内涵性全抽象”（Intensional Full-Abstraction）。这个问题有两个不同的答案，但它们不约而同从博弈论出发，开创了博弈论在编程语言语义中的有效应用。Samson Abramsky、Radha Jagadeesan 和 Pasquale Malacaria 发明了 AJM 博弈 [48]，Martin Hyland、Luke Ong 和 Hanno Nickau 发明了 HO 博弈 [49]。在许多方面，博弈语义更适合与操作语义结合的域理论——域的作用被博弈替代，而连续函数则被策略所替代。博弈的作用从函数式编程拓展到了指令式编程。

为了让大家更好理解程序的博弈语义，我们将博弈和策略用事件结构来表示，这是一种不太常规的方法。相关示例请见第 8 节提到的并行博弈和并行策略实例。首先，我们用事件结构描述布尔类型相关的博弈。在该博弈中，对手（代表程序执行的环境）首先做出行动，请求一个值（事件⊟，后续用该符号表示对手事件）。然后玩家（代表程序本身）可以做出行动回应对手的行动，给出一个真值（事件$\boxplus_T$，后续用该符号表示玩家事件）或假值（$\boxplus_F$，该事件与$\boxplus_T$为相互冲突的事件）。



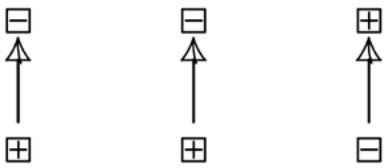
为了说明博弈中的类型结构，将博弈简化。先看一个单值类型（该类型只有一个值，例如函数式程序设计中的Unit）。在这个类型对应的博弈中，对手首先请求一个值（事件⊟）。然后，玩家可以做出行动回应，给出一个值（事件⊞）。将此博弈用事件结构表示如下：

接下来用博弈来表示一个二元组类型，可以简单地将单值博弈的两个副本并行放置，得到如下事件结构：

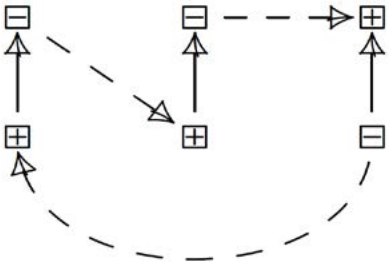


对手可以只请求左边的值或者右边的值，也可以同时请求两个值。玩家可以回应也可以不回应。

*下一步，用博弈来表示一个函数类型，这个函数类型将二元组映射到单值。接下来我们要考虑博弈中的策略（Strategy）。在博弈论中，大多数时候只关注单个博弈**中**的策略，**跨**博弈的策略的并不太为人所熟知。但事实上，跨博弈的策略有相当广泛的应用[1]。基于这一想法，以下面这个博弈中的策略为例，其中的策略就是从二元组到单值的博弈策略。在第 8 节将对中此进一步解释。*



此博弈为两个博弈的并行组合（其中一个为二元组类型，用于获得输入；另一个为单值类型，用于获得输出）。相比于前面提到的二元组类型博弈，此博弈中二元组类型博弈的对手和玩家角色发生了互换。这个博弈中的策略，本质上是一个程序。以下面这个事件结构为例。此结构代表的策略（或程序），首先会检查并行博弈组合中的左分支，然后检查右分支，最后产生输出值：



[1] 在 20 世纪 70 年代初 John Conway 在与剑桥大学 Archimedeans 数学协会一次关于数字与博弈的交流中，笔者第一次有了这一想法。

一般而言，在上述博弈中的某个策略如果回应了对手的第一步，它的回应要么是输出值，要么会请求更多的输入以做出进一步行动。希望这样的描述能够帮助大家理解如何用策略来刻画一个程序。虽然描述编程语言时，策略需要获胜条件（比如要确保程序确实在执行或能够正确运行），但许多传统的博弈语义并不涉及获胜条件。

博弈论的故事远不止于此。一方面，AJM 和 HO 两种博弈语义如何才能走向一致，从一开始就没有明确的方案。另一方面，更重要的是，博弈语义是基于玩家与对手的顺序交替行动。

博弈论过于看重玩家与对手的顺序行为，这反过来阻碍了博弈论的发展。在**博弈论**中，这一问题导致了不同类型的博弈，每种博弈都支持其中某一些特性。这样对并行性的处理方式非常零碎，通常是用额外的结构来记录信息（但是这些信息通常是不完整的），限制了博弈论中博弈与策略的组合方式。虽然博弈语义与结构和组合密切相关，但博弈大多数情况下是顺序进行的。多数情况下，并行性间接体现在博弈参与者原子动作与原子动作之间的交替。这样就很难有效处理系统分布性的特点，阻碍对其因果依赖关系的分析，往往还伴随着补偿性、**临时**的公平性假设。

缺少一种为分布式博弈 / 并行博弈而生、表达能力更强的代数理论，将玩家和对手当作团队成员，这些成员分布在不同的位置，可以做出行动、相互交流。虽然早期研究中出现了这种理论的雏形 [50–54]，但渐渐地，它们各自使用的博弈和策略体系逐渐被事件结构所替代，最终这些早期研究走向了统一的方向 [55, 56]。

8 并行策略

分布式博弈 / 并行博弈让我们可以重新思考博弈在计算机科学中的交互理论和模型中的基本原理，赋予这类博弈更加灵活的基础。其中的核心思想是用事件结构代替传统开发中的博弈树。

用事件结构表示**并行博弈**时，事件带有极性标记，用于区分玩家和对手的行动：玩家（系统或程序）的行动记为+，对手（未知环境）的行动则记为−。

博弈中通常还有获胜条件、收益函数等要素。获胜条件是玩家获胜需要的（事件结构）构造，而收益函数则为每个构造分配奖励。简单起见，假设博弈是**非竞争性的**，也就是说玩家和对手的行动之间不存在直接冲突。即使玩家和对手之间存在冲突，也肯定是源自玩家两次行动或对手两次行动之间的冲突。

用事件结构来表示博弈，博弈的过程就不再是一个行动的序列（可以理解为一个全序），而是行动之间的因果关系的偏序。从全序到偏序的这种转变，如果不能有效处理，

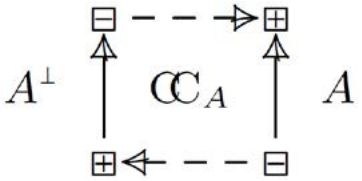
会带来技术上的困难，以及不必要的复杂性。幸运的是，我们还有为交互进程开发的数学工具可以利用，特别是在处理事件结构时 [32，33]。

在两方博弈中，有两个非常重要的基本操作。一个是**对偶**博弈的形成，玩家和对手的角色发生互换形成对偶博弈。对于具有极性的事件结构A，通过反转其事件极性，可以得到它的对偶$A^{\perp}$。在博弈中，我们通常说的“策略”其实指的就是玩家的策略，对手策略称为“应对策略”（Counterstrategy）。任意博弈A中的应对策略，一定是其对偶的博弈$A^{\perp}$中的一个策略。另一个操作是在事件结构A和B上实现博弈的**并行组合**，只需将A和B并列，使不同分支中的事件之间不存在冲突，将组合此记作$A \| B$。

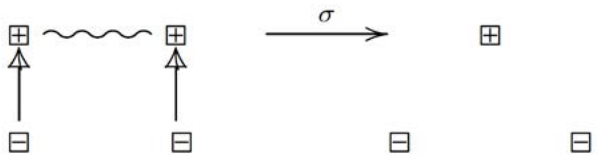
Conway 和 Joyal 认为 [57, 58]，**从**博弈A**到**博弈B的策略σ可以看作是复合博弈$A^{\perp} \| B$**中的**策略。给定另一个从博弈B到博弈C的策略τ，在共同博弈B上让两个策略相互对抗，并隐藏这种交互，就得到了策略**组合**$\tau \odot \sigma$。

并行博弈中的策略又是什么样的？ [55] 提供了详细的解答和说明。本质上讲，策略定义选择是让模仿策略成为是策略组合中的单位元（Identity）。并行博弈中的策略决定了玩家的行动及其行动的依赖关系，二者都应遵循博弈的限制，且对对手行为的限制不会超过博弈本身的限制。关于策略的公理需要联系第 5 节提到过的预层模型，通过预层模型建立起与 Scott 域的联系 [59]。一般来说，策略可能是非确定性的，但我们可以将范围限制为确定性策略，则所有的非确定性行为都来自于对手。

举例来说，考虑博弈$A^{\perp} \| A$中的**模仿**（Copycat）策略，这个策略名副其实，玩家原原本本地复制另一个分支中对手的行动。要构建这样的模仿策略，可以向$A^{\perp} \| A$添加额外的因果依赖关系，使得任意玩家行动都依赖于其对手在组合中另一半（也就是自己的拷贝）的行动。将这一模仿策略记作$\mathbb{C}_A$。如下所示，在简单博弈A中，玩家每一次的行动都依赖于一次对手的行动，表示如下：



实际上，要构建一个策略，有时候仅仅通过增加额外的因果依赖关系是不够的。通常来说，博弈A中的策略用事件结构$\sigma : S \to A$的映射来表示，描述事件结构S做出的玩家行动选择。例如，一个博弈中两个对手行动与一个玩家行动并行，玩家的策略是，如果对手行动就行动。这个例子表示如下：



这一策略将左边两个相冲突的玩家行动映射到右侧单个玩家行动，并将左边每次对手的行动映射到右边相应的对手行动。

*并非所有事件结构的映射$\sigma : S \to A$都是策略。映射还需要满足两个公理，才可以被视作策略：**Receptivity**（或称为接受性）和 **(Linear) Innocence**。通俗地说，Receptivity 和 (Linear) Innocence 使玩家对对手行为的限制不会超出博弈本身允许的范围。Receptivity 指的是对手在可到达位置可能发生的任意行动，在策略中应被标记为可能的行动。（也就是说，凡对手有动作，策略必须能够反应。）Innocence 则表示，策略只能增加新的形为$\boxminus \rightarrow \boxplus$的因果依赖关系，在这一因果依赖关系中，玩家除了从原始博弈中继承的行动之外，只能等待对手的行动。Silvain Rideau 和笔者指出，当上述两个公理都满足时，模仿策略可以认为是策略组合中的单位元 [55]。*

***从**博弈A**到**博弈B的策略是复合博弈$A^{\perp} \| B$**中**的策略。因此，策略可以表示为映射关系$\sigma : S \to A^{\perp} \| B$。给定另一个从博弈$B$到博弈$C$的策略$\tau : T \to B^{\perp} \| C$，用两个策略在博弈$B$上对抗，就得到了**组合**$\tau \odot \sigma$。为了更准确地实现这一点，可以进行与事件结构映射相关的两个操作：(1) **回调**，生成**交互**$\tau \circledast \sigma : T \circledast S \to A^{\perp} \| B \| C$，“同步”博弈$B$上与$S$和$T$匹配的行动；(2) 对事件结构映射进行**部分 – 完整分解**，隐藏同步，根据其定义部分生成策略**组合**$\tau \odot \sigma$，即$T \odot S \to A^{\perp} \| C$。*

*如果S中所有冲突都是源于对手行动（这来源于对于对手行动的因果依赖，此时如果对手行动之间存在冲突，则可能在S引入冲突），那么我们定义策略$\sigma : S \to A$是**确定性的**。策略的确定性是可组合的，换言之，确定性策略的组合仍然是确定性的。模仿策略是确定性的，所以对于**非竞争性的**博弈，模仿策略是一个确定性组合中的单位元。需要注意的是，上文提到的策略并非确定性策略。*

8.1 获胜条件

博弈A的获胜条件给出了其获胜构造的的一个子集W，满足W时，玩家获胜。而所谓的**获胜**策略，则是指无论对手如何行动（或根本不行动），玩家最终都能够获胜（到达某一个获胜构造）[60]。

*用形式化的方式描述，如果$\sigma : S \to A$对于S所有满足 +−maximal 的事件构造x来说，都有$\sigma x \in W$，那么就可以称σ为一个**获胜**策略。此处的 +−maximal 指的是玩家无法再在该构造中做出其他行动。这就相当于，σ针对对手应对策略的所有后手都能够让玩家获胜。*

对具有获胜条件W的博弈A，将玩家和对手角色互换，得到博弈$A^{\perp}$，其获胜条件为W的补集。该博弈为A的对偶博弈。在有获胜条件的简单并行博弈组合$A \| B$中，若玩家在

任何一个分支中获胜，则认为玩家在该博弈组合中获胜。有了这些拓展，可以认为当博弈A和博弈B均有获胜条件，从博弈A到B的获胜策略一定也是博弈$A^{\perp} \parallel B$中的获胜策略。这些选择确保了获胜策略的组合也是获胜策略。由于博弈不存在竞争，模仿策略肯定是一种获胜策略。

8.2 不完全信息

具有**不完全信息**的博弈中（玩家对对手信息没有充分的了解），某些行动会被屏蔽（或不可做出）。显然此时我们应当排除所有依赖于不可见行动的策略。通过与上一节类似的方法，并行博弈和并行策略可以自然地扩展到不完全信息博弈的领域 [61]。我们可以为博弈中的每一个行动都分配一个访问级别，全局中所有的访问级别是严格有序的。博弈中的行动或策略的因果只能依赖于相同级别或更低级别的行动。

*详细来说，假定有**级别**的固定先序$(\Lambda, \leq)$。Λ**博弈**中包括博弈A和**级别函数**$l : A \to \Lambda$，若$a \leq_A a'$，则$l(a) \leq l(a')$对A中所有行动a, a'都成立。Λ博弈中的Λ**策略**为策略$\sigma : S \to A$，若$s \leq_S s'$，则$l\sigma(s) \leq l\sigma(s')$对$S$中所有$s, s'$都成立。博弈组成对偶或并行博弈组合时，博弈中所有行动的访问级别都不会改变。从Λ博弈A到Λ博弈B的Λ策略，在博弈$A^{\perp} \parallel B$中也是Λ策略。因此Λ策略组合在一起也是Λ策略。*

8.3 新壶装旧酒

用树表示博弈，人们对相关概念更为熟悉。事件结构比树更复杂，但这并不妨碍我们将并行博弈策略与熟悉的树形博弈概念联系起来。事件结构中隐含着树的概念。如果两个事件相冲突，或者其中一个事件因果依赖于另一个，那么事件结构为**树形**结构。这种情况下，这个博弈中所有的事件结构可以构成一棵树（树的根节点是一个空的事件构造），其中的边代指事件结构的蕴含关系。

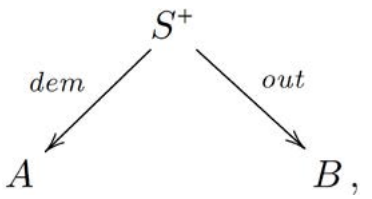
树形博弈是指底层事件结构为树的博弈。因为我们假设博弈中是没有竞争的，所以树形博弈的任何有限构造中，所有下一步行动（如果有的话）要么全部来自玩家，要么全部来自对手。对于前者，我们说的树形博弈处在玩家位置，后者自然处在对手位置。在属于玩家的位置上，确定性策略有两种选择，要么做出唯一行动，要么保持原地不动。和其他很多博弈的表示方法相比，并行策略不会强迫玩家做出行动（但可以通过合理的获胜条件鼓励玩家做出行动）。获胜条件是指让玩家获胜的构造。因此，在树形博弈中，这些条件可以是构造树中上的有限和无限分支。

很明显，树形博弈的对偶也是树形的。应对策略（同时也是对偶博弈中的策略）根据对手的构造选择对手的行动。当应对策略在所有对手构造中都是确定的，策略可以选择留在原地或采取行动。正如我们的预期，确定性策略构造树上一条有限或无限的分支，当存在获胜条件时，该分支可以使两个玩家的其中一个获胜。

树形博弈中，我们可以找到熟悉的概念。更令人惊喜的是，从并行博弈更丰富的（树形）结构中，我们可以找到熟悉的范式。这些范式原本与博弈也无关联；如果有关联，也只是某种程度的非正式关联。

例如，如果将范围缩小到并行博弈之间的确定性策略（博弈中的所有行动均为玩家行动），我们可以发现 Berry 的稳定域理论的身影，而 Girard 的定性域和相干空间是稳定域中的特例。对于这种受限博弈，一般的非确定性策略与对应**稳定生成空间**存在对应关系，而后者是一种在非确定性数据流及其组合中极为常见的语义模型。

*为了更详细地说明，以一个从纯玩家博弈A到另一个纯玩家博弈B的策略σ为例。该策略的映射表示为$\sigma : S \to A^{\perp} \parallel B$，具有 Receptivity 和 Innocence 性质。请注意，在$A^{\perp} \parallel B$中，所有对手行动都在$A^{\perp}$中，所有玩家行动都在B中。由 Receptivity 性质可以推定，A的任何事件构造都被复制并作为S中的初始输入；由 Innocence 性质可以推定，$A^{\perp}$和B中除了原有因果关系之外，所有可以引入S的新因果关系，都是存在于$A^{\perp}$的对手行动到B的玩家行动之间的。除了从两个博弈继承而来的因果依赖关系，S只能让B中玩家某个的行动因果依赖于$A^{\perp}$中行动的一个有限子集。因此，S中的任意玩家行动s同时与B中的输出事件$out(s)$和对输入的请求$dem(s)$（s发生所需要的A的有限构造）相关。纯玩家博弈之间的策略对应于**稳定生成空间**，其中需要用到事件结构S^{+}中的两个（特殊）函数。*



*这些函数是通过对S^{+}进行约束而得到的，具体来说，就是将S限制为玩家行动的集合。对于一般策略来说，特定输入的输出事件可能存在冲突——输出是非确定性的。当σ是确定性的时，所有冲突都继承自对手行动之间的冲突，则策略σ对应从A的构造域到B的构造域的**稳定函数**。*

由两个分支组成的并行博弈比起纯玩家博弈稍显复杂，该并行博弈中的一个分支为纯玩家博弈，另一个为纯对手博弈。这两个博弈之间的确定性策略可以产生 Abramsky–Jagadeesan 模型，适用于建立在稳定函数上的 GoI [39]。

将获胜条件和不完全信息等要素加入这些博弈中，让对手可以看到玩家的行动，但反之不行，我们可以在这样一种设置中找到确定性策略中的辩证范畴（Dialectica Category）[62]（即 Gödel 提出的“辩证解释”）。Valeria de Paiva 在剑桥大学攻读博士期间研究的是辩证

范畴，标志着“透镜”在函数式编程中的早期应用。这些概念当时被独立地发明出来，并用来实现数据结构上的可组合局部变化 [64, 65]。

将辩证博弈稍加泛化，就变成了“容器类型”的例子，确定性策略就变成了“依赖透镜”[66]。在通用张量积场景中，“透镜”泛化为“函数式引用”[67]。如果策略可以是非确定性的，那就可以找到基于稳定生成空间的函数式引用。

概率和量子程序语义 [68–71] 丰富了并行博弈和并行策略的内涵，内涵上的丰富仍然适用于这些特殊情况，例如为辩证博弈间的策略添加概率。在 [4] 中对此进行了详细的讨论，在 [3] 中有更简单易懂的介绍。

上述例子展示了函数式编程中对交互的处理方法，这些方法的发展相互独立。任何有关交互的组合理论都绕不开系统和系统所处环境的二分关系，并行博弈和并行策略通过事件的极化巧妙地处理了这种二分关系。从例子中可以看出，函数式方法处理二分关系需要利用输入输出之间的粗略区别，更加巧妙地通过函数及其参数来处理基本的交互。在并行博弈中，我们可以更清楚地看出是什么将不同的范式区分开来，又是什么将它们联系起来。

这些例子并未覆盖所有将函数拓展处理交互的方式。本文也没有涉及编程语言中的作用理论（Effect Theory）。作用理论用单子和代数理论完善了 Eugenio Moggi 的研究成果，用改进版的计算树来描述计算 [72, 73]。为了解决概率和量子计算问题，并行策略也进行了相应的改进。在并行博弈和并行策略联系效果理论方面，还需要进行更多的研究。

9 结语

本文探讨了域理论的局限性，也展示了其经久不衰的生命力。并行博弈和并行策略提供了一种交互的通用模型，这种通用性为构建或改进（域）模型需要的形式提供了指导。在某些特例中，并行博弈和并行策略可以简化为更简单的域模型。一方面，并行策略有助于构建域模型，另一方面，在已有域模型的情况下，可以极大地简化并行策略。如果明明有更简单的模型，却依然用复杂的模型，不是一种聪明的做法。据我们所知，在许多情况下，域理论提供的模型都是最简单的。

致谢

诚挚感谢各位匿名评审提出的宝贵建议。

参考文献

[1] Winskel, G.: The formal semantics of programming languages, an introduction. MIT Press. In Italian (UTET Libreria, 1999). In Chinese (China Machine Press, CITIC Publishing House, 2004). (1993)

[2] Abramsky, S., Jung, A.: Domain theory. Handbook of logic in computer science **3** (1994) 1–168

[3] Winskel, G.: Domain theory and interaction. Newsletter BCS-FACS FACTS Issue 2021-2 July (compressed) (2021)

[4] Winskel, G.: Making concurrency functional. CoRR **abs/2202.13910** (2022)

[5] Scott, D.S.: A type-theoretical alternative to ISWIM, CUCH, OWHY. Theor. Comput. Sci. **121**(1&2) (1993) 411–440

[6] Scott, D.S.: Mathematical concepts in programming language semantics. In: American Federation of Information Processing Societies: AFIPS Conference Proceedings: 1972 Spring Joint Computer Conference, Atlantic City, NJ, USA, May 16-18, 1972. Volume 40 of AFIPS Conference Proceedings., AFIPS (1972) 225–234

[7] Scott, D.S.: Data types as lattices. SIAM J. Comput. 5(3) (1976) 522–587

[8] Milner, R.: Implementation and applications of scott's logic for computable functions. In: Proceedings of ACM Conference on Proving Assertions About Programs, Las Cruces, New Mexico, USA, January 6-7, 1972, ACM (1972) 1–6

[9] Gordon, M.J.C., Milner, R., Morris, L., Newey, M.C., Wadsworth, C.P.: A metalanguage for interactive proof in LCF. In Aho, A.V., Zilles, S.N., Szymanski, T.G., eds.: Conference Record of the Fifth Annual ACM Symposium on Principles of Programming Languages, Tucson, Arizona, USA, January 1978, ACM Press (1978) 119–130

[10] Landin, P.J.: The next 700 programming languages. Commun. ACM 9(3) (1966) 157–166

[11] Manna, Z., Vuillemin, J.: Fix point approach to the theory of computation. Commun. ACM 15(7) (1972) 528–536

[12] Strachey, C.S., Wadsworth, C.P.: Continuations: A mathematical semantics for handling full jumps. High. Order Symb. Comput. 13(1/2) (2000) 135–152

[13] Plotkin, G.D.: A powerdomain construction. SIAM J. Comput. **5**(3) (1976) 452–487

[14] Kahn, G.: The semantics of a simple language for parallel programming. In Rosenfeld, J.L., ed.: Information Processing, Proceedings of the 6th IFIP Congress 1974, Stockholm, Sweden, August 5-10, 1974, North-Holland (1974) 471–475

[15] Brock, J.D., Ackerman, W.B.: Scenarios: A model of non-determinate computation. In Díaz, J., Ramos, I., eds.: Formalization of Programming Concepts, International Colloquium, Peniscola, Spain, April 19-25, 1981, Proceedings. Volume 107 of Lecture Notes in Computer Science., Springer (1981) 252–259

[16] Winskel, G.: Events, causality and symmetry. Comput. J. **54**(1) (2011) 42–57

[17] Plotkin, G.D.: LCF considered as a programming language. Theor. Comput. Sci. **5**(3) (1977) 223–255

[18] Kahn, G., Plotkin, G.D.: Concrete domains. Theor. Comput. Sci. **121**(1&2) (1993) 187–277

[19] Berry, G.: Stable models of typed lambda-calculi. In: ICALP. Volume 62 of Lecture Notes in Computer Science., Springer (1978) 72–89

[20] Girard, J.: The system F of variable types, fifteen years later. Theor. Comput. Sci. **45**(2) (1986) 159–192

[21] Girard, J.: Linear logic. Theor. Comput. Sci. **50** (1987) 1–102

[22] Berry, G., Curien, P.: Sequential algorithms on concrete data structures. Theor. Comput. Sci. **20** (1982) 265–321

[23] Curien, P.: On the symmetry of sequentiality. In Brookes, S.D., Main, M.G., Melton, A., Mislove, M.W., Schmidt, D.A., eds.: Mathematical Foundations of Programming Semantics, 9th International Conference, New Orleans, LA, USA, April 7-10, 1993, Proceedings. Volume 802 of Lecture Notes in Computer Science., Springer (1993) 29–71

[24] Rabinovich, A.M., Trakhtenbrot, B.A.: Communication among relations (extended abstract). In Paterson, M., ed.: Automata, Languages and Programming, 17th International Colloquium, ICALP90, Warwick University, England, UK, July 16-20, 1990, Proceedings. Volume 443 of Lecture Notes in Computer Science., Springer (1990) 294–307

[25] Russell, J.: Full abstraction for nondeterministic dataflow networks. In: 2013 IEEE 54th Annual Symposium on Foundations of Computer Science, Los Alamitos, CA, USA, IEEE Computer Society (nov 1989) 170–175

[26] Saunders-Evans, L., Winskel, G.: Event structure spans for nondeterministic dataflow. Electr. Notes Theor. Comput. Sci. 175(3): 109-129 (2007)

[27] Milner, R.: A Calculus of Communicating Systems. Volume 92 of Lecture Notes in Computer Science. Springer (1980)

[28] Brookes, S., Hoare, C.A.R., Roscoe, A.W.: A theory of communicating sequential processes. J. ACM **31** (1984) 560–599

[29] Nielsen, M., Plotkin, G., Winskel, G.: Petri nets, event structures and domains. TCS **13** (1981) 85–108

[30] Winskel, G.: Events in computation. (1980) PhD thesis, University of Edinburgh.

[31] Winskel, G.: Event structure semantics for CCS and related languages. In: ICALP'82. Volume 140 of LNCS., Springer, A full version is available from Winskel's Cambridge homepage (1982)

[32] Winskel, G.: Event structures. In: Advances in Petri Nets. Volume 255 of LNCS., Springer (1986) 325–392

[33] Winskel, G., Nielsen, M.: Models for Concurrency. In: Handbook of Logic in Computer Science 4. OUP (1995) 1–148

[34] Joyal, A., Nielsen, M., Winskel, G.: Bisimulation from open maps. Inf. Comput. **127**(2) (1996) 164–185

[35] Hyland, M.: Some reasons for generalising domain theory. Mathematical Structures in Computer Science **20**(2) (2010) 239–265

[36] Cattani, G.L., Winskel, G.: Profunctors, open maps and bisimulation. Mathematical Structures in Computer Science **15**(3) (2005) 553–614

[37] Girard, J.: The system F of variable types, fifteen years later. Theor. Comput. Sci. 45(2) (1986) 159–192

[38] Girard, J.: Towards a geometry of interaction. Contemporary Mathematics **92** (1989) 69–108

[39] Abramsky, S., Jagadeesan, R.: New foundations for the geometry of interaction. Inf. Comput. **111**(1) (1994) 53–119

[40] Gonthier, G., Abadi, M., Lévy, J.J.: The geometry of optimal lambda reduction. In: POPL '92. (1992)

[41] Scott, D.S.: Domains for denotational semantics. In Nielsen, M., Schmidt, E.M., eds.: Automata, Languages and Programming, 9th Colloquium, Aarhus, Denmark, July 12-16, 1982, Proceedings. Volume 140 of Lecture Notes in Computer Science., Springer (1982) 577–613

[42] Winskel, G., Larsen, K.G.: Using information systems to solve recursive domain equations effectively. In Kahn, G., MacQueen, D.B., Plotkin, G.D., eds.: Semantics of Data Types, International Symposium, Sophia-Antipolis, France, June 27-29, 1984, Proceedings. Volume 173 of Lecture Notes in Computer Science., Springer (1984) 109–129

[43] Abramsky, S.: Domain theory in logical form. In: Proceedings of the Symposium on Logic in Computer Science (LICS '87), Ithaca, New York, USA, June 22-25, 1987, IEEE Computer Society (1987) 47–53

[44] Plotkin, G.D., Winskel, G.: Bistructures, bidomains and linear logic. In Abiteboul, S., Shamir, E., eds.: Automata, Languages and Programming, 21st International Colloquium, ICALP94, Jerusalem, Israel, July 11-14, 1994, Proceedings. Volume 820 of Lecture Notes in Computer Science., Springer (1994) 352–363

[45] Ehrhard, T.: Hypercoherences: A strongly stable model of linear logic. Math. Struct. Comput. Sci. **3**(4) (1993) 365–385

[46] O'Hearn, P.W., Riecke, J.G.: Kripke logical relations and PCF. Inf. Comput. **120**(1) (1995) 107–116

[47] Loader, R.: Finitary PCF is not decidable. Theor. Comput. Sci. **266**(1-2) (2001) 341–364

[48] Abramsky, S., Jagadeesan, R., Malacaria, P.: Full abstraction for PCF. Inf. Comput. 163(2): 409-470 (2000)

[49] Hyland, J.M.E., Ong, C.H.L.: On full abstraction for PCF: I, II, and III. Inf. Comput. 163(2): 285-408 (2000)

[50] Abramsky, S., Melliès, P.A.: Concurrent games and full completeness. In: LICS '99, IEEE Computer Society (1999)

[51] Laird, J.: Game semantics for higher-order concurrency. In Arun-Kumar, S., Garg, N., eds.: FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings. Volume 4337 of Lecture Notes in Computer Science., Springer (2006) 417–428

[52] Melliès, P.A., Mimram, S.: Asynchronous games : innocence without alternation. In: CONCUR '07. Volume 4703 of LNCS., Springer (2007)

[53] Ghica, D.R., Murawski, A.S.: Angelic semantics of fine-grained concurrency. In: FOSSACS'04, LNCS 2987, Springer (2004)

[54] Faggian, C., Piccolo, M.: Partial orders, event structures and linear strategies. In: TLCA '09. Volume 5608 of LNCS., Springer (2009)

[55] Rideau, S., Winskel, G.: Concurrent strategies. In: LICS 2011. (2011)

[56] Winskel, G.: ECSYM Notes: ECSYM notes: event structures, stable families and concurrent games. https://www.cl.cam.ac.uk/~gw104/ecsym-notes.pdf (2016)

[57] Conway, J.: On numbers and games. Wellesley, MA: A K Peters (2000)

[58] Joyal, A.: Remarques sur la théorie des jeux à deux personnes. Gazette des sciences mathématiques du Québec, 1(4) (1997)

[59] Winskel, G.: Strategies as profunctors. In: FOSSACS 2013. Lecture Notes in Computer Science, Springer (2013)

[60] Clairambault, P., Gutierrez, J., Winskel, G.: The winning ways of concurrent games. In: LICS 2012: 235-244. (2012)

[61] Winskel, G.: Winning, losing and drawing in concurrent games with perfect or imperfect information. In: Festschrift for Dexter Kozen. Volume 7230 of LNCS., Springer (2012)

[62] de Paiva, V.: The Dialectica categories. PhD Thesis, University of Cambridge (1988)

[63] Avigad, J., Feferman, S.: Gödel's functional ("dialectica") interpretation. (1999) 337–405

[64] Oles, F.J.: A category theoretic approach to the semantics of programming languages. PhD Thesis, University of Syracuse (1982)

[65] Foster, J.N., Greenwald, M.B., Moore, J.T., Pierce, B.C., Schmitt, A.: Combinators for bidirectional tree transformations: A linguistic approach to the view-update problem. ACM Trans. Program. Lang. Syst. **29**(3) (2007) 17

[66] Abbott, M.G., Altenkirch, T., Ghani, N.: Containers: Constructing strictly positive types. Theor. Comput. Sci. **342**(1) (2005) 3–27

[67] Pickering, M., Gibbons, J., Wu, N.: Profunctor optics: Modular data accessors. Art Sci. Eng. Program. **1**(2) (2017) 7

[68] Winskel, G.: Distributed probabilistic and quantum strategies. Electr. Notes Theor. Comput. Sci. 298: 403-425 (2013)

[69] Paquet, H., Winskel, G.: Continuous probability distributions in concurrent games. In Staton, S., ed.: Proceedings of the Thirty-Fourth Conference on the Mathematical Foundations of Programming Semantics, MFPS 2018, Dalhousie University, Halifax, Canada, June 6-9, 2018. Volume 341 of Electronic Notes in Theoretical Computer Science., Elsevier (2018) 321–344

[70] Clairambault, P., de Visme, M., Winskel, G.: Game semantics for quantum programming. Proc. ACM Program. Lang. **3**(POPL) (2019) 32:1–32:29

[71] Clairambault, P., de Visme, M.: Full abstraction for the quantum lambdacalculus. Proc. ACM Program. Lang. **4**(POPL) (2020) 63:1–63:28

[72] Moggi, E.: Computational lambda-calculus and monads. In: Proceedings of the Fourth Annual Symposium on Logic in Computer Science (LICS '89), Pacific Grove, California, USA, June 5-8, 1989, IEEE Computer Society (1989) 14–23

[73] Plotkin, G.D., Power, A.J.: Computational effects and operations: An overview. Electron. Notes Theor. Comput. Sci. **73** (2004) 149–163

开源策略的制定与社区构建的度量

侯培新，李自，王晔晖
开源管理中心

摘要

随着数字产业的发展，越来越多的企业拥抱开源，并将开源纳入企业的发展战略。本文从企业开源策略的制定、企业级开源社区的构建等方面阐述企业如何利用开源发展生态经济、推动行业数字化转型发展，并提出面向社区开发者和管理者的社区度量方法，支撑企业构建高成长性、高粘性的开源社区，实现企业的开源战略。

关键词

开源策略，开源社区，社区度量

1 引言

在早期形态的软件中，源码公开并传播已广泛存在于学术科研机构之间，甚至供应商、客户上下游交付也经常用源码方式进行，但当时这种方式还不能称为“开源软件”，因为它缺乏诸如开源许可证、开源社区、协作贡献等一系列开源软件的基本要素。

在“开源软件”这个名字出现之前，自由软件已开始大行其道，并演化成为一场运动，形成基本原则（自由四要素），一些相关组织、团体以及明星项目也相继涌现，如自由软件基金会（Free Software Foundation，FSF）及 GNU（GNU's Not Unix!）。后来为了进一步获得商业组织的支持并更加聚焦其在商业世界中的实用价值，“开源软件”的名字被正式提出[1]。在此，我们无意深挖“开源软件”与“自由软件”背后的理念源头与政治站队[2]，目前绝大数人在大部分场合中提及的“开源软件”其实是这二者的合集。

开源软件已无所不在，业界对开源软件的认识也不断加深，并且呈现出螺旋式上升的趋势。开源的主要参与者也从之前的个人为主逐渐转变为企业为主，从而真正形成了项目产品利润项目（Project->Product->Profit->Project）这样的商业闭环，并不断推动开源的可持续发展。当今企业拥抱开源已经不单纯是为了降低成本、快速开发新产品——尽管这还是很多企业的主要动机（图 1）——近年来，随着生态经济、行业数字化转型、软件定义一切的发展，越来越多的企业希望通过主动开源来加速上述目标的实现（图 2）。

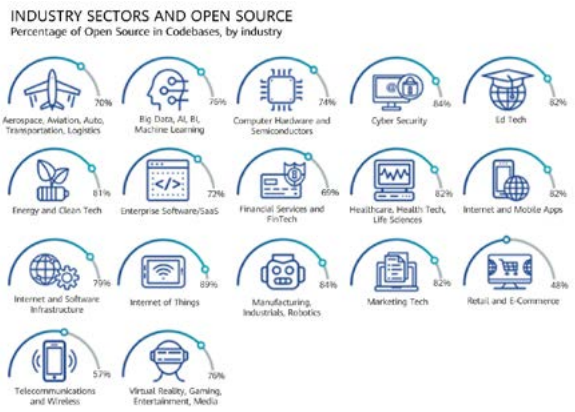


图 1 OSSRA 各行业使用开源占比

图 2 垂直行业主动开源占比

本文从企业开源策略的制定、企业级开源社区的构建等方面阐述企业如何利用开源来发展生态经济、推动行业数字化转型，并提出面向社区开发者和管理者的社区度量方法，以支撑企业构建高成长性、高粘性的开源社区，实现其开源战略。

2 开源策略的制定

一般情况下，开源策略的制定需要考虑如下六个要素[3]：商业模式、知识产权策略、开源技术策略、项目治理与社区平台策略、生态构建策略、成功度衡量。在这六个维度中，前五个将是本节阐述的主要内容，而最后一个维度则在第 3 节“社区构建的度量”中着重阐述。

2.1 要素一：商业模式

开源要支撑怎样的商业模式，如何进行商业变现，这是支撑其他要素的基础，因此需要首先确定下来。企业开源不可能是做慈善，没有商业变现的支撑，就无法保证持续的投入与发展。

业界的开源通常支持以下几种商业模式：

- 软件订阅与支持：售卖围绕开源代码的订阅服务并提供商业支持，如红帽企业 Linux（Red Hat Enterprise Linux，RHEL）、Pivotal Cloud Foundry；
- 特性增强并收费：开源基础版本，但对专有代码的附加组件进行收费，如 GitLab Enterprise 以及大量设备商的产品，包括华为的高斯 DB 和基于 OpenStack、K8S 等开源软件构建的商业软件。
- 关联产品（引流模式）：通过开源软件增加用户触达，并为其他业务导流，例如，安卓开源项目（Android Open Source Project，AOSP）及 Mozilla Firefox 项目为 Google 搜索等业务提供导流。
- 软件双许可：一般同一软件会有一个免费许可，具备传染性的 copyleft 开源许可，如 GNU GPL（General Public License），用于吸引广大用户及共建者；另一个则是私有软件许可，收费，但用户可以不受开源许可证（比如 GNU GPL）的限制，并享受商业公司提供的 SLA 保障与服务，如 Linux 桌面系统 Qt。
- （云）服务：以云服务的方式售卖开源软件，如 Amazon OpenSearch Service、Google Kubernetes Engine 等；
- 硬件：售卖硬件产品，但将硬件相关软件开源，以吸引并方便开发者使用，例如 Intel Clear Linux 开源项目对 Intel x86 CPU 的支持与使能。

值得注意的是，开源项目不等于商业产品，哪怕产品的软件源码与开源项目完全一致。商业产品基于开源项目构筑商业模式时需做如下考虑：

- 相对于开源项目，产品要有足够的商业价值；
- 相对于开源项目，产品需要更明确的定位；
- 产品要和开源项目中的软件竞争；
- 不同的公司将使用相同的开源项目软件进行竞争。

这里以 RHEL 为例来说明以上几点。RHEL 的所有源码均来自于上游社区——RHEL 一直秉承着“上游优先”甚至“上游 Only”的原则，即使是自己开发的特性，也会先合入上游社区再拉取回 RHEL 产品。

第一，独特的商业价值。Linux 的服务器操作系统非常复杂，RHEL 通过订阅与支持为客户在使用、管理、配置、维护等方面提供企业级服务，以及丰富的南北向认证，这是开源项目无法提供的独特商业价值。

第二，明确的产品定位。RHEL 将企业所关注的操作系统稳定性、长期维护及支持与社区先进特性综合在一起，实现二者之间的均衡，这与上游开源项目本身主要关注新特性开发以及有限版本、较短时长的维护相比，产品定位更加明确，也更贴近企业用户需求。

第三，尽管 RHEL 相比上游社区项目具有独特的价值与定位，但难免还是会有用户直接从上游社区项目获取软件，自己集成、安装、维护。RHEL 的做法是与其任由这些用户自由发展，还不如基于 RHEL 版本构建（收购）CentOS 社区，去除所有 RHEL 的商标痕迹，并定期维护、更新 RHEL。用户不仅可以享受到与 RHEL 基本等同的产品质量与维护，而且不用支付任何费用。表面上看，CentOS 似乎与 RHEL 形成了竞争，但实际上它吸引并留存了 RHEL 的潜在客户，也防止了相当数量的用户流向自下载构建操作系统或流向竞争对手。

第四，与其他基于相同开源项目的产品竞争。在 Linux 商业发行版这个领域内，除了 RHEL，还有 SUSE Linux Enterprise、Ubuntu 等竞品，因此 RHEL 也需要应对来自于这些产品的挑战。这些产品依赖同样的开源项目，因此在特性上大同小异，但在长期竞合中逐渐形成了各自的客户群及生态系统。虽然 RHEL 一家独大，但也能长期共存。

2.2 要素二：知识产权策略

开源软件拥有完整的知识产权属性，因此一般意义上的公有领域（Public Domain）的智力成果不是开源软件，因为没有与之相关联的知识产权。

开源软件知识产权的制定涉及专业的法律问题，因此就如何制定知识产权策略，本文不会提供任何具体的指导。读者可以从如下 4 个维度进行思考，并以此为切入点向知识产权律师或法务专业人员提出诉求，以制定相关策略：

- 开源许可证

  主动对外开源的开源软件必须选定一个许可证（为适配不同的应用场景或与不同的开源软件集成，有时同一个开源软件也可能选择多个许可证）。一般而言，一旦软件首创者选定了某个开源许可证，后续其他贡献者贡献的代码也自然会采用选定的这个许可证。但有一种情况比较特殊，就是集成发行版。以 openEuler 为例，其作为操作系统的发行版，集成了数千款原生上游社区已有的开源软件。这些软件的许可证可能存在差异，因此，许可证是否兼容就决定了这些软件能否集成到一起。考虑到开发者后续可能还会引入新的开源软件以增强操作系统功能，许可证的合规策略一般还会逐单审视、定期刷新。一般集成发行版社区都会采取许可证黑白名单方式，如 Fedora、openEuler 社区都有类似机制（https://fedoraproject.org/wiki/Licensing:Main#Good_Licenses）。

  此外，开源许可证还存在一些其他挑战，包括：软件中残缺许可证的识别、许可证应用颗粒度的多样性（软件级、模块级、文件级）、许可证场景的多样性（同一许可证在用户态 / 内核态、静态链接 / 动态链接等不同场景下的权利和义务也不相同）、软件许可证变更等。这些都需要相关的工具系统（https://compliance.openeuler.org/）及法律专业人员的深度介入。

- 著作权

  开源软件不论是托管给开源基金会进行开放治理还是由公司控制治理，著作权所有权（ownership）一般归属于原创开发者或其雇佣者，不会随开源贡献而发生转移。开源只是在许可证或其他类似贡献者协议（CLA 或 DCO）的约束下，赋予使用者（包括开源基金会）非常广泛的著作权权利，比如使用、复制、分发、修改等，有时这些权利甚至是永久的、非排他的、不可召回的（详情可以参考各开源许可证文本及 OSI 对于开源软件的定义），但这并不意味着将著作权所有权转移给了其他人。尤其是当原创者或原创企业将开源软件捐赠给开源基金会进行开放治理的时候，这点往往会引起一些误解。这也是各大主流开源基金会所形成的实践（包括但不限于 Linux 基金会、Apache 基金会、OpenStack 基金会等）。当然，也有一些开源基金会要求转移著作权，FSF 就是如此（https://www.gnu.org/licenses/why-assign.en.html），但这更多是出于“软件自由”的理念以及维权便利性的考虑，而非业界（尤其是大企业）主动采用的开源著作权策略。这里要特别指出，即便是在 FSF，这种著作权的转移也不是强制的，只有当贡献者希望由 FSF 全权处理维权时才会转移著作权。近来，我们

也看到 FSF 旗下的一些项目已经取消了这一策略，例如 GCC 已不再要求转移著作权（https://news.slashdot.org/story/21/06/05/0246247/gcc-will-no-longer-require-copyrights-be-assigned-to-the-fsf）。

- 专利权
  - 有些许可证有明确的专利免费使用授权，要注意由于开源而丧失专利起诉权利的场景；
  - 对于我方重点布局的专利领域，贡献开源及参与开源需要谨慎。
  - 最彻底的专利隔离方式是成立独立公司（如高通、爱立信、微软），专职做开源贡献。
- 商标
  - 如果开源项目捐赠给了开源基金会，项目商标的所有权一般也会转移；
  - 项目商标与产品商标若存在冲突，在贡献给基金会前要提前准备好新的项目商标；
  - 项目商标的使用权，尤其在开放治理的场景下（托管给基金会），一般会对所有使用者一视同仁，无法保留专有权利。

2.3 要素三：开源技术策略

开源技术策略分两个维度：

- 技术卖点

  当前开源项目数不胜数，如何在众多项目中脱颖而出，关键在于技术竞争力及先进性。

  要解决真实世界的问题：

  - 从无到有（O3DE）：在开放 3D 引擎（Open 3D Engine，O3DE）引入之前，市场上主流的商业软件都是闭源的，有限的几款开源引擎要么是 2D 引擎，要么无法提供 AAA 级游戏主机画面质量。AWS 在收购 CryEngine 代码后重新进行改造、增强并开源，希望打破原有产业格局并快速牵引 O3DE 算力上云。
  - 从有到优（AOSP）：AOSP 出现之前，基于 Linux 的移动操作系统有几大痛点，包括但不限于：1）关键组件缺乏高质量的开源软件（如电话子系统 RIL、浏览器、Java 虚拟机等都需要从商业公司购买）；2）各开源组件版本及选型碎片化严重，无法形成统一的应用开发接口，上层应用开发无法归一化，也没有统一的应用市场进行应用分发与变现；3）没有对接新兴的云服务（如地图、搜索服务等）；4）不具备当时代表了下一代操控体验（电容触屏滑动体验）的 MMI（iPhone 已具备），或需要自己开发或购买第三方组件。而 AOSP 的推出解决了上述所有问题，因此迅速吸引了大量南北向开发者，生态快速成长。
  - 巨大愿景（ECOMP）：电信网络云化，见图 3。

图 3 电信网络云化的愿景

- 开什么、闭什么

  以 RHEL 为例，基于“上游优先”、“上游 Only”原则，RHEL 的代码完全来自上游社区，要如何构筑差异化竞争力呢?

  因此，RHEL 对如下内容不做开源：

  - 构建工具链，包括编译选项（形成竞争力 1：RHEL 商业版本的二进制优化能力）；
  - 打造测试集及相关工具（形成竞争力 2：RHEL 商业版本的缺陷解决效率）。

2.4 要素四：项目治理与社区平台策略

- 项目治理：分为开放治理、封闭治理两种方式，图 4 中横轴代表参与者更多是个人导向还是公司导向，以及社区是否接受来自所属公司以外的开发者，与是否由公司控制并无耦合关系 [3]。
- 代码工程与社区运营平台：包括代码托管及相关的软件工程、运营平台工具（DevOps 流水线、CICD、贡献检查及门禁、开发者门户、官网等），见图 5。

图 4 开放治理模型

图 5 代码工程与社区运营平台

有些工具可能存在业务连续性风险，常见的迁移策略包括：

- 商业工具 / 服务逐步切换成可信供应商；
- 开源组件转国内商业服务；
- 风险开源组件逐渐形成独立维护与演进。

2.5 要素五：生态构建策略

作为社区的主导者，我们需要引导不同类型的生态伙伴发现参与社区的价值，这里针对 4 种常见类型的生态伙伴分别给出对应的策略 [3]：

- 用户：
  - 通过增加采用量来发展用户生态系统；
  - 针对以用户为中心的项目，创建一个完整的发行版；
  - 专注于推动下载，集成到其他开源生态系统中去。
- 开发者：
  - 通常情况下，最好是将开源项目贡献给开源基金会，以实现开放治理，吸引更多开发者参与进来。
  - 专注于框架、平台或工具，而不是完整的发行版，这样有助于提高开发者的生产力；
  - 专注于与其他开源开发的轻松整合。
- 其他公司（含友商）：
  - 选择现有的基金会，减少他人加入所带来的法务工作。
  - 先以项目开发为主，而后建立社区适配，例如：Kubernetes -> CNCF，OpenStack -> OpenStack 基金会，Ceph -> Ceph 基金会，Apache 孵化器 -> Apache 项目。
  - 通过优秀的观众渠道启动项目，例如，Kubernetes 是在 2014 年的 DockerCon 大会上启动的，而 OpenStack 则是在 2010 年的 OSCON 开源大会上启动的。

- 在组建前先宣布组建意向，以招募成员。例如，Linux 基金会曾于 2015 年宣布其组建 Hyperledger 的意向，但在次年才正式启动组建。
- 与供应商一起招募终端用户，确保项目能提供价值并得到推广。例如，Hyperledger 项目就是 与 CME Group、Deutsche Börse、State Street、Wells Fargo 等联手推出的。
- 预先与其他公司沟通并就架构主导与看护达成一致（图 6 以 Cisco OpenDayLight 项目为例）。

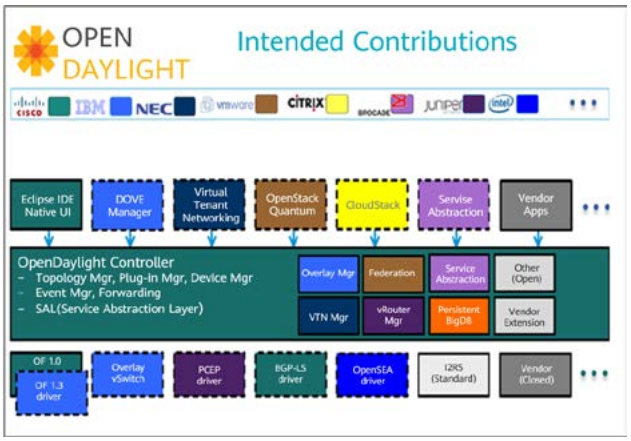


图 6 OpenDayLight 项目的贡献主体示意图

- 标准与产业组织
  - 开源的参考实现（EFSP (JCP) -> Jakarta EE）
  - 开源引领标准化（Docker -> OCI）

2.6 要素六：成功度衡量

成功度衡量可以从最终商业结果与社区发展过程这两个维度进行。商业结果度量维度相对简单，也是闭环开源策略制定与执行效果的终极度量，但往往需要较长时间才能得出结果，不利于掌握过程并及时进行调整。围绕社区发展过程这个维度，本节主要给出一些常用的中间态结果度量，并指出其对后续商业决策的潜在影响：

- 用户数度量（网站 portal 或用户调查）：
  - 用户数或者下载量
  - 社区大小
  - 用户数或者公司数（公司 X 下载了我们项目 200 次，所以我们可以尝试将产品卖给它）
  - 用户的地理位置（中国区的用户下载量占 30%，所以我们应该在中国区开设一个销售办事处）
- 其他生态活动度量（基金会度量平台或社区洞察工具）：
  - 成员
  - 活动
  - 贡献者
  - 培训
  - 开发者活跃度
  - 下游项目

3 社区构建的度量

3.1 社区度量的两个角度

“社区”指的是一群因共同利益而聚集在一起的人或组织。从根本上来说，创建一个成功的社区就是创建一个生态系统，成员们可以在社区中从事有意义的工作，快速提升自我，保持持续成长的动力，这样的社区才能长盛不衰。

在确定了项目开源策略之后，就可以对外发布并建立社区了，最终目标是建立一个高成长性、高粘性的可持续发展的开源社区。当然在社区构建和运营过程中，会面临许多问题，例如社区 Issue 长时间无人答复、开发者满意度差、开发者流失等。要解决这些问题，就必须构建一套完善的社区度量体系。

社区度量的意义在于衡量社区是否达成了既定的阶段性目标。具体说来，需要制定有效的量化方案，根据度量标准设计合理的预期，及时洞察度量结果以制定改进方案，确保社区健康的发展。

一个社区是否成功，可以从两个角度来衡量：一个是开发者角度，一个是社区管理者角度。

3.2 面向开发者的度量

社区的核心力量是社区成员，而开发者又是社区成员中最关键的角色，高效的开发者是社区成功的关键因素之一。

为了建立面向开发者的度量，我们先简要阐述构建成员画像的几个基本原则 [6]：

- 根据社区的愿景使命及阶段性目标，确定主要成员的画像类型，如用户画像、布道者画像、活动组织者画像、Issue 支持者画像、开发者画像等，有些画像的角色可能是重叠的。由于构建画像需要耗费大量人力，我们应该更关注画像类型的质量而不是数量。
- 在构建成员画像时，需要重点关注以下几个要素：能力、经验、动机、关心的事情、期待的奖励以及关注的领域。

下面我们将详细介绍构建开发者画像的过程。

3.2.1 构建开发者画像

开发者画像是用户画像技术在社区开发场景中的一种实践。开发者画像通常由社区开发者的多项特征以及与其他开发者的关联关系构成：

- 开发者特征通常是一系列的标签。如图 7 所示，我们可以通过标签来代表一位开发者在社区中所展现出来的技能特征。

图 7 开发者技能云

另外，我们还可以构建开发者的贡献度特征。“贡献度”是指开发者在特定社区中对该社区作出的贡献，主要考察开发者在社区中的活跃度和熟练度。这些贡献覆盖了开发者的各项活动，如在 Gitee 开源社区中参与项目或提交代码，在 Stack Overflow 问答社区中回答问题，等等。如图 8 所示，我们可以构建开发者参与社区的贡献度特征，图中的标签代表该开发者参与的所有项目，圆形直径代表开发者对项目的贡献度，直径越长，贡献度越高。

图 8 一名开发者参与贡献的所有项目标签集合

- 与其他开发者的关联关系可以用开发者关系图谱来表示，通过将开发者之间的交互可视化，来量化社区的协作强度。利用开发者关系图谱，我们可以识别开发者参与社区的趋势——特别是核心开发者——还可以判断开发者是否有从社区流失的倾向等。如图 9 所示，开发者关系图谱通常可以分为社交关系、直接协作关系和间接协作关系 [4]。

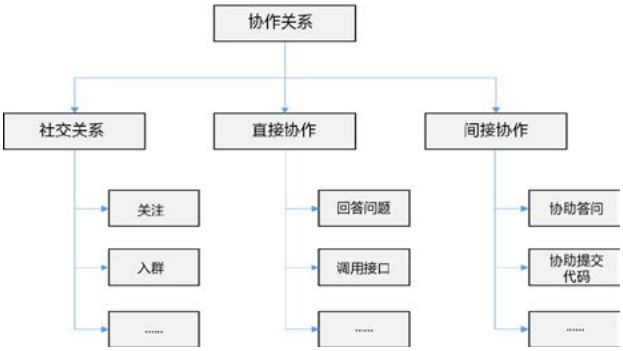


图 9 开发者关系图谱

每种关系下又包含了详细的协作关系，关系类型和关系强度将视为协作关系的属性。在开发者的开发活动中，“社交关系”是指在特定社区中开发者之间的人际关系。虽然与开发活动并不直接相关，但研究发现，开发者加入好友所在的项目，其贡献率将会大幅提升 [5]。社交关系又可以细分为关注关系和同组织关系，分别表示开发者之间的相互关注和开发者隶属于同一个组织。“直接协作”是指开发者之间存在直接的交互关系，即两个开发者面对同一任务需要紧密的沟通合作。例如开发者 D1 与开发者 D2 之间若存在 Answer to 和 Call API 的关系，则表示 D1 回答了 D2 提出的问题、在实现某些功能时 D1 调用了 D2 提供的接口，这些协作的频度用数字表示。此外，开发者共同修改代码文件、对同一代码的开发和测试也属于这种紧密型的协作关系，其他类似的协作关系还有很多，恕不一一列举。最后，“间接协作”是指开发者之间存在间接的交互关系。相对于直接协而言，间接协作是一种较弱的协作关系，但对于整体的开发任务也会起到一定的作用。例如开发者 D1 与开发者 D2 都回答了某个问题，或者都向某个开源项目提交了自己的代码，则他们之间就构成了 Co-answer 和 Co-commit 的关系。

3.2.2 建立开发者社区的参与旅程框架

在完成开发者画像的构建之后，接下来需要建立开发者在社区的参与旅程框架（如图 10 所示），方便度量开发者在不同阶段为社区创造的价值 [6]。

该框架涵盖了社区参与的三个关键部分。不同的成员画像类型，需要不同的参与旅程框架：

- 新人引导：五角星左边代表社区帮助新人以最简单的方式尽快创造价值，价值本身既是新人自己的，也是社区的。

图 10 开发者社区的参与旅程框架

- 参与状态转换：五角星右边代表开发者开始在访客 – 常客 – 核心成员这三个关键身份之间转换。
- 激励：为了促进社区发展、保持成员的参与度，我们会采取一系列措施（蓝色方块），帮助成员积累经验、培养新技能并维持积极性。

在介绍完开发者画像及社区参与旅程框架后，下面我们可以引出社区画像成熟度模型（如图 11 所示），该模型提供了一套工具来定义开发者画像在各阶段的成功：

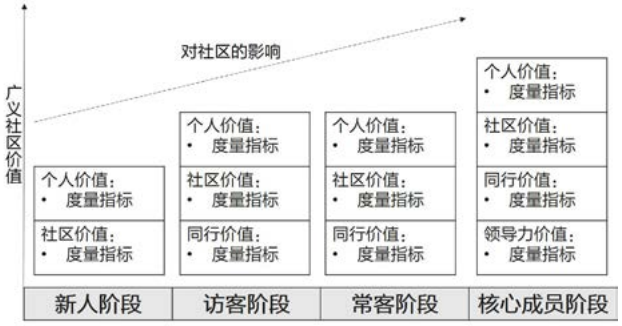


图 11 社区画像成熟度模型

使用方法如下：沿着横轴，可以看到社区参与旅程框架的各个阶段，包括新人、访客、常客和核心成员阶段。针对这些阶段，分别定义各个阶段的成功标准，并利用不同维度的指标来进行度量。

- 个人价值：某成员画像应该为他 / 她本人带来哪些可度量的价值？例如，对于新人阶段，Issue 支持者画像的成功标准可以定义为“在社区提出新 Issue 并获得答案”。
- 社区价值：某成员画像为社区带来了哪些可度量的价值？例如，对于新人阶段，Issue 支持者画像的成功标准可以定义为“解决了社区其他成员提出的 Issue”。
- 同行价值：某成员画像为社区其他成员带来了哪些可度量的价值？例如，对于常客阶段，成功标准可以是“为社区不同成员提供一对一指导或者培训”。
- 领导力价值：某成员画像在社区领导力方面带来了哪些可度量的价值？

3.2.3 构建开发者 NPS 度量模型

当我们完成了开发者画像的构建，并设计了开发者在整个社区的参与旅程框架，要衡量他们在不同阶段（新人 / 访客 / 常客 / 核心成员）为社区创造的价值，就需要从开发者角度出发构建净推荐度（Net Promoter Score，NPS）度量模型 [7]——通过度量和洞察，让社区与开发者为彼此带来更多价值。NPS 是业界用来衡量用户体验的通用指标，是管理用户体验、提升客户满意度的有效工具。通过一个终极问题——你有多大可能会将我们的社区推荐给同事和亲朋好友？为什么？——NPS 可以清晰地将用户分成三类，并在此基础上计算出一个简单易懂的值，然后制定出计划，以期改进并获得成功（如图 12 所示）。

图 12 NPS 定义

开发者 NPS 度量模型（图 13）由两部分组成：第一部分是 NPS 度量体系，第二部分是 NPS 反馈体系。通过 NPS 度量体系，结合开发者画像，我们可以通过 NPS 问卷调查的形式感知社区存在的问题以及开发者的诉求。通过 NPS 反馈体系，及时给出解决方案，调整社区的相关治理机制。最终提升开发者满意度，以及开发者对社区的忠诚度。

图 13 开发者 NPS 度量模型

3.3 面向社区管理者的度量

作为社区的管理者，开源社区有许多度量指标及度量模型可以衡量社区是否成功。但问题是开源项目面临着浩瀚

如烟的数据，任何可以获得数据的对象，都能收集、跟踪。每个组织跟踪的度量标准——以及他们如何处理数据——在很大程度上取决于自身社区独有的战略目标，以及他们在市场和开源社区中的独特挑战。特别是对于大型组织来说，有太多的项目，不可能跟踪所有的对象并保证此举有意义。为摆脱这一困境，在选择度量指标和度量领域时，需要结合社区的战略目标，投入自己的思考 [8]。

3.3.1 选择度量指标

设定社区的战略目标包括：构建社区的远景使命、明确社区为人 / 组织提供的价值、制定阶段性目标。具体执行过程中，我们需要有针对性地衡量社区的工作成果。

在制定具体指标时，我们需要保证目标能够清晰、准确地衡量，切忌模棱两可。为此，我们需要遵循以下四个规则 [6]：

- 衡量关键维度

  社区主要有七个关键维度：

  - 增长率：有多少新人加入社区？如何随时间变化？
  - 留存率：有多少人在持续地参与社区活动？
  - 社区内成员的活跃度；
  - 组织成员与社区成员的协作活跃度；
  - 交付能力：社区内不同的贡献者是否按照其职责提供了相应的价值；
  - 出席率：线上、线下活动的出勤情况；
  - 高效性：整个社区的运转机制是否畅通无阻？

- 同时度量“质”与“量”

  跟踪不同类型的贡献数量是衡量活跃度和交付率的好办法，但同时也要验证贡献的质量，做到“质”与“量”兼顾。

- 度量可量化

  在度量目标是否达成时，我们应能清晰地回答出“是”或“否”，而不是“也许”。

- 限制度量指标的数量

  我们需要做到精简指标数量，只跟踪关键指标，这样可以保持关注，简化讨论。

3.3.2 核心组织的支持

作为社区的初创者及主导者，核心组织想要真正成功地创建一个社区，需具备从计划、发展、互动到优化社区体验的一整套技能。不仅需要发展这些技能，也需要在组织内部建立支持系统。

组织发展这些技能的过程可分为三个阶段 [6]：

1. 孵化新技能：营造一个兼具资源、教育、战略和执行力的环境，该环境可以引入新技能、为人员提供培训、在人们应用这些技能时提供支持。
2. 建立机制：在吸取孵化阶段的经验教训后，需要制定机制、加深团队理解、完善和发展这些标准。
3. 整合：需要将这些机制推广到组织内所有团队中去。

我们将这三个阶段与专业领域的目标技能相结合，构建出了如下组织能力成熟度模型：

组织能力 →

	1. 孵化阶段	2. 建立机制阶段	3. 整合
战略	配合完成战略计划、执行	定期的结构化计划周期	战略演化与执行
管理	明确社区管理风格	团队成长与辅导	组织全面支持
增长	制定初期增长战略	验证增长模式假设	识别增长模式并推动其继续发展
参与	社区团队/员工参与	其他关键部门参与	整个组织参与
领导	特定团队/员工的领导	组织和社区的利益相关者参与领导	形成独立领导团队
工具	可以使用社区工具	将社区工具整合到更广阔的组织资源中	社区工具成为产品/工程核心工作的关键部分
指标	跟踪目标成员画像的相关指标	定期制定改进与执行	社区指标成为组织KPI的重要组成部分

图 14 组织能力成熟度模型

如图 14 所示，组织能力成熟度模型从上到下分为七行，每行代表一个专业领域，每个领域皆跨越了技能发展的三个阶段。

在使用该模型时，组织需定期复盘，向相关部门收集反馈意见，并将改进计划整合到下一个迭代中去。

3.4 社区体验改进

基于社区健康度度量体系和开发者画像的关系，我们与华东师范大学王伟教授和上海交大曹健教授的团队一起联手，在华为主导的 openEuler/MindSpore 社区实施了社区体验改进。

3.4.1 开发者画像助力社区维系开发者

在 OpenEuler 社区中，如图 15 所示，我们通过构建开发者在不同 SIG 组中的技能标签，识别出大多数开发者都参与过 CVE 漏洞的修复工作。对于社区管理者而言，制定更简单的漏洞修复流程、引入自动化漏洞感知和修复工具可以大幅提升开发体验。

图 15 openEuler 开发者标签

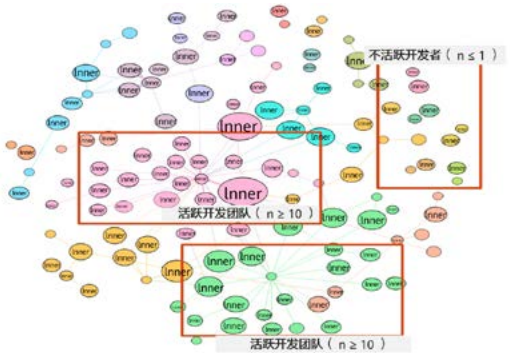


图 16 MindSpore 社区协作者关系图谱

在 MindSpore 社区中，我们构建了开发者 Issue 交互关系图谱，协助社区管理者识别了 10+ 不活跃开发者，并通过问卷、线下沟通等方式重新激活了部分存在流失倾向的开发者（如图 16 所示）。另外，通过构建开发者兴趣图谱，在 MindSpore 社区试点社区安全、前端、数据、科学计算 SIG 组，提升了社区的整体活跃度。

3.4.2 开发者 NPS 度量模型推动 openEuler 社区改进开发者体验

依据 3.2 节的讨论，现在给出开发者 NPS 度量模型在 openEuler 社区的实际应用。

如表 1 所示，我们详细定义了开发者通过引新活动（例如 meetup）成为社区新人之后可能会经历的所有社区旅程。在每个旅程节点，根据开发者角色（新人 / 访客 / 常客 / 核心成员）构建了不同的开发者画像。然后引入 NPS 调查问卷，对这些画像精准投放不同的问题，以获得开发者明确、清晰的反馈。通过这种方式，帮助社区及时发现问题，做针对性的管理与改进，最终提升开发者对社区的忠诚度。

4 结语

当前，越来越多的企业主动拥抱开源，并将开源纳入企业的发展战略。但“为什么开源”、“如何开源”仍是企业不得不面对的实际挑战。从开源策略六要素出发，本文从商业、项目、生态等多个角度全面审视了企业的开源活动，帮助企业通过社区度量掌握开源健康度，构建高成长性、高粘性的开源社区，并最终支撑企业真正实现其开源战略。

表 1 开发者旅程

开发者旅程	引新	使用	签署CLA	加入SIG	新建 Issue	Issue 响应	Issue 关闭	本地开发	新建 PR	CI	代码评审	PR 合入	版本构建	版本发布
画像构建	新人画像	用户画像	CLA签署者画像	SIG成员画像	Issue 建立者画像	Issue 响应者画像	Issue 关闭者画像	开发者画像			代码评审者画像	提交人画像	发布者画像	
画像角色	新人	核心成员 常客 访客	核心成员 常客 访客		核心成员 常客 访客			核心成员 常客 访客		核心成员 常客 访客	核心成员 常客 访客	核心成员 常客 访客	核心成员 常客 访客	
开发者关系图谱	新人 - 布道者 导师	用户 - 导师	CLA签署者 - 导师	SIG成员 - 维护人	Issue 建立者 - 维护人	Issue 响应者 - 维护人/ 贡献人	Issue 关闭者 - 维护人	开发者 - 维护人 Issue建立者		开发者 - Infra 责任人	代码评审者 - 开发者 提交人 维护人	提交人 - 提交人 维护人 Issue建立者	发布者 - 维护人 TC 委员会	
NPS 调查问卷	新人NPS	用户NPS	SIG NPS		Issue NPS			本地开发NPS		构建NPS	提交人 NPS		发布NPS	
	1.社区心智份额 2.引新渠道	1. 使用满意度	1.CLA签署满意度	1. SIG满意度	1.Issue创建文档满意度	1.Issue响应时长 2.Issue支持满意度	1.Issue闭环周期	1. 本地开发文档满意度 2. 示例代码 3. 开发环境搭建 4. 本地调式 5. 本地构建，测试，部署	1.PR创建文档满意度 2.PR与Issue关联率	1.构建效率 2.构建满意度	1.评审响应时长 2.评审满意度	1.PR闭环周期	1.发布信息满意度 2. 版本使用满意度	
NPS角色	推荐者 被动者 贬损者	推荐者 被动者 贬损者	推荐者 被动者 贬损者		推荐者 被动者 贬损者			推荐者 被动者 贬损者		推荐者 被动者 贬损者	推荐者 被动者 贬损者		推荐者 被动者 贬损者	
竞品NPS	开展第三方NPS调查，与竞品社区进行比较，识别相对NPS，找机会点，做针对性管理。													
度量指标	基于开发者旅程及NPS调查问卷建立度量指标。													
反馈机制	根据NPS调查问卷及及度量平台实时监控，支撑社区运营管理，提升开发者对社区忠诚度。													

参考文献

[1] Raymond and Eric S (1999), "The cathedral and the bazaar: Musings on Linux and open source by an accidental revolutionary," O'Reilly Media. ISBN 1-56592-724-9.

[2] Richard Stallman, "FLOSS and FOSS," Accessed 9 October 2022.

[3] Bryan Che, "OpenSource policy"

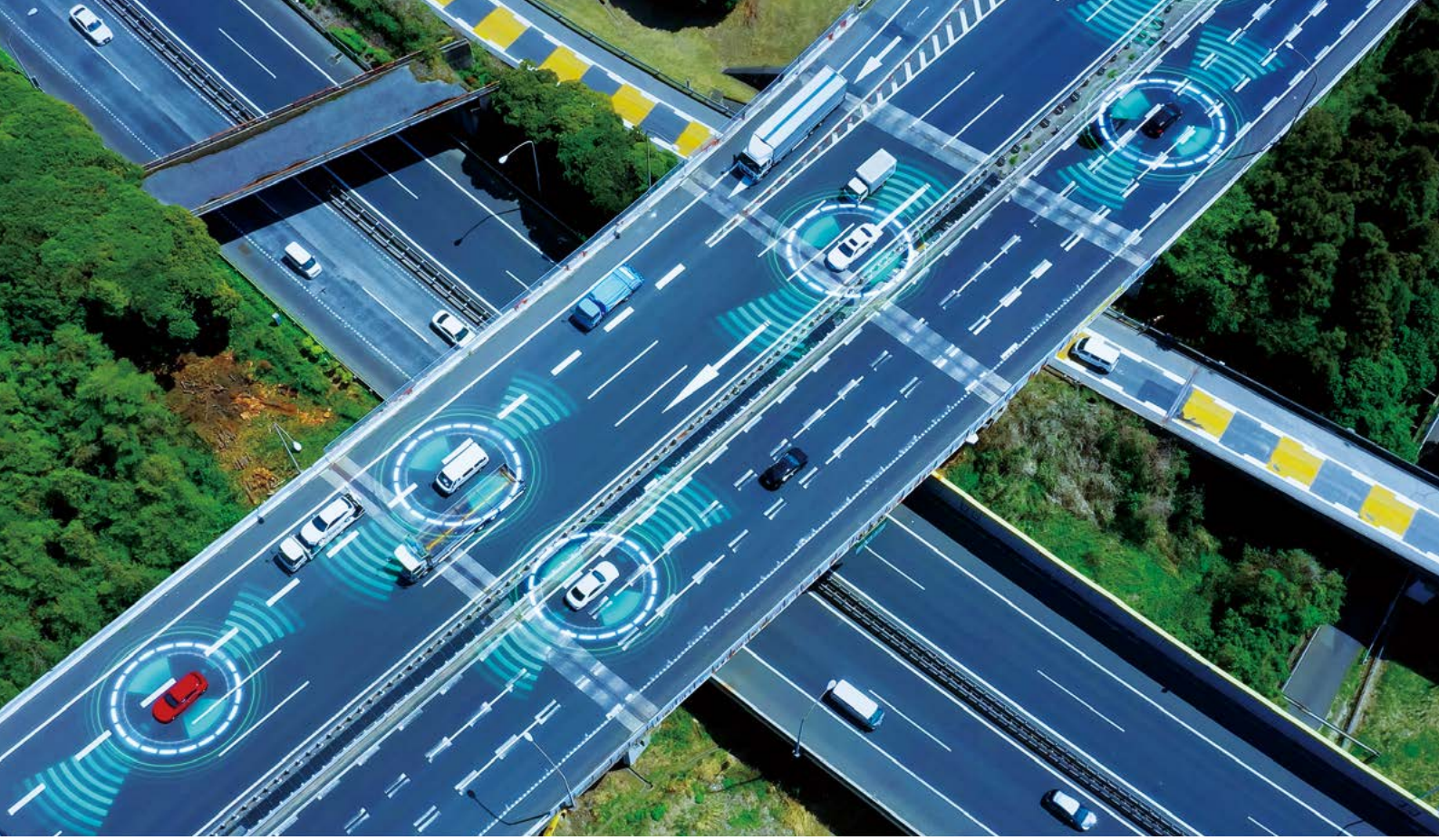
[4] 张建，孟祥鑫，孙海龙，王旭，刘旭东，" 数据驱动的软件开发者智能协作技术 ," 大数据 , 202101: 76–93, 2021.

[5] 杨程，范强，王涛，尹刚，王怀民，" 基于多维特征的开源项目个性化推荐方法 ," 软件学报 , 2017, 28(6):1357–1372, 2017

[6] Jono Bacon (2019), "People powered," HarperCollins Focus. ISBN 9781400214884.

[7] Fred Reichheld and Rob Markel, "The ultimate question 2.0 (revised and expanded edition): How net promoter companies thrive in a customer-driven world," 2011

[8] Christine Abernathy, Chris Aniszczyk, Joe Beda, Sarah Novotny, and Gil Yehuda, "Measuring your open source program's success," Accessed 9 October 2022.



智能汽车 LiDAR 辅助 GNSS-RTK 定位

Han Gao[1]，Weisong Wen[2]，Li-Ta Hsu[2]，Yongliang Wang[1]

[1] 黎曼实验室
[2] 香港理工大学航空及民航工程学系

摘要

全球导航卫星系统动态实时差分（Global Navigation Satellite System Real-Time Kinematic，GNSS-RTK）定位可以基于固定解实现高精度定位，是自动驾驶车辆（Autonomous Driving Vehicles，ADV）绝对定位的必备技术。但城市建筑物对信号的阻挡、反射和衍射会使 GNSS-RTK 定位的性能大幅下降。因此，本文提出一种新方法，利用 3D LiDAR 和惯性传感器生成局部环境映射，排除 GNSS 潜在非视距（Non-Line-of-Sight，NLOS）接收信号，进一步提升城市峡谷场景下 GNSS-RTK 定位的性能。首先，通过基于因子图优化的 LiDAR/ 惯性积分构建局部环境描述，即 3D 点云图（Point Cloud Map，PCM）。然后，在 GNSS-RTK 定位前采用 3D PCM 检测并排除 GNSS 潜在 NLOS 接收信号。最后采用优化后的 GNSS-RTK 定位校正 LiDAR/ 惯性积分构建的 3D PCM 所产生的漂移。实验利用低成本车载 GNSS 接收机在不利于定位的城市环境中采集测试数据集，验证了该方法的有效性。

关键词

GNSS，RTK，NLOS，LiDAR，城市峡谷

1 引言

全球导航卫星系统动态实时差分（Global Navigation Satellite System Real-Time Kinematic，GNSS-RTK）广泛应用于高精度航空测绘 [1] 以及 L4 等级的全自动驾驶车辆（Autonomous Driving Vehicle，ADV）定位 [2]。GNSS-RTK 定位通常包括两个步骤：（1）根据接收的 GNSS 测量数据估计浮点解。（2）基于估计的浮点解，采用最小二乘算法（如 LAMBDA [3]）计算整周模糊度，并作为初始猜测值。通过双差分载波和码测量可以在露天区域基于固定解实现厘米级定位。但在城市峡谷场景中，周边建筑物对 GNSS 信号的反射和阻挡会产生 NLOS 和多径接收信号，使 GNSS-RTK 定位的精度大幅下降。问题的根本原因是 GNSS NLOS 接收信号。因为部分 GNSS 接收信号被严重污染，产生大量噪声。根据以往研究 [4]，高度城市化地区中大部分 GNSS 接收信号都是多径信号或 NLOS 信号。因此，基于差分载波和码测量估计的浮点解，其精度有所下降，模糊度更难求解。

近几十年，[5-7] 开展了大量研究以提升城市峡谷场景下 GNSS-RTK 定位的性能。[6] 提出采用多天线提升 GNSS-RTK 离群值测量的鲁棒性，但该方法的有效性取决于 GNSS 污染信号量（多径信号或 NLOS 接收信号量）。最近，[5] 提出通过 3D 建筑模型来排除 GNSS 污染信号，以提升城市峡谷场景下 GNSS-RTK 定位的性能。该方法可以通过视距（Line-of-Sight，LOS）测量提高固定率，但需要精确的 3D 建筑模型和 GNSS 接收机位置的初始猜测值。另一个研究方向是通过传感器融合来增加传感器。GNSS-RTK 和惯性测量单元（Inertial Measurement Unit，IMU）的结合能够带来更多增益，因此已被广泛研究。根据 [8]，在信号较好的典型城市场景中，高级双频多 GNSS-RTK 定位在 1 小时车程内的正确整周模糊度固定率可达 76.7%。但总体定位性能很大程度上取决于 GNSS 中断期间 IMU 传感器的成本 [9]。[10] 和 [11] 提出了在不利于 GNSS 定位的环境中将 GNSS-RTK 与视觉测量紧密结合，以提升定位性能。但是，视觉测量对动态对象的照明条件和密度比较敏感，而且视觉测量的比例复原在很大程度上取决于 GNSS 测量数据的质量。[4，13-15] 没有采用视觉测量，而是提出通过 3D LiDAR 传感器不断提升城市峡谷场景下 GNSS 单点定位（Single Point Positioning，SPP）的性能，因为 3D LiDAR 传感器鲁棒性较高，且不受照明条件影响。该方法利用 3D LiDAR 传感器生成的 3D 点云来描述周边环境，进一步排除 [4] 或校正 GNSS NLOS 接收信号 [13]。[16] 的最新成果结合了 GNSS NLOS 接收信号的校正与重构，通过增量构建环境描述（3D PCM）提升城市场景下 GNSS SPP 定位的性能。但该方法只采用了码测量，没有对载波相位测量进行探讨。

本文基于 [4] 和 [13] 的 3D LiDAR 辅助 GNSS SPP 定位，提出通过 3D LiDAR 传感器从根本上解决信号反射和阻挡引起的定位精度下降问题，从而提升城市峡谷场景下 GNSS-RTK 定位的性能。首先，根据 [17] 的最新成果，通过因子图优化（Factor Graph Optimization，FGO）将 LiDAR 和 IMU 测量松散结合，实现 LiDAR/ 惯性里程计（LiDAR/Inertial Odometry，LIO），以估计两个历元间的相对运动，并生成 3D PCM（即局部环境描述）。再基于 [16]，利用 PCM 检测和排除 GNSS 潜在 NLOS 接收信号，从而提升 GNSS 测量数据的质量。然后利用未排除的 GNSS 卫星测量信号估计浮点解，并采用 LAMBDA 算法求解模糊度。最后，结合 GNSS-RTK 定位固定解的估计值和 LIO，进一步校正 3D 点云漂移。简而言之，该方法有效结合了 LIO 和 GNSS-RTK 定位。LIO 在短周期内局部精度高，并为 GNSS NLOS 信号检测提供了环境描述。GNSS-RTK 定位提供了无漂移的全局参考位置，但结果受 GNSS NLOS 接受信号的影响。本文的研究成果如下：

（1）通过 LiDAR/ 惯性积分检测和排除 GNSS 潜在 NLOS 接收信号，进一步提升 GNSS-RTK 定位的性能。这是 LiDAR/ 惯性积分被首次应用于排除此类信号。

（2）通过优化后的 GNSS-RTK 定位校正 3D 点云漂移，提高整体定位精度。

（3）利用低成本 GNSS 接收机采集的数据集来评估该方法的有效性。

以下是本文各部分简介。第 2 节提供方法概述。第 3 节介绍如何生成局部环境描述。第 4 节介绍 NLOS 信号检测和 GNSS-RTK 定位。第 5 节采用香港城市峡谷场景中采集的数据集进行实验，评估方法的有效性。第 6 节得出结论，并提出未来研究方向。

2 方法概述

方法概述如图 1 所示，由两部分组成：（1）基于 3D LiDAR 和 IMU 在云上生成实时环境描述，并校正 GNSS-RTK 解。（2）基于实时环境描述检测和排除 GNSS NLOS 信号，并基于未排除的卫星信号进行 GNSS-RTK 定位。本文用加粗大写字母表示矩阵，用加粗小写字母表示向量，用斜体小写字母表示可变标量，用小写字母表示常量标量。GNSS 接收机状态和卫星位置均采用东北天坐标系（East, North, Up，ENU）表示。

为使描述更加清晰，本文定义了以下几种符号：

- 给定历元 k 内卫星 s 接收的伪距测量值为 $\rho^{s}_{r,k}$。下标 r 表示 GNSS 接收机，k 表示时间索引，上标 s 表示卫星索引。
- 给定历元 k 内卫星 s 接收的载波相位测量值为 $\psi^{s}_{r,k}$。
- 地心地固坐标系（Earth-centered, Earth-fixed,

ECEF）或 ENU 坐标系中的变量用上标 G 和下标 L 表示。例如，在 ENU 坐标系和 ECEF 坐标系中，位姿和位置的转换为 $\mathbf{T}_L^G = [\mathbf{R}_L^G, \mathbf{t}_L^G]$，其中 $\mathbf{R}_L^G$ 和 $\mathbf{t}_L^G$ 分别表示旋转和平移。

- AHRS、LiDAR 和 GNSS 接收机的体坐标系用上标 BI、BL 和 BR 表示。例如，$\mathbf{P}_k^{BL}$ 表示历元 k 内的一个 3D LiDAR 点云坐标系。
- GNSS 接收机和 3D LiDAR 之间的外部参数为 $\mathbf{T}_{BL}^{BR} = [\mathbf{R}_{BL}^{BR}, \mathbf{t}_{BL}^{BR}]$。IMU 和 3D LiDAR 之间的外部参数为 $\mathbf{T}_{BL}^{BI} = [\mathbf{R}_{BL}^{BI}, \mathbf{t}_{BL}^{BI}]$。
- ECEF 坐标系中，卫星 s 在给定历元 k 的位置为 $\boldsymbol{\rho}_k^s = [\rho_{k,x}^s, \rho_{k,y}^s, \rho_{k,z}^s]^{\mathrm{T}}$。
- ECEF 坐标系和 ENU 坐标系中，GNSS 接收机在给定历元 k 的位置为 $\mathbf{p}_{r,k}^G = [p_{r,k,x}^G, p_{r,k,y}^G, p_{r,k,z}^G]^{\mathrm{T}}$ 和 $[p_{r,k,x}^L, p_{r,k,y}^L, p_{r,k,z}^L]^{\mathrm{T}}$。ENU 坐标系的旋转为 $\mathbf{R}_{r,k}^L = [\alpha_{r,k,x}^L, \beta_{r,k,y}^L, \gamma_{r,k,z}^L]^{\mathrm{T}}$。
- GNSS 接收机在给定历元k的钟差为 $\delta_{r,k}$，单位为米。$\delta_{r,k}^s$ 表示卫星钟差，单位为米。

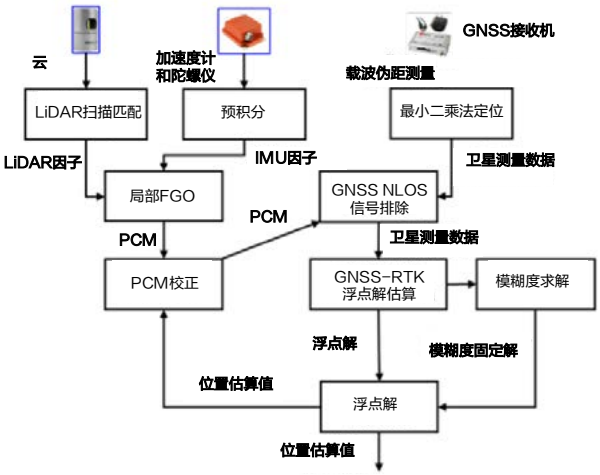


图 1 方法概述。输入为 IMU、3D LiDAR 和 GNSS 接收机的原始测量数据，输出为 GNSS 接收机的估计状态。

3 生成局部环境描述

本节介绍如何基于 LiDAR/ 惯性传感器生成局部环境描述，以及如何进行 GNSS-RTK 校正。首先，采用 LiDAR/ 惯性积分通过局部 FGO 生成 3D PCM（如图 1 所示）。然后利用 3D PCM 检测 GNSS NLOS 接收信号。但是 LiDAR/ 惯性积分会随着时间的推移产生漂移。3D PCM 漂移可以通过全局 FGO 结合 GNSS-RTK 和 LIO，利用优化后的 GNSS-RTK 定位结果进行校正（如图 1 所示）。PCM 采用 ENU 坐标系表示，ENU 坐标系和原始 LiDAR 坐标系之间的外部参数通过前几个 GNSS-RTK 固定解进行校正 [18]。

3.1 基于局部 FGO 的 LiDAR/ 惯性积分

LiDAR/ 惯性积分是 [17] 最近发表的研究（该文献探讨了基于滑窗的 FGO 松耦合积分），并非本文的主要研究成果。但为了论述的完整性，本节提供了 LiDAR/ 惯性积分的简介。世界坐标系中第 k 个 IMU 状态，即捕捉到 LiDAR 点云第 k 个坐标系时的 IMU 状态可表示为：

$$\mathbf{x}_k^{BL^{(0)}} = \left[\mathbf{p}_{BI_k}^{BL^{(0)}}, \mathbf{v}_{BI_k}^{BL^{(0)}}, \mathbf{q}_{BI_k}^{BL^{(0)}}, \mathbf{ba}_k, \mathbf{bg}_k\right] \tag{1}$$

其中，$\mathbf{x}_k^{BL^{(0)}}$包括位置、速度、四元数旋转、加速度偏差$\mathbf{ba}_k$和陀螺仪偏差 $\mathbf{bg}_k$。因此，FGO 局部窗口内的状态集（$\mathbf{X}^{BL^{(0)}}$）可表示为：

$$\mathbf{X}^{BL^{(0)}} = \left[\mathbf{x}_0^{BL^{(0)}}, \mathbf{x}_1^{BL^{(0)}}, \cdots, \mathbf{x}_K^{BL^{(0)}}\right] \tag{2}$$

其中，K 表示 FGO 滑窗的大小。

3.1.1 IMU 预积分模型

IMU 单元用于测量 IMU 体坐标系中系统的角速度和比力。加性噪声以及加速度和陀螺仪偏差缓变会使测量结果产生偏差 [19]：

$$\tilde{\mathbf{a}}^{BI} = \mathbf{R}_{BL^{(0)}}^{BI}(\mathbf{a}^{BL^{(0)}} - \mathbf{g}^{BL^{(0)}}) + \mathbf{ba} + \mathbf{n}_a \tag{3}$$

$$\tilde{\boldsymbol{\omega}}^{BI} = \boldsymbol{\omega}^{BI} + \mathbf{bg} + \mathbf{n}_g \tag{4}$$

$\tilde{\mathbf{a}}^{BI}$ 是 IMU 体坐标系的原始 IMU 加速度测量值，$\mathbf{a}^{BL^{(0)}}$是 ENU 坐标系中系统的无噪加速度。$\mathbf{g}^{BL^{(0)}}$是世界坐标系中的重力。$\mathbf{R}_{BL^{(0)}}^{BI} \in SO_3$是从 ENU 坐标系到 IMU 体坐标系的旋转矩阵。$\mathbf{ba}$表示加速度偏差缓变，其导数服从高斯分布。$\mathbf{n}_a \sim N(0, \boldsymbol{\sigma}_a^2)$是加速度的加性噪声。$\tilde{\boldsymbol{\omega}}^{BI}$是 IMU 体坐标系中的原始 IMU 陀螺仪测量值，$\boldsymbol{\omega}^{BI}$是无噪旋转速率。$\mathbf{bg}$是$\boldsymbol{\omega}^{BI}$的偏差，同样假设其导数对象服从高斯分布。$\mathbf{n}_g \sim N(0, \boldsymbol{\sigma}_g^2)$是 $\boldsymbol{\omega}^{BI}$ 的加性噪声。考虑到 IMU 测量的高频特性，采用 IMU 预积分技术将多个 IMU 测量值堆叠到单个因子中 [19]，则局部坐标系 $\{\cdot\}^{B_k}$ 中的 IMU 预积分公式为：

$$\boldsymbol{\alpha}_{BI_{k+1}}^{BI_k} = \iint_{t_k}^{t_{k+1}} \mathbf{q}_{BI_t}^{BI_k}(\tilde{\mathbf{a}}^{BI_t} - \mathbf{ba}_k)\delta t^2 \tag{5}$$

$$\boldsymbol{\beta}_{BI_{k+1}}^{BI_k} = \int_{t_k}^{t_{k+1}} \mathbf{q}_{BI_t}^{BI_k} (\tilde{\mathbf{a}}^{BI_t} - \mathbf{ba}_k) \delta t \tag{6}$$

$$\boldsymbol{q}_{BI_{k+1}}^{BI_k} = \int_{t_k}^{t_{k+1}} \mathbf{q}_{BI_t}^{BI_k} \otimes \begin{bmatrix} 0 \\ \frac{1}{2}(\tilde{\boldsymbol{\omega}}^{BI_t} - \mathbf{bg}_k) \end{bmatrix} \delta t \tag{7}$$

其中，$\boldsymbol{\alpha}_{BI_{k+1}}^{BI_k}$、$\boldsymbol{\beta}_{BI_{k+1}}^{BI_k}$ 和 $\boldsymbol{q}_{BI_{k+1}}^{BI_k}$ 分别是位置、速度和旋转的预积分。实际上，IMU 测量是离散的，并通过线性插值与 LiDAR 坐标系同步。本文不采用上述的连续数值积分，而是采用中位积分。假设BI_t和BI_{t+1}是两个历元BI_k和BI_{k+1}之间的两个连续时间点，δt 是 BI_t 和 BI_{t+1} 之间的时间间隔。

中位积分作为两个 IMU 状态间的相对运动测量值，可以起到约束作用。残差 $\mathbf{r}_{IMU}(\mathbf{z}_{BI_{k+1}}^{BI_k}, \mathbf{X}^{BL^{(0)}})$ 可定义为 [19]：

$$\mathbf{r}_{IMU}(\mathbf{z}_{BI_{k+1}}^{BI_k}, \mathbf{X}^{BL^{(0)}}) = \begin{bmatrix} (\mathbf{q}_{BI_k}^{BL^{(0)}})^{-1}(\mathbf{p}_{BI_{k+1}}^{BL^{(0)}} - \mathbf{p}_{BI_k}^{BL^{(0)}} - \mathbf{v}_{BI_k}^{BL^{(0)}}\delta t - \frac{1}{2}\mathbf{g}^{BL^{(0)}}\delta t^2) - \boldsymbol{\alpha}_{BI_{k+1}}^{BI_k} \\ 2\left[(\mathbf{q}_{BI_{k+1}}^{BI_k})^{-1} \otimes ((\mathbf{q}_{BI_k}^{BL^{(0)}})^{-1} \otimes \mathbf{q}_{BI_{k+1}}^{BL^{(0)}})\right]_{xyz} \\ (\mathbf{q}_{BI_k}^{BL^{(0)}})^{-1}(\mathbf{v}_{BI_{k+1}}^{BL^{(0)}} - \mathbf{v}_{BI_k}^{BL^{(0)}} - \mathbf{g}^{BL^{(0)}}\delta t) - \boldsymbol{\beta}_{BI_{k+1}}^{BI_k} \\ \mathbf{ba}_{k+1} - \mathbf{ba}_k \\ \mathbf{bg}_{k+1} - \mathbf{bg}_k \end{bmatrix} \tag{8}$$

其中，运算符 $[\cdot]_{xyz}$ 用于提取四元数的虚部。$\mathbf{z}_{BI_{k+1}}^{BI_k}$ 表示预积分测量值，由 $\boldsymbol{\alpha}_{BI_{k+1}}^{BI_k}$、$\boldsymbol{\beta}_{BI_{k+1}}^{BI_k}$ 和 $\boldsymbol{q}_{BI_{k+1}}^{BI_k}$ 组成。

3.1.2 LiDAR 扫描匹配测量模型

本节采用 [20] 提出的扫描匹配法推导 3D 点云连续坐标系之间的相对运动。对这些相对运动进行累加，则 LiDAR 里程计在给定历元 k 的位姿估计值为 $\mathbf{T}_{BL_k}^{BL^{(0)}}$，计算公式如下 [17]：

$$\mathbf{T}_{BL_k}^{BL^{(0)}} = \left[\tilde{\mathbf{p}}_{BL_k}^{BL^{(0)}}, \tilde{\mathbf{q}}_{BL_k}^{BL^{(0)}}\right] \tag{9}$$

LiDAR 扫描匹配的残差 $\mathbf{r}_L(\mathbf{T}_{L_k}^W, \mathbf{X})$ 可表示为 $\mathbf{T}_{BL_k}^{BL^{(0)}}$ 和 $\mathbf{x}_k^{BL^{(0)}}$ 之间的差值：

$$\mathbf{r}_L(\mathbf{T}_{BL_k}^{BL^{(0)}}, \mathbf{X}^{BL^{(0)}}) = \begin{bmatrix} \mathbf{p}_{BL_k}^{BL^{(0)}} - \tilde{\mathbf{p}}_{BL_k}^{BL^{(0)}} \\ 2\left[(\tilde{\mathbf{q}}_{BL_k}^{BL^{(0)}})^{-1} \otimes \mathbf{q}_{BL_k}^{BL^{(0)}}\right]_{xyz} \end{bmatrix} \tag{10}$$

[17] 介绍了扫描匹配的详细推导过程。

3.1.3 求解局部 FGO

根据残差推导结果，可通过以下组合目标函数优化滑窗内的状态集 $\mathbf{X}^{BL^{(0)}}$ [21]：

$$\begin{aligned} \mathbf{X}^{BL^{(0)*}} = \arg\min \frac{1}{2} \{ & \sum_{k=0}^{K} \rho\left(\left\|\mathbf{r}_L(\mathbf{T}_{BL_k}^{BL^{(0)}}, \mathbf{X}^{BL^{(0)}})\right\|_{\mathbf{C}_{L_k}}^2\right) \\ & + \sum_{k=0}^{K-1} \rho\left(\left\|\mathbf{r}_{IMU}(\mathbf{z}_{BI_{k+1}}^{BI_k}, \mathbf{X}^{BL^{(0)}})\right\|_{\mathbf{C}_{BI_{k+1}}^{BI_k}}^2\right) \} \end{aligned} \tag{11}$$

其中，$\mathbf{r}_L(\mathbf{T}_{BL_k}^{BL^{(0)}}, \mathbf{X}^{BL^{(0)}})$ 表示 LiDAR 扫描匹配因子的残差，$\mathbf{z}_{BI_{k+1}}^{BI_k}$ 和 $\mathbf{r}_{IMU}(\mathbf{z}_{BI_{k+1}}^{BI_k}, \mathbf{X}^{BL^{(0)}})$ 分别表示 IMU 预积分因子的测量值和残差，$\mathbf{X}^{BL^{(0)*}}$是待估计的最优状态。$\mathbf{C}_{L_k}$ 表示基于 [17] 推导的 LiDAR 扫描匹配因子的协方差矩阵。$\mathbf{C}_{BI_{k+1}}^{BI_k}$ 表示 IMU 预积分因子的协方差矩阵。$\rho(*)$ 表示基于柯西核 [22] 的鲁棒性损失函数。最后，采用 Ceres Solver[23] 求解这个非线性问题，通过 Levenberg-Marquardt（L-M）算法 [24] 迭代最小化代价函数（11）。基于状态集$\mathbf{X}^{BL^{(0)}}$，在局部 FGO 滑窗内累加原始 3D 点云可以生成 PCM，结果表示为 $\mathbf{M}_k^{BL^{(0)}}$。

3.2 LIO 与基于全局 FGO 的 GNSS-RTK 积分

全局 FGO 是 LIO 位姿估计（见 3.1 节）和 GNSS-RTK 定位的融合。与 LIO 类似，全局 FGO 优化了滑窗内的状态集。ENU 坐标系中 IMU 坐标系的第 k 个状态可表示为：

$$\mathbf{x}_k^L = \left[\mathbf{p}_{BI_k}^L, \mathbf{q}_{BI_k}^L\right], \tag{12}$$

其中，$\mathbf{x}_k^L$ 由 ENU 坐标系中的位置和四元数旋转组成。FGO 局部滑窗内的状态集（$\mathbf{X}_k^L$）可表示为：

$$\mathbf{X}^L = \left[\mathbf{x}_0^L, \mathbf{x}_1^L, \cdots, \mathbf{x}_K^L\right], \tag{13}$$

其中，K 表示 FGO 滑窗的大小。假设优化后的 GNSS-RTK 位姿估计公式如下：

$$\mathbf{z}_{r,k}^G = \left(p_{r,k,x}^G, p_{r,k,y}^G, p_{r,k,z}^G\right)^{\mathrm{T}} \tag{14}$$

其中，$\mathbf{z}_{r,k}^G$ 用 ECEF 坐标系表示。则 $\mathbf{z}_{r,k}^G$ 的残差为：

$$\mathbf{r}_{\mathrm{GNSS}}(\mathbf{z}_{r,k}^G, \mathbf{X}^L) = \left[(\mathbf{T}_L^G)^{-1}(\mathbf{z}_{r,k}^G - \mathbf{t}_L^G) - \mathbf{p}_{BI_k}^L\right] \tag{15}$$

其中，$\mathbf{r}_{\mathrm{GNSS}}(\mathbf{z}_{r,k}^G, \mathbf{X}^L)$ 表示与 $\mathbf{z}_{r,k}^G$ 关联的残差。根据 LIO，给定历元 k 的观测结果是 $\mathbf{x}_k^{BL^{(0)}}$，给定历元 $k-1$ 的观测结果是 $\mathbf{x}_{k-1}^{BL^{(0)}}$。残差计算公式如下：

$$\mathbf{r}_{\mathrm{LIO}}(\mathbf{x}_{k-1}^{BL^{(0)}}, \mathbf{x}_k^{BL^{(0)}}, \mathbf{x}_{k-1}^L, \mathbf{x}_k^L) = \begin{bmatrix} (\mathbf{P}_1 - \mathbf{P}_2) - (\mathbf{p}_{BI_k}^L - \mathbf{p}_{BI_{k-1}}^L) \\ 2\left[((\mathbf{R}_2)^{-1}\mathbf{R}_1) \otimes (\mathbf{p}_{BI_{k-1}}^L \otimes \mathbf{p}_{BI_k}^L)\right]_{xyz} \end{bmatrix} \tag{16}$$

其中：

$$\begin{cases} \mathbf{P}_1 = (\mathbf{p}_{BL_{(0)}}^{L})^{-1}((\mathbf{R}_{BL}^{BI})^{-1}(\mathbf{p}_{BL_k}^{BL_{(0)}} - \mathbf{t}_{BL}^{BI}) - \mathbf{t}_{BL_{(0)}}^{L}) \\ \mathbf{P}_2 = (\mathbf{p}_{BL_{(0)}}^{L})^{-1}((\mathbf{R}_{BL}^{BI})^{-1}(\mathbf{p}_{BL_{k-1}}^{BL_{(0)}} - \mathbf{t}_{BL}^{BI}) - \mathbf{t}_{BL_{(0)}}^{L}) \\ \mathbf{R}_1 = (\mathbf{q}_{BL_{(0)}}^{L})^{-1}\mathbf{q}_{BL_k}^{BL_{(0)}}(\mathbf{R}_{BL}^{BI})^{-1} \\ \mathbf{R}_2 = (\mathbf{q}_{BL_{(0)}}^{L})^{-1}\mathbf{q}_{BL_{k-1}}^{BL_{(0)}}(\mathbf{R}_{BL}^{BI})^{-1} \end{cases} \tag{17}$$

为使推导过程更加清晰，本文定义了 $\mathbf{P}_1$ 和 $\mathbf{P}_2$。根据残差推导结果，可通过以下组合目标函数优化滑窗内的状态集 $\mathbf{X}^L$ [21]：

$$\mathbf{X}^{L^*} = \arg\min \frac{1}{2}\{\sum_{k=0}^{K}\rho(\left\|\mathbf{r}_{\mathrm{GNSS}}(\mathbf{z}_{r,k}^{G}, \mathbf{X}^{L})\right\|_{\mathbf{C}_{\mathrm{GNSS}}}^{2}) + \sum_{k=0}^{K-1}\rho(\left\|(\mathbf{x}_{k-1}^{BL_{(0)}}, \mathbf{x}_{k}^{BL_{(0)}}, \mathbf{x}_{k-1}^{L}, \mathbf{x}_{k}^{L})\right\|_{\mathbf{C}_{\mathrm{LIO}}}^{2})\} \tag{18}$$

其中，$\mathbf{r}_{\mathrm{GNSS}}(\mathbf{z}_{r,k}^{G}, \mathbf{X}^{L})$ 表示 GNSS 因子的残差，$\mathbf{r}_{\mathrm{LIO}}(\mathbf{x}_{k-1}^{BL_{(0)}}, \mathbf{x}_{k}^{BL_{(0)}}, \mathbf{x}_{k-1}^{L}, \mathbf{x}_{k}^{L})$ 表示 LIO 因子的残差，$\mathbf{X}^{L^*}$ 是待估计的最优状态。$\mathbf{C}_{GNSS}$ 表示 GNSS 因子的协方差矩阵。$\mathbf{C}_{LIO}$ 表示 LIO 因子的协方差矩阵。与局部 FGO 类似，通过 Ceres Solver[23] 利用 L-M 算法 [24] 求解公式 (18)。基于状态集 $\mathbf{X}^L$，在全局 FGO 滑窗内累加原始 3D 点云可以校正 PCM，表示为 $\mathbf{M}_k^L$。与 [16] 相比，通过 GNSS-RTK 定位积分生成的 PCM 没有漂移，这是本文的研究成果之一。

4 NLOS 信号排除与 GNSS-RTK 定位

4.1 基于 PCM 的 GNSS NLOS 信号排除

GNSS 接收机 $\rho_{r,k}^{s}$ 的伪距测量值可表示为 [25]：

$$\rho_{r,k}^{s} = r_{r,k}^{s} + c(\delta_{r,k} - \delta_{r,k}^{s}) + I_{r,k}^{s} + T_{r,k}^{s} + \varepsilon_{r,k}^{s} \tag{19}$$

其中，$r_{r,k}^{s}$ 是卫星和 GNSS 接收机之间的几何范围。$I_{r,k}^{s}$ 表示电离层延迟距离，$T_{r,k}^{s}$ 表示对流层延迟距离。$\varepsilon_{r,k}^{s}$ 表示多径效应、NLOS 接收信号、接收机噪声和天线相位相关噪声引起的误差。大气效应（$T_{r,k}^{s}$ 和 $I_{r,k}^{s}$）通过 RTKLIB 的传统模型（Saastamoinen 和 Klobuchar 模型）进行补偿 [26]。由于 ENU 坐标系中的 PCM、卫星仰角和方位角可以基于伪距测量值通过最小二乘法计算，因此可以通过 [16] 提出的快速搜索方法检测 GNSS NLOS 接收信号。GNSS NLOS 接收信号检测如图 2 所示。

假设历元 k 内接收的卫星集为 $\mathbf{SV}_{r,k} = \{SV_{r,k}^{1}, \cdots, SV_{r,k}^{i}, \cdots, SV_{r,k}^{N}\}$，其中包括 LOS 和 NLOS 卫星。变量 N 表示历元 k 内接收的卫星数量。$SV_{r,k}^{i}$ 表示卫星 i 的测量值。检测和排除 GNSS NLOS 接收信号后，未排除的卫星集为 $\mathbf{SSV}_{r,k} = \{SV_{r,k}^{1}, \cdots, SV_{r,k}^{i}, \cdots, SV_{r,k}^{M}\}$，其中变量 M 表示未排除的卫星数量。

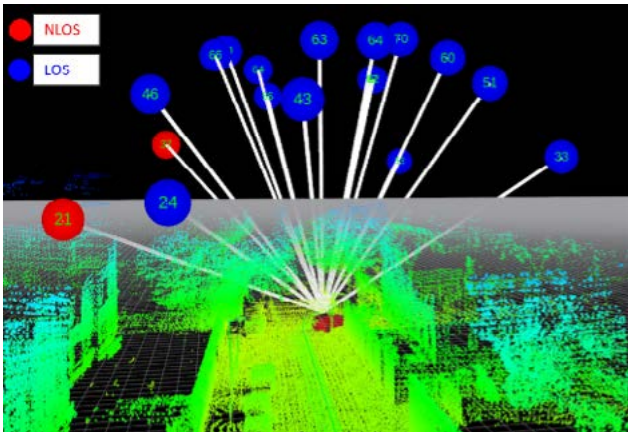


图 2 基于 PCM 检测 NLOS 接收信号。红圈表示 GNSS NLOS 卫星，蓝圈表示 LOS 测量数据，圈内数字表示卫星仰角。

4.2 基于未排除卫星的 GNSS-RTK 定位

GNSS 接收机的每次载波相位测量 $\psi_{r,k}^{s}$ 表示为 [25]：

$$\lambda\psi_{r,k}^{s} = r_{r,k}^{s} + c(\delta_{r,k} - \delta_{r,k}^{s}) + I_{r,k}^{s} + T_{r,k}^{s} + \lambda B_{r,k}^{s} + \delta\psi_{r,k}^{s} + \varepsilon_{r,k}^{s} \tag{20}$$

变量 λ 表示载波波长。变量 $\delta\psi_{r,k}^{s}$ 表示载波相位校正项，其中包括天线相位偏移和变化、固体潮引起的站位移、相位转绕效应和卫星时钟相对论性校正。$\varepsilon_{r,k}^{s}$ 表示多径效应、NLOS 接收信号、接收机噪声和天线时延引起的误差。$B_{r,k}^{s}$ 是载波相位偏差，其计算公式如下。[26] 介绍了载波相位校正的详细公式。

$$B_{r,k}^{s} = \psi_{r,0,k} - \psi_{0,k}^{s} + N_{r,k}^{s} \tag{21}$$

变量 $\psi_{r,0,k}$ 表示接收机本振的初始相位。类似地，变量 $\psi_{0,k}^{s}$ 代表卫星发射的导航信号的初始相位。变量 $N_{r,k}^{s}$ 表示载波相位的整周模糊度。

卫星 s 的双差（Double Difference，DD）伪距测量（$\rho_{DD,k}^{s}$）公式如下：

$$\rho_{DD,k}^{s} = (\rho_{r,k}^{s} - \rho_{b,k}^{s}) - (\rho_{r,k}^{w} - \rho_{b,k}^{w}) \tag{22}$$

变量 $\rho_{b,k}^{w}$ 和 $\rho_{b,k}^{s}$ 代表参考站接收的伪距测量值，参考站用下标 b 表示。仰角最高的卫星其多径和 NLOS 误差往往最小。因此以仰角最高的卫星 w 为主卫星。将 DD 应用于伪距测量后，推导出来的 $\rho_{DD,k}^{s}$ 没有钟差，且不受大气效应的影响 [26]。类似地，卫星 s 的 DD 载波相位测量（$\psi_{DD,k}^{s}$）公式如下：

$$\psi^s_{DD,k} = (\psi^s_{r,k} - \psi^s_{b,k}) - (\psi^w_{r,k} - \psi^w_{b,k}) \tag{23}$$

变量 $\psi^s_{b,k}$ 和 $\psi^w_{b,k}$ 代表参考站接收的载波相位测量值。同样，$\psi^s_{DD,k}$ 也没有括钟差，且不受大气效应的影响。变量 $\psi^s_{DD,k}$ 包括待估计的 DD 模糊度 [26]。

GNSS-RTK 的浮点解可表示为：

$$\mathbf{x}^{float}_{r,k} = (\mathbf{p}^G_{r,k}, \Delta N^1_{rb,k}, \Delta N^2_{rb,k}, \cdots, \Delta N^{M-1}_{rb,k})^{\mathrm{T}} \tag{24}$$

其中，变量 $\mathbf{x}^{float}_{r,k}$ 表示 GNSS 接收机在历元 k 内的状态，该状态由 ECEF 坐标系中的位置（$\mathbf{p}^G_{r,k}$）和 DD 模糊度组成。变量 $\Delta N^{M-1}_{rb,k}$ 表示卫星 $M-1$ 的 DD 载波相位模糊度偏差。换句话说，每个 DD 载波相位测量值都有特定的模糊度偏差。因此，DD 伪距测量的观测模型（$\rho^s_{DD,k}$）可表示为：

$$\rho^s_{DD,k} = \mathbf{h}^s_{DD,\rho,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b) + \omega^s_{DD,\rho,k} \tag{25}$$

$$\begin{aligned}\mathbf{h}^s_{DD,\rho,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b) = (\left\|\mathbf{x}_{r,k} - \mathbf{p}^s_k\right\| - \|\mathbf{p}_b - \mathbf{p}^s_k\|) \\ - (\left\|\mathbf{x}_{r,k} - \mathbf{p}^w_k\right\| - \|\mathbf{p}_b - \mathbf{p}^w_k\|)\end{aligned} \tag{26}$$

变量 $\omega^s_{DD,\rho,k}$ 表示与 $\rho^s_{DD,k}$ 关联的噪音。函数 $\mathbf{h}^s_{DD,\rho,k}(*)$ 表示计算 GNSS 接收机状态和 DD 测量值 $\rho^s_{DD,k}$ 的观测函数。DD 伪距测量的误差因子如下：

$$\begin{aligned}&\left\|e^s_{DD,\rho,k}\right\|^2_{\Sigma^s_{DD,\rho,k}} \\ &= \left\|\rho^s_{DD,k} - \mathbf{h}^s_{DD,\rho,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b)\right\|^2_{\Sigma^s_{DD,\rho,k}}\end{aligned} \tag{27}$$

变量 $\Sigma^s_{DD,\rho,k}$ 表示与变量 $\rho^s_{DD,k}$ 关联的协方差，该变量通过收到的卫星测量信号的仰角和信噪比（Signal-to-Noise Ratio，SNR）计算得出。同样，DD 载波相位测量的观测模型可表示为：

$$\psi^s_{DD,k} = \mathbf{h}^s_{DD,\psi,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b) + \omega^s_{DD,\psi,k} \tag{28}$$

$$\begin{aligned}\mathbf{h}^s_{DD,\psi,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b) = (\left\|\mathbf{x}_{r,k} - \mathbf{p}^s_k\right\| - \|\mathbf{p}_b - \mathbf{p}^s_k\|) \\ - (\left\|\mathbf{x}_{r,k} - \mathbf{p}^w_k\right\| - \|\mathbf{p}_b - \mathbf{p}^w_k\|) + \Delta N^s_{rb,k}\end{aligned} \tag{29}$$

变量 $\omega^s_{DD,\psi,k}$ 表示与 $\psi^s_{DD,k}$ 关联的噪声。变量 $\Delta N^s_{rb,k}$ 表示载波相位测量的 DD 模糊度。则 DD 载波相位伪距测量的误差因子如下：

$$\begin{aligned}&\left\|e^s_{DD,\psi,k}\right\|^2_{\Sigma^s_{DD,\psi,k}} \\ &= \left\|\psi^s_{DD,k} - \mathbf{h}^s_{DD,\psi,k}(\mathbf{x}_{r,k}, \mathbf{p}^s_k, \mathbf{p}^w_k, \mathbf{p}_b)\right\|^2_{\Sigma^s_{DD,\psi,k}}\end{aligned} \tag{30}$$

变量 $\Sigma^s_{DD,\psi,k}$ 代表与 $\psi^s_{DD,k}$ 关联的协方差。则 GNSS-RTK 浮点解估计的目标函数为：

$$\mathbf{x}^{float*}_{r,k} = \arg\min \sum_{s,k} (\left\|e^s_{DD,\psi,k}\right\|^2_{\Sigma^s_{DD,\psi,k}} + \left\|e^s_{DD,\rho,k}\right\|^2_{\Sigma^s_{DD,\rho,k}}) \tag{31}$$

变量 $\mathbf{x}^{float*}_{r,k}$ 表示浮点解的最优估计值。通过 Ceres Solver [23] 求解上述目标函数（31），可得出当前历元的 GNSS-RTK 浮点解。得到 GNSS-RTK 浮点解后，使用模糊度求解算法估计固定解。载波相位测量值没有噪声时，变量 $\Delta N^s_{r,k}$ 应为整数。本文通过广泛应用的 LAMBDA 算法 [28] 求解整周模糊度，得到固定解 $\mathbf{x}^{fix}_{r,k}$。然后将固定解代入全局 FGO 来校正 3.2 节中的 PCM 漂移。

5 实验结果及相关讨论

5.1 实验设置

本节通过香港典型城市场景中采集的数据集验证本文方法的有效性。实验采用多传感器数据采集车，如图 3 所示。测试场景如图 3 右下角所示，街道两侧有高楼和树木，不利于 GNSS-RTK 定位。

实验采用 u-blox M8T GNSS 接收机以 1Hz 的频率采集 GPS/ 北斗的原始测量数据，并部署了一个 3D LiDAR 传感器（Velodyne 32）以 10 Hz 的频率采集原始 3D 点云数据，同时采用 Xsens Ti-10 IMU 以 200 Hz 的频率收集数据。此外，还采用了 NovAtel SPAN-CPT 提供地面真值，这是一个 GNSS（GPS、GLONASS 和北斗）RTK/INS（光纤陀螺仪、FOG）组合导航系统。FOG 陀螺仪偏差运行稳定性为 1 度 / 小时，随机游动为 0.067 度 / 小时。数据采集车与 GNSS 基站间的基线距离约为 5 公里。所有数据均采用机器人操作系统（Robot Operation System，ROS）[29] 来收集和同步。实验前已对所有传感器间的坐标系统进行校正。

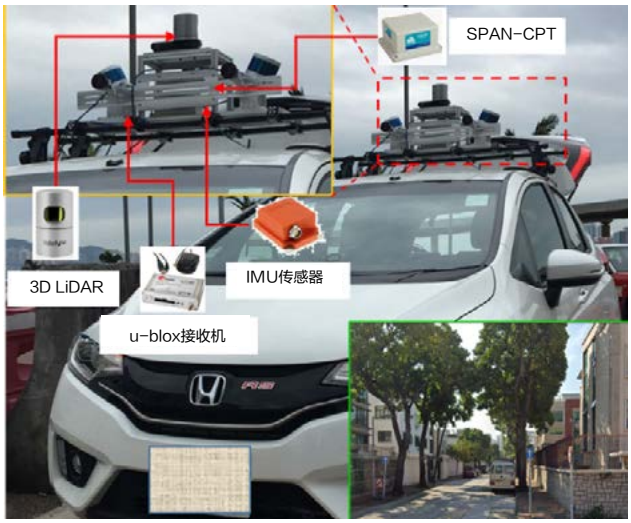


图 3 数据采集车和测试场景

下面通过比较三种方法分析 GNSS-RTK 定位性能，验证本文方法用于提升 GNSS-RTK 定位性能的有效性。分析采用 ENU 坐标系评估精度，以第一个点作为参考位置。

（a）GNSS-RTK：基于 u-blox M8T 接收机原始测量数据的传统 GNSS-RTK 定位方案 [26]。模糊度逐历元求解。

（b）u-blox 接收机：基于 u-blox M8T 接收机、无参考站校正的商用 GNSS 定位方案

（c）GNSS-RTK-C：基于 u-blox M8T 接收机的原始测量数据以及本文 GNSS NLOS 信号排除法的 GNSS-RTK 定位方案。模糊度逐历元求解。

5.2 城市峡谷场景性能评估

以上三种方法的结果对比如表 1 所示。表格第一行是用于定位估计的卫星数量。GNSS 定位平均需要 17 颗卫星的接收信号，而本文的 GNSS NLOS 信号排除法平均只需要 15.8 颗卫星的接收信号。第二列为 u-blox 接收机的 2D 定位误差，定位结果基于 u-blox 接收机的标准 NMEA 消息。平均误差为 6.25 米，标准差为 7.31 米，最大误差为 38.53 米。整个过程都采用了 GNSS 方案。第三列为传统 GNSS-RTK 定位的结果。u-blox 接收机和参考站的原始测量数据使平均误差降低到 2.43 米，标准偏差和最大误差分别下降到 1.16 米和 7.20 米。由于测量数据的质量问题，固定率只有 1.0%。本文的 GNSS NLOS 信号排除法（GNSS-RTK-C）使平均误差降至 1.95 米，标准偏差为 1.003 米，但最大误差为 12.40 米，与 GNSS-RTK 的 7.20 米相比大幅增加。不过固定率小幅度上升至 1.6%。实验结果证明，本文方法能够有效减少 NLOS 信号对定位的影响。

表 1 定位性能对比

项目	u-blox 接收机	GNSS-RTK	GNSS-RTK-C
平均卫星数	17	17	15.8
平均误差	6.25 米	2.43 米	1.95 米
标准差	7.31 米	1.16 米	1.003 米
最大误差	38.53 米	7.20 米	12.40 米
固定率	不涉及	1.0%	1.6%

上述三种方法的轨迹以及地面真值如图 4 所示，整个实验的定位误差如图 5 所示。在图 5 圆圈标出的历元 A 附近，本文方法引起的定位误差明显大于传统的 GNSS-RTK，其原因是过度排除 GNSS NLOS 信号。图 4 中部为接近历元 A 的场景，其中树木较多。图 5 下方为排除的卫星数量，采用生成的 PCM 检测到其中四颗卫星属于 NLOS 卫星。排除 NLOS 卫星后，卫星的几何分布严重扭曲，导致定位误差增加。[4]、[30] 和 [31] 也发现了类似现象。

总体而言，本文方法（GNSS-RTK-C）与传统方法（GNSS-RTK）相比，能够有效检测和排除 GNSS 潜在 NLOS 信号，从而提高定位性能。但两种方法的固定率都不高（GNSS-RTK：1.0%，GNSS-RTK-C：1.6%），对于 GNSS-RTK-C 而言，这主要是因为排除 GNSS NLOS 信号会使卫星的几何分布变差。

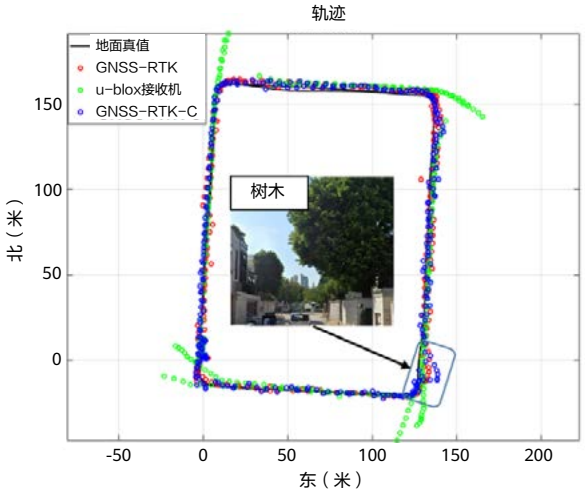


图 4 三种方法的轨迹。黑色曲线表示地面真值，红色、绿色和蓝色曲线分别表示 GNSS-RTK、u-blox 接收机和 GNSS-RTK-C 方案。

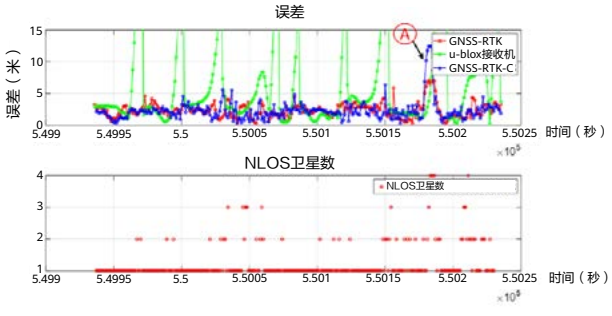


图 5 上方为三种方法的误差。红色、绿色和蓝色曲线分别表示 GNSS-RTK、u-blox 接收机和 GNSS-RTK-C 方案。下方为排除的 GNSS NLOS 卫星数量。

6 结语

GNSS-RTK 定位目前仍然是实现自动驾驶定位的必备技术，自动驾驶需要精确的绝对定位。但是由于信号反射和卫星几何分布较差，GNSS-RTK 定位难以广泛应用于城市峡谷场景。本文提出一种新方法，通过车载传感（LiDAR/惯性积分）检测和排除 GNSS 潜在 NLOS 信号，从而提升 GNSS-RTK 定位的性能。对于实验数据集，本文的 NLOS 信号排除法使 GNSS-RTK 定位的精度从 2.43 米提高到 1.95 米。

但由于排除 GNSS NLOS 信号会使卫星的几何分布变差，因此 GNSS-RTK 固定率仍然较低。未来，更多的环境特征将作为伪卫星被用于优化卫星的几何分布，进一步提高城市峡谷场景下GNSS-RTK定位的整周模糊度固定率。多个 3D LiDAR 也将被用于生成更全面、更详细的 PCM，从而增大环境重构的视场角。

参考文献

[1] P. J. Teunissen and A. Kleusberg, "GPS for geodesy," *Springer Science & Business Media*, 2012.

[2] G. Wan et al., "Robust and precise vehicle localization based on multi-sensor fusion in diverse city scenes," in 2018 *IEEE International Conference on Robotics and Automation (ICRA)*, 2018: IEEE, pp. 4670-4677.

[3] P. J. Teunissen, "Least-squares estimation of the integer GPS ambiguities," in I*nvited lecture, section IV theory and methodology*, IAG general meeting, Beijing, China, 1993.

[4] W. Wen, G. Zhang, and L. T. Hsu, "Correcting NLOS by 3D LiDAR and building height to improve GNSS single point positioning," *Navigation*, vol. 66, no. 4, pp. 705-718, 2019.

[5] R. Furukawa, N. Kubo, and A. El-Mowafy, "Prediction of RTK-GNSS performance in urban environments using a 3D model and continuous LOS method," in *Proceedings of the 2020 International Technical Meeting of The Institute of Navigation*, 2020, pp. 763-771.

[6] P. Fan, W. Li, X. Cui, and M. Lu, "Precise and robust RTK-GNSS positioning in urban environments with dual-antenna configuration," *Sensors*, vol. 19, no. 16, p. 3586, 2019.

[7] T. Li, H. Zhang, Z. Gao, Q. Chen, and X. Niu, "High-accuracy positioning in urban environments using single-frequency multi-GNSS RTK/MEMS-IMU integration," *Remote Sensing*, vol. 10, no. 2, p. 205, 2018.

[8] G. C. Chung, S. T. Su, and M. Y. Alias, "Adaptive windowed statistical selection rake for long ultra-wideband multipath channels," (in English), *Wireless Pers Commun*, vol. 98, no. 1, pp. 453-466, Jan 2018, doi: 10.1007/s11277-017-4878-8.

[9] H. T. Zhang, "Performance comparison of kinematic GPS integrated with different tactical grade IMUs," *CALGARY, ALBERTA: THE UNIVERSITY OF CALGARY*, 2006.

[10] P. Henkel, A. Blum, and C. Günther, "Precise RTK positioning with GNSS, INS, Barometer and Vision,"

in *Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017)*, 2017, pp. 2290-2303.

[11] T. Li, H. Zhang, Z. Gao, X. Niu, and N. El-Sheimy, "Tight fusion of a monocular camera, MEMS-IMU, and single-frequency multi-GNSS RTK for precise navigation in GNSS-challenged environments," *Remote Sensing*, vol. 11, no. 6, p. 610, 2019.

[12] X. Bai, W. Wen, and L.-T. Hsu, "Performance analysis of visual/inertial integrated positioning in typical urban scenarios of Hong Kong," in *Proceedings of 2019 Asian-Pacific Conference on Aerospace Technology and Science*, 2019.

[13] W. Wen, G. Zhang, and L.-T. Hsu, "GNSS NLOS exclusion based on dynamic object detection using LiDAR point cloud," *IEEE Transactions on Intelligent Transportation Systems*, 2019.

[14] G. Zhang, Weisong Wen, and L.-T. Hsu, "Correcting GNSS NLOS by 3D LiDAR and building height," *ION GNSS+ 2018*, Miami, Florida, USA, 2018.

[15] W. Wen, G. Zhang, and L.-T. Hsu, "Exclusion of GNSS NLOS receptions caused by dynamic objects in heavy traffic urban scenarios using real-time 3D point cloud: An approach without 3D maps," in *Position, Location and Navigation Symposium (PLANS), 2018 IEEE/ION*, 2018: IEEE, pp. 158-165.

[16] W. Wen, "3D LiDAR aided GNSS and its tightly coupled integration with INS via factor graph optimization," in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, 2020, pp. 1649-1672.

[17] J. Zhang, W. Wen, F. Huang, and L.-T. Hsu, "Loosely coupled Lidar-inertial odometry for urban positioning and mapping (to be submitted)," *Remote Sensing*, 2021.

[18] W. Wen, G. Zhang, and L.-T. Hsu, "Object-detection-aided GNSS and its integration with Lidar in highly urbanized areas," *IEEE Intelligent Transportation Systems Magazine*, vol. 12, no. 3, pp. 53-69, 2020.

[19] C. Forster, L. Carlone, F. Dellaert, and D. Scaramuzza, "On-manifold preintegration for real-time visual--inertial odometry," *IEEE Transactions on Robotics*, vol. 33, no. 1, pp. 1-21, 2016.

[20] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Robotics: Science and Systems*, 2014, vol. 2, no. 9.

[21] F. Dellaert and M. Kaess, "Factor graphs for robot perception," *Foundations Trends in Robotics*, vol. 6, no. 1-2, pp. 1-139, 2017.

[22] Z. Zhang, "Parameter estimation techniques: a tutorial with application to conic fitting," *Image vision Computing*, vol. 15, no. 1, pp. 59-76, 1997.

[23] S. Agarwal and K. Mierle. "Ceres Solver." http://ceres-solver.org (accessed 6 January 2021).

[24] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," in *Numerical analysis: Springer*, 1978, pp. 105-116.

[25] E. Kaplan and C. Hegarty, *Understanding GPS: principles and applications*. Artech house, 2005.

[26] T. Takasu and A. Yasuda, "Development of the low-cost RTK-GPS receiver with an open source program package RTKLIB," in *International symposium on GPS/GNSS*, 2009: International Convention Center Jeju Korea, pp. 4-6.

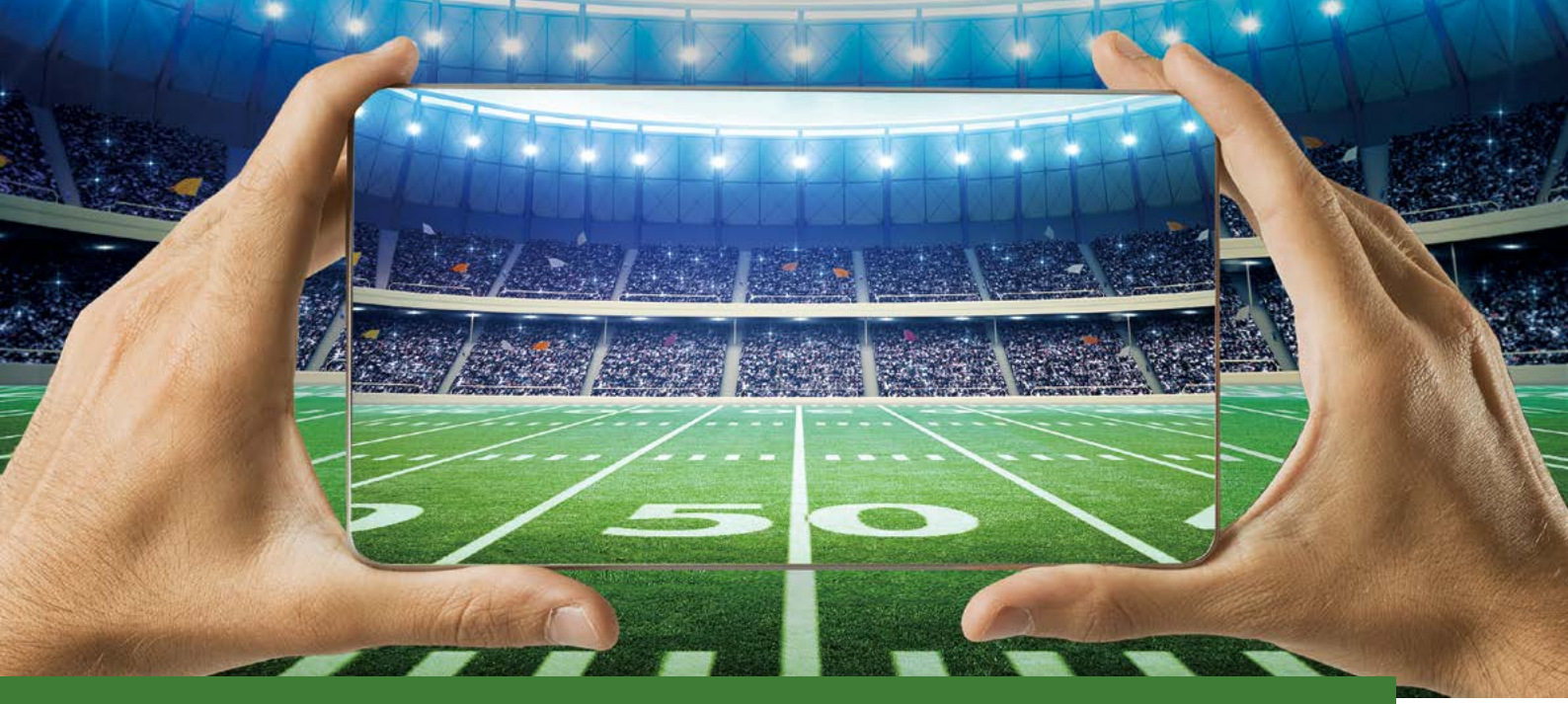
[27] A. M. Herrera, H. F. Suhandri, E. Realini, M. Reguzzoni, and M. C. de Lacy, "goGPS: open-source MATLAB software," *GPS Solution*, vol. 20, no. 3, pp. 595-603, 2016.

[28] P. Teunissen, "Theory of integer equivariant estimation with application to GNSS," *Journal of Geodesy*, vol. 77, no. 7-8, pp. 402-410, 2003.

[29] M. Quigley et al., "ROS: an open-source Robot Operating System," in I*CRA workshop on open source software*, 2009, vol. 3, no. 3.2: Kobe, Japan, p. 5.

[30] X. Bai, W., Zhang, G., Hsu, and Li-Ta, "Real-time GNSS NLOS detection and correction aided by sky-pointing camera and 3D LiDAR," presented at the *Proceedings of ION Pacific PNT 2019*, Honolulu, HA, USA, 2019.

[31] X. Bai, W. Wen, and L.-T. Hsu, "Using Sky-pointing fish-eye camera and LiDAR to aid GNSS single-point positioning in urban canyons," I*ET Intelligent Transport Systems*, vol. 14, no. 8, pp. 908-914, 2020.



技术使能艺术：新一代 HDR Vivid 视频技术标准

余全合[1]，徐巍炜[1]，王弋川[1]，张继武[1]，陈虎[2]，袁乐[3]

[1] 中央媒体技术院

[2] 慕尼黑音视频实验室

[3] 上海海思信源技术开发部

摘要

目前电视、电脑、手机、移动平板等显示设备的显示能力参差不齐，对于同一内容的显示效果不尽相同。常常会出现制作时的画面质量和终端呈现的画面质量不一致的情况，使消费者无法完全感知创作意图。为了缓解这一问题，创作者在创作时不得不做一些妥协；业界也制定了一些相关标准。尽管如此，消费者的体验和创作者想要表达的内容还是无法做到一致。为了解决这一难题，HDR（High Dynamic Range）Vivid 通过分析制作端作品内容的特征，生成元数据，基于这些元数据在终端侧根据人眼视觉系统的特性，结合终端的显示能力进行亮度、对比度、以及色彩感知保持的显示适配，从而最大限度还原创作意图。该技术通过优化色彩、亮度、对比度等画面要素，精准调色，从而呈现更卓越的视觉效果，并在不同的消费终端呈现出相似的创作意图，搭建了创作者和消费者之间的“桥梁”。

关键词

HDR，色调映射，视觉感知，亮度和对比度保存，质量控制，色彩感知，人眼感知模型，优化

1 引言

高质量影像主要取决于以下五个要素：分辨率、位深度、帧速率、色域、动态范围。

分辨率指数字图像中的像素数量 [1]。对于给定的显示设备尺寸，分辨率越高，所包含的像素越多，对细节的表现也就更加精细。国际电信联盟（International Telecommunication Union，ITU）发布的 BT.709 [2] 标准规定了高清影像的分辨率为 1920×1080。2012 年，ITU 又发布了 BT.2020 [3] 标准，将 4K（3840×2160）、8K（7680×4320）分辨率标准化。

位深度表示每个像素可以显示的颜色比特数 [4]。位深越大，可显示的颜色数量就越多，整幅画面的渐变就越自然。BT.709 标准规定了高清影像的 8 bits 编码格式，为适应 4K/8K 影像要求，BT.2020 标准将位深提升到了 10/12 bits。

帧速率指单位时间（1s）内连续显示的图像的数量 [5]。电影的帧速率一般为 24p（每秒 24 帧）；BT.709 标准规定高清影像的最高帧速率为 60p；针对 8K 广播电视节目，BT.2020 标准规定的最高帧速率可达 120p，在该帧速率下，运动图像的平滑程度在视觉上几乎与真实世界无异。

色域代表可以显示的所有颜色的范围合集 [6]。自然界中可见光谱的颜色组成了最大的色域空间，包含了人眼所能看到的所有颜色。显示设备的色域越大，能够再现的颜色就越多。BT.2020 标准相比 BT.709 标准，将色域从 Rec.709 扩展到 Rec.2020，进一步扩大了颜色范围。

动态范围指某一事物最大值与最小值之间的差异，比例关系是其中的一种体现方式 [7]。在影像领域，动态范围即是亮度的差别。动态范围越大，图像上同时记录的亮部细节与暗部细节就越丰富，给观看者的感受会更加真实。BT.2020 标准大幅推动了以上四个要素的发展，但在动态范围上并没有给出提升的建议。直到 2016 年，ITU 推出的 BT.2100 [8] 标准才正式定义了高动态范围（High Dynamic Range，HDR）影像的相关参数。

HDR 技术的发展对显示终端的能力提出了更高的要求。根据 UHD 联盟针对电视机等终端设备认证提出的 UltraHD Premium 标准 [9]，以亮度要求为例，若需通过 HDR 认证，显示设备亮度特性需要满足以下要求：最大峰值亮度达到 1000 nits（cd/m^2），最大黑度值（屏幕能够表现的仅次于关闭状态的最暗的亮度）不超过 0.05 nits（cd/m^2）；或者，最大峰值亮度达到 540 nits（cd/m^2），最大黑度值不超过 0.0005 nits（cd/m^2）。然而，创作者制作的高质量 HDR 影像往往具有更大的动态范围。如何让它们能够在更低动态范围的显示设备上正确显示，同时忠实地还原创作者的意图，成为了需要解决的问题。降低动态范围

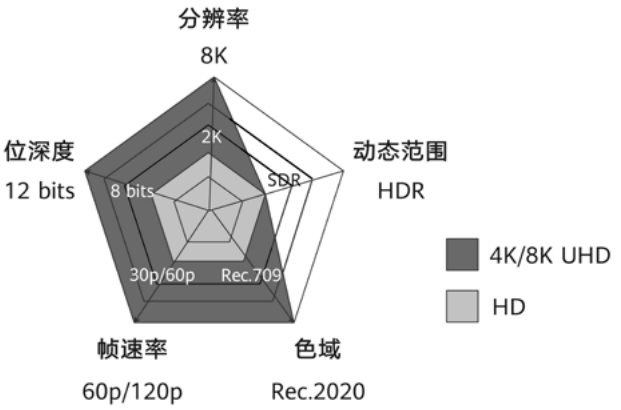


图 1 高质量影像要素

的同时尽量保持原始影像对比度、细节等特征的过程被称为色调映射（Tone Mapping）[10]。

学术界早在多年前就开始对色调映射算法进行研究。Reinhard [11] 基于人眼视锥细胞的光感受响应，构建出全局色调映射算子；Kim [12] 基于人类视觉灵敏度，提出遵循高斯分布并取决于场景的平均对数亮度映射模型；随着人工智能的发展，深度学习技术也被应用在色调映射研究中。Patel [13] 利用生成对抗网络，进行有监督的色调映射学习。Zhang [14] 采用多尺度的卷积神经网络，使得 HDR 图像的全局信息与局部信息在映射过程中均得以最大程度的保留。

为了进一步提升终端显示质量，工程实践与标准化过程中进一步采用“元数据”的概念，元数据包含了原始影像的特征信息，传输到显示端后，再根据相应的标准规范，指导终端基于元数据进行色调映射。电影电视工程师协会（Society of Motion Picture and Television Engineers，SMPTE）于 2014 年颁布了 ST 2086 标准 [15]，对 HDR 的静态元数据进行了规定。静态元数据包含了艺术家在调色时使用的监视设备的信息。

本文提出的 HDR Vivid 技术与传统技术相比，其创新性主要体现在以下几个方面：

1. 采用高分辨率、宽位深、高帧率、广色域、高动态范围的策略，制作高质量的 HDR Vivid 内容。
2. 提出了一种亮度对比度和色彩保持方法，保证 HDR 内容的灵活性，稳定性和合理性，精确还原各种场景下的亮度对比度和色彩。
3. 采用动态元数据方案，结合人眼感知，确保 HDR 内容在具有不同显示能力的终端设备上均可以得到较为一致的呈现效果。
4. 借助内容生成制作工具，开放动态元数据的调整接口，从而给予创作者更大的艺术创作空间。

HDR Vivid 相关技术已经被中国超高清视频产业联盟采纳，并成为高动态范围视频技术标准，本文将对 HDR Vivid 涉及的科学原理和技术方案做详细介绍。

2 端到端 HDR 系统

创作者通常利用现代技术创作作品，整个创作过程主要包括制作、分发和显示与感知。

图 2 端到端 HDR 系统

整个 HDR 创作过程如图 2 所示。首先，创作者使用摄像机和调色软件等工具制作出母版；然后，分发过程将母版进行压缩和传输，形成压缩码流；最后，显示和感知过程将压缩码流进行解码和显示。这三个过程相互联系，每个过程的效果都会影响下一个过程。下面将对视觉艺术和 HDR 制作、HDR 分发和 HDR 呈现和感知做详细介绍。

2.1 视觉艺术与 HDR 制作

视觉艺术是指运用一定的物质、材料、技术手法，创作可供人观看欣赏的艺术作品。从广义上说，雕塑、绘画、摄影等艺术门类都属于它的范畴，它不仅创作方式多样，造型手法也十分多样 [16]。视觉艺术作为一种传达信息的“语言”，由视觉基本元素和设计原则两部分构成的一套传达意义的规范或符号系统 [17]。基本元素包括：线条、形状、明暗、色彩、质感、空间。艺术家通过组织和运用这些基本元素实现创作意图。艺术家根据需要选择相应的材料和表现形式（雕塑或是绘画、具象或是抽象等等），运用一定的原则和方法，在一定的范围内控制各种元素之间的关系，最后形成能够传达特定信息的图像 [18]。

绘画是一种古老的视觉艺术 [19]，其发展受到技术的限制。洞穴人利用土系颜料制作洞穴壁画；文艺复兴时期从天青石中提取得到明亮的深蓝色颜料，使得该时期的圣母像呈现出高雅的蓝色；十九世纪新型色素井喷式的发展以及金属管的发明，为画家们带来了出门绘画的可能性与便捷性，带来了绘画史上的外光写生的发展。小孔成像的引入使绘画的逼真感大幅提升 [20, 21]，同时也推动了十九世纪摄影的诞生。

摄影随着盖达尔的银版摄影术诞生。1889 年，美国依斯曼柯达公司以硝化纤维素软片作为基材，大规模生产照相胶卷，推动了照相机的小型化。胶片利用银盐颗粒感光，银盐感光颗粒的光学特性近似于 Gamma 曲线 [22]。根据相纸的感光特性，辅以增厚剂和减薄剂，使用光谱特性狭窄的相纸等等才能获得满意的照片 [23]。二十世纪末，数码相机的发明使摄影技术的发展进入了新的阶段。根据摄影师 [24] 测试，胶片相机可接受的宽容度为 8，而数码相机可接受的宽容度有 9，在很多情况下胶片的噪声和细节很难分辨，而数码相机的表现更好。

从固定的石壁，到雕塑、绘画，再到电子显示设备，技术的发展深刻地改变了人创造和感知视觉艺术的方式。阴极射线管（Cathode Ray Tube，CRT）是广泛使用的电子显示设备，其使用的 P22 荧光粉色域接近 BT.709，最高亮度接近 100 nits。后来，液晶显示器（Liquid Crystal Display，LCD）快速发展，替代了 CRT，提供了更宽的色域范围以及更高的亮度 [25]。近年的 OLED（Organic Light-Emitting Diode）和 micro-LED 以及激光显示更是将色域表现拓展至 DCI-P3（Digital Cinema Initiative-P3）甚至 BT.2020，峰值亮度提升至 3000-10000 nits，显示位深提升至 10 bits。就像是矿物颜料提纯技术的发展促进了绘画的发展，HDR 技术的发展也来了更好感知体验。HDR Vivid 支持 12 bits、4000-10000 nits 的动态范围、BT.2020 色域（BT.2020 基本包络了视觉系统可感知的颜色范围）以及元数据控制，为艺术创作带来了更大的空间，为艺术的呈现带来了更灵活的手段。

2.2 HDR 分发

为了将 HDR 制作的作品进行有效分发，需要考虑合理的编码方式，其中位深度和线性到非线性转换是两个最重要因素。由于数字影像是离散的，为了保留更多、更细腻的信息，最简单的方式就是增加位深。在最大亮度不变的情况下，BT.2020 标准使用 10/12 bits 编码，记录的信息比 BT.709 标准使用的 8 bits 要丰富得多。

此外，对原始 HDR 内容最直观的编码方式是采用线性编码，但人类视觉感知系统并非是线性的，这种编码方式会造成亮部占用色阶过多，而暗部信息记录不足，不利于被人类视觉系统所感知。在视频领域，Gamma 输入输出特性曲线除了指代 CRT 显示器时代的“伽玛校正”，还有更为广泛的含义。采集端光传感器对所记录数据的转换称为光－电转换函数（Opto-electronic Transfer Function，OETF），而编码的数据在显示端被还原并显示出来的转换称为电－光转换函数（Electro-optical Transfer Function，EOTF）。任何类型的 OETF 与对应的 EOTF 曲线均可被认为是广义上的伽玛曲线。然而应该设计什么样的 Gamma 曲线才能够符合人类视觉感知呢？学术界对此有相当多的研究与成果。

早在 18 世纪，对数均匀的特性就被 Bouguer 等人提出，即亮度的对数均匀增加时，对人眼感知来说也是均匀增加的。后来，韦伯 – 费希纳定律（Weber–Fechner Law）[26] 被普遍接受，这个模型可以简单表示如下：

$$\frac{\Delta L}{L} = k \tag{1}$$

其中，L 表示绝对亮度，ΔL 表示在亮度 L 时人眼能够刚好感受到差别的亮度差，k 称为韦伯分数，是一个定值，通过求解可以得到：

$$L(p) = Ce^{kp} \tag{2}$$

其中，p 表示亮度的主观感受，C 为常数，由此可见这个模型是符合对数均匀特性的。常用的 Log 编码就是基于韦伯 – 费希纳定律来制定的。在大部分情况下，对数均匀模型都比较精确，然而当出现较暗（如 1 nit（cd/m^2）以下）或者较亮场景（如 1000 nits（cd/m^2）以上）时，另一种幂率均匀模型的表现会更好。同样，这一概念在 19 世纪就被 Plateau 等人提出，直到 1957 年 Hunt [27] 从视神经刺激出发，证明了其与光照呈现出幂函数的关系。之后，Stevens [28] 总结出史蒂文斯定律（Stevens' Power Law），指出在很大范围内，人眼对亮度的感知是随着亮度立方根均匀增加的。被普遍接受的 Gamma 编码就是基于史蒂文斯定律来制定的，只不过使用的幂次各不相同。例如仅考虑暗处时，由于对光照更加敏感的视杆细胞开始活动，DeVries–Rose 模型 [29] 指出此时感知应该是近似平方根均匀的。

随着动态范围的不断提升，传统方法容易导致色阶断裂或者码字的严重浪费，基于对比度阈限模型的新编码方式开始出现并使用。根据现实的需要，主要有两大设计思路：精确地设计编码获得对于人眼对比度 + 感知的精确适应；粗略地设计编码以换取兼容性。前者主要基于 Barten's Mode[30] 亮度感知模型，包括瞳孔大小、散粒噪声、视网膜细胞的侧向抑制、视神经噪声、时间与空间对视觉感知的影响等因素，最终用对比度灵敏度函数（Contrast Sensitivity Function，CSF）来描述结果：

$$S(X_0,u,L) = \frac{1}{m_t} = \frac{\frac{M_{opt}(u,L)}{k}}{\sqrt{\frac{2}{T}\left(\frac{1}{X_0^2}+\frac{1}{X_{max}^2}\right)\left(\frac{1}{\eta pE(L)}+\frac{\Phi_0}{1-e^{-\left(\frac{u}{u_0}\right)^2}}\right)}} \tag{3}$$

其中，m_t 为对比度灵敏度函数的倒数，L 表示亮度 (nits)，u 表示对比度的空间频率（单位为 cycles/deg），X_0 表示视角（单位为度，deg）、其余变量均在 ITU 的 BT.2446 [31] 中被标准化。在 BT.2446 中，选择 X_0 为 60deg，因此将该函数简化为二元函数 $S(u, L)$。该模型最早于 2001 年被医学数字成像和通信（Digital Imaging and Communications in Medicine，DICOM）标准 [32] 使用，仅仅利用了 u = 4cycles/deg 的截面，推出了一套 0.05–4000 nits（cd/m^2）的编码方案；2008 年，Aydın [33] 等人提出 PU（Perceptually Uniform）编码，针对每一个亮度，选择对比度敏感度最大的空间频率点，将敏感度 S 转化为 L 的单值函数。但由于使用 LUT 进行编码，所以应用场景相对受限。直到 2014 年，SMPTE 发布了 ST.2084 [34] 标准，正式推出 PQ（Perceptually Quantizer）编码函数。PQ 选取 40° 视角的 CSF，提供了 0–10000 nits，满足 10–12 bits 编码的方案。与 PU 不同的是，PQ 使用一个相对简单且易于求逆的近似函数取代了 PU 的 LUT，极大提升了其实用性。2016 年，PQ 被 BT.2100 确定为高动态范围节目分发推荐使用的两条 Gamma 曲线之一，并逐步成为互联网视频、电影等领域中 HDR 内容分发的主流格式。

然而，精确编码要求编码后的数据必须精确对应绝对亮度，不能像传统编码那样针对显示器的最高亮度做 0–1 之间的映射。因此，在不进行亮度调整的情况下，PQ 编码无法保证 HDR 内容在所有显示器上都能够得到正常显示，因为严格按照亮度解码后，高于设备最大显示能力的高亮信息会被直接截断。考虑到现有设备的兼容性，允许不同显示能力的设备都可以使用一样的信号源，一种更加简单、略显“粗糙”的编码方式应运而生，这就是混合对数编码（Hybrid–Log Gamma，HLG）[35]。HLG 由英国广播公司 BBC 联合日本 NHK（Nippon Hoso Kyokai）电视台共同提出，其思想类似于融合了传统的视觉感知模型，即在暗部使用近似平方根编码，亮部使用对数编码。由于该方法同时兼容 SDR（Standard Dynamic Range），且目前消费级显示器的最大亮度并未达到很高水平，采用相对亮度的 HLG 编码函数同样也被 BT.2100 确定为高动态范围节目分发推荐使用的 Gamma 曲线，在广播电视、视频直播等领域广泛使用。

2.3 HDR 显示和感知

显示设备根据自身显示能力和显示策略来显示内容。显示的内容经过视觉系统、视觉神经以及大脑视觉，最终在大脑中形成对于显示内容的感知。

眼睛作为一个光学系统，可以用傅里叶光学很好地描述。视网膜图像可以表示为输入图像与点扩散函数（Point Spread Function，PSF）的卷积。但是眼睛并不是一个完美的系统，其点扩散函数是一个在较大视角范围内起作用的模糊内核，如图 3 所示。点扩散函数由瞳孔孔径、角膜和晶状体像差决定，而广角部分由散射引起。PSF 的中心部分会导致模糊，而尾部则因为将亮点光源叠加到其余像素上从而引起图像对比度降低。同时，折射率对波长的依赖性会引起色差，长波长和短波长光将在视网膜上不同的深度成像，蓝色通常更加模糊。但是眼睛中的介质对于到达视网膜的波长相当于带通滤波器，能够使透射波长与光感受器的灵敏度更好地匹配，会更多地吸收蓝光。视觉系统也会通过神

经网络进行像差补偿。所以，虽然眼睛的光学质量相对较低，但也是一个高度优化的光学系统 [36]。

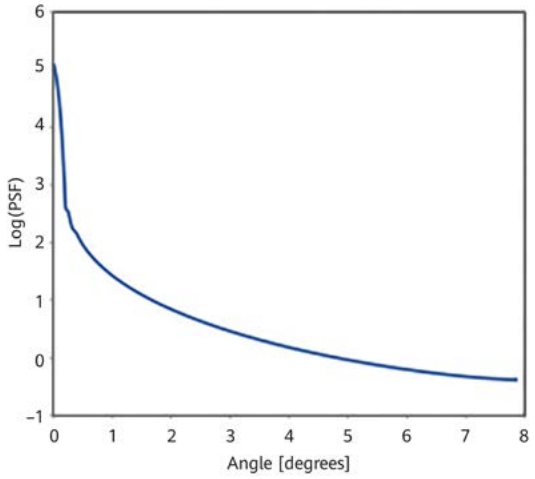


图 3 视觉系统点扩散函数 [36]

视网膜厚为 0.25 mm，由五层组成：外核层包含光感受器的细胞体。外丛状层包含光感受器的突触终末以及双极和水平细胞的树突。内核层包含双极、水平和垂直的细胞体。内丛状层包含双极细胞的轴突终末。神经节细胞层包含所有视网膜神经节细胞，视网膜神经节细胞的轴突形成视神经，通过视盘离开视网膜并与大脑连接 [37]。大约 90% 的视神经纤维终止于外侧膝状体核（Lateral Geniculate Nucleus，LGN），这是两个大脑半球丘脑的一小部分。从视网膜开始的平行通路通过 LGN 到达初级视觉皮层（V1）。

视觉系统对于内容和观看环境都有一定的适应性，主要包含亮度和光的适应和对比度的适应。光适应在视网膜中完成，其效果相当于将亮度除以平均亮度，因此视网膜可以将大范围视觉信号编码为小范围神经信号；对比度适应始于视网膜，在 LGN 和视觉皮层得到加强，其效果相当于将相应像素位置的对比度除以局部对比度 [38]。

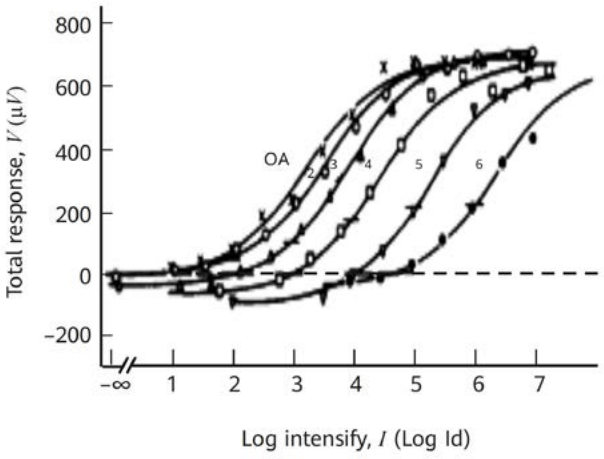


图 4 光感受器响应曲线 [39]

神经系统的响应如图 4 所示，通常表现为 S 型曲线纳卡－拉什顿方程：

$$R = R_{max}\frac{I^n}{I^n + I_s^n} \tag{4}$$

其中，R 表示输入信号 I 的响应，R_{max} 表示最大响应，I_s 表示半饱和系数，n 是固定的幂次方 [38]。n 在视锥细胞的相应中通常为 0.75，在视网膜中通常为 1.0，在 LGN 通常为 1.1，在 V1 通常为 2。一般来说，对于小输入，S 形近似为线性，对于中等输入，类似于幂律，对于高输入，类似于对数。

3 HDR Vivid 技术

3.1 HDR Vivid 端到端系统

对于传统的静态 HDR，创作者创作视频内容时只能考虑当时制作的环境和显示设备上呈现的画质，无法控制作品通过媒体介质传播之后在终端上的播放效果。解决此难题的办法是创作者在所有播放终端上针对各种不同的观看环境手动调试多个版本，但这个方法相对耗时，且不切实际。HDR Vivid 技术帮助创作者在不同终端上复现原始创作意图，其端到端系统框图如图 5 所示。在每一个制作过程中，都涉及 HDR Vivid 技术，标准和生态。HDR Vivid 技术保证终端能还原创作意图，标准保证技术互通互联，而生态则保证对于技术和标准的实现和支持。

为了还原创作意图，需要保证图 5 中 1 和 2 的结果尽可能相似，因此数学建模如图 6 和公式 (5) 所示：

$$\underset{\theta}{\operatorname{argmin}} E\left(\left\|\mathrm{T}(\mathrm{I}) - \mathrm{T}(\tilde{I})\right\|_2^2\right) \tag{5}$$

公式 (5) 是一个最小化求解问题，学术界为了解决这一数学问题，进行了各种尝试。Z. Mai[40] 假设 $\tilde{I}$ 只有亮度损失，没有色度损失，并把亮度损失的处理等效为将输入亮度分为 N 等分，每等分进行不同的亮度处理，利用 KKT 优化理论，可以得到最优解析解。此外，基于不同的视觉实验，多个模型被相继提出，主要有 CSF 模型 [41]、CAM 模型 [42]、Hunt 模　型 [43]、CIVDM（Contrast Invariant Visual Distortion Model）模型 [44] 以及其他人眼感知模型 [45-47] 等。总体来说，人眼感知模型基于人类视觉感知系统，模拟了人脑对于图像的真实响应。因此完善人眼感知模型需要考虑人眼对于颜色刺激与亮度刺激的响应、显示设备亮度、和视频亮度色彩等因素。

HDR Vivid 充分利用了人眼感知模型研究中对于效果影响较大的因素，对 HDR 制作和呈现技术进行了详细的设计。在艺术创作时，HDR Vivid 技术会对内容进行人眼感知建模通过科学分析产生 HDR Vivid 元数据，HDR Vivid

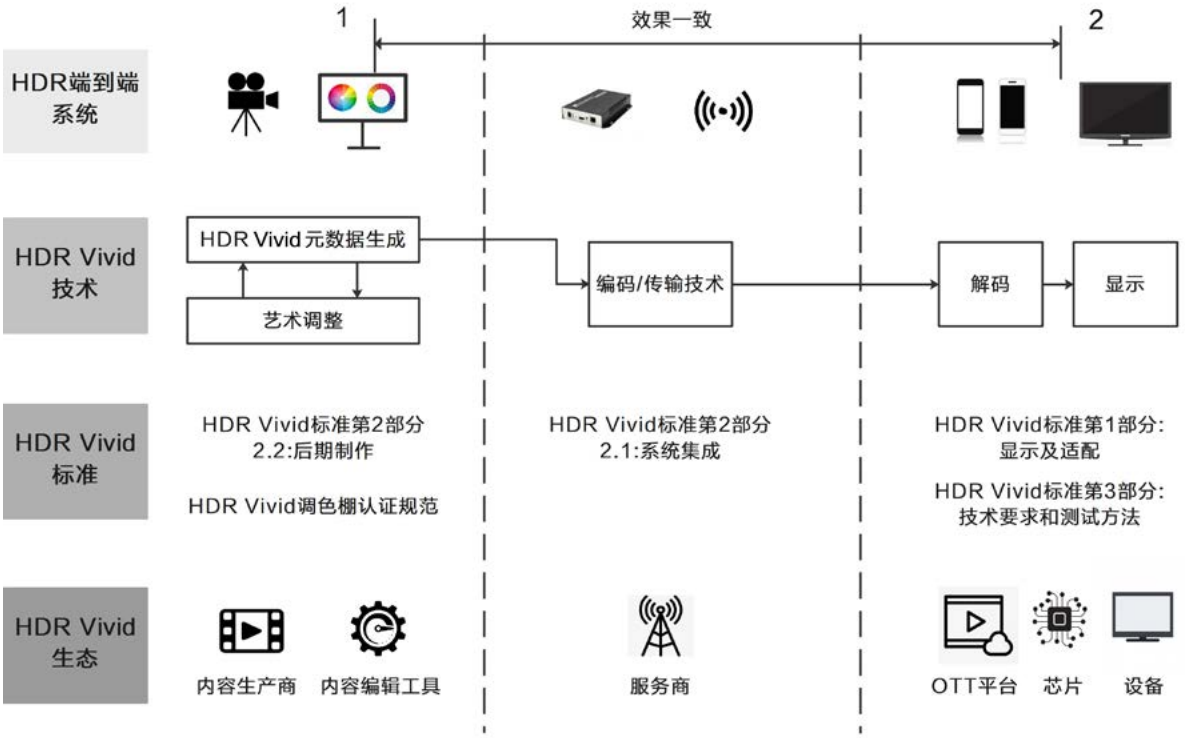


图 5　端到端 HDR Vivid 技术使能艺术系统框图

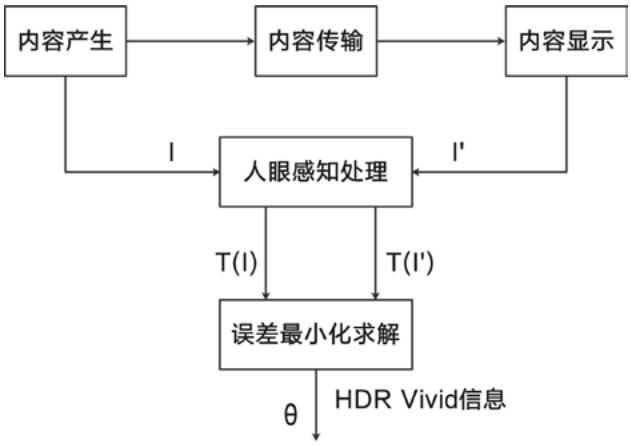


图 6　人眼感知系统最小化求解

元数据包含了亮度曲线信息和色彩曲线信息。经过分发之后，解析这些 HDR Vivid 元数据，在显示过程中利用这些元数据，然后再根据自身显示能力进行创作意图还原，从而呈现正确的艺术效果。HDR Vivid 信息包含了每一个画面的光影信息和色调信息，贯穿了整个 HDR Vivid 端到端系统，主要包括呈现技术和制作技术。

3.2 HDR Vivid 呈现技术

3.2.1 亮度和对比度保持的色调映射

显示设备的显示能力与原始内容的实际动态范围通常不同，因此需要对于图像内容进行色调映射处理，使待显示内容的动态范围与设备显示能力相匹配。在这个过程中，需要使待显示的内容与原始内容的亮度和对比度感知尽量接近。

以往的研究表明，视觉神经系统的响应跟随刺激值的最大值、平均值、方差变化而变化。视觉神经系统的响应通常描述为 S 型的纳卡 − 拉什顿方程形式 [48]：

$$R = R_{max}\frac{L^n}{L^n + Ls} \tag{6}$$

其中，指数 n 在视锥细胞中通常为 0.75，在视网膜中通常为 1.0，在 LGN 通常为 1.1，在 V1 通常为 2。一般来说，对于低亮输入，s 形近似为线性，对于中等输入，类似于幂律，对于高亮输入，类似于对数 [49]。在不同测试环境中可表现为以下形式 [50]：

$$R = R_{max}\frac{(g(L-c))^n}{(g(L-c))^n + 1} \tag{7}$$

其中，g、n、c 会根据测试内容进行变化，L 为亮度的 log 值。对于高亮值，史蒂文模型符合人眼对于亮度的感知，最终形成了立方根形式的亮度感知函数：

$$R = c\,(a * L + b)^{0.33} + d \tag{8}$$

其中 L 是输入，R 是输出，a、b、c、d 是参数。

在色调映射前后保持亮度感知一致，需要使得图像原始亮度值 L_{org} 和图像色调映射后的亮度值 L_{TM} 的亮度感知保持一致：

$$R\,(L_{org}, MaxL_{org}, avgL_{org}) = R\,(L_{TM}, MaxL_{TM}, avgL_{TM}) \tag{9}$$

其中，$MaxL$ 是当前图像的最大值，$avgL$ 是当前图像的平均值。

研究表明，对比度感知的变化受到刺激值空域频率的影响 [51]。通常使用傅里叶变换核来模拟视觉系统中的卷积核，进而研究对比度敏感度函数 CSF。对比度感知灵敏度是对比度感知阈值的倒数，对比度感知阈值表示在均匀背景上最小可感知的对比度。Barten 利用亮度、视角、背景、眼睛瞳孔大小以及光感受器灵敏度等，建立了非常精细的 CSF 模型 [52]。根据该模型，在不同的亮度下，对比度敏感度均存在一个极大值。DICOM、PU、PQ 曲线均采用 CSF 构建了对比度均匀的光电和电光转换曲线，使得不同灰度值之间的量化步长小于人眼恰可感觉失真（Just Noticeable Difference，JND）。为了保持对比度感知均匀，HDR Vivid 在 PQ 域的曲线应满足平均值附近以及主体内容分布亮度范围的曲线接近线性（狭长型的纳卡 – 拉什顿方程基本能满足要求）。结合亮度感知和对比度感知，HDR Vivid 的基本曲线形态为：

$$m\_a \times \left(\frac{m_p \times PQ(L)^{m_n}}{(K1 \times m_p - K2) \times L^{m_n} + K3} \right)^{m\_m} + m\_b \tag{10}$$

以往的研究表明，视觉系统总体神经的响应会随刺激值的最大值、平均值、方差变化而变化。根据史蒂文模型，较亮区域的亮度感知为立方根均匀分布；根据 DeVries–Rose 模型，较暗区域的亮度感知为平方根均匀分布，与纳卡 – 拉什顿方程在不同位置响应的近似结果基本相同。纳卡 – 拉什顿方程对于低亮输入，S 形近似为线性；对于中等输入，类似于幂律；于高亮输入，类似于对数，实际还是比较复杂的。部分研究 [53, 54] 发现，仅使用纳卡 – 拉什顿方程并不能很好地描述这种情况下的亮度感知，如图 7 所示。

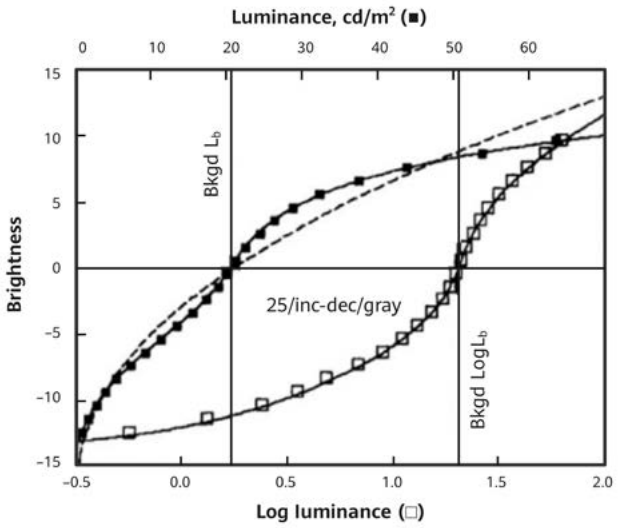


图 7 平均亮度变化后人眼的响应 [53]

由上图可以看出在平均亮度附近，响应仍满足纳卡 – 拉什顿方程，然而在暗区和亮区两端的位置，其响应形态并不相同。这点可以用人眼的结构特征来解释，人眼视网膜包含两种类型的感光细胞：视锥细胞和视杆细胞，两种细胞在亮暗区域有不同响应 [55]。在 10^{-6} 到 10^{-2} nits 之间只有视杆细胞响应，在 10 nits 以上视锥细胞的响应较强。对于中等亮度，随着亮度增加，视锥细胞的响应逐渐增强。

因此，HDR Vivid 标准在暗区使用了样条函数对暗区的色调映射进行了处理，以拟合暗区的响应：

$$a_{n+1}\,(x - T_n)^3 + b_{n+1}\,(x - T_n)^2 + c_{n+1}\,(x - T_n)^1 + d_{n+1} \quad T_n \le x < T_{n+1} \tag{11}$$

其中 x 为输入亮度，a、b、c、d 为样条参数。

最终，HDR Vivid 色调映射曲线如公式 (12)，形态如图 8 所示。

$$Phoenix = \begin{cases} a_1 x + b_1 & 0 \le x < T_1 \\ a_{n+1}(x - T_n)^3 + b_{n+1}(x - T_n)^2 + c_{n+1}(x - T_n)^1 + d_{n+1} & T_n \le x < T_{n+1} \\ m\_a \times \left(\dfrac{m\_p \times L^{m\_n}}{(K1 \times m\_p - K2) \times L^{m\_n} + K3} \right)^{m\_m} + m\_b & T_k \le x < T_{k+1} \end{cases} \tag{12}$$

函数主要分为三部分，第一部分为一次样条曲线，它能够准确反应暗区的动态范围。第二部分是三次样条曲线，它能根据 HDR 特征，在任意位置段输出任意想要的形状，保证曲线能够覆盖所有的 HDR 场景。第三部分是基础曲线部分，该部分能保证画面的主体亮度和对比度；此外，用户通过少量的控制参数可以得到较好的视觉效果。

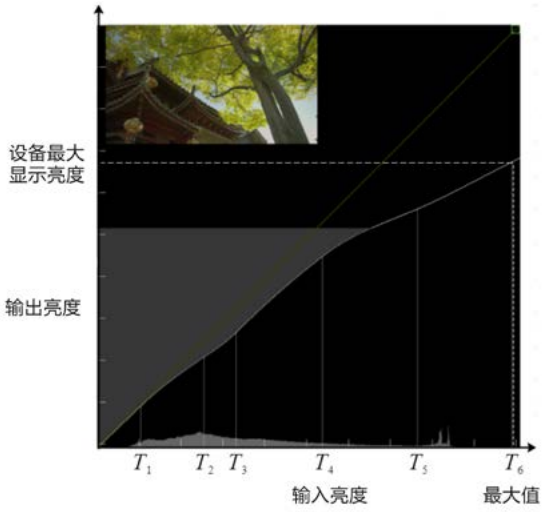


图 8　HDR Vivid 亮度处理曲线

3.2.2 色彩感知保持的饱和度调整

为了保持颜色的准确性，HDR Vivid 色调映射在亮度上获得增益值 gain，但是将增益值以相同比例对于 R、G、B 三个分量同时进行处理：

$$\begin{aligned} Rnew &= R * gain \\ Gnew &= G * gain \\ Bnew &= B * gain \end{aligned} \tag{13}$$

XYZ 与线性 RGB 之间互相转换为线性变换，因此：

$$\begin{aligned} Xnew &= X * gain \\ Ynew &= Y * gain \\ Znew &= Z * gain \end{aligned} \tag{14}$$

在 CIE xy (x, y, Y) 空间中，可得：

$$\begin{aligned} xnew &= \frac{X * gain}{X * gain + Y * gain + Z * gain} = x \\ ynew &= \frac{Y * gain}{X * gain + Y * gain + Z * gain} = y \end{aligned} \tag{15}$$

从而避免亮度映射时发生较大的色彩偏差。

在亮度发生变化的时候，色度也会发生变化。在 Tumblin and Turk 的研究中 [56]，根据亮度的变化对于色彩进行调整：

$$Cnew = \left(\frac{C}{L}\right)^{s} * L_{TM} \tag{16}$$

在之后的研究中 [57, 58] 进一步演化为：

$$Cnew = \left(\frac{L_{TM}}{L}\right) * C \tag{17}$$

在我们的研究中发现，因为 Hunt 效应 [59] 等色貌现象，物体的色貌随着整体的亮度变化发生明显的改变。色度随着亮度的变化而变化，一定程度上影响了色彩的感知。变化的程度和场景本身的色彩亮度特征有关，因此需要对于调整的强度进行控制，综合起来得到：

$$Vnew = \left(\frac{L_{TM}}{L}\right)^{C0} * V \tag{18}$$

$$Unew = \left(\frac{L_{TM}}{L}\right)^{C0} * U \tag{19}$$

较好的补偿了亮度变化后引起的色度的变化。

同时，在实验中发现，与中等亮度区域相比，高亮区域的色度跟随亮度变化的斜率更大，如图 9 所示：

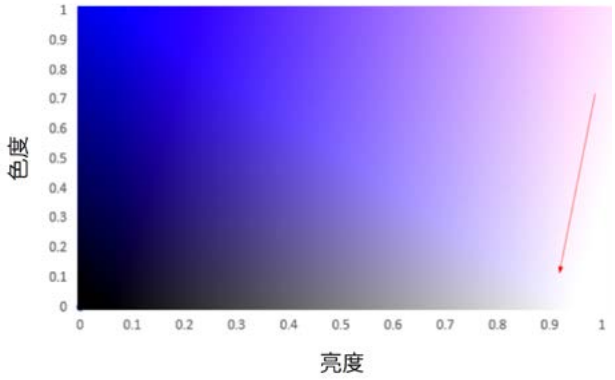


图 9　亮度与色度

因此，对于高亮区域进行特殊处理：

$$S_{ca} = \begin{cases} B - C1 \times SatR \times \left(\dfrac{f_{MAX}[i] - A \times RML}{RML - A \times RML}\right)^{M} & TML < f_{MAX}[i] < RML \\ B - C1 \times SatR & f_{MAX}[i] \geq RML \end{cases} \tag{20}$$

结合亮度色调映射的增益，总体饱和度调整曲线如图 10 所示：

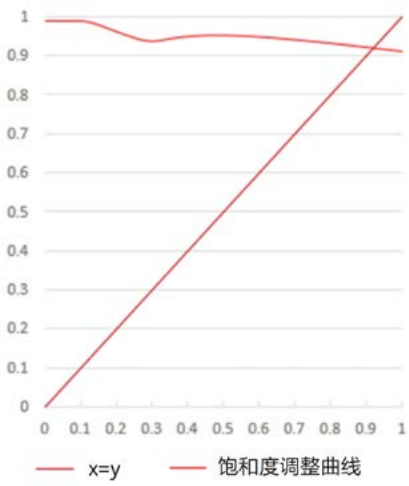


图 10　HDR Vivid 色彩处理过程

3.3 HDR Vivid 制作技术

HDR Vivid 制作技术是整个过程中的关键。利用端到端传递的动态元数据，可以实现不同终端显示效果的一致性，同时允许创作者融入自己喜好的艺术风格。具体的流程如图 11 所示。

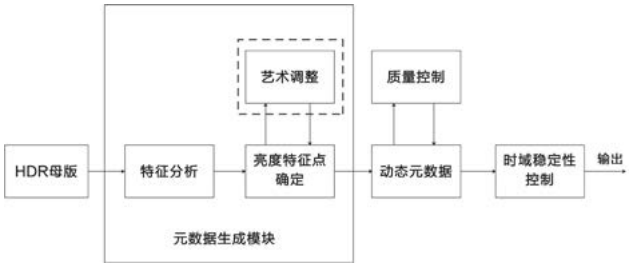


图 11 HDR Vivid 制作流程

首先，对 HDR 母版内容进行特征分析，包括最直接的图像亮度特征、直方图分布等，根据不同场景的特性，通过一系列算法确定映射曲线的关键控制点，也称为亮度特征点。此时创作者也可以根据需求对亮度特征点进行调整。最终，在亮度特征点确定之后，再按照 HDR Vivid 标准将相关信息转换为符合标准形式的动态元数据。以上部分共同构成了图 11 中的元数据生成模块。质量控制模块的目的是为了在无人工干预的情况下，对算法生成的动态元数据进行质量评价与反馈。而时域稳定性控制模块可以有效地防止因引入动态元数据产生视频闪烁现象。下文将分别对这三个模块中的相关技术进行介绍。

3.3.1 元数据生成

图像的亮度最大值、最小值、平均值、方差（变化范围）是图像特征最直接的体现。此外，直方图作为一种二维统计图表，是对数据分布情况的图形表示。这个概念最初由英国统计学家 Karl Pearson 提出 [60]，后来在各领域都得到了广泛的应用。在图像亮度直方图中，横坐标表示像素的亮度值，纵坐标表示落在该亮度值上的像素数目。亮度直方图可以直观体现图像的明暗分布特性，对后续亮度特征点的确定起到关键的作用。

在色调映射以及直方图均衡的研究中，为了能够达到对比度与明暗协调的效果，通常根据直方图将图像划分为多个子区域，分别进行处理。[61] 中的方法通过估计图像中不同部分的亮度和反射率，将图像划分为暗区和亮区分别处理。针对自然界广泛存在的不均匀光照，[62] 中的方法进一步将图像划分为：欠曝（暗区）、曝光正常、过曝（亮区）三个部分，分别进行处理。但是，如果对于正常曝光区域的操作不够精细，会引起正常曝光区域的对比度损失。[63] 中进一步将正常曝光区域划分为：低正常曝光、中正常曝光、高正常曝光区域，以提升正常曝光区域的效果。随着区域划分逐渐精细，图像效果进一步提升，但是过去探讨的大多是过暗或过亮的图片，不符合当前需求。HDR Vivid 元数据生成采用了基于内容和光照精细划分的曲线生成算法，不但根据亮度特征对图像或直方图进行划分，还根据不同内容统计特征的亮度分布对划分进行约束。

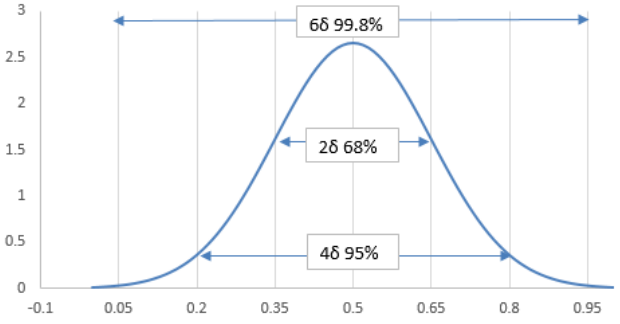


图 12 高斯分布

自然图像的亮度满足高斯分布，根据这个原理，JPEG 使用变换核将图像变换到变换域进行量化，从而实现图像压缩。高斯分布由德国的数学家和天文学家 Moivre 于 1733 年首次提出，德国数学家 Gauss 率先将其应用于天文学研究。该分布的正态曲线呈钟型，如图 12 所示：

总体为 6 段，占总像素分布的 99.8%。根据图像的亮度分布，HDR Vivid 元数据生成将图像整体也分为 6 段，包含极暗区、暗区、低亮区、中亮区（肤色区）、漫反射白区、高光反射区。极暗区代表人眼暗视觉区域，颜色缺失严重，噪声重，易发生死黑。暗区为比极暗区更亮的区域，颜色暗沉但是可见，纹理清晰，而极暗区基本没有纹理。中亮区为肤色等反射率接近 18% 的物体亮度集中的区域。漫反射白区为 SDR 的极亮区，HDR 的次亮区。高光反射区为自然界中反射率较高的材质反射光线造成的，具有强烈亮度感知的区域。HDR Vivid 典型的映射曲线如图 13 所示，由若干段平滑连接的曲线构成，分段曲线的连接点即为需要确定的亮度特征点。从图中可见，映射曲线一共有 6 个亮度特征点，下面将介绍每段曲线的形态、设计原理、亮度特征点的确定方式。

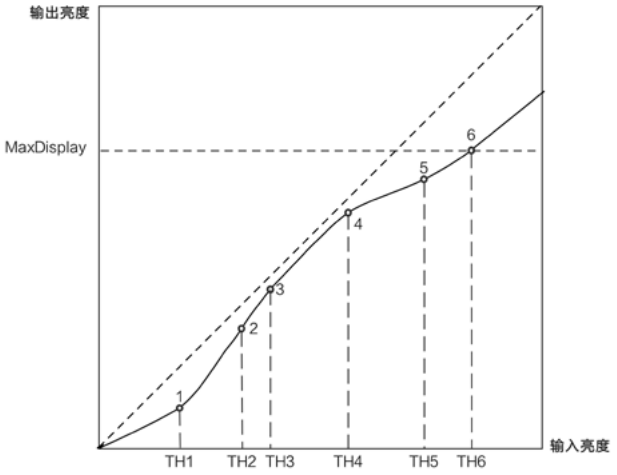


图 13 HDR Vivid 典型映射曲线

亮度特征点 1 的横坐标 TH1 设置为 0.15（经 PQ 编码后的值，若无特殊说明，本节之后出现的所有亮度参考值均为 PQ 编码值），这是基于 Mantiuk 等人的研究 [44]。该研究表明，人眼负责感知色彩的视锥细胞亮度感知下限约为 1 nit（0.15），低于这个下限，无法感知色彩的视杆细胞会占据视觉主导地位。通常，曲线的起始点会设置为原点 (0,0)，起始点和亮度特征点 1 之间采用一次样条进行连接，这样设计的优势在于只需调整斜率就可对该段曲线形态进行控制。根据亮度直方图中位于 TH1 以下的像素分布情况，可对斜率进行动态调整。对该区间内像素分布很少的场景，可以适当减小斜率，换取更好的对比度；若像素分布较多，则需要尽量增加斜率，避免暗部被压缩过多造成细节丢失。斜率确定之后，即可求得亮度特征点 1 的纵坐标值。

亮度特征点 3 与亮度特征点 4 之间采用基础曲线进行连接，其公式如下：

$$F(L)=m\_a\times\left(\frac{m\_p\times L^{m\_n}}{(K1\times m\_p-K2)\times L^{m\_n}+K3}\right)^{m\_m}+m\_b \quad (21)$$

该段曲线决定了映射后图像的整体风格，形态较为灵活。根据实验结果，通常可固定 m_m、m_n、m_b、$K1$、$K2$、$K3$（如 $m\_m=2.4$, $m\_n=K1=K2=K3=1$, $m\_b=0$），根据不同场景特征调整 m_p 与 m_a 的取值，从而达到最优效果。m_p 主要与场景的整体亮度有关，而 m_a 主要与输入内容的亮度最大值及显示设备的最大显示能力有关。因此，可以使用 (Maxsource, MaxDisplay)、(Avgsource, AvgLight) 两点坐标来求解 m_p 与 m_a。其中，Maxsource 为图像特征中的亮度最大值，MaxDisplay 为显示设备的最大显示能力，Avgsource 为图像特征中的亮度平均值，AvgLight 的计算方法如下：

$$AvgLight=\frac{\sum_{i=0}^{N_{frame}-1}q(i)}{N_{frame}} \quad (22)$$

$$q(i)=\begin{cases}MaxDisplay & L[i]\geq MaxDisplay\\ L[i] & 其他\end{cases} \quad (23)$$

N_{frame} 表示一帧图像中包含的像素总数，$L[i]$ 为像素索引值为 i 处的亮度值。值得注意的是，当 Avgsource 较小时（通常阈值下限预设值为 0.25），结合 Mantiuk 的研究成果 [44]，点 (Avgsource, AvgLight) 用 (Perceptual_1nit, 0.15) 代替求解更加符合人眼的视觉特性。此外，还可以通过划分若干个亮度区间，利用直方图均衡化的方法 [64]，求解能够使局部对比度达到最大的两点坐标，进而得到 m_p 与 m_a 的取值。Perceptual_1nit 的计算如公式 (24) 所示：

$$N(Perceptual\_1nit)=(N(0.25)-N(0.15))\times Rate+N(0.15) \quad (24)$$

其中，$N(x)$ 表示亮度直方图中 PQ 编码值小于 x 的像素数量，$Rate$ 为预设比率，通常取 30%。

根据 ITU 的 ITU 的 BT.2408 [65] 标准，人物肤色的亮度范围下限为 0.35，故将亮度特征点 3 的横坐标 TH3 设置为 0.35，可以实现对肤色的有效保护。将 TH3 代入已经确定的基础曲线函数，即可求得亮度特征点 3 的纵坐标值。亮度特征点 2 的横坐标由 [TH1, TH3] 区间内的像素分布情况确定，具体计算方法如公式 (25)、(26) 所示：

$$TH2=\frac{\sum_{i=0}^{N_{frame}-1}q(i)}{Num} \quad (25)$$

$$q(i)=\begin{cases}L[i] & TH1\leq L[i]\leq TH3\\ 0 & 其他\end{cases} \quad (26)$$

N_{frame} 与定义同上，Num 表示亮度直方图中位于 [TH1, TH3] 区间的像素数量。初始亮度特征点 2 位于亮度特征点 1 与亮度特征点 3 的连直线上，已知横坐标可以计算出亮度特征点 2 的纵坐标值。考虑到对比度的提升，还可以根据位于 [TH1, TH2] 区间的像素数量 $Num1$ 与 [TH2, TH3] 区间的像素数量 $Num2$ 之间的关系，对亮度特征点 2 的纵坐标值给予一定的偏移量。为提高灵活性和有效保护暗部细节，亮度特征点 1 与亮度特征点 3 之间选择两条三次样条曲线进行平滑连接。三次样条曲线的具体形式如公式 (27) 所示：

$$F(L)=a_{n+1}(L-T_n)^3+b_{n+1}(L-T_n)^2+c_{n+1}(L-T_n)^1+d_{n+1}$$
$$T_n\leq L<T_{n+1} \quad (27)$$

亮度特征点 6 横坐标 TH6 通常设置为图像特征中的亮度最大值 Maxsource，纵坐标设置为显示设备的最大显示能力 MaxDisplay，代表充分利用显示端设备的显示能力，建立图像最大亮度与显示器最大亮度之间的映射关系。当然，也可以根据亮度直方图在接近 Maxsource 附近的分布情况，适当地调整亮度特征点 6 的横坐标值 TH6。为了计算亮度特征点 4 的横坐标 TH4，可以先将 [TH3, TH6] 平均划分 U 为个区间，预设值为 $U=6$。则 TH4 初始值可以通过公式 (28) 计算：

$$TH4 = TH3 + \left(\frac{TH6 - TH3}{U}\right) \times (U - 2) \tag{28}$$

然后，在 [TH4, TH6] 区间上利用直方图均衡化的思想，计算得到更新后的 TH4 及亮度特征点 5 的横坐标 TH5。由于亮度特征点 4 也在基础曲线上，故将 TH4 代入已经确定的基础曲线函数，即可求得亮度特征点 4 的纵坐标值。与亮度特征点 2 类似，亮度特征点 5 的初始纵坐标值也可以通过将 TH5 代入亮度特征点 4 与亮度特征点 6 的连直线方程中求解。此外，亮度特征点 2 纵坐标值的偏移机制对亮度特征点 5 同样适用。

通过符合 HDR Vivid 动态元数据标准形式的要求将特征点计算结果和求解过程中的相关参数进行转换，得到动态元数据。

艺术调整模块主要依赖于 HDR 内容的前端制作工具。为了适应创作者的习惯，降低工具的使用复杂度，HDR Vivid 的动态元数据调整并非直接针对元数据，而是采用一种“黑盒”机制。创作者在工具界面上看到的是熟悉且常用的艺术调整功能，如暗部细节、中灰亮度、高光等。拖动相应滑块后，由内部算法将其映射到亮度特征点的位置，从而得到更新后的动态元数据。一个简单的界面示例如图 14 所示：

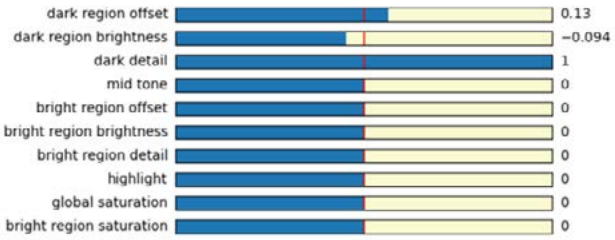


图 14 HDR Vivid 艺术调整参考界面

3.3.2 质量控制

HDR Vivid 质量控制系统如图 15 所示，这是一个具有自适应调整机制的智能控制系统，能使系统智能地输出最优的 HDR Vivid 元数据。同时，用户也可以根据个人的喜好手动调节相关参数，输出特殊风格的艺术效果。

图 15 中的显示模型使用公式 (29)[66] 所示：

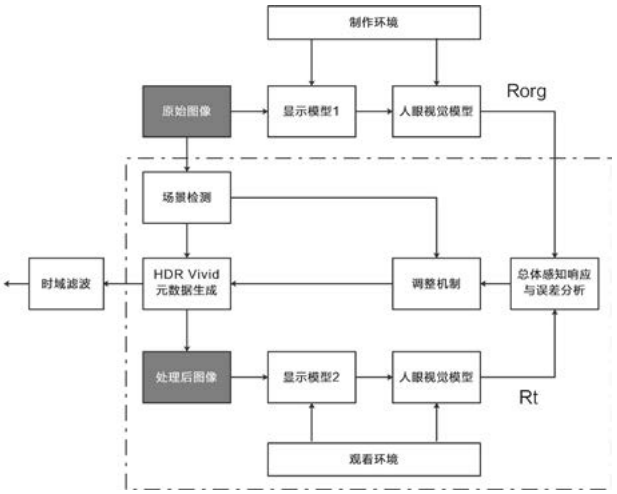


图 15 HDR Vivid 质量控制系统

$$L_d(L') = OETF^{-1}(L') \cdot (L_{max} - L_{black}) + L_{black} + L_{refl} \tag{29}$$

其中，L_d 是显示器的发光亮度，L' 是输入的亮度，一般为解码码流之后电信号数值。$OETF^{-1}(L')$ 是对 L' 信号进行电光转换，转换函数通常有 gamma、HLG 和 PQ 信号的转换。L_{max} 是显示器的峰值亮度，L_{black} 是显示器的黑电平。L_{refl} 是环境光反射到屏幕上的亮度。一般通过公式 (30) 来近似：

$$L_{refl} = \frac{k}{\pi} E_{amb} \tag{30}$$

E_{amb} 是环境光的亮度，k 是显示器反射率。

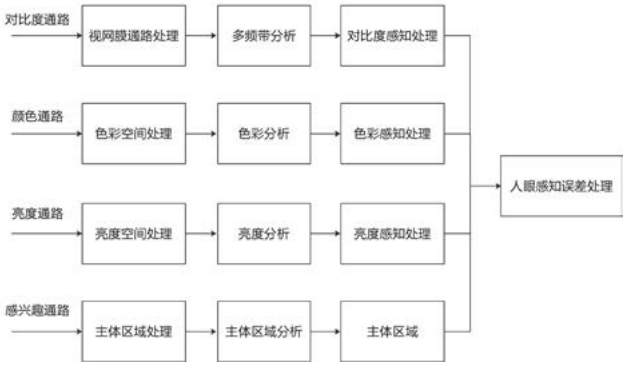


图 16 HDR Vivid 人眼感知通路流程图

人眼感知模型模拟人眼对于视频的真实反映，其流程如图 16 所示，它在 CIVDM 对比度模型上进行了扩展，增加了色彩、亮度、感兴趣通路，充分考虑了 HDR Vivid 内容在传递和显示过程中的损失，以及 HDR Vivid 内容本身的特点。颜色通路用于保持 HDR 广色域的特点，特别是高饱和颜色区域。亮度通路用于复现 HDR 高光细节部分，感兴趣通路使 HDR Vivid 的重点内容不受损失。人眼感知误差处理过程是对对比度通路、颜色通路、亮度通路和感兴趣通路的综合考虑，选择最重要的部分，确保能最大限度还原艺术创作意图。

在对比度通路中，视网膜通路处理过程是首先考虑屏幕大小，距离等因素，同时将输入信号转换为更加容易被人眼感知的 LMSR 信号，在多频带分析过程中将 LMSR 信号进行拉普拉斯频带分解，分解为多个频带信号；对比度感知模型采用相应的 CSF 函数对每个频带信号进行处理，并将多个频带的 CSF 进行叠加处理。在颜色通路中，色彩空间处理过程将信号分解为 lab 和 XYZ 信号，色彩分析过程将 lab 和 XYZ 信号的颜色分量 ab 和 xy 计算色彩饱和度和色调以及在马蹄形色域图中的位置，色彩感知处理过程根据色彩分析中的结果计算色彩偏移的偏移量和位置以及纠正的方向。在亮度通路中，亮度空间处理过程将信号转换为 lab 信号以及 Ycbcr 信号，并将 Y 经过 guilder filter 分类出细节层和亮度层，亮度分析过程将亮度层和 lab 中 l 信号的高光细节进行提取，舍弃一些不包含细节的高光部分。亮度感知处理过程根据亮度层和 l 层计算亮度细节，从而得出损失的高光细节。在感兴趣通路中，主体区域处理过程将图像边缘提取处理，主体区域分析过程检测图像场景，例如是人物，植物还是动物等，根据边缘检测的结果确认主体感兴趣部分，主体区域处理过程计算主体感兴趣部分的损失。

人眼感知误差处理是将上一个模块输出的对比度感知响应，色彩感知响应，亮度感知响应和感兴趣区域感知响应进行非线性组合，从而得出感知的总体误差，例如当图像对比度较强时，重点关注处理后的图像是否有对比度的损失，当图像高光较多且高光细节较多时，模型会重点关注处理后的图像的亮度细节是否有损失，当图像出现高饱和颜色，模型会重点关注是否出现颜色偏离较大的情况。

调整机制是通过判断图像损失主要出现在哪方面，进行相应的调整过程。当某个图像区域的对比度损失较多时，会重点增加该区域的对比度曲线斜率，增强对比度，通过分析我们可以得出在中亮区域的像素点分布较多，如果处理过程使该区域的对比度减弱，则需要调整响应参数，使此图像在该区域的对比度能够与原来一致，如图 17 所示，主体亮度对比度得到了较大提升。

图 17 HDR Vivid 主体对比度调整

图 18 HDR Vivid 高光细节调整

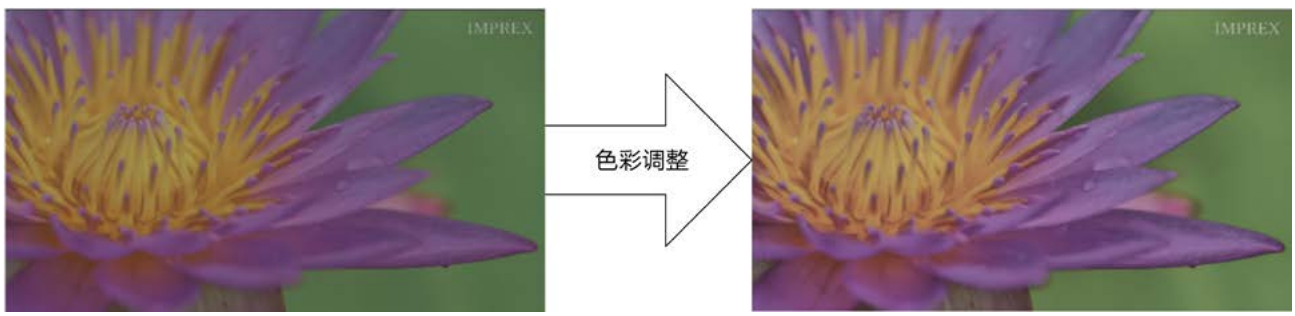


图 19 HDR Vivid 色彩调整

当检测到出现高光细节损失较多时，会调整高亮的处理方式，使高光细节能够更加明显，如图 18 所示，高光雪场景的细节得到了很好的改善。

当出现饱和度损失较大时，会调整颜色饱和度因子，使颜色更加趋近与原始颜色，如图 19 所示，颜色和原始图像更加接近。

人工调节模型是将上面结果进行人工干预，该模型会改变总体感知响应和误差分析模块，使输出结果更加偏重某方面损失。例如可以使对比度损失输出结果更大。此外，人工调节模型可以直接修改调整机制，使输出结果达到艺术创作者或者调色师满意效果。

3.3.3 时域稳定性控制

图像亮度最大值、最小值、平均值、方差（变化范围）的统计值在色调映射中是影响色调映射曲线形态最重要的因素。在显示适配过程中，通常会根据图像内容亮度的方差，把图像内容亮度的平均值映射到屏幕中接近亮度中间值的位置；图像内容亮度的最大值映射到屏幕最高显示亮度；图像内容亮度的最小值映射到屏幕最低显示亮度。但是 HDR 内容包含的亮度范围较大，除了包含漫反射高光，还包含大量的镜面反射高光。高光部分通常产生在光线照射在反射率较高的材质上产生，并且与光线入射角度相关性很强。物体的轻微移动或者形变都可能引起图像亮度的这些统计值发生改变（如图 20 所示），从而引起色调映射曲线的变化，并引起色调映射后图像前后帧差异。在最小可察觉失真的研究中 [67]，人眼可感知到的失真与对比度敏感度函数、眼睛的移动 [68]、局部亮度掩蔽效应以及对比度掩蔽效应等因素均有关系。如果前后帧的差异被人眼感知到，就会发生闪烁现象 [69]。

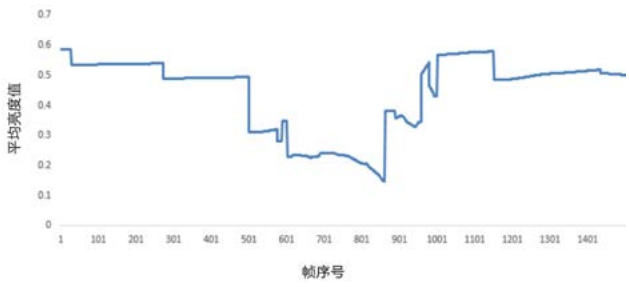


图 20 一个视频片段内图像亮度平均值的时域变化情况

在图像色调映射过程中，如果使用了基于频域的一些局部色调映射处理，在频域分割过程中亮度和细节层的分割、亮度层映射、细节层融合这些环节，都有可能导致局部区域处理结果不够稳定。

为了提升视频的稳定性，需要对于前后帧的色调映射曲线进行限制，消除时域的震荡，同时使其变化引起的前后帧像素值差异不超过人眼可感知的程度。为了消除时域的亮度抖动，保证同一场景中的处理过程稳定，其处理过程如公式 (31) 所示。

$$hdrvivid\_dy\_info\_filter = \frac{\sum_{i=0}^{hdr\_dy\_info\_fifo\_Num-1} w[i] * hdr\_dy\_info\_fifo[i]}{\sum_{i=0}^{hdr\_dy\_info\_fifo\_Num-1} w[i]} \quad (31)$$

其中 $hdrvivid\_dy\_info\_filter$ 是滤波后的 HDR Vivid 信息，$hdr\_dy\_info\_fifo\_Num$ 为 HDR Vivid 信息中的同一场景有效信息数量，$hdr\_dy\_info\_fifo[i]$ 是同一场景中以前 HDR Vivid 信息和当前帧的 HDR Vivid 信息，$w[i]$ 为时域滤波核。

一般来讲，同一个场景中的不同图像处理之后的差异不宜过大，否则会出现抖动或者闪烁的现象。场景检测模块可以进一步检测人物，植物，动物等场景，根据这些场景来指导调整机制以及人眼感知模型关注的通路。例如，在人物场景中需要保证人脸部分在正常范围之内，避免处理后的图像过暗或者过亮，导致和真实人脸肤色存在较大差距。可通过以下公式进一步计算前后帧区域差异值是否可察觉：

$$hdrvivid\_diff = \frac{\sum_{i=0}^{HisNum} His[i] * f_{hdrvivid\_dy\_info\_filter}(L,T)}{\sum_{i=0}^{HisNum} His[i]} \quad (32)$$

其中 His[] 为全局或区域的直方图信息，$f_{hdrvivid\_dy\_info\_filter}(L,T)$ 为根据滤波后参数获取的色调映射值变化是否可被察觉的概率函数。

在视频中，如果前后两帧的内容发生巨大变化，而这两帧的前后连续帧的内容都是相似的，这种情况称为场景切换。如果无法将两个不同场景的片段分开，时域滤波会导致场景中前后帧发生变暗或者变亮的现象，视觉效果类似于长时间闪烁，如图 21 所示。

所以动态元数据提取需要鲁棒性较高的场景切换检测。

在视频中，除了时域亮度抖动和场景切换之外，还有整体图像亮暗构成变化以及渐变（淡入淡出）引起的时域不稳定现象。对于渐变场景，原始的像素亮度并不相同，但是经过色调映射之后的值可能会相同，产生奇怪的视觉效果，如图 22 所示。

通过以下方法可以在一定程度上进行优化：加入时域变化的检测，拉长元数据分析的窗口，将元数据的生成从单帧拉长至多帧，直至变化结束为止。对于整体图像亮暗构成变化的场景也可以采取同样的处理方式。

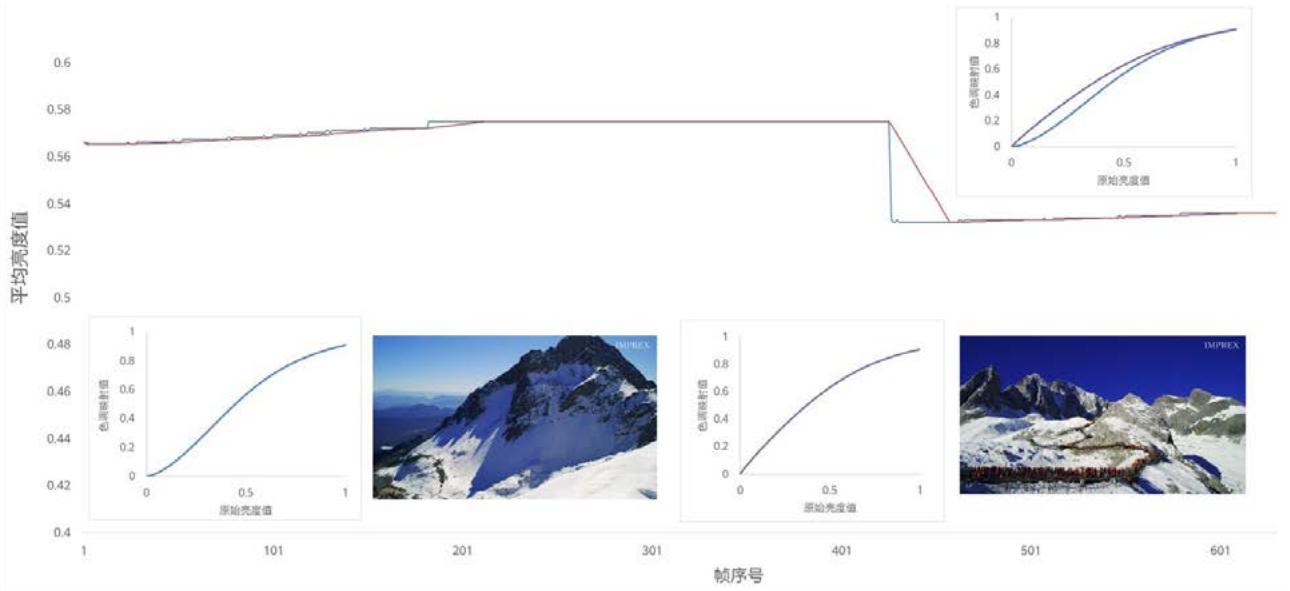


图 21　未检测出场景切换时的时域滤波情况对比

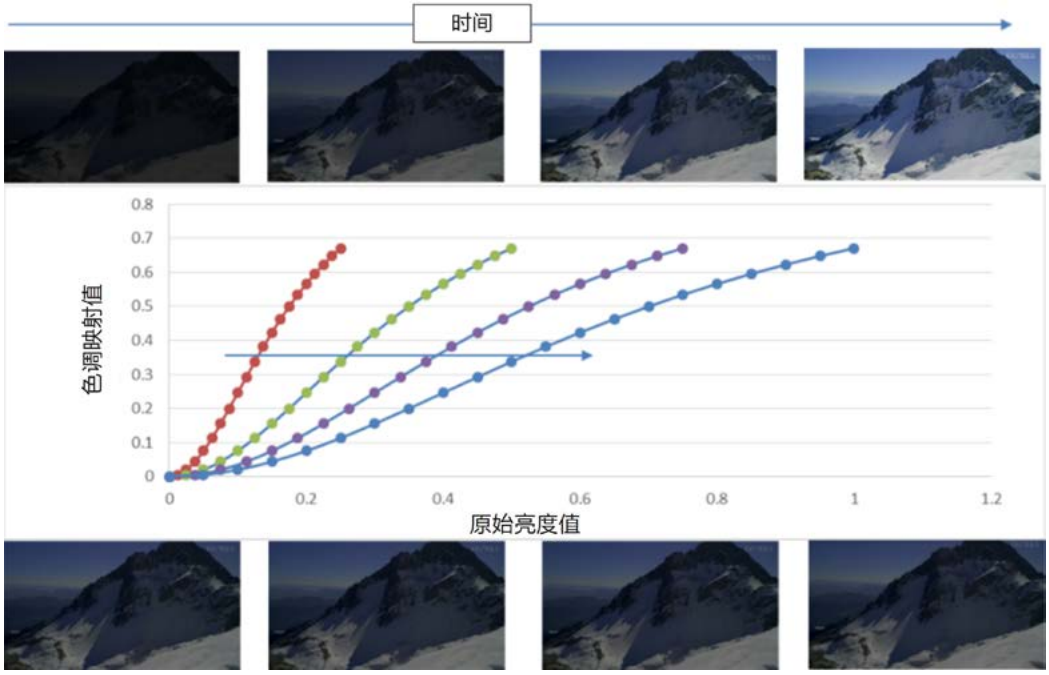


图 22　逐渐变亮场景的色调映射

4 HDR Vivid 结果

4.1 HDR Vivid 曲线特性分析

亮度和对比度保持的色调映射处理需要满足两个基本条件，第一是处理是平滑和连续的：$f'(x) \geq 0$（假设处理函数为 $f(x)$），第二是对 HDR 内容保留亮度最大值的处理能和显示设备最大值 v_{max} 相对应 $f(x_{max}) = v_{max}$，$f(x)$ 可以有许多种形式定义，但是需要足够灵活来应对 HDR 中的不同场景。同时需要具备一定的稳定性，在处理完之后图像特征和处理之前应该尽可能相似。不同帧之间参数的微小变化造成的局部调整不能使画面出现大幅波动。最后，不能有过多的复杂参数，否则设备厂商在实现过程中容易出现错误。下面根据灵活性、稳定性、合理性和易用性这四点对于曲线进行进一步分析。

灵活性表示曲线能够根据 HDR 场景能够出现任意形状，曲线的形态受参数的影响，每一个不同的参数集都对应一条不一样的曲线。用 $\{f_1 \ldots f_n\} \in \emptyset$ 表示所有有效的曲线。

$$\emptyset = \{f(x) \le v_{max};\ G'(L) \ge 0\} \tag{33}$$

我们使用蒙特卡洛采样方法，随机采样 10000 个曲线参数样本点，在这些随机点中取属于 $\emptyset$ 空间的有效数据点。如图 23 所示，HDR Vivid 的有效曲线能够覆盖更多空间，具有较大的灵活性。

$G_t(Y_i, p_1 \ldots p_n)$ 表示亮度处理曲线，其中 $p_1 \ldots p_n$ 是曲线参数，Y_i 是输入亮度，Y_o 是输出亮度。用 $\Delta p_1 \ldots \Delta p_n$ 表示两帧之间曲线参数的变化，确切的说，$p_1 \ldots p_n$ 表示时间 t 的曲线参数，$p_1 + \Delta p_1 \ldots p_n + \Delta p_n$ 是时间 $t+1$，则动态映射的结果差异为：

$$G_{t+1}(Y_i, p_1 + \Delta p_1 \ldots p_n + \Delta p_n) - G_t(Y_i, p_1 \ldots p_n) = \sum_{k=1}^{k=n} \frac{\partial G}{\partial p_k}(Y_i, p_1 \ldots p_n)\Delta p_k \tag{34}$$

通常会在对数域评估结果，则：

$$D(Y) = In(Y_i, p_1 + \Delta p_1 \ldots p_n + \Delta p_n) - In(Y_i, p_1 \ldots p_n)$$

$$= \frac{1}{G_t(Y_i, p_1 \ldots p_n)} \sum_{k=1}^{k=n} \frac{\partial G}{\partial p_k}(Y_i, p_1 \ldots p_n)\Delta p_k \tag{35}$$

因此，稳定性计算如下：

$$S(Y) = 1 - \int_{Ymin}^{Ymax} D(Y)dY \tag{36}$$

如果输入输出归一化到 [0, 1] 区间，$S(Y)$ 的取值最终也落在 [0, 1] 区间。最大值 1 表示稳定性最高，最小值 0 表示稳定性最低。图 24 为亮度每增加 10 nits 计算前后曲线的变化，得到在目标亮度为 500 nits 和 100 nits 下的曲线稳定性，从图中可以看出 HDR Vivid 曲线具有较好的稳定性。

经过亮度处理后的视频应和原始视频的分布特征一致，以保证总体感受的较为自然。因此我们对输入符合高斯分布的随机采样样条，针对我们提出的曲线，S 型曲线和分段曲线分布对这样随机样本点进行处理，得到输出值。

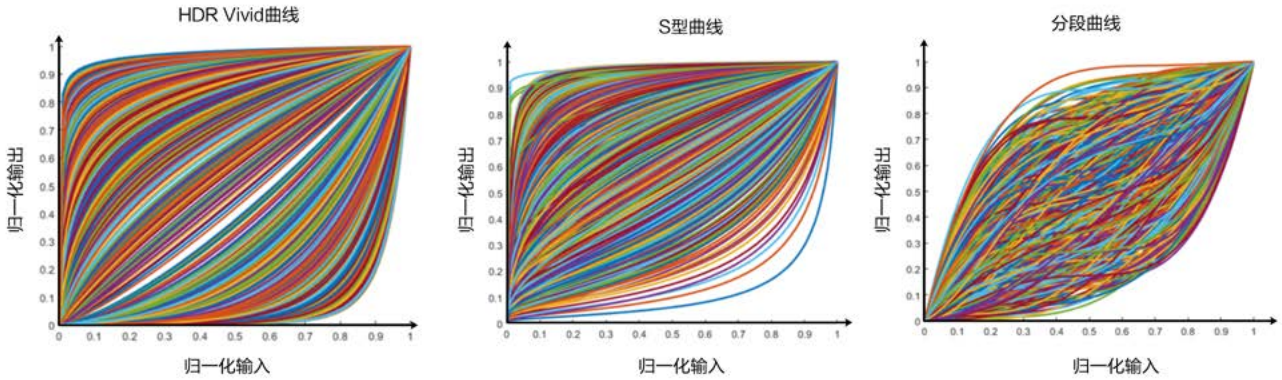


图 23　基于蒙特卡洛方法的有效曲线空间

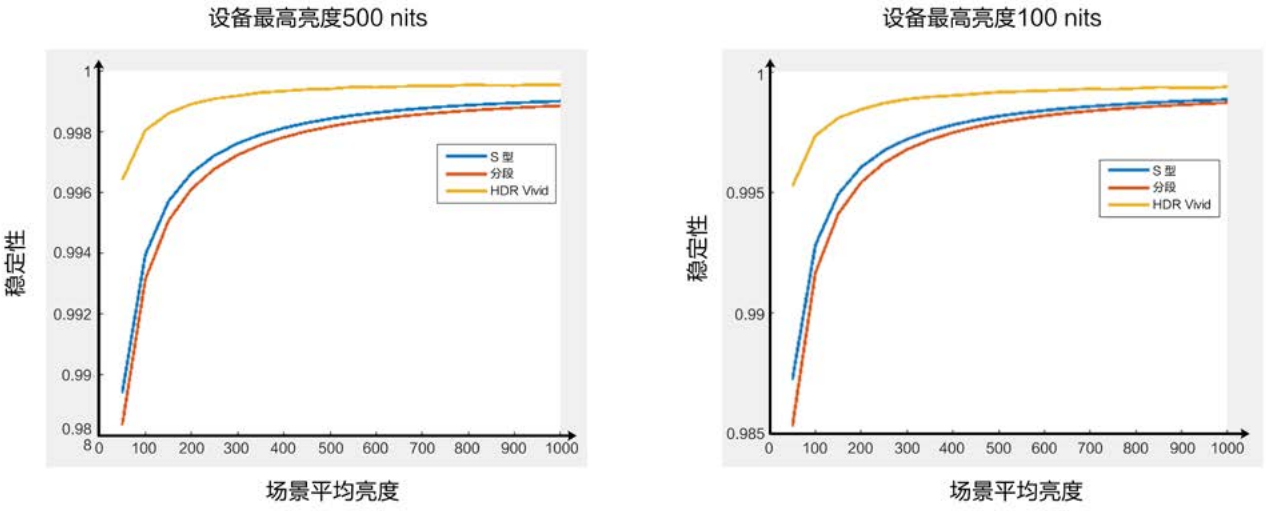


图 24　稳定性和场景平均亮度的关系

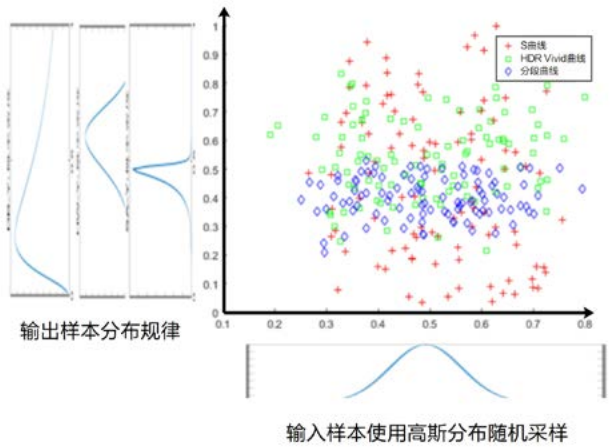


图 25　不同曲线处理的散点分布

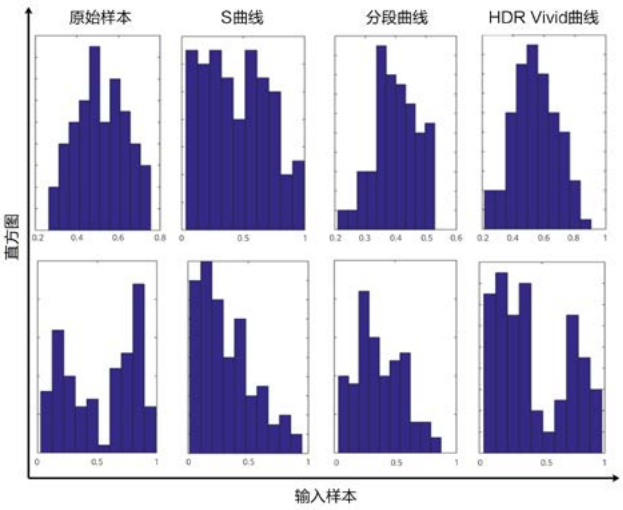


图 26　不同曲线处理的直方图分布

如图 25 所示，输入和输出都统一归一化到 [0, 1.0] 区间。从图 25 中可以看出本文提出的曲线的输出样条点形态和输入样本点形态最相似。S 型曲线修改了输入样本点，会导致总体感受和原始视频相比存在偏差；分段曲线压缩了输入样本点形态分布，会导致部分区域取值概率低。图 26 显示了不同的曲线直方图分布情况，其横坐标为线性值，纵坐标为该线性值对应的直方图，上面一行为单峰高斯分布，下面一行为双峰高斯分布，从图中可以看出 S 型曲线和本文提出的曲线处理之后的直方图分布和原始分布最为相似，S 型曲线对于直方图进行了扩展，而分段曲线则对直方图进行了收紧，因此本文提出的曲线能很好地保持原始视频的特征，从而复现创作者意图。

4.2 HDR Vivid 曲线效果分析

我们使用 HDR 视频在不同的终端显示设备上测试了 HDR Vivid 效果。选取的 HDR 视频峰值亮度覆盖 500–10000 nits 范围之内，最小亮度达到 0.001 以下，色域范围在 BT.2020 区间，终端显示设备包括 SDR 设备和 HDR 设备。SDR 设备显示峰值亮度为 100nit，输出格式为 gamma 2.2，色域范围为 BT.709。HDR 设备峰值亮度为 400 nits 和 1000 nits 两种，输出格式为 PQ，色域范围为 DCI−P3。结合 3.3.1 节 HDR Vivid 元数据生成的流程，可以基于亮度特征点选取情况对最终映射效果的影响进行分析。首先选择整体对比度进行分析。图像映射的主要风格由中间两个亮度特征点的位置确定，即基础曲线部分。其中，m_a 与 m_p 是关键参数。当两个亮度特征点的横坐标固定时，纵坐标之间的间隔大小影响了整幅图像对比度的强弱。如图 27 所示，图 1a 方案因设置的 m_p 过大而 m_a 过小，导致基础曲线部分略显“扁平”，整体对比度不足；

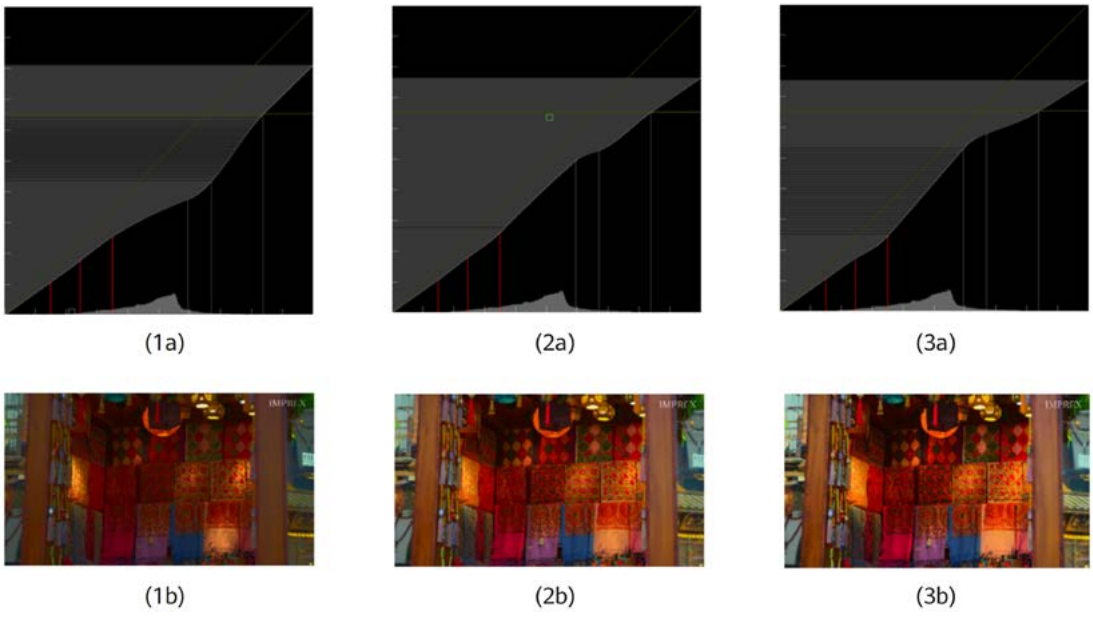


图 27　HDR 设备亮度为 400 nits 时 m_p、m_a 对整体对比度的影响

图 2a 通过减小 m_p 并增大 m_a，对比度得以提升；图 3a 中映射曲线的形态较为合理，可以最大程度地保持对比度。图 1b、2b 与 3b 分别对应三条映射曲线的结果。

暗部亮度主要取决于第一个亮度特征点的位置。根据 3.3.1 节内容，亮度特征点 1 的横坐标由前人研究发表的人眼视锥细胞感知亮度下限来确定，故其纵坐标，即一次样条曲线的斜率确定较为关键。如图 28 所示，由于一次样条曲线覆盖部分存在较多的图像像素，故图 1a 中映射曲线的形态较为合理，而图 2a 中方案将斜率设置过低（亮度特征点 1 的纵坐标过小），导致该部分像素被压缩过多，损失大量细节。图 1b 与 2b 分别对应两条映射曲线的结果与局部放大图。

暗区细节与亮区细节类似，都由两段三次样条中间连接点的纵坐标决定。修改偏移量的大小与方向可以影响暗区 / 亮度的对比度与细节表现，这里以亮区细节为例。如图 29 所示，图 1a 为根据后三个亮度特征点构成的两个区间内亮度直方图反映的像素数量关系，在中间亮度特征点纵坐标初始值的基础之上给予一定量的负向（即向下）偏移，与图 2a 中不做调整相比，亮区的对比度与细节会呈现得更好。反之，若偏移方向选择不当，如图 3a 中进行相同量正向（即向上）偏移，效果反而会比图 2a 更差。图 1b、2b 与 3b 分别对应三条映射曲线的结果与局部放大图。

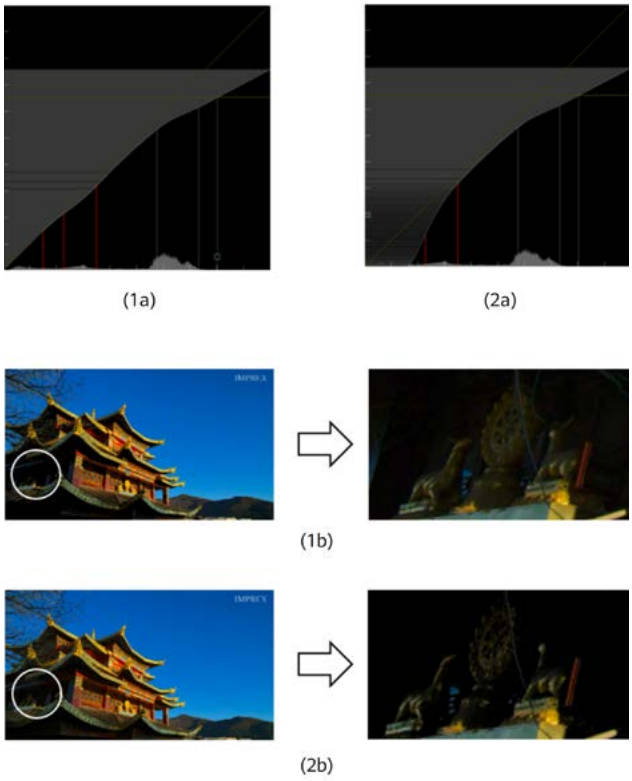


图 28 HDR 设备亮度为 400 nits 时一次样条曲线斜率对暗部亮度的影响

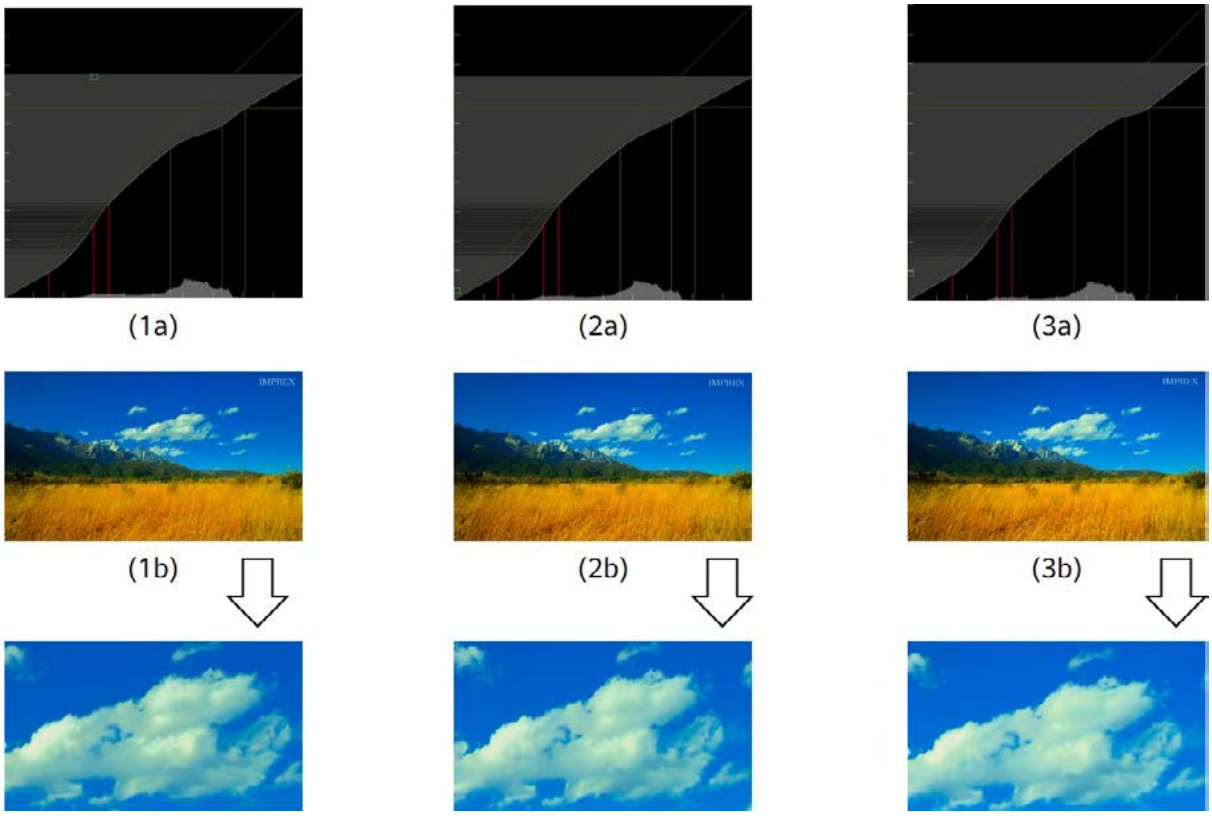


图 29 HDR 设备亮度为 400 nits 时三次样条曲线中间亮度特征点纵坐标对亮区细节的影响

对高光内容的保持也是色调映射中需要考虑的重要因素。结合显示设备的最大显示能力，HDR Vivid 通过选择最后一个亮度特征点的横坐标（纵坐标设置为显示器最大能显示的亮度值）来保持高光内容。如图 30 所示，图 1a 中亮度特征点 6 的横坐标取值为该帧图像的亮度最大值，保证在映射后可以完整保留所有高光信息，而图 2a 中因其取值过小，导致大量像素映射后亮度超出显示设备能力，即会出现“过曝”现象，高光内容会变得不可见。图 1b 与 2b 分别对应两条映射曲线的结果与局部放大图。

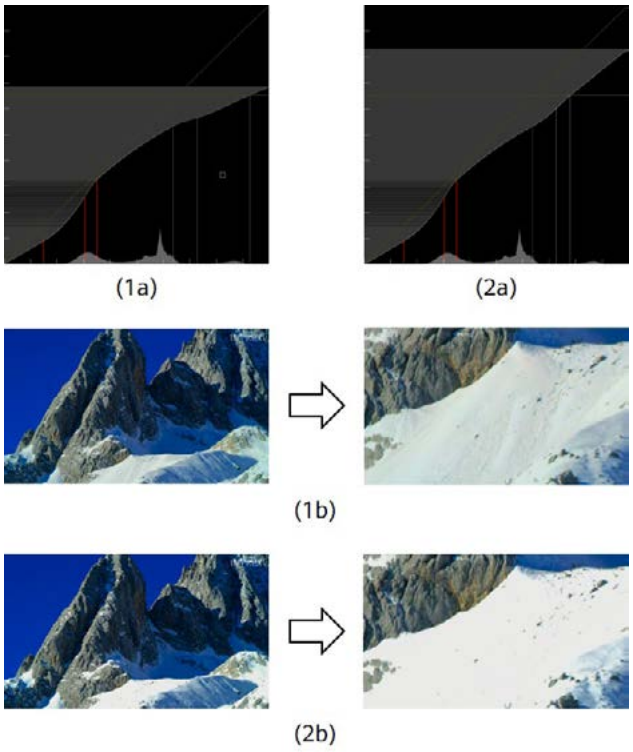


图 30 HDR 设备亮度为 400nit 时最后一个亮度特征点横坐标对高光内容的影响

在色彩感知方面，使用色彩补偿技术之前和之后的色彩具有相近的感知，如图 31 所示，其他方案处理效果的色彩偏差较大，饱和度过高，而经过 HDR Vivid 处理之后的色彩感知和原始图像较相似。

图 31 HDR Vivid 色彩保持比较

4.3 HDR Vivid 端到端效果对比

此外，我们还对比了常用的 5 种色调映射方法 [70–74]，并针对每一个场景将这 5 种方法中最好的主观结果选择出来和 HDR Vivid 作比较，如图 32~34 所示。图 32 中，原始图像具有暗部细节较多和整体平均亮度较高特点，HDR Vivid 在这种场景的处理上，在保持主体亮度的情况下，对于暗部细节也有了较好的复现，这主要得益于 HDR Vivid 的亮度处理过程中暗部细节部分有专门的一次样条函数来保证暗部细节能够得到很好还原。图 33 和图 34 中，原始图像具有高光细节较多和整体平均亮度较高特点，HDR Vivid 在这种场景处理上，对于亮度细节能够得到很好的还原，并且能保证主体对比度不受损失，这主要得益于 HDR Vivid 可以利用倒 S 曲线处理高光，利用正 S 曲线处理主体部分。

5 结语

本文介绍了 HDR Vivid 标准及相应的原理和技术。HDR Vivid 提供了一个端到端的解决方案，打通了制作、传输、显示和感知个链路，能够较好地传递创作者的创作意图，使 HDR Vivid 终端用户能更好的感受创作者的想的表达艺术效果。该方案使创作者能够使用更灵活的手段进行创作，而无需担心终端呈现效果不可控。具体地，HDR Vivid 技术通过结合视觉系统特征，让用户在不同的显示设备都能得到近似的观看体验。HDR Vivid 使电视、电脑、手机、移动平板等显示设备的消费者看到前所未见的丰富色彩与细节、更强的明暗对比、更鲜明的层次和更立体的维度，感受光影的亦幻亦真。

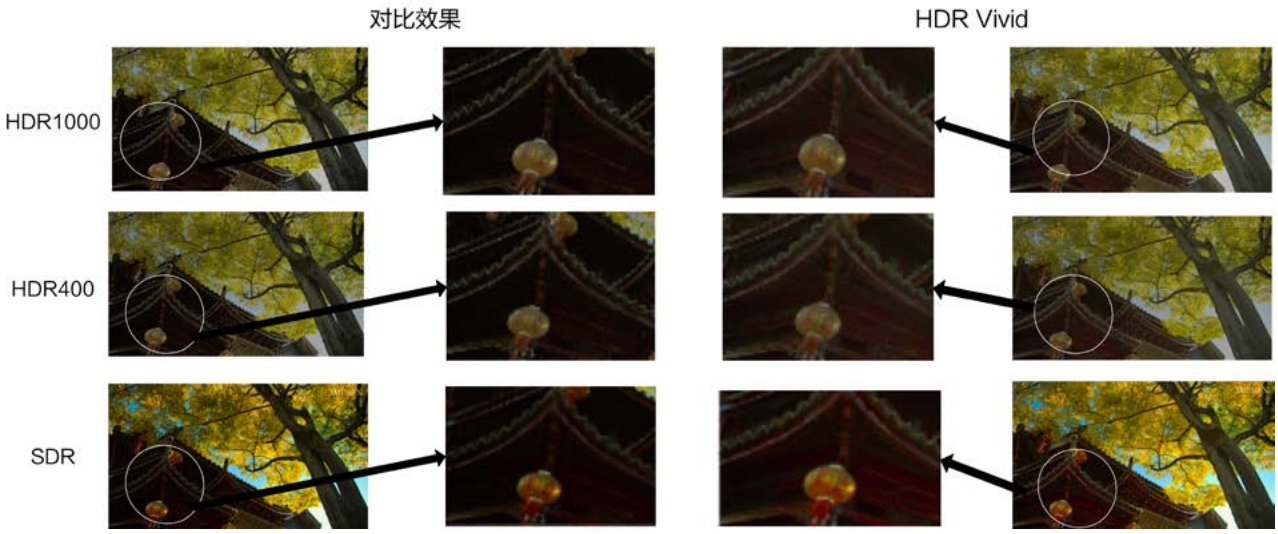


图 32　HDR Vivid 暗区损失比较

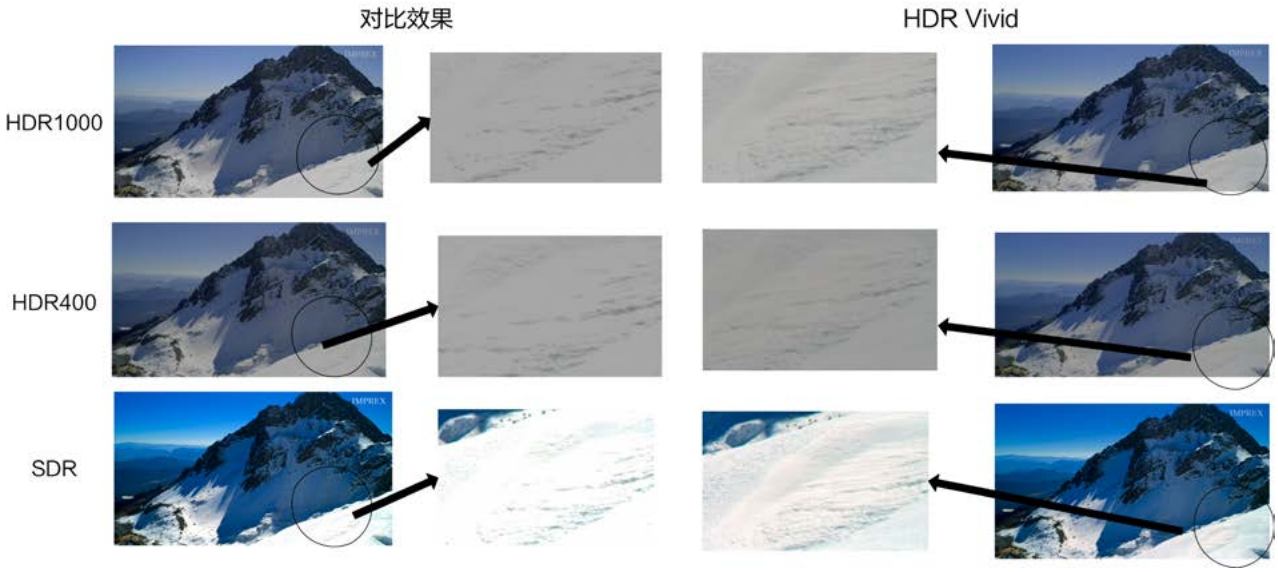


图 33　HDR Vivid 亮区损失比较

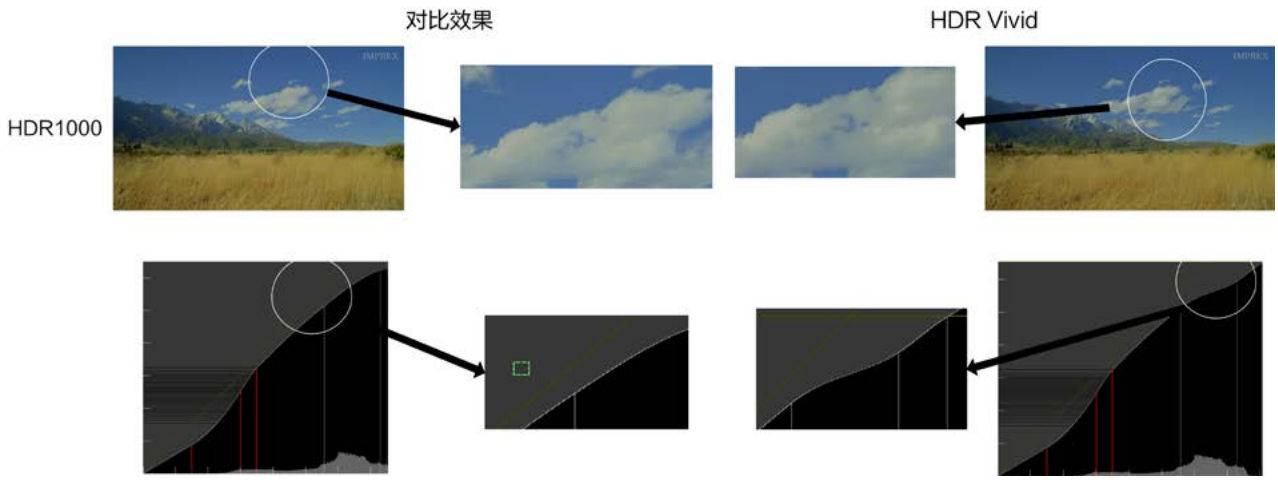


图 34　HDR Vivid 高光细节比较

参考文献

[1] *Photography – digital still cameras – guidelines for reporting pixel-related specifications*. ANSI/I3A IT10.7000–2004.

[2] *Recommendation ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange*.

[3] *Recommendation ITU-R BT.2020-2: Parameter values for ultra-high definition television systems for production and international programme exchange*.

[4] G.J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand (May 25, 2012), "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on Circuits and Systems for Video Technology*, Retrieved May 18, 2013.

[5] Sean Whaley (21 November 2018), "What is frame rate and why is it important to PC gaming?," Retrieved 5 August 2021.

[6] Thomas De Quincey (1854), *De Quincey's works*. James R. Osgood. p. 36. gamut-of-hues 0-1856.

[7] "Dynamic range", *Electropedia*, IEC, archived from the original on 2015-04-26.

[8] *Recommendation ITU-R BT.2100-2: Image parameter values for high dynamic range television for use in production and international programme exchange*.

[9] https://alliance.experienceuhd.com/uhd-premium-features.

[10] L. Meylan, "Tone mapping for high dynamic range images," *EPFL*, Tech. Rep., 2006.

[11] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Trans*. Vis. Comput. Graphics, vol. 11, no. 1, pp. 13–24, Jan. 2005.

[12] M. H. Kim and J. Kautz, "Consistent tone reproduction," in Proc. 10th IASTED Int. Conf. Comput. Graphics Imaging, D. Thalmann, Ed. Anaheim, CA, USA: ACTA, 2008, pp. 152–159.

[13] V. A. Patel, P. Shah, and S. Raman, "A generative adversarial network for tone mapping HDR images," in Proc. 6th Nat. Conf. Comput. Vis., Pattern Recog., Image Process. Graph. (NCVPRIPG), Apr. 2018, pp. 220–231.

[14] Z. Zhang, C. Han, S. He *et al*., "Deep binocular tone mapping," Vis. Comput., vol. 35, nos. 6–8, pp. 997–1011, Jun. 2019.

[15] ST 2086:2014 - *SMPTE Standard - Mastering Display Color Volume Metadata Supporting High Luminance and Wide Color Gamut Images*.

[16] 吴颖 . " 艺术管理与市场 ," 中国传媒大学出版社，2017 年 , 第 106 页。

[17] 郭茂来 , " 视觉艺术概论 ," 人民美术出版社 , 2000 年 , 第 12 页。

[18] [美] 帕特里克・弗兰克著，视觉艺术原理（第八版），上海人民美术出版社，2008.2，第 9 页。

[19] [美] 帕特里克・弗兰克著 , 视觉艺术史，上海人民美术出版社，2008.3。

[20] 阿森纳，H.H.H，彼得 . 卡尔布 . 现代艺术史。

[21] 波塔尔，自然的魅力，1558。

[22] https://www.ilfordphoto.com/amfile/file/download/file/1748/product/735/?___store=ilford_brochure&___from_store=ilford_brochure.

[23] Mees, C.E.K. *The Theory of the Photographic Process. Macmillan*, 1942: 542-551.

[24] https://www.youtube.com/watch?v=idepJM8iHpY&t=54s.

[25] https://link.springer.com/article/10.3758/s13414-012-0281-4.

[26] https://en.wikipedia.org/wiki/Weber%E2%80%93Fechner_law.

[27] HUNT, "R. W. G., light energy and brightness sensation," https://libgen.is/scimag/10.1038%2F1791026a0.

[28] Stevens, S. S. (1957). "On the psychophysical law," *Psychological Review*, 64(3), 153–181.

[29] https://coggle.it/diagram/V-hiK5vlWIhPCd_F/t/weber's-and-devries-rose-law.

[30] P.G. J. Barten, "Physical model for the Contrast Sensitivity of the human eye," Proc. SPIE 1666, 57-72 (1992).

[31] ITU-R BT.2246-5: *The present state of ultra-high definition television*.

[32] PS 3.14-2001. "Part 14: grayscale standard display function," *Digital Imaging and Communications in Medicine (DICOM)*.

[33] Aydın, T. O., Mantiuk, R., and Seidel, H.-P. (2008), "Extending quality metrics to full luminance range images," *Human Vision and Electronic Imaging XIII*.

[34] *Society of Motion Pictures and Television Engineers (SMPTE)*. SMPTE ST 2084:2014 – High Dynamic Range Electro-Optical Transfer Function of Mastering Reference Displays. Technical report, Society of Motion Pictures and Television Engineers (SMPTE), White Plains, NY, August 2014.

[35] "High dynamic range," *European Broadcasting Union*. Retrieved 2015-11-01.

[36] Artal P, "Image formation in the living human eye," *Annual Review of Vision Science*, 2015, 1: 1–17.

[37] Lee BB, Martin PR, and Grünert U, "Retinal connectivity and primate vision," *Progress in Retinal and Eye Research*, 2010, 29(6): 622–39.

[38] Mante V, Frazor RA, Bonin V, Geisler WS, and Carandini M. "Independence of luminance and contrast in natural scenes and in the early visual system," *Nature Neuroscience*, 2005, 8(12): 1690.

[39] Valeton J and Norren DV, "Light adaptation of primate cones: an analysis based on extracellular data," *Vision Research*, 1983, 23(12): 1539–47.

[40] Zicong Mai, "Optimizing a tone curve for backward-compatible high dynamic range image and video compression," *IEEE Transactions on Image Processing*, Institute of Electrical and Electronics Engineers, 2017, ff10.1109/TIP.2017.

[41] DALY, S. 1993, "The visible differences predictor: an algorithm for the assessment of image fidelity." in *Digital Image and Human Vision*, Cambridge, MA: MIT Press, A. Watson, Ed., 179–206.

[42] KUANG, J., JOHNSON, G. M., AND FAIRCHILD, and M. D. 2007, iCAM06: "A refined image appearance model for HDR image rendering," *Journal of Visual Communication and Image Representation*, 18, 5, 406–414.

[43] HUNT, R. 2004, "The reproduction of colour in photography," *Printing and Television: 6th Edition*. John Wiley & Sons.

[44] Rafal Mantiuk, Kil Joong Kim, Allan G.Rempel, and Wolfgang Heidrich, HDR-VDP-2: "A calibrated visual metric for visibility and quality predictions in all luminance conditions," in *ACM Transactions on Graphics*, article no. 40, 2011.

[45] E. Reinhard and K. Devlin, "Dynamic range reduction inspired by photoreceptor physiology," *IEEE Trans*, Vis. Comput. Graphics, vol. 11, no. 1, pp. 13–24, Jan. 2005.

[46] M. H. Kim and J. Kautz, "Consistent tone reproduction," in Proc. 10th IASTED Int. Conf. Comput. Graphics Imaging, D. Thalmann, Ed. Anaheim, CA, USA: ACTA, 2008, pp. 152–159.

[47] L. Lenzen, M and Christmann, "Subjective viewer preference model for automatic HDR down conversion," *Electron. Imaging*, vol. 12, pp. 191–197, 2017.

[48] Mante V, Frazor RA, Bonin V, Geisler WS, and Carandini M, "Independence of luminance and contrast in natural scenes and in the early visual system," *Nature Neuroscience*, 2005, 8(12): 1690.

[49] Billock VA and Tsou BH, "To honor Fechner and obey Stevens: relationships between psychophysical and neural nonlinearities," *Psychological Bulletin*, 2011, 137(1): 1.

[50] Radonjic A, Allred SR, Gilchrist AL, and Brainard DH, "The dynamic range of human lightness perception," *Current Biology*, 2011, 21(22): 1931–6.

[51] Campbell FW and Robson J, "Application of Fourier analysis to the visibility of gratings," *The Journal of Physiology*, 1968, 197(3): 551–66.

[52] Barten PG, "Contrast sensitivity of the human eye and its effects on image quality," vol. 21. Bellingham, WA: Spie Optical Engineering Press; 1999.

[53] Whittle P, "Brightness, discriminability and the

"crispening effect"," *Vision Research*, 1992, 32(8): 1493–507.

[54] Nundy S and Purves D, "A probabilistic explanation of brightness scaling," *Proceedings of the National Academy of Sciences*, 2002, 99(22): 14482–7.

[55] D. C. Hood and M. A. Finkelstein, *Handbook of Perception & Human Performance*, Wiley, 1986.

[56] Jack Tumblin and Greg Turk, "LCIS: a boundary hierarchy for detail-preserving contrast reduction," *Computer Graphics Proceedings*, SIGGRAPH 99, pp. 83-90.

[57] Li. Y., Sharan and L.m Adelson, E., "Compressing and companding high dynamic range images with subband architectures," *ACM trans*. Graph.23(3). 2005.

[58] Lubbe, E., *Colours in the Mind - Colour Systems in Reality*, 2008.

[59] Hunt, "R. W. G., a model of colour vision for predicting colour appearance," *Color Res*. Appl. 7, 95-112. 1982.

[60] Pearson, K. (1895), "Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 186: 343–414.

[61] T. L. Kong and N. A. Mat Isa, "Bi-histogram modification method for nonuniform illuminationand low-contrast images," *Multimedia Tools Appl.*, vol. 77, no. 7, pp. 8955–8978, 2018.

[62] S. Fong Tan and N. Ashidi Mat Isa, "Exposure based multi-histogram equalization contrast enhancement for non-uniform illumination images," *IEEE Access*, vol. 7, pp. 70842–70861, 2019.

[63] N. H. Saad, N. A. M. Isa, and A. A. M. Salih, "Local neighbourhood image properties for exposure region determination method in nonuniform illumination images," *IEEE Access*, vol. 8, pp. 79977–79997, 2020.

[64] Acharya and Ray, *Image Processing: Principles and Applications*, Wiley-Interscience 2005 ISBN 0-471-71998-6.

[65] *Recommendation ITU-R BT2408-3: Guidance for operational practices in HDR television production*.

[66] Rafał Mantiuk, Scott Daly, Louis Kerofsky, "Display adaptive tone mapping," *ACM Transactions on Graphics*, Vol. 27, No. 3, Article 68, Publication date: August 2008.

[67] Jia Y T, Lin W S, and Kassim A A, "Estimating just-noticeable distortion for video," *IEEE Transactions on Circuits and Systems for Video Technology*, 2006, 16(7): 820-829.

[68] Recasens, A., Khosla, A., Vondrick, C., and Torralba, A., *Where are they looking?*, NIPS 2015.

[69] B. Guthier, S. Kopf, M. Eble, and W. Effelsberg, "Flicker reduction in tone mapped high dynamic range video," SPIE, 2011.

[70] R Mantiuk, K Myszkowski, and HP Seidel, "A perceptual framework for contrast processing of high dynamic range images," *ACM Transactions on Applied Perception (TAP)*, 3 (3), 286-308.

[71] S. Ferradans, M. Bertalmio, E. Provenzi, and V. Caselles, "An analysis of visual adaptation and contrast perception for tone mapping," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10): 2002–2012, Oct 2011.

[72] Durand F and Dorsey J, "Fast bilateral filtering for the display of high-dynamic-range images" [J]. *ACM Transactions on Graphics*, 2002, 21(3): 257-266.

[73] F. Durand and J. Dorsey, "Fast bilateral filtering for the display of high-dynamic-range images," *ACM Trans*. Graph, 21(3): 257–266, 2002.

[74] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans*. Graphics, 21(3): 249–256, July 2002.


OpenEuler
面向数字基础设施的开源操作系统
支持多样性计算，满足服务器、云、边缘和嵌入式全场景
主流计算架构100%覆盖
ARM、x86、RISC-V、SW-64、LoongArch 、NPU、GPU、DPU、100+ 整机、300+ 板卡
服务器
云计算
边缘计算
嵌入式
支持多样性计算
信息技术
通信技术
运营技术
CRM
ERP
BSS/OSS
NFV
DCS
SCADA
PLC
……
云原生
大数据
CDN
主流应用场景100%支持
MEC
工业控制
…
覆盖全场景应用
欧拉 DevKit
系统构建和部署
多设备构建工具
兼容/迁移/调优工具
SDK
欧拉 SDK
应用全场景开发
多架构机密计算SDK
应用兼容性评估SDK
完整开发工具链
openEuler公众号
openEuler官网

华为技术有限公司
深圳市龙岗区坂田华为总部办公楼
电话：+86 755 28780808
邮编：518129