大数据技术与系统国家地方联合工程研究中心
服务计算技术与系统教育部重点实验室
集群与网格计算湖北省重点实验室

并行与分布式计算通部

BING XING YU FEN BU SHI JI SUAN TONG XUN

2025年第3期 总第62期 2025年09月

封面人物：冯思乐——步履不停：在荆棘中寻找科研的答案

2024–2025年度实验室各研究方向成果展示

实验室举行2025级新生见面会

第一届先进计算技术与系统论坛暨2025年实验室暑期年会顺利召开

地 址：武汉市华中科技大学东五楼二楼
邮 编：430074
电 话：15387112483
E-mail：yanv@hust.edu.cn
Homepage：http://grid.hust.edu.cn

风采

实验室举行2025级新生见面会

第一届先进计算技术与系统论坛暨2025年实验室暑期年会顺利召开

2024-2025年度实验室各研究方向成果展示

在2024-2025年度，“大数据技术与系统国家地方联合工程研究中心”、“服务计算技术与系统教育部重点实验室”以及“集群与网格计算湖北省重点实验室”和“湖北省大数据安全工程技术研究中心” 全体师生围绕国家重大战略需求和产业创新链，面向云计算、大内存、大数据、区块链等基础技术，持续深化图计算、大模型、算力网络、云原生等前沿方向，取得了一系列具有重要影响力的原创成果，在本期季刊中进行回顾与总结。

分布式系统组在分布式系统软件、区块链、边缘计算及算网融合方向取得重要进展。1）在分布式系统软件方面，提出了基于WASM的高效UDF执行环境WAF，通过数据布局调整前移和共享内存机制降低数据传输开销；实现了有状态Pod 实时迁移方案KubeSPT，通过网络状态同步和热数据懒恢复机制将停机时间减少86%-93%；设计了多粒度分布式镜像仓库策略MIS，通过协同决策存储粒度与镜像分布实现存储空间高效利用与镜像拉取时延降低；2）在区块链技术方面，研发了分支逻辑感知的图式细粒度预执行机制Seer，通过两级分支预测与检查点机制提升预执行准确性；提出了高效可验证的动态查询索引系统FlexIM，结合强化学习优化索引选择及轻量化验证结构；设计了基于乐观路径的图式拜占庭协议Remora，在乐观情况下实现与传统非DAG协议相当的3δ延迟；提出了适应混合网络的多值拜占庭共识协议Pako，通过快速路径与异步二值共识提升网络自适应能力；3）在边缘计算方面，研制了基于不确定性图的分层注意力网络系统，通过不确定性聚焦变换强化关键区域建模，提升复杂场景建筑物分割精度；开发了基于摄像头与IMU传感器的多模态神经网络，有效解决遮挡、动作相似等复杂工况下的工人行为识别问题；提出了基于高排队时延和计算迁移的计算任务保障机制，通过早退出技术动态平衡排队与迁移开销；研发了高效分割式联邦学习系统Hourglass，通过数据并行与特征优化显著降低存储开销并提升模型收敛速度；4）在算网融合方面，设计了弹性流水线训练框架DynPipe，通过动态建模迭代时间与模型陈旧度实现自适应分区；研制了MoE训练加速系统PopFetcher，基于专家热度预测与通信–计算重叠机制降低通信开销；提出了流水线化纵向联邦学习框架BS-VFL，通过交织模型更新与统计信息交换减少通信开销；开发了动态图神经网络训练框架PipeTGL，通过图式拓扑调度优化批次依赖关系；设计了基于深度强化学习的联邦聚合算法FedAA，通过DDPG算法实现聚合权重精细控制与客户端选择；建立了隐私保护LLM推理的隐私–效用权衡理论，为模型部署提供理论指导。

系统软件与体系结构组围绕内存计算、图计算系统、大模型推理系统和数据流架构等方向开展深入研究。1）在内存计算方面，提出了基于亲和感知租约机制的高性能事务化有状态无服务器工作流系统RTSFaaS，通过动态租约分配和转移机制显著降低远程通信开销；研发了面向分离式内存架构的高性能事务系统HDTX，采用快速提交协议和基于RDMA的卸载机制，大幅提升事务处理吞吐量并降低延迟；设计了数据类型感知预取方案DTAP，通过软硬件协同实现高效预取，平均加速达1.37倍；开发基于智能SSD的键值分离存储系统AegonKV，通过垃圾回收卸载全面优化吞吐率、尾延迟与空间使用率；2）在图计算系统方面，研制了基于ReRAM的异步图处理加速器ASGraph，通过依赖感知的子图构造和价值驱动的调度机制减少冗余计算；提出了以数据为中心的自适应基数树硬件加速器DCART，显著降低锁竞争与遍历开销；设计了GPU链驱动时序图计算框架TempGraph，通过时间依赖链转换和捷径构建实现高效并行与快速收敛；3）在大模型推理系统方面，构建了首个大模型智能体效率评测基准AgentRace，实现对主流框架在运行时性能、扩展性、通信开销等方面的系统评估；研发了基于差分缓存的MoE推理加速架构Diff-MoE，通过优先级驱动的专家缓存策略显著提升推理吞吐与内存效率；4）在数据流体系结构方面，提出了面向新型异构数据流架构的亚核级多算子交叉调度系统，通过细粒度调度和数据共享优化资源利用率；设计了PE级自适应任务映射与并行调度策略，依据PE阵列资源和数据流特性实现高效资源分配；研发了基于异质数据流图的多领域融

合执行方法，支持复杂计算过程的精准描述与调度；开发了面向RAG的异构存内计算加速系统HeterRAG，通过局部性感知检索和生成策略实现低延迟与高能效。

大数据组在大数据基础理论、处理技术与分析应用方面成果显著。1）在理论方面，提出了基于全同态加密的低延迟分布式矩阵乘法算法FHE4DMM，通过块中间布局表示和通信计算联合优化，实现最高16.62倍的加速比；研究了面向非独立同分布数据的差分隐私联邦学习收敛性分析与自适应优化机制DPNFL/AdDPNFL，通过截断集中差分隐私技术和无放回部分客户端采样，在隐私–效用间实现更好平衡；提出了基于汉密尔顿蒙特卡洛的分布外样本生成与检测方法HamOS，通过马尔可夫链采样和分布外似然程度估计提升模型在开放环境中的可靠性；2）在处理技术方面，开发了基于性能剖析的大模型算子内并行训练系统，通过代表性程序性能剖析和真实执行代价建模，优化并行策略搜索；研制了基于GPU的椭圆曲线密码学高吞吐量框架gECC，通过批处理架构、内存层级优化及指令级重构，实现数字签名算法4.18-5.56倍的性能提升；设计了基于FPGA的流式数据专用硬件加速器，通过可动态调整的循环逻辑长度结构和高压缩存储策略，实现高吞吐与低延迟处理；3）在分析应用方面，构建了大规模真实网页设计到代码数据集WebCode2M，涵盖256万条样本，为前端自动化开发提供重要基准；提出了Layout-as-Thought的代码生成方法LaTCoder，通过布局感知分块和CoT提示策略，在复杂网页布局保持方面显著提升性能；研发了神经–符号融合的类型推断模型Nester，通过数据流引导和模块化程序执行，将Top-1精确率提升至70.7%；开展了图神经网络公平性攻击与防御研究，提出了节点注入攻击方法NIFA和混合公平性优化框架LibraGNN，实现可控的混合公平性定义；开发了时间序列分布外分类方法ITSR，通过正交性保证不变特征和相关特征的低相似性，提升跨域泛化能力；研制了基于自监督补丁匹配的胶囊内镜图像拼接方法S2P-Matching，通过改进的自监督对比学习和Transformer模型，提升图像匹配精度；提出了面向早期病灶的精细分类框架，通过候选病灶定位和跨图像注意力融合机制，实现高精度医学影像分析。

网络空间安全组在软件与系统安全、人工智能安全、密码学等方向取得多项突破。1）在软件安全方面，提出了反序列化引导的调用图构建工具Flash，通过混合分派策略和基于可控性的反射分析，显著降低Gadget Chain检测的误报率和漏报率；研发了基于因果学习的漏洞检测模型鲁棒性增强框架CausalCode，通过因果数据增强和不变性表示学习提升模型抗对抗攻击能力；开展了自动漏洞修复方法的系统化研究，构建了首个面向C/C++程序的漏洞修复基准数据集Vul4C，对主流AVR工具进行了全面评估；2）在智能体与恶意代码检测方面，设计了基于LLM的恶意NPM包检测器MalPacDetector，实现了动态特征生成与更新机制，以1.3%的误报率和7.5%的漏报率优于现有检测器；提出了持续攻击下的安卓恶意软件对抗样本检测防御方案HagDe，通过对样本施加迭代扰动和损失函数异常增幅分析，有效识别对抗样本；3）在协议与硬件安全方面，研发了基于有限状态机引导的网络协议模糊测试方法，通过状态机建模和引导测试提升协议漏洞发现能力；提出了基于克罗内克积的MPUF建模攻击框架，实现更高精度、更低数据需求和更高效的PUF建模攻击；4）在人工智能安全方面，提出了针对目标检测器的通用对抗样本生成方法NumbOD，通过空间–频率融合攻击实现高效隐蔽的攻击效果；研发了全同态加密软硬件协同加速系统Athena，通过算法优化和专用Kernel设计，在KeySwitch与Bootstrapping操作上实现1.8-4.7倍的性能提升。

在世界百年未有之大变局中，实验室全体师生秉持创新思想，不断提升技术实力。回顾2024-2025年度，实验室各团队坚持深化基础理论研究，突破关键系统技术，推动应用创新落地，形成了系列具有影响力的研究成果，多项技术在实际系统中得到验证与应用。党的二十大报告明确指出，要加快实现高水平科技自立自强，加快建设科技强国。未来，实验室将继续围绕国家急需与行业变革性技术，聚力攻坚，为发展新质生产力、实现高水平科技自立自强贡献更大力量。

王　雄

二〇二五年九月

主　编：金　海
本期执行主编：王　雄
编　委：（按姓氏拼音排序）陈汉华、戴小海、丁晓锋、杜冰倩、段卓辉、耿　聪、顾　琳、何　强、胡胜山、华强胜、黄晨明、黄　航、黄　宏、黄　禹、黄　卓、蒋文斌、李钦宾、李婷婷、李　珍、李　志、廖小飞、刘海坤、刘英书、陆　枫、罗瑞坤、毛伏兵、邵志远、石宣化、陶　莉、万　瑶、王多强、王虹飞、王　雄、文　明、吴　松、吴月明、肖　江、徐　鹏、姚德中、姚鹏程、叶晨成、余　辰、余庚花、袁　斌、张书豪、张　腾、张晓今、张　宇、赵　进、郑　龙、郑　然、邹德清
责任编辑：燕　燕

地　址：武汉市华中科技大学东五楼二楼
邮　编：430074
电　话：15387112483
E-mail：yany@hust.edu.cn
Homepage：http://grid.hust.edu.cn
（此刊仅供内部交流学习）

卷首语

热点

封面人物

专栏

声音

动态

推荐

交流

Marvel携Structera完成全平台CXL互操作验证，巩固生态领导地位

（黄振业 整理）

2025年9月2日，Marvell宣布其内存扩展与近内存计算产品Structera，已通过内部与第三方实验室严格测试，成功兼容领先CPU架构和主流内存平台，这使得Structera成为唯一一个在领先的CPU架构和所有三大内存供应商之间完成互通测试的CXL 2.0产品系列。

随着以数据为中心的应用程序的复杂性增加，内存在性能方面发挥着越来越大的作用。而为了满足超大规模企业对跨不同内存和CPU技术无缝集成的需求，兼容性成为衡量产品优劣的重要指标。

本次测试主要包括以下平台：CPU侧覆盖Intel、AMD等主流芯片厂商；内存侧则与三星、SK海力士、美光三大厂商的DDR4和DDR5内存无缝兼容。为了支持不同的系统架构，Structera IP可用于集成到定制芯片设计中。该IP产品依赖设计的灵活性，以及利用Marvell开发的成熟CXL 生态系统支持，使其能够支持SoC、加速器等多种平台。基于此，客户能够将Marvell的CXL技术直接嵌入到他们的芯片中，从而优化特定工作负载性能、能效和部署成本。

Structera产品线包括两个CXL设备系列，各自侧重于不同应用领域。Structera A CXL近内存加速器集成了16个Arm® Neoverse® V2内核和多个内存通道，可满足深度学习推荐模型（DLRM）和机器学习等高带宽内存应用的需求。Structera X CXL内存扩展控制器能够将TB 级内存添加到通用服务器中，并满足内存数据库等大容量内存应用的需求。Structera CXL设备系列是业界首款支持四个存储通道、集成内联LZ4压缩并使用5nm制造工艺的设备系列。

（参考链接：https://investor.marvell.com/2025-09-02-Marvell-Extends-CXL-Ecosystem-Leadership-with-Structera-Interoperability-Across-All-Major-Memory-and-CPU-Platforms）

Spectrum-XGS 以太网技术

（黄一凡 整理）

近日，NVIDIA在美国斯坦福大学举办的Hot Chips 2025大会上进行许多重要议程。其中，最重要消息之一是通过以太网络为基础串联多座数据中心的Spectrum-XGS 以太网互联技术。

随着 AI 需求的激增，单个设施内的数据中心功率和容量已达到极限。现有的商用以太网网络基础设施因高延迟、高抖动及性能的不可预测而无法满足需求。而Spectrum-XGS技术能打破数据中心的距离限制，将不同地理位置的运算节点整合为单一超大型算节，除了能提高整体运算性能之外，也可以扩展可用内存容量，以容纳量体更大、参数更多的数据集或AI模型。从而做到跨数据中心串联，达到远程分布式运算的效果。

NVIDIA将这种技术概念称为Scale-Around。相较于提升单一运算节点性能的Scale-Up，或是串联运算多个节点以提高整体性能的Scale-Out，Scale-Around的概念更着重于串联为于不同数据中心的远程运算节点，提供超低延迟、高带宽的数据交换渠道，能够编排多个数据中心的GPU对GPU之间的庞大数据集的运算，以满足超大量体AI运算需求。

从理论上来看，Scale-Around概念由于数据传输距离较远，与Scale-Out相比一定会有延迟

较高的缺点，但是它的优点则是能打破距离的隔阂，并且因多个数据中心为于不同地区，所以能舒缓单一电网供电的压力，并且可以灵活调度不同数据中心进行集成运算，可以视各数据中心的负载情况、时区进行优化资源调度，最终具有更高的使用弹性。

NVIDIA创办人暨首席执行官黄仁勋表示，AI工业革命正在发生中，而规模更大的AI工厂是必要的基础建设，我们通过Spectrum-XGS 以太网技术将位于不同成市、国家、大陆的数据中心汇集为超大量体的超级AI工厂。

（参考链接：https://blogs.nvidia.cn/blog/nvidia-introduces-spectrum-xgs-ethernet-to-connect-distributed-data-centers-into-giga-scale-ai-super-factories/）

Agentic AI（自主代理式人工智能）

（徐语骋 整理）

随着大语言模型的不断发展，人工智能正逐步从“感知—生成”阶段迈向“自主代理”阶段。作为一种结合大语言模型推理能力与任务执行机制的全新技术范式，Agentic AI 被认为是继生成式 AI 之后的又一重大突破。其运行方式类似于“AI 代理”，不仅能回答问题，还能分解任务、调用工具、执行操作，甚至在复杂环境中自主规划步骤，完成完整的工作流。

与传统的 AI 助手不同，Agentic AI 的核心优势在于它的 “模拟人类工作方式”。例如，OpenAI 发布的“Operator”尝试让 AI 直接操作计算机界面，Anthropic 的“Computer Use”功能则支持 AI 在软件界面中执行任务。这意味着 AI 不再只是信息提供者，而是能直接完成执行环节，成为真正的“虚拟同事”。

在应用案例方面，Agentic AI 已展现出广阔潜力。企业可以利用它完成供应链调度、财务合规审查、客服自动回复等任务；科技公司则在尝试将其与 ERP、CRM 等企业软件融合，推动业务流程自动化。与此同时，部分初创公司也专注于开发面向垂直行业的代理系统，例如制造业中的质量检测代理、金融领域的自动合规代理等。

然而，围绕 Agentic AI 的质疑也同样存在。一方面，来自大语言模型能力的限制，当前 LLM 的推理稳定性和泛化性尚不足以支持完全自主的复杂任务执行。另一方面，Agentic Ai面临一些产业化挑战，包括算力成本、系统集成难度、安全与合规风险。例如，多步推理带来的资源消耗和可能的错误执行，都需要企业投入额外的监管与审计机制。

总体而言，Agentic AI 代表了人工智能从“知识生成”到“自主执行”的演进路径。尽管目前仍处在探索和试点阶段，但其“数字员工”化的潜力已经引起工业界的高度关注。未来随着技术的进一步成熟和安全规范的完善，Agentic AI 有望成为推动企业智能化转型的关键力量，为智能工作模式带来深远影响。

（参考链接：https://www.theverge.com/the-stepback-newsletter/767376/ai-agents-jarvis-what-can-they-do）

分布式计算与大模型研究

（王焜尧 整理）

大模型研究已成学术界主流热点，人工智能正从“大模型竞争”转向“分布式智能协同”。随着模型参数迈向万亿级，单一节点已难承载存储、计算与通信压力，分布式计算成为研发基石。学界聚焦效率、可用性与安全三

大挑战，推动训练与推理技术跃升。

CVPR 2025 Oral论文《UniAP: Unifying Inter- and Intra-Layer Automatic Parallelism by Mixed Integer Quadratic Programming》中研发了高效能分布式训练算法 UniAP，该高效能分布式训练算法会比低效能分布式训练算法成本低，最高可能会节省数倍甚至数十倍以上的算力成本。UniAP 是首个能实现层内并行策略（张量并行等）和层间并行策略（流水线并行等）联合优化的工作。给定模型和硬件平台，UniAP 能够通过自动搜索找到高效能的分布式训练方案，既解决了效率和成本问题（实验中，比已有的最好方法最高快 3.8 倍，比不采用并行策略优化的算法最高快 9 倍），也解决了很多人在大模型分布式训练时因为超参数设置不合理而无法成功运行训练过程的问题，即易用性问题。此外，还实现了 UniAP 跟国产 AI 计算卡的适配。相关工作为大模型训练的降本增效提供了核心技术、（国产）平台和框架。

中国电信研究院于今年发表的文章《支持大模型分布式训练的光传输网络技术探索》，着眼于训练大规模模型所需的计算资源和时间呈现爆炸式增长，智算中心集群规模向着十万卡级甚至百万卡级加速演进的现状，并针对适用于多数据中心分布式训练的光传输网络技术进行分析与展望。该研究分析了跨数据中心分布式训练对光网络的需求，探讨800G等超高速传输技术、组网成本与误码容忍问题，为底层网络提供技术展望。

此外，北京大学崔斌团队研发高效分布式框架，针对负载不均设计细粒度模型切分与并行策略搜索算法，依托昇腾算力统一接口管理任务，精算算力、内存、通信后智能拆解模型，按模块差异分配策略，较3D并行等模板方案再提效15%。团队还利用昇腾高速总线优化通信分组，实现计算通信重叠，最大化带宽利用率与流水线效率，相关成果已在NeurIPS、ICLR、AAAI发表三篇论文，为自主算力应用树立范式。

（参考链接：https://arxiv.org/abs/2307.16375）

大语言模型的智能体推理框架研究进展与应用全景

（陈　伟　整理）

近年来，随着大语言模型（LLMs）内在推理能力的突破性进展，LLM驱动的智能体系统已成为学术界和产业界共同关注的焦点。

当前研究的核心挑战在于如何系统化理解不同推理框架的运作机制。现有工作主要沿着单智能体方法、工具增强方法和多智能体系统三个维度展开创新。单智能体方法通过提示工程（如角色扮演、环境模拟）和自优化机制（如反思式学习、迭代优化）提升个体认知能力。

在科学发现领域，智能体系统正重塑研究范式。数学证明方向，MA-LoT通过“证明生成–纠错”双智能体协作，将Lean4验证器的定理证明效率提升40%；生物化学领域，PharmAgents 构建虚拟药物研发团队，将靶点发现到临床前评估的全流程缩短至传统方法的1/3时间。

医疗健康场景凸显了框架设计的严谨性需求。诊断辅助系统如KG4Diagnosis通过知识图谱增强的分层多智能体架构，将罕见病诊断准确率提高至92%；临床管理领域，TAO框架采用三级安全代理实现端到端监管，在MedSafetyBench测试中误诊率低于0.5%。

软件工程领域见证了从代码生成到全生命

周期管理的跨越。测试驱动开发框架AgentCoder在HumanEval基准上达到96.3%的通过率，其创新在于将测试用例生成与执行纳入迭代循环；程序修复系统AutoCodeRover-v2结合频谱缺陷定位和上下文检索，在SWE-bench-lite基准上实现30.67%的问题解决率。全周期开发平台MetaGPT通过标准化操作流程(SOPs)协调需求分析、系统设计等阶段，在1.98B参数规模下仍保持87.7%的代码生成准确率。

社会经济仿真开辟了复杂系统研究新路径。社交模拟平台SocioVerse整合1000万真实用户画像，成功复现信息茧房形成过程；金融领域FinRobot实现全周期分析自动化，其多模态决策模块在波动市场中的年化收益超越基准15%。这些系统通过双重交互机制（个体–环境/个体–个体）涌现出群体动力学特征，为政策制定提供量化依据。

未来研究将聚焦动态推理框架、跨模态具身交互等方向。仿真平台如OASIS已支持百万级智能体并行交互，而生物化学领域ProtAgents通过蛋白质设计–模拟–优化的闭环系统，在3周内完成传统团队半年的研究任务。随着GPT-4o、Claude-3.5等多模态模型的出现，GUI操作、战略推理等新兴场景正在形成下一个研究前沿。学术界需建立更完善的评估体系，如Scientist-Bench、SurveyBench等领域专用基准，以推动研究向可复现、可比较的方向发展。

（参考链接：https://arxiv.org/pdf/2508.17692）

Native Sparse Attention：基于硬件优化的原生可训练稀疏注意力机制

（彭　彬　整理）

其提出了一种名为NSA（Native Sparse Attention）的新型稀疏注意力机制，旨在解决长上下文建模中的计算效率问题。传统全注意力机制在处理长序列时计算成本极高，而稀疏注意力通过选择性计算关键的查询–键对，显著减少计算开销，同时保持模型性能。NSA结合算法创新与硬件优化，提出动态分层稀疏策略，通过压缩粗粒度token、选择保留关键token以及滑动窗口捕捉局部信息，兼顾全局上下文感知与局部精度。将压缩粗粒度令牌用于全局上下文扫描，选择关键令牌用于精确的局部信息检索，滑动窗口处理局部上下文，有效地平衡了模型能力和计算效率。同时，其硬件对齐设计优化了块状稀疏注意力，使其适配现代硬件，实现算术强度平衡的算法设计，并减少内存访问瓶颈，同时支持端到端训练，降低预训练计算成本。实验结果显示，在通用基准测试、长上下文任务和指令推理任务中，使用NSA预训练的模型表现出色。在训练和推理阶段，NSA均展现出显著的速度优势。训练时，随着上下文长度增加，速度提升越发明显，在64k上下文长度时，前向和后向传播速度分别最高可达9.0倍和6.0倍。解码时，NSA内存访问效率高，随着解码长度增加，延迟显著降低，在64k上下文长度时，速度提升最高可达11.6倍。

（参考链接：https://arxiv.org/pdf/2502.11089）

首个“AI勒索软件”出现：恶意行为代码由大模型动态生成

（朱文哲　整理）

随着人工智能技术的快速发展，大语言模型（LLMs）正被越来越多地滥用到网络犯罪活动中。近期，思科Talos和ESET研究团队分别披

露了相关研究成果：一方面，恶意大模型正在暗网兴起并被黑客积极利用；另一方面，首个结合本地大模型的勒索软件“PromptLock”已经出现，显示恶意软件正在向动态化、智能化方向演进。

PromptLock是首个利用本地大模型生成恶意组件的勒索软件样本，由Golang编写，目前发现有Windows和Linux两个变种。其最大特点在于动态代码生成：该软件并非预置恶意逻辑，而是通过硬编码的提示词调用本地运行的gpt-oss:20b模型（通过Ollama API接口），让模型充当“Lua代码生成器”，在受害者设备上即时生成恶意脚本。

这些提示词驱动模型生成的Lua脚本涵盖多个环节：系统枚举（收集操作系统信息、用户名、主机名、工作目录），文件系统扫描（查找目标文件并分析内容，尤其是涉及个人隐私或敏感数据的文件），数据窃取与加密（利用Lua脚本完成数据外泄和后续加密操作）。值得注意的是，Lua语言轻量且跨平台，使得生成的脚本能够在Windows、Linux甚至macOS上运行，大幅拓展了攻击范围。加密环节则采用轻量级的SPECK 128位分组加密算法，进一步提高了灵活性和效率。

目前，PromptLock仍处于概念验证（PoC）阶段，部分功能尚未完善。例如其代码中定义了数据销毁函数，但并未真正实现。此外，研究人员还发现一些异常信息，如提示词中出现的比特币地址，看似与比特币匿名创始人中本聪相关，可能是混淆或占位符。尽管如此，ESET认为该发现揭示了未来趋势：恶意软件将不再依赖静态逻辑，而是通过本地AI模型在攻击现场动态生成恶意代码，从而更难被传统检测手段发现。

（参考链接：https://www.secrss.com/articles/80901）

大模型驱动的网络攻击演进与风险

（邱士煜 整理）

今年以来，多项研究陆续揭示大模型在网络攻击中的不同应用。2025年3月份，赛门铁克曾展示OpenAI Operator Agent在提示调整后可以自动完成一次完整的钓鱼流程，包括识别目标、推测邮箱格式、生成脚本并发送邮件。

随后6月份有研究披露WormGPT新变种依托xAI Grok与Mistral模型，通过Telegram渠道售卖，能够在越狱后生成钓鱼邮件和窃取凭据的脚本。同月，思科Talos报告进一步指出，攻击者正在滥用大模型，常见方式包括使用无审查模型、贩售定制模型如WormGPT，以及通过越狱提示绕过安全限制，同时还出现针对模型供应链的投毒与后门问题。

紧接着，在7月份，Forescout对50个大模型进行测评，结果显示近半数模型无法完成漏洞识别和利用生成任务，即便能产出可用结果也需要大量人工引导。

8月份，ESET披露了一种名为PromptLock的新型勒索软件，被认为是首个利用本地大模型动态生成恶意代码的样本。它通过调用gpt-oss:20b模型生成Lua脚本，能够收集系统信息、扫描文件并执行加密，尽管仍处于概念验证阶段，但展示了勒索软件从静态逻辑向动态生成的重要转变。

这些事实表明，大模型驱动攻击正在快速演进，涉及自动化、越狱利用、黑市运营与漏洞研究等多个层面。

（参考链接：https://www.secrss.com/articles/76648）

步履不停：在荆棘中寻找科研的答案

光阴荏苒，我在东五实验室的日子悄然近四载。回望走过的道路，那段日子充满了挑战与疑惑，也正是这些迷雾般的经历，让我逐步认识到科研不仅仅是技术的攻坚，更是一场心灵的历练。感谢实验室的季刊约稿，让我有机会回顾这段宝贵的经历，并分享一些科研路上的心得与感悟。希望我的故事能为师弟师妹们提供一些参考和启发。

迷雾初现：荆棘中的迷途与顿悟

我的第一个研究课题聚焦于基于多步过滤的克隆漏洞检测系统。起初，我信心满满地想要设计一个创新的布隆过滤器方案，试图让传统布隆过滤器这一仅支持精确匹配的结构能够支持模糊匹配。然而，实际的实验进程远比预期复杂：动态位数组设计、权重因子调节、甚至引入自然语言模型的尝试，都未能带来理想的效果。连续数周，我与代码周旋，效果不仅没有改善，还失去了布隆过滤器最大的优势“快速”。正当我几近迷失方向时，吴月明师兄的一句提醒让我豁然开朗：“试着换个角度，思考一下模糊匹配的本质。”这一句话像一道亮光，驱散了我心中的迷雾，让我意识到自己过于执着于技术细节，而忽略了问题的全局。正是在那样的迷惘与顿悟中，我学会了放下固有思路，拥抱更广阔的学术视野，为后续的探索奠定了坚实的心理基础。

破冰之旅：跨界思考与创新突围

在组会上，我鼓起勇气如实陈述研究中遭遇的困境。本以为会迎来批评，邹德清老师却笑了笑：“科研不是孤军奋战。你现在的瓶颈，恰恰是因为读的论文还不够‘杂’。”吴月明师兄听到后，随即推荐了几篇看似与克隆漏洞检测无关的论文，从数学的信息论到恶意代码检测，甚至包含一篇社交网络的跨领域研究。“试试看，别人的‘钥匙’或许能开你的锁。”我抱着半信半疑的心态翻看那些论文。一篇关于JavaScript恶意行为检测的研究中提到了降噪策略，突然让我灵光一现，我们提取到的语法信息中的冗余信息会带来噪声，我们是否也能够通过某种降噪的方法来提高信息密度，从而增强处理效率？于是，我与吴师兄一同讨论。经过反复试验与调整，新思路果然使实验效果显著提升。这一跨界融合的经历不仅解决了技术难题，也让我深刻体会到跨领域借鉴的重要性。正如邹德清老师所言，科研的钥匙往往隐藏在不同学科的边缘，只要肯跨出舒适区，创新便无处不在。

从困境到顶会：论文路上的锤炼与蜕变

新方案取得初步成功后，我便投身于论文撰写的艰辛征程。起初，我以为只需将实验数据简单罗列便可完成论文，但实际写作却让我认识到：科研成果的传递同样需要严密的逻辑与精炼的文字。第一稿中，我的叙述像流水账般冗长散乱，缺乏清晰的论证。吴师兄直言不讳地指出：“这不是一篇成熟的学术论文，而更像是一份实验记录。”在他的指点下，我重新规划论文结构，明确问题的重要性，逐步引入现有方法的局限，再用严谨的数据证明新方案的优越性。论文修改的过程充满了反复推敲与不断完善。每一处论点、每一组数据都要经过反复验证，确保逻辑严谨、表述准确。面对

审稿人提出的数据集偏倚和可扩展性问题，我和吴师兄邹老师积极调整实验方案，扩充数据样本，并引入大规模测试，最终赢得了评审们的认可。这段历程不仅锤炼了我的科研能力，更让我深刻体会到严谨治学的精神和不断自我超越的重要意义。

穿过丛林：在CGCL文化中扎根

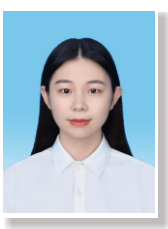
东五实验室的独特文化，早已成为我们科研路上最坚实的基石。实验室提出的“穿过丛林（CGCL）”精神——创新、卓越、沟通、友爱，不仅镌刻在实验室的墙上，更融入了每个东五人的血脉。金海老师对学术的极致追求让我第一次体会到科学家精神的重量。尽管他需要管理数百人的团队，却坚持亲自审核每一篇投稿论文。他曾逐字逐句地帮我审查论文中的细节错误。我清晰地记得有一次，我提交论文后收到了金海老师的邮件回复。他在邮件中严肃地指出：“参考文献的格式存在严重问题，细节最能反映一个人的科研态度。”短短一句话，却如同当头棒喝，让我深刻意识到科研不仅关乎理论创新，更体现在严谨的学术规范之中。看到这封邮件的那一刻，我心中一颤，回想起自己在整理参考文献时的疏忽，不禁感到惭愧。从那时起，我不再把格式视作微不足道的细节，而是以更严谨的态度对待每一篇论文的每一个部分，力求精益求精。金老师的这句话，提醒我始终保持对学术的敬畏与严谨。实验室的文化在细节中滋养着我，每年的CNCC会议，让我得以站在巨人的肩膀上眺望领域前沿；实验室的年会，更是不同领域在宏观层面上的交流与碰撞。这些经历不断塑造着我的科研思维，也让我深刻体会到科研不仅是个人的探索，更是在学术共同体中的成长与进步。

星光不灭：同行者的温暖与未来的展望

当论文最终被顶级会议接收，内心的激动难以言表，但我更珍视那段与团队共同奋斗的时光。实验室里，邹老师的指点、师兄们的激励、同学们在走廊里热烈讨论的声音，无一不构成了我科研旅程中最宝贵的记忆。每当深夜灯火通明，我总会想起那些在黑暗中依旧闪耀的灵感微光——那是团队凝聚的智慧和心血的结晶。科研的路途固然艰难，但正是因为有了这些同行者的支持与陪伴，我才能在困境中寻找到突破的契机。

展望未来，我坚信，只有不断吸收新知、敢于跨界尝试，才能在这条荆棘丛生的科研之路上走得更远。科研不只是个体的努力，更是集体智慧与文化传承的结晶。正如东五实验室的精神所昭示，每一次挑战都是迈向更高峰的起点，星光虽微，却能照亮前行的路途。如果说科研是一场冒险，那么东五实验室便是最坚实的后盾：它教会我如何将焦虑转化为探索的动力，如何在黑暗中捕捉微光，更让我明白——真正的突破，从来不是一个人的奇迹。

读研期间在USENIX Security 2024、ICSE 2024、FSE 2023、ASE 2022等网络安全领域国际顶会发表多篇论文。



冯思乐
2023级博士研究生
研究方向：克隆漏洞检测、供应链安全
Email：fengsiyue@hust.edu.cn

系统软件与体系结构组典型成果介绍

董雨康、赵建军、段卓辉、赵　进、黄　禹、姚鹏程、王庆刚、李钦宾、张书豪、
毛伏兵、叶晨成、郑　龙、张　宇、刘海坤、邵志远、蒋文斌、廖小飞

关键词： 内存计算，图计算系统，
大模型推理系统，数据流架构

1 介绍

包括电子商务、社交舆情、企业运维、科学计算等在内的各类大数据应用日益复杂、多元。新型的计算模式和计算范式，如图计算、内存计算、稀疏计算、数据流架构等，不断涌现并迅猛发展，为高效的大数据分析处理提供了强大的推动力，同时也引出了大量值得深入研究和探索的科学问题。基于上述背景，本小组重点研究适合上述需求的新型体系结构及其相关系统软件，研发新型处理器及软硬件配套、探索新型存储器件及运行环境，为上层各类大数据应用提供有效的支撑。基于所承担的国家重点研发计划、国家自科基金等项目，依托华中科技大学–华为技术有限公司“数据中心架构创新中心”等联合研究中心，这一年度，本小组主要在如下四个方面进行了研究，包括：内存计算、图计算系统、大模型推理系统、数据流架构，下面简要介绍一下相关的代表性研究工作：

1）在内存计算方面，提出了基于亲和感知租约机制的高性能事务化有状态无服务器工作流系统（RTSFaaS），采用基于租约的并发控制协议，在工作节点之间动态分配和转移租约，使得大多数事务能够以对缓存对象的强关联性进行执行，从而减少远程通信开销并提升缓存效益；提出了面向分离式内存架构的高性能事务系统（HDTX），以应对在RDMA支撑的分离式内存池系统实施分布式事务时面临的处理延迟高、网络开销大及内存节点算力不足等问题；提出了数据类型感知预取方案（DTAP），通过让硬件预取器识别数据对象类型，精准定位指针链并动态调整预取深度，实现高效预取；提出了一种基于智能SSD的键值分离LSM存储系统（AegonKV），充分利用智能SSD进行垃圾回收（GC）卸载，而无需与LSM索引竞争带宽和CPU资源，从而全面改进键值分离系统的吞吐率、尾延迟和空间使用率。

2）在图计算系统方面，提出了一种基于ReRAM的异步图处理加速器（ASGraph），通过依赖感知的子图构造、价值驱动的调度机制和混合处理策略，有效减少了冗余计算；针对现有自适应基数树（ART）系统面临的大规模冗余树节点遍历与高额同步开销两大性能瓶颈，提出了一种数据中心化（data-centric）的硬件加速器（DCART），用于高效支持ART操作；提出了一种高效的GPU链驱动（Chain-driven）时序图计算框架TempGraph，通过将时序图转换为互不相交的时间依赖链并构建捷径以解耦时序依赖，实现了高效的GPU并行计算与快速收敛，显著提升了时序图计算中的资源利用率和处理性能；针对现有的 HGNN 系统采用以超边为中心的数据流模型存在大量冗余计算的问题，引入“微边”概念，构建了微边中心化模型，并提出了一种基于异步调度的并行优化框架（RePAG）实现了更细粒度的任务并行性，设计了专用硬件加速器 MeHyper以充分释放RePAG模型潜力。

3）大模型推理系统方面，推出了首个专为

系统评估大模型智能体框架效率设计的基准测试平台AgentRace，对主流大模型智能体框架进行运行时性能、扩展性、通信开销及工具调用延迟的可控复现对比；为充分利用主流MoE模型推理过程中专家激活存在的局部性，提出了Diff-MoE架构，在GPU内存中设计了一套差分缓存并集成轻量级专家激活预测器，显著提升了MoE模型的推理吞吐率与内存效率。

4）在数据流体系结构方面，提出了一种面向新型异构数据流架构（DFU）的亚核级多算子交叉调度系统，通过将算子细化为共享指令的亚核（Subkernel），以支持不同算子的亚核交叉或顺序调度执行，通过片上存储（SPM）实现多算子间输入数据的高效共享，减少片外访存开销；针对单一数据流图在表达交叉领域时的局限性，提出异质数据流图的构建方法，通过流图中不同类型节点之间的协同工作来更加全面、准确地描述领域融合中的复杂计算过程，同时进一步将流图映射到异质数据流抽象机实现其在硬件层面的调度、分配和执行；针对检索增强型生成（RAG）系统的内存瓶颈问题，设计了一种异构 PIM 系统(HeterRAG),利用 HBM 的高带宽、低功耗以及DIMM的大容量、低成本来加速 RAG，此外，HeterRAG还通过融合三种软硬件协同优化技术（局部性感知检索、局部性感知生成和细粒度并行流水线）来进一步提高性能。

2 内存计算

2.1 基于亲和感知租约机制的高性能事务化有状态无服务器工作流系统

无服务器计算（Serverless Computing）近年来受到广泛关注，其核心优势在于通过函数即服务（Function-as-a-Service，FaaS）范式简化了可扩展云应用的编程、部署与运维过程。现有的有状态 FaaS 平台通常依赖外部数据存储（如 DynamoDB 和 S3）来管理应用状态。虽然这种计算与存储的分离方式在一定程度上实现了共享状态管理，但频繁的远程状态访问往往带来较大的性能开销，从而使有状态 FaaS 平台在保证强一致性的同时面临性能受损的挑战。首先，并发控制通常会因频繁的远程锁状态访问而导致较高的通信开销。其次，并发控制协议往往会削弱有状态 FaaS平台中缓存机制的效率。

为了应对这些挑战，提出了基于亲和感知租约机制的高性能事务化有状态无服务器工作流系统，RTSFaaS。如图1所示，RTSFaaS采用基于租约的并发控制协议，在工作节点之间动态分配和转移租约。与传统的租约机制不同，该方法允许某个工作节点以独占方式在分布式缓存中缓存并控制对特定对象的访问。由此，从全局角度来看，每个对象仅存在一个副本，并由单一工作节点独占缓存。通过这种方式，RTSFaaS使得大多数事务能够以对缓存对象的强关联性进行执行，从而减少远程通信开销并提升缓存效益。具体而言，RTSFaaS融合了两个关键设计，以应对当前FaaS平台在处理事务化有状态无服务器工作流时面临的上述挑战。

首先，RTSFaaS提出了一种亲和感知的租约分配机制，通过将一组相互依赖的对象分配给单个工作节点并赋予其独占租约，从而提升缓存的利用效率。RTSFaaS维护一张统计表，用于记录每个工作节点的请求总量和数据访问频率。在接收到请求时，RTSFaaS基于统计表采用分值驱动的调度策略，将请求分配给与该请求具有最强数据亲和性的工作节点。随后，在将一批请求分配给不同工作节点后，RTSFaaS会利用最新的统计信息为每个对象指定租约持有者；若某个工作节点对某对象的访问频率最高，则该节点即成为该对象的租约持有者。通过这种方式，RTSFaaS能够使更多函数在本地缓存中完成执行，从而显著提升缓存效益并减

少远程内存访问。

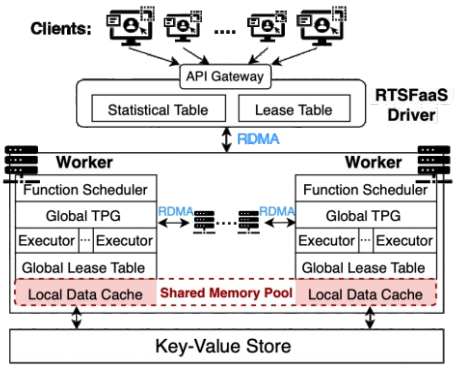


图1 RTSFaaS架构图

其次，RTSFaaS提出了一种RDMA驱动的动态租约转移机制，以显著降低锁操作带来的网络通信开销。在每一批请求中，RTSFaaS将函数执行与一致性保证解耦为两个不重叠的阶段：规划阶段与执行阶段。在规划阶段，工作节点识别函数之间的数据依赖关系，并协同构建全局任务优先图（Task Precedence Graph, TPG），用于确定函数的串行化执行顺序。在执行阶段，RTSFaaS允许租约持有者以独占方式缓存并控制对特定对象的访问，这些租约持有者会预取相应对象至本地缓存并执行函数。同时，它们会根据全局TPG动态地将对象的租约转移给其他节点，以保证数据访问的顺序性。当某个工作节点需要访问未在本地缓存的对象时，它会检查远程对象的租约状态，并通过单边RDMA原语直接访问其他工作节点的缓存。

实验结果表明，如图2所示，RTSFaaS在不同输入吞吐量下始终保持较低的延迟，相比Boki和Beldi展现出更优的性能。当中位延迟为700ms时，RTSFaaS的吞吐量在Movie Review场景下比Boki提升2.0倍（比Beldi提升6.0倍），在Travel Reservation场景下比Boki提升4.0倍（比Beldi提升20倍），在Banking Service场景下比Boki提升5.0倍（比Beldi提升17倍）。

2.2 基于RDMA分离式内存的高性能分布式事务系统

云计算基础设施呈现模块化与弹性化发展趋势，促使应用对资源弹性供给、异构硬件兼容性及故障隔离的需求日益提升。计算存储分离架构获得学术界与产业界的共同关注。该架构通过高速网络互联将传统单体服务器解构为独立计算节点与内存节点。计算节点配置本地内存资源，其功能定位于远程内存池的高速缓存；内存节点则仅保留基础算力，执行集群内存分配及网络初始化任务。资源池化模式显著优化了资源利用率，同时兼具弹性扩展与故障域隔离优势。

在访问分离式内存池数据时，计算节点需借助分布式事务机制保障数据完整性与一致性。在面向单体服务器的现有RDMA分布式事务方案设计中，存储节点仍需消耗大量算力执

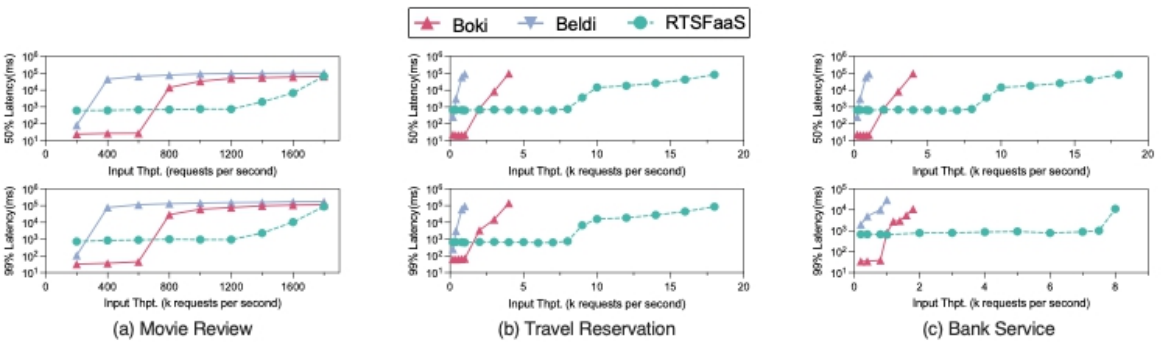


图2 RTSFaaS、Boki和Beldi在Movie Review、Travel Reservation和Banking Service三个应用上的延迟与吞吐量对比

行高开销操作（如缓冲区轮询、锁管理及数据拷贝）。这导致此类系统难以适配计算资源受限的内存节点。

在RDMA支撑的分离式内存池系统实施分布式事务时存在三个核心挑战。首先，典型分布式事务方案需历经执行、锁定、验证及主从节点提交等五个环节处理单次事务。先进的FORD系统虽针对分离式内存特性优化协议流程，但仍需四个处理阶段方可提交事务请求，引入显著网络时延。其次，传统架构依赖内存节点的本地算力执行高效数据同步，然而资源受限的分离式内存节点迫使计算节点发起两轮数据传输执行提交操作，高负载场景下将加剧RDMA带宽压力。最后，当前盲目重试与先进先出的锁机制无法保障关键任务的低延迟需求，而单体服务器优先级调度机制受限于分离式内存环境算力匮乏问题，无法直接部署运行。

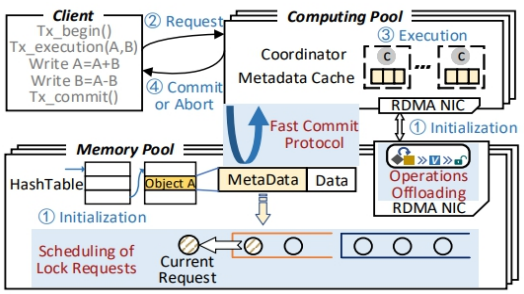


图3 HDTX架构图

为了应对这些挑战，提出了面向分离式内存架构的高性能事务系统HDTX。如图3所示，HDTX系统的计算池内的协调者节点负责接收并处理客户端发起的事务请求。内存池采用哈希表结构存储应用数据。计算节点上的协调者直接通过RDMA网络访问远程内存节点。

HDTX使用快速提交协议以实现网络轮次最小化目标。为了加速分布式事务并维持强数据一致性，快速提交协议结合重做日志与可见性控制技术，将验证、备份提交和主节点提交三个阶段合并优化。快速提交协议首先将备份提交及主节点提交合并为提交阶段与异步释放阶段。由于验证阶段与提交阶段间缺乏数据依赖性，快速提交协议进一步融合二者以减少网络通信开销。在执行与加锁阶段完成后，计算节点通过单次网络往返即可实现事务提交。

HDTX通过基于RDMA的卸载机制来将释放阶段卸载至内存节点的RDMA网卡。鉴于提交阶段已持久化存储包含最新数据的重做日志至内存节点，HDTX利用RDMA写与RDMA原子原语实现同步任务卸载。通过编排RDMA等待与使能原语，RNIC能够自主执行释放流程。由此，计算节点与内存节点间避免了冗余的数据传输，从而缓解了RDMA网络带宽争用问题。

HDTX还采用去中心化的优先级锁机制来调度事务请求。在加锁阶段，各计算节点借助RDMA FAA原子原语为读写集申请写锁。对于任务关键型事务，可动态提升其锁请求的优先级值，加速事务的锁获取过程，无需内存节点CPU介入即可调度延迟敏感型事务。

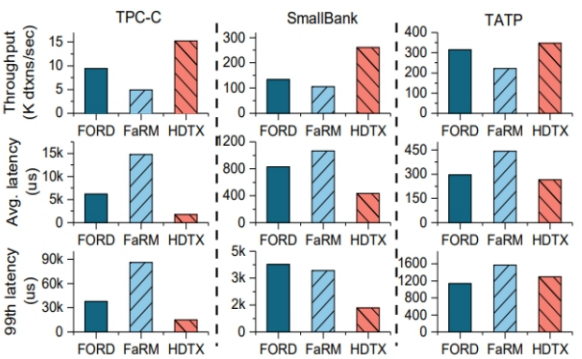


图4 在线事务处理工作负载下各事务系统的性能

实验结果表明，如图4所示，在TPC-C、SmallBank和TATP在线事务处理工作负载下，相较典型RDMA事务系统FaRM与FORD，HDTX将平均事务延迟分别降低88.3%与72.1%，99分位延迟最大降幅达82.7%和60.9%，事务吞吐率则最高提升2.08倍与84.7%。

2.3 数据类型感知的硬件预取技术DTAP

在树、图等链接数据结构的查询场景中，程序常因依赖且随机的指针追踪内存访问，面临频繁缓存缺失与显著性能损耗，这一问题在键值存储、在线数据分析、机器学习等现代应用中尤为突出。现有硬件预取技术难以有效应对该挑战，空间预取器（Spatial Prefetcher）依赖内存空间局部性，却因链接数据结构对象在内存中随机分布而失效；时间预取器（Temporal Prefetcher）需大容量片上存储保存历史访问轨迹，且面对不规则访问时准确率大幅下降；指针追踪预取器虽能识别指针并递归预取，但编译器方案难以动态确定预取深度，硬件方案片上存储开销显著。为此，提出了数据类型感知预取方案—DTAP，旨在通过软硬件协同实现高效预取。

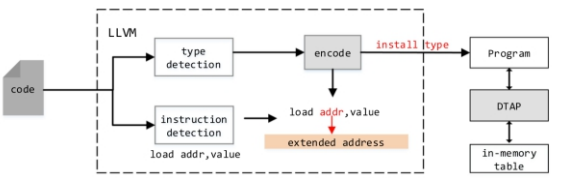


图5 DTAP的工作流程

如图5所示，DTAP 的核心思路是让硬件预取器识别数据对象类型，精准定位指针链并动态调整预取深度，整体架构分为软件编译器扩展与硬件 ISA 及预取器架构两部分。软件层面，编译器扩展承担类型信息提取、编码与传递的关键职责。它首先分析源代码，筛选出含指针的复合类型（结构体、类），为每个类型分配 15 位整数 ID，再对类型内每个指针进行 32 位编码，包含目标类型 ID、指针偏移与目标对象大小等关键信息。随后生成内存类型表，由类型索引表（128KB）映射类型 ID 与类型数组位置）和类型数组（存储编码后类型数据）组成，同时修改加载指令，将 15 位类型 ID 嵌入虚拟地址保留位，实现类型信息向硬件的传递。硬件层面，DTAP 通过 ISA 扩展构建高效预取执行机制。两个非特权寄存器分别存储内存中类型索引表与类型数组的虚拟地址，方便用户态程序配置。128 项的片上类型表采用 16 路组相联三元内容可寻址存储器（Ternary Content Addressable Memory，TCAM）结构，仅保留高频有效指针类型，通过频率和置信度的替换策略管理条目，确保资源高效利用。类型表遍历器在片上类型表未命中时，异步访问内存类型表加载信息，且不阻塞 CPU 前端。扩展预取引擎从加载指令提取类型 ID，匹配片上类型表后计算指针地址，递归预取目标对象，预取缓冲则通过命中情况更新片上类型表相关参数。此外，DTAP 还能基于预取准确率、覆盖率与内存带宽利用率，动态调整预取深度，通过一系列规则实现性能优化。

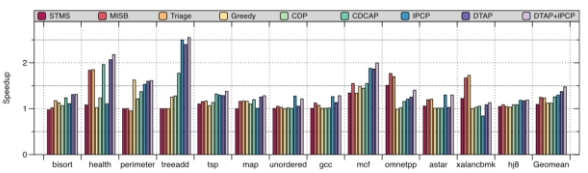


图6 不同预取方案下IPC提升比例

使用 Gem5模拟器，以 13 个内存密集型应用（含 Olden 基准、SPEC CPU 系列、C++ STL 组件及哈希索引）为负载，与多种主流预取方案对比。实验结果如图6所示，DTAP 平均加速 1.37 倍，DTAP 与 IPCP 预取器的组合方案加速比达 1.46 倍，比其他预取器平均提升5.9%到25.5%。

2.4 基于智能SSD的键值存储加速机制

键值分离架构通过将值的存储从LSM结构中分离，大幅减少了合并操作的开销，在大值存储的场景中获得了显著的性能提升。为了减少空间冗余并确保最佳的数据检索性能，键值分离架构在值区域的管理中引入了垃圾回收（GC）机制。它将包含大量过期数据的文件

合并到新的有序文件中，这些文件按键排序并包含最新版本的数据。此操作旨在实现两个目标：回收存储空间和提高扫描性能。然而，垃圾回收机制的引入增加了额外的I/O和计算开销。

值得注意的是，现有的工作主要集中在键值数据组织的架构优化上，仅关注垃圾回收在计算和I/O开销之间的权衡，这使得难以在吞吐率、尾延迟和存储开销这三个方面同时表现出色。通过对这些工作的分析，发现垃圾回收设计中缺乏结构化调度和计算优化会间歇性地阻碍系统吞吐率，但为了系统性能而牺牲垃圾回收的及时性或准确性会导致冗余空间使用呈指数级增长，或者合并开销和写停顿显著增加。因此，优化垃圾回收操作以在键值分离系统的各个方面实现出色性能仍然是一个核心挑战。

为此，提出了一种基于智能SSD的键值分离LSM存储系统（AegonKV），AegonKV充分利用智能SSD进行GC卸载，而无需与LSM索引竞争带宽和CPU资源，从而全面改进键值分离系统的吞吐率、尾延迟和空间使用率。

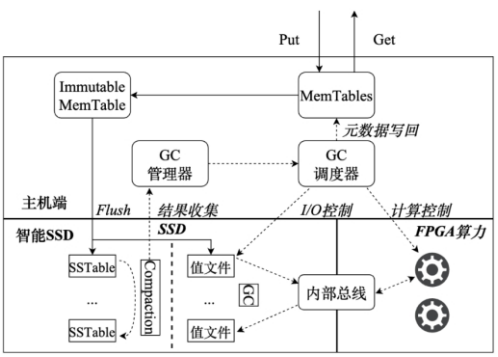


图7 AegonKV架构图

图7显示了AegonKV的架构图，AegonKV基于软硬件协同提出了四个新的设计来支持高效可用的GC卸载：首先，AegonKV为GC流程设计了ValidMap数据结构，通过在GC管理器中扩展利用合并结果并构建ValidMap，AegonKV大幅减少了GC过程中重复读写外部索引数据的次数和数据量。第二，为了高效利用智能SSD的计算资源，AegonKV在GC调度器中构建了三个不同的模块（I/O控制、计算控制和元数据写回），这些模块负责统一调度和协调数据通道、资源分配和数据移动，从而优化资源利用率。第三，为了解决GC过程中额外数据移动带来的带宽争用问题，AegonKV提出了一种无需验证即可直接批量写入GC元数据的方法。最后，AegonKV在智能SSD的FPGA算力硬件上实现了高效的GC计算操作，该流程包括数据读写操作、文件编解码、数据过滤和合并等单元模块。

基于代表性键值分离系统Titan实现了AegonKV，并且对AegonKV进行了全面评估，包括YCSB和实际生产工作负载，并将其性能与Titan、DiffKV、BlobDB和RocksDB进行比较。结果表明，与现有键值分离系统相比，AegonKV实现了1.28~3.3倍的吞吐率提升，尾延迟降低37%~66%，空间开销减少15%~85%。

3 图计算系统

3.1 基于ReRAM的高效异步迭代图处理加速器

图处理在许多实际应用中占据核心地位，但由于其计算与通信比率低和数据局部性差，传统基于冯·诺依曼架构的加速器难以有效解决图处理中的内存瓶颈问题。为此，基于阻变随机存取存储器（ReRAM）的加速器被广泛研究，通过利用ReRAM的并行计算能力来提升图处理的性能。然而，现有基于ReRAM的图处理加速器仍面临着冗余计算开销大的问题，这主要是因为子图顶点的状态更新依赖于其他子图，导致图顶点状态在ReRAM中多次重复处理。

现有基于ReRAM的加速器通常以子图为

基本处理单元，每次迭代中每个子图最多被处理一次。然而，由于子图间的顶点状态更新具有不规则性和并发性，新的顶点状态需要多次迭代才能传播到其他子图，导致收敛速度下降。此外，依赖链之间的子图可能分布在不同的ReRAM交叉阵列上并行处理。这种并行性可能导致子图在处理时使用过时的邻居状态进行更新。

通过对基于稀疏矩阵–向量乘法（SpMV）的图处理特性分，发现依赖感知的子图构建能够显著优化状态传播。具体而言，若根据顶点状态间的依赖关系构建子图，新的状态可以在子图内部快速传递到其后继顶点。这种紧密连接的子图结构能够减少跨子图的通信开销，从而降低冗余计算的频率。进一步研究发现，依赖感知的状态传播策略能够有效提升计算效率。例如，优先处理积累了大量待传播状态或能影响更多邻居的子图，能够使状态更新尽可能覆盖更多顶点，避免因无序处理导致的重复计算。

基于上述观察，提出了一种基于ReRAM的异步图处理加速器ASGraph，通过依赖感知的子图构造、价值驱动的调度机制和混合处理策略，有效减少了冗余计算。ASGraph根据顶点状态之间的依赖关系动态构建子图，并优先处理高价值子图，同时采用混合处理方案加速紧密连接子图的状态传播。

图8展示了ASGraph的系统概览，它由三个主要组件组成：依赖感知子图构造器（DSC）、值驱动子图调度器（VSS）和混合处理引擎（HPE）。其中，DSC用于根据顶点状态之间的依赖关系动态构建矩阵格式的子图；VSS用于评估构建的子图的价值，并优先将高价值的子图分配给HPE处理；HPE由多个定制的基于ReRAM的交叉阵列组成，这些阵列采用HP方案来处理同一SCC内每行子图。

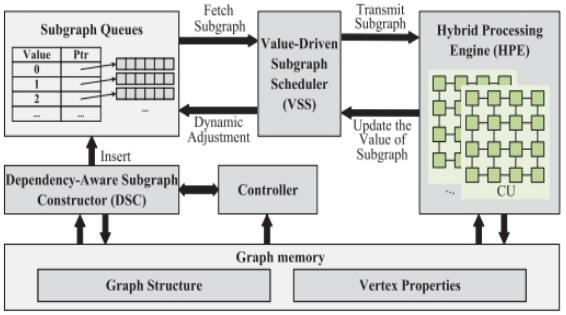


图8 ASGraph硬件架构

ASGraph依赖主机将原始图转化为DAG，并按拓扑顺序重新排序顶点，再划分为一系列子矩阵。控制器按照拓扑顺序分配子矩阵，每个子矩阵仅分配一次。分配时，DSC从活跃顶点追踪其后继依赖关系，提取紧密相连的顶点并构建子图，随后插入子图队列，实现活跃顶点在ReRAM阵列上的快速状态传播。子图构建完成后，VSS计算其价值以支持价值驱动调度，并在运行时动态维护子图的值。VSS从队列中选取价值最高的子图，存入缓冲区并传输至HPE处理。由于同一行子图共享相同的值，仅需计算一次，故一行子图可同时传输到HPE。HPE接收子图后，优先处理紧密连接的子图，将其转为邻接矩阵并加载到计算单元（CU），迭代至状态收敛。之后，同一行的其他子图也由CU处理，每行源顶点可累积更多状态再传播到其他行顶点，从而减少冗余计算。处理后，HPE会更新顶点状态，并触发VSS根据最新状态更新子图价值。

通过将ASGraph并与现有的基于ReRAM基的处理加速器以及基于CPU和GPU的图处理系统进行了对比。实验结果显示，与最先进的ReRAM基图处理加速器相比，ASGraph在性能和能耗方面均有显著提升，分别实现了平均25.5倍和70.8倍的性能提升以及平均4.8倍和2.2倍的能耗降低。

3.2 以数据为中心的自适应基数树硬件加速器

自适应基数树（Adaptive Radix Tree，ART）是一种在数据库、键值存储等领域被广泛采用的高效树型索引结构。然而，在多线程并发执行各类操作（如查询、插入、更新）时，现有ART系统仍面临大规模冗余树节点遍历与高额同步开销两大性能瓶颈。

冗余树节点遍历问题源于，不同操作在访问ART时，往往需要独立遍历相同的节点路径，这导致大量重复的节点访问和无效的片外通信。实验结果表明，在SMART结构中，冗余节点遍历比例超过77.8%，而在ART和Heart中，这一比例甚至高达86.1%与82.5%。这种遍历过程通常伴随着大量不规则且不可预测的内存访问，导致随机访问比例高、数据局部性差。更严重的是，ART中的部分键和子节点指针通常仅占1B和8B，远小于通用处理器的64B缓存行，平均缓存行利用率仅约20.2%，最终造成内存带宽的严重浪费与片外访问的碎片化。

现有ART系统的并发控制多依赖基于锁的算法（如节点级互斥锁），在访问热点节点时易产生严重锁竞争，尤其写操作比例较高时更为突出。Heart和SMART尝试用CAS缓解，但由于ART局部性差、缓存未命中频繁，CAS访问内存的延迟比L1缓存高出15倍以上。实测结果表明，在真实工作负载IPGEO中，并发控制开销随操作数增加在Heart和SMART中由16.2%升至62.1%，在ART中更是从24.1%升至71.3%；当写比例升高时，系统性能急剧下降。

深入研究发现，在ART的真实工作负载中，大多数操作集中于少量节点，这些节点在短时间内被频繁访问，呈现显著的时间相似性和空间相似性。基于时间相似性，可将访问同一节点的操作合并并且一次性遍历触发，减少冗余遍历与同步开销；基于空间相似性，可优先将高频访问节点及其部分匹配结果缓存于片上存储，降低片外访问与数据传输延迟。因此，提出了一种数据中心化（data-centric）的硬件加速器——DCART，用于高效支持ART操作。

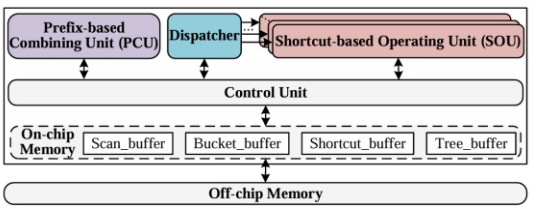


图9 DCART硬件架构图

DCART整体架构由三个关键单元和片上缓存组成：前缀合并单元（Prefix-based Combining Unit，PCU）负责扫描到达的操作，根据键的前缀将访问同一节点的操作动态分配到不同的独立桶（Bucket）中，实现按前缀的操作合并。这样，同一节点上的多个操作只需获取一次锁即可顺序执行，显著减少锁竞争调度器（Dispatcher）调度器将各个桶分发给不同的捷径执行单元（SOU）并行处理，确保访问同一节点的操作由同一SOU处理，从硬件层面消除节点级同步开销。捷径执行单元（Shortcut-based Operating Unit, SOU）在处理桶中操作时，可直接利用已维护的捷径表（Shortcut Table）快速定位目标节点及父节点，避免重复的自顶向下键匹配。若当前操作首次访问该节点，则在执行后生成对应的捷径记录以供后续操作复用。对于片上缓存设计的设计，DCART采取了四类片上缓存（树缓存、扫描缓存、桶缓存、捷径缓存）以隔离不同数据访问，减少片外通信。针对高频访问节点（高价值节点），树缓存采用基于价值的替换策略，优先保留命中频率高的节点，避免缓存抖动。另外，为了进一步提高系统的执行效率，PCU与SOU采用批处理方式实现流水化：PCU在合并第i+1批操作的同时，SOU正在处理第i批，从而隐藏操作合并的运行时开销。

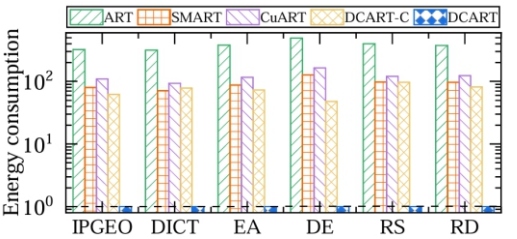


图10 DCART与其他系统的运行能耗对比

实验结果表明，DCART在多项指标上均显著优于现有方案。与ART、SMART和CuART相比，DCART将锁竞争次数降低至3.2%-19.7%，部分键匹配次数降低至3.2%-15.9%，并通过硬件流水线与片上缓存机制，将执行性能提升至分别高出123.8x-151.7x、35.9x-44.2x和21.1x-31.2x，同时在三类真实工作负载下实现了更低的P99延迟和更高的吞吐率。在能耗方面，如图10所示，DCART相比ART、SMART、CuART 和软件版 DCART-C 分别节省315.1x-493.5x、92.7x-148.9x、71.1x-126.2x和48.1x-97.6x 的能量；敏感性分析进一步显示，随着操作数量或写比例增加，DCART的性能优势愈加明显，充分验证了其在高并发、高写入场景下的优越性与可扩展性。

3.3 面向GPU的高效链驱动时序图计算框架

时序图在建模顶点和边随时间演化的复杂关系中具有重要意义，广泛应用于交通预测、金融建模、社交网络和流行病学等领域。近年来，已有大量面向CPU平台的时序图计算引擎提出，主要遵循两类执行模型。第一类是在静态图算法（如Bellman-Ford或Dijkstra）中加入时间约束，但其额外操作成本高，导致访问冗余和计算开销大。为此，出现了第二类基于转换的模型，即将时间信息嵌入顶点，将时序图转化为有向无环图（DAG），从而以较低开销保证时间约束。尽管该模型优于静态执行方式，但直接应用于GPU时仍存在效率不足，因时间约束限制了计算能力和带宽利用率。

进一步分析表明，资源利用率低下主要源于状态沿时间依赖链的顺序传播特性。一方面，时间约束使得每次迭代仅有极少顶点处于活跃状态（低于6.2%），并行度差，导致大量GPU线程空闲。另一方面，长时间依赖链需要多次迭代才能将状态传播至间接邻居，从而收敛缓慢。这些问题使GPU的并行性和带宽优势难以发挥，整体利用率偏低（不足26.4%）。

基于以上分析，提出了一种高效的GPU链驱动（Chain-driven）时序图计算框架TempGraph。具体而言，它将时序图转换为一组互不相交的时间依赖链，这些链能够揭示顶点之间的时序依赖关系，同时便于在GPU上沿这些链进行快速路径探索。此外，TempGraph采用一种新型的生成—激活—计算（Generate-Activate-Compute，GAC）执行模型，通过为不同链维护一组快捷方式来解耦它们之间的时序依赖关系，不同链对应的捷径可利用大量GPU线程并行生成。通过这些生成的捷径，每条链的头顶点可直接将其状态传播至该链的尾顶点。随后，从该尾部顶点发起的其他链可立即被激活并计算，无需等待沿时间顺序的缓慢状态传播。因此，每个顶点可通过远少于传统方法的迭代次数将状态传播至其他顶点，且大量时间图链可由GPU线程并行处理，从而在处理时间路径问题时确保快速收敛速度和高数据并行性。此外，TempGraph采用了一种考虑时间依赖性的分区调度方法，并结合异步CPU-GPU数据传输，以高效支持大规模时间图的GPU内存外处理。

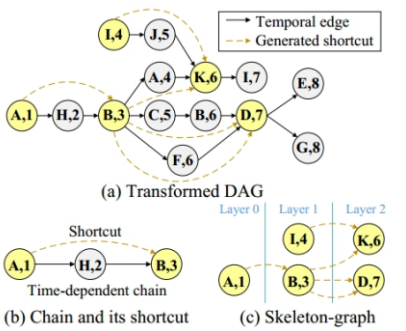


图11 GAC并行执行模型的描述

链驱动的GAC执行模型由三个阶段构成：骨架图生成（Skeleton-graph Generating）、捷径引导的链激活（Shortcut-guided Chain Activating）以及基于链的并行计算（Chain-based Parallel Computing）。如图11所示，执行模型通过对时序图进行DAG转换，并构建依赖链与捷径，最终形成骨架图；而具体的执行过程则由TempGraph的核心模块实现。整体架构如图14所示，包含三个关键组件：基于链的时序图转换器（Chain-based Temporal Graph Transformer）、骨架图生成器（Skeleton-graph Generator）以及捷径引导的链调度器（Shortcut-guided Chain Scheduler）。

首先，Chain–based Temporal Graph Transformer（基于链的时序图转换器）负责将原始时序图转化为时间依赖链。输入时序图通常以边列表的形式给出，系统会根据时间戳信息将其展开为有向无环图（DAG）。随后，转换器识别hub节点，并以这些hub节点为根并行遍历DAG，从而构建出一系列互不相交的时间依赖链。完成后，这些链被加载至GPU内存中供后续计算使用。值得强调的是，时序图只需进行一次转换，生成的时间依赖链可在不同应用中复用，有效降低了预处理开销。

其次，Skeleton-graph Generator（骨架图生成器）通过为时间依赖链生成捷径（shortcuts），构建骨架图（skeleton-graph）。在实现中，系统利用大量GPU线程并行计算捷径权重，以在保证效率的同时支持大规模图数据处理。对于仅包含两个顶点的短链，系统直接使用原始时序边；而对于更长的链，则生成相应的捷径来简化依赖关系。最终得到的骨架图会存储在GPU内存中，用于指导后续的链并行计算。虽然捷径生成会引入一定的运行时开销，但其大规模并行化特性能够有效摊薄成本，并显著提升链计算的并行度与整体性能。

最后，Shortcut-guided Chain Scheduler（捷径引导的链调度器）负责在执行阶段协调骨架图与时间依赖链的计算。TempGraph设计了两个独立的计算引擎：一个用于处理骨架图Gs，另一个用于处理时间依赖链Ch。在执行过程中，骨架图引擎按照拓扑顺序遍历捷径，从而快速激活对应的时间依赖链，并将链的ID写入Chain_Queue。随后，链计算引擎从队列中取出已激活的链，并将其分配给GPU线程并行处理，每条链由一个线程独立负责。

通过这种机制TempGraph能够充分发挥GPU的大规模并行能力，实现高效的时序图计算。

对真实世界和合成数据集进行了广泛的实验。实验结果表明，在NVIDIA A100 GPU 上运行的TempGraph与基于CPU的最新解决方案TeGraph相比，在128核Intel CPU 机器上实现了33.9-368.9x 倍的加速。此外，与基于GPU的最新解决方案（即基于GPU的静态图处理系统Tigr、Gunrock、LargeGraph和HyTGraph，这些系统均采用了最先进的时序图计算技术）相比，TempGraph在NVIDIA A100 GPU上实现了3.0-16.2x的性能提升。

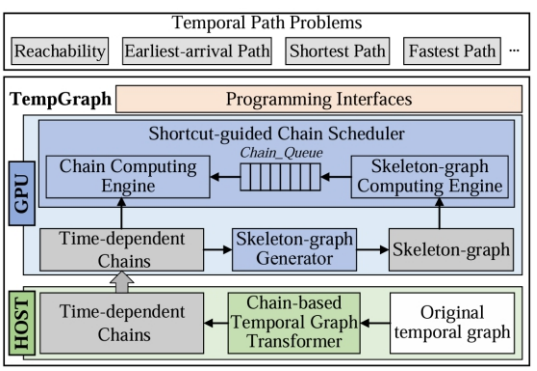


图12 TempGraph整体架构图

3.4 以微边为中心的超图神经网络加速器

超图神经网络作为一种新兴的深度学习框架，利用超图结构实现信息的传递与聚合，显著增强了数据建模和模式识别的能力。现有的

HGNN 系统采用以超边为中心的数据流模型，首先为每个超边收集其关联顶点的特征向量以更新超边表示，随后再利用超边特征向量反向更新相关顶点的表示。然而，在执行过程中，由于多个聚合任务之间存在重复的顶点参与，导致计算与访存操作出现严重冗余。实验分析表明，这类冗余计算在整体计算开销中占据极高比例，其上界可达 91.95%。

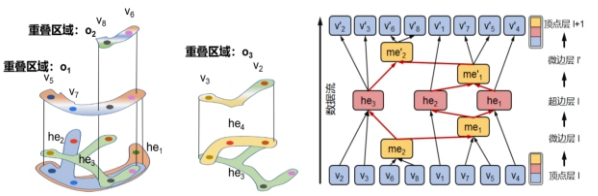


图13 重叠区域（微边）与微边中心化模型

研究揭示了 HGNN 中存在一种隐含的数据流依赖关系，可连接原本孤立的粗粒度聚合任务，从而有效减少冗余计算，如图13所示。在超图中，超边间常共享部分顶点（即重叠区域），这些区域之间还可能形成嵌套结构。将此结构抽象为数据流路径后，细粒度重叠区域的聚合结果可复用于其他超边或更大区域的聚合过程，显著降低超边聚合阶段的冗余度；反向利用该路径亦可减少顶点聚合中的重复计算。为显式建模这些依赖关系，提出“微边”概念，作为表示超边或重叠区域内部共享子结构的细粒度抽象单元。微边由共享顶点子集构成，可用于刻画超边之间的重叠、重叠区域之间的嵌套，以及超边与重叠区域的交集。

利用微边，将传统的超边中心化数据流模型（顶点→超边→顶点）转变为新型的微边中心化模型（顶点→微边→超边→微边→顶点），并将新的数据流图称之为微边聚合流图（MAG），如图13。在MAG中，每个顶点、微边和超边都代表一个独立且无冗余的聚合任务（AG），并通过唯一标识符进行明确区分。MAG通过其边结构显式地刻画了这些任务间的依赖关系流，从而彻底消除了计算冗余。为了进一步挖掘 MAG 中潜在的计算并行性，提出 RePAG 模型，即一种基于异步调度的并行优化框架。RePAG 的核心机制在于：动态识别所有前驱任务均已完成的“可运行任务”，并对其进行并行调度，以提升整体计算效率。

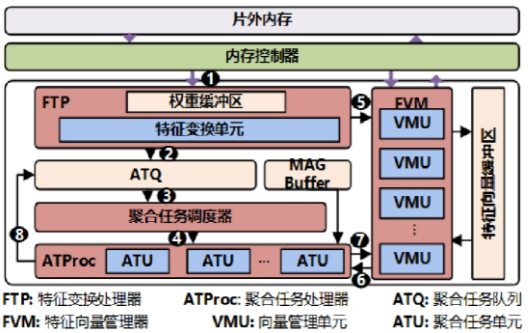


图14 MeHyper加速器

相较于传统的超边中心化执行模型，RePAG 实现了更细粒度的任务并行性，但仍面临两大核心挑战：其一，图遍历式的任务激活与调度流程涉及大量不规则计算和随机内存访问；其二，微边中心化的数据流将产生大量顶点、微边与超边的中间特征表示，受限于片上存储容量，需频繁访问片外内存。为解决上述问题并充分释放 RePAG 的模型潜力，设计了专用硬件加速器 MeHyper，如图14所示。MeHyper 采用模块化设计，包含四个核心模块：特征变换处理器，通过脉冲阵列完成线性映射；聚合任务调度器，动态平衡各聚合单元的任务负载，实现任务的高效派发；聚合任务处理器，由多个并行聚合单元组成，分别执行 MAG 中的任务，并通过特征向量管理器访问所需数据；特征向量管理器，协调顶点、微边和超边特征的读写请求，维持高效的数据流通。此外，MeHyper 引入了解耦合的双子流水线架构，分别面向计算（ReP）与任务管理（AG），实现流水线级内部并行；并结合对任务依赖图的分析，提出层次化特征向量管理策略，依据任务的后继信

息有选择地释放中间特征，进一步降低存储压力与片外访问频率。

在对Mehyper进行了全面评测，在多个真实数据集和HGNN模型上的实验表明，MeHyper在性能和能效方面均显著优于现有GPU平台。与现有HGNN加速系统HyperGef相比，MeHyper在推理任务上可实现最高10.51倍的性能提升。

4 大模型推理系统

4.1 大模型智能体框架性能评测基准AgentRace

随着技术进步，大模型智能体（LLM Agent）这种能够通过智能交互执行复杂任务的自主实体已成为极具前景的研究与实践方向。要实现大模型智能体在未来现实场景中的广泛部署，其框架的运行效率至关重要。高效执行、良好扩展性和最小化通信开销对于确保及时响应与实际可用性具有决定性意义，特别是在资源受限和延迟敏感的环境中。

为此，推出首个专为系统评估大模型智能体框架效率设计的基准测试平台AgentRace，对主流大模型智能体框架进行运行时性能、扩展性、通信开销及工具调用延迟的可控复现对比。如图15所示，该平台由框架（Frameworks）、工作流（Workflows）、数据集（Datasets）和分析（Analysis）四大相互关联的模块组成，旨在全面覆盖多样化的智能体框架、执行工作流、任务复杂度及性能分析。

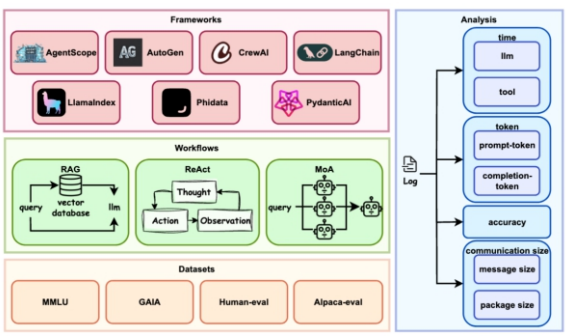


图15 AgentRace框架的结构

数据集模块定义了基准测试的核心任务体系，其核心设计目标是通过多样化现实场景全面评估LLM智能体框架，框架采用了GAIA、HumanEval、MMLU和AlpacaEval四个具有差异化特性的代表性数据集。这些数据集共同构成了从单轮查询、精确代码生成到多步推理和协同任务执行的完整光谱，可系统检验智能体框架在工具使用、记忆管理、检索集成和跨智能体通信等维度的性能表现；工作流模块的设计兼顾了现实任务执行策略的多样性和确保与现有智能体框架的广泛兼容性两大目标。通过ReAct、RAG和MoA三大主流工作流范式实现智能体实例化。这些工作流代表了三种本质不同的协同机制：顺序提示、检索增强回答和分布式多智能体协作。通过同时支持这三种范式，该基准测试能够全面评估智能体框架在不同推理风格、系统架构和性能约束下的表现；框架模块整合了当前主流的开源大模型智能体框架（包括LangChain、AutoGen、AgentScope、CrewAI、LlamaIndex、Phidata、PydanticAI），覆盖多样化的设计理念、运行时环境和抽象层级。所有框架均在统一的数据集、提示词、工具接口和工作流条件下进行评估，确保对比实验的公平性；分析模块定义了评估LLM智能体框架系统效率的核心指标，包括执行耗时、令牌消耗、通信量和准确率。该指标体系通过多维度量化智能体性能，在计算效率与通信效率的评估中保持任务目标保真度。通过精确测量这些权衡关系，本基准测试既支持框架间的原则性比较，又能为系统优化提供可操作的改进方向。

通过大量系统性的实验，整合了各框架性能瓶颈的细粒度分析，并揭示现有智能体框架低效的关键成因，为从业者和研究者优化高效大模型智能体部署提供可行建议，相关内容如下：

（1）大模型推理时间在各类智能体框架中普遍占据主导地位（如图16所示），而低效的提示工程会显著增加延迟和成本，主要表现在两个方面：一是向提示词中附加不必要的对话历史，二是使用过于冗长的提示内容。

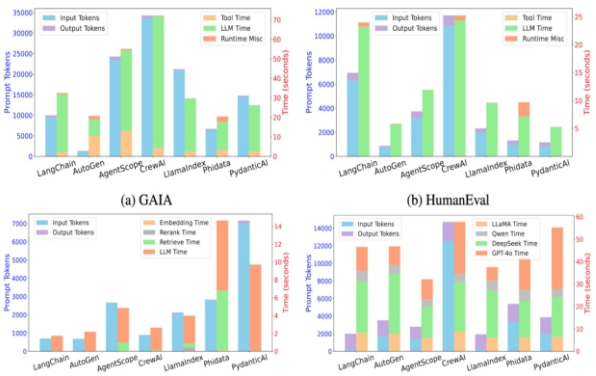


图16 不同框架的每请求令牌消耗量和执行时间

（2）各智能体框架在工具执行效率方面存在显著差异，其中搜索类和图像类工具因其高延迟特性成为主要瓶颈。具体表现为：轻量级工具（如文本和文档处理工具）效率差异较小，而中延迟工具（如PDF和Python工具）则因框架实现和I/O策略不同呈现中等程度差异。

（3）当前多智能体框架在信息检索时通常依赖外部数据库，但数据库性能常被忽视，而向量数据库是更优选择。RAG的工作流性能主要受嵌入和检索延迟影响：AgentScope因初始化时调用大型嵌入模型作为独立大模型请求，导致向量化延迟较高；Phidata则因采用两步处理流程同样存在向量化速度慢的问题。

（4）在多智能体系统中，低效的通信架构和包设计会导致较高的通信开销。例如，CrewAI 等集中式框架由于采用顺序子智能体协调机制，每次交互时提示词和消息体量都会增长，从而增加通信成本。而 Phidata 则因重复传输消息内容并包含额外元数据，进一步扩大了通信量。

（5）大语言模型若完全缺乏输出约束可能导致工具调用失败，但过度严格的输出验证又会显著增加令牌开销并降低响应成功率。如，LlamaIndex等框架容易因GPT-4o的结构化输出不稳定而出现工具调用失败，因其要求严格的输入格式，在GAIA等复杂数据集上准确率较低，除非强制规范输入格式。

这些发现为LLM智能体的设计与部署指明了优化方向。希望所提出的框架AgentRace能为未来开发高效、可扩展且鲁棒的智能体系统提供指导，并计划随着大模型智能体生态的发展持续扩展该基准测试体系。

4.2 基于优先级驱动的批量 MoE 推理的差分专家缓存

随着ChatGPT、DeepSeek等大规模语言模型（LLMs）的快速发展与广泛应用，混合专家模型（Mixture-of-Experts, MoE）凭借稀疏激活特性有效降低了推理成本，有效降低了推理过程中的计算开销。然而，MoE模型因其巨量专家参数的存储需求，对资源受限的单GPU部署环境提出严峻挑战。

为缓解GPU内存瓶颈，当前主流MoE推理方案通常将专家参数卸载至主机内存，并在推理过程中按需加载活跃专家。现有方法主要分为两类：基于预取的卸载方案和基于缓存的卸载方案。前者依赖专家激活预测，提前传输专家参数，但在大批量推理时，GPU与主机之间的通信开销会迅速增长，成为系统瓶颈；后者则通过缓存部分专家减少重复加载，但受限于有限的缓存效率和命中率，难以充分利用专家局部性。虽然这些方案在单样本推理下表现尚可，但在批处理场景下，数据传输延迟将显著限制端到端吞吐率。

通过对主流MoE模型推理过程的深入分析，发现专家激活具有明显的全局局部性和时间局部性：一方面，少数专家在整个推理阶段

被频繁激活，覆盖了大量推理样本的需求；另一方面，部分专家在连续的解码步骤中反复被激活。然而，现有方法尚未充分利用这些局部性特征来优化专家缓存与数据传输策略。

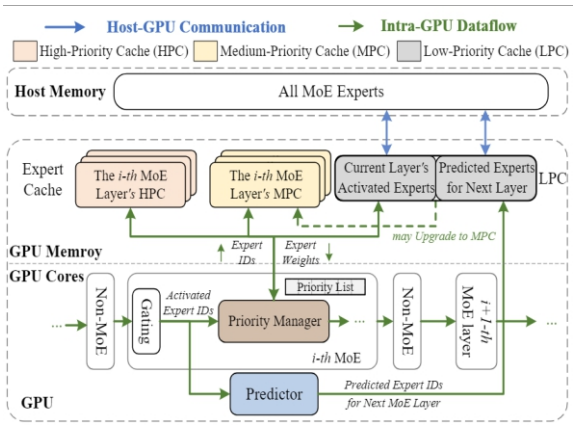


图17 Diff-MoE架构图

为充分挖掘并利用上述局部性，提出了Diff-MoE架构，在GPU内存中设计了一套差分缓存。Diff-MoE为每个专家动态维护一个优先级分数，根据专家在历史和当前推理过程中的激活情况不断调整。首先在离线微调阶段根据全局激活统计筛选出全局热门专家，在推理前将其优先加载至每个MoE层专属的高优先级缓存（HPC）中，并在推理过程中保持常驻，保证高频参数始终可用。对于其他专家，在推理过程中动态调整其优先级分数，并据此划分出局部热门专家和冷门专家。

每个MoE层拥有独立的高优先级缓存（HPC）和中优先级缓存（MPC）。其中HPC专门存放全局热门专家，MPC动态保存近期活跃且具备强时间局部性的局部热门专家。所有MoE层还共享一个低优先级缓存（LPC）作为临时缓冲区。在推理过程中，当路由网络激活的专家不在当前MoE层的HPC或MPC中时，会将其从CPU加载到LPC中，随后执行计算。计算结束后，Diff-MoE依据优先级分数驱动的缓存替换策略，判断是否将其晋级至MPC中，未晋级的专家会立刻从LPC中移除，确保缓存空间高效复用。

此外，Diff-MoE集成了一个轻量级的专家激活预测器，能够基于历史轨迹，预测下一MoE层可能被激活的专家，并提前将高概率专家参数从CPU加载至GPU上的共享LPC内。通过将数据迁移与计算过程充分重叠，进一步降低通信带来的延迟和性能损失。

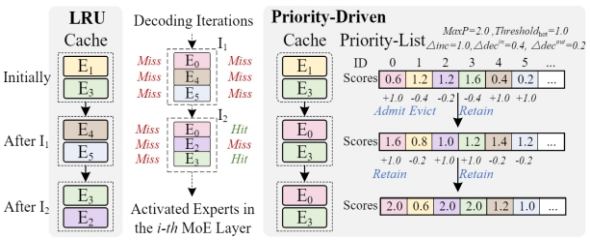


图18 优先级驱动的缓存替换策略

评估表明，Diff-MoE在端到端吞吐率、内存效率、Cache命中率、预测准确率在不同批次大小下均表现优异。与SOTA工作的 DeepSpeed、Pre-gated MoE 和 MoE-Infinity 相比，Diff-MoE的推理吞吐量分别提高了 2.74 倍、2.22 倍和1.55 倍。

5 数据流架构

5.1 基于数据流芯片的亚核级多算子交叉调度系统

新型异构数据流架构DFU，由传统CPU和异构众核处理器DFU组成。该架构通过将数据流图映射部署到PE阵列上计算，显著提升任务计算效率。

然而，当前DFU架构在多算子执行场景下存在以下不足：顺序执行限制：现有运行时系统采用串行算子执行模式，PE阵列每次仅支持单一亚核（Subkernel）运行，缺乏多算子并行调度能力；数据复用效率低：多算子间缺乏有效的数据共享机制，频繁的片外访存导致内存访问开销显著，限制了系统性能；因此，如何在新型异构数据流架构上设计一种高

效的多算子并行调度方法，以实现动态算子选择，提升系统整体性能，成为亟待解决的技术难题。

为克服上述不足，提出了一种基于数据流芯片的亚核级多算子交叉调度系统，通过细粒度的亚核调度、动态资源分配和数据复用机制，显著提升多算子任务的执行效率。核心创新点包括：（1）亚核级并发与交叉调度：将计算任务分解为多个算子，并进一步细化为共享指令的亚核（Subkernel），支持不同算子的亚核交叉或顺序调度执行，以提升指令级并行性和算子并发度；（2）动态算子分组与调度队列：根据数据流图和运行时特征，将算子动态分组并维护调度队列，依据PE阵列资源状态动态分配亚核，实现高效并行执行；（3）高效数据复用与访存优化：通过片上存储（SPM）实现多算子间输入数据的高效共享，减少片外访存开销。

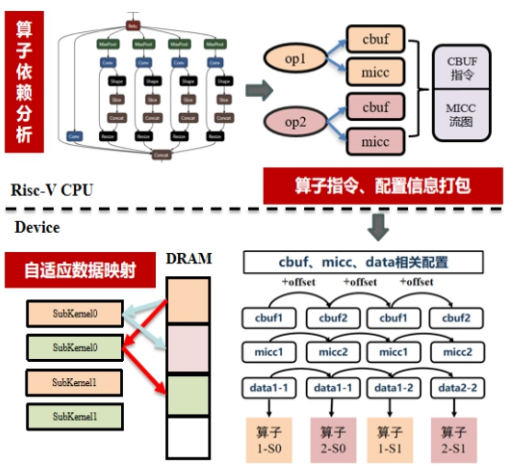


图19 任务配置系统功能示意图

系统由任务配置子系统和交叉调度子系统两大核心模块组成，具体如下：1）任务配置子系统：分析用户提交的计算任务，提取算子间数据依赖关系，生成统一的数据流结构；融合不同算子的CBUF（控制缓冲区）、MICC（微指令控制单元）及输入数据配置文件，确保指令和数据高效加载至PE阵列。2）交叉调度子系统：根据数据流图和运行时资源状态，将算子分组并维护调度队列；基于细粒度依赖关系及系统可用资源状态，动态调度多算子亚核，实现交叉或顺序调度执行，最大化指令并行性；通过SPM共享多算子亚核输入数据，减少主存与SPM间的数据传输开销。

基于上述设计，实现了基于数据流芯片DFU的亚核级多算子交叉调度系统。实验表明，通过合并多算子数据传输并使用数据共享机制，系统能有效消除冗余访存，实现高达75%的存储开销节省。通过动态调度多个算子的亚核并同时部署至PE阵列上执行，系统实现了约21.5%的执行速度提升。随着可并行执行算子数量的增加，多算子并行启动的效率增益将进一步放大。

5.2 一种基于异质数据流图的多领域融合执行方法

数据流被视作攻克多领域融合难题的一种行之有效的途径。从本质上讲，数据流图是一种对计算过程进行抽象描述的模型，它以数据的流动和处理过程为核心，能够清晰地展现数据在系统中的流动路径以及在各个处理节点上的操作情况。

常见的数据流图类型包含指令级数据流图、线程级数据流图和程序块级数据流图。需要注意的是，任何一种单一的数据流图在面对交叉领域的复杂情况时，都难以对其进行全面而有效的表达。鉴于单一数据流图在表达交叉领域时的局限性，构建异质数据流图就显得尤为重要。在异质数据流图中，不同的节点被赋予了不同的表现类型，这些不同类型的节点能够各自适应交叉领域中不同的计算场景和数据处理需求，从而巧妙地应对当前领域融合所面临的各种挑战。一些节点可以专门用于处理类似指令级数据流图所擅长的细粒度操作，而另一些节点则可以负责处理程序块级数据流图所

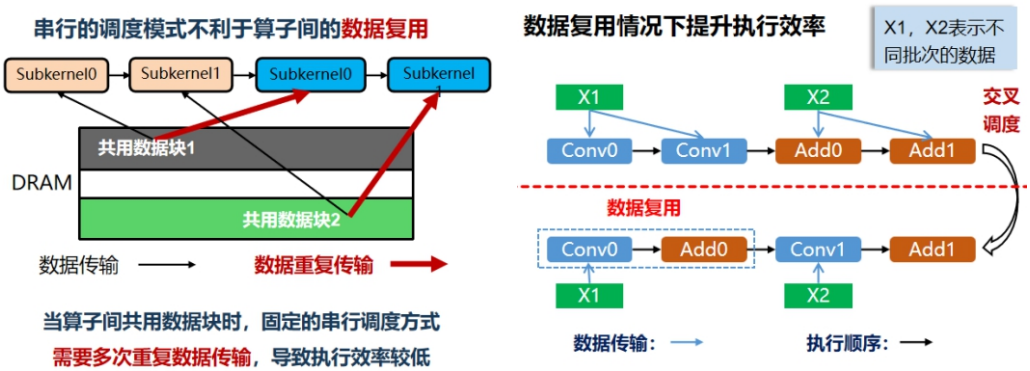


图20 多算子数据复用示意图

对应的宏观数据交互，通过不同类型节点之间的协同工作，异质数据流图能够更加全面、准确地描述领域融合中的复杂计算过程，为解决多领域融合中的数据处理问题提供了一种更为有效的解决方案。

异质数据流图的构造过程如图21所示。首先将一个应用程序抽象为指令级数据流图，指令级数据流图中的每个节点代表一项指令级数据流节点，连接节点间的边则象征节点之间的数据依赖关系；随后识别指令级数据流图中的串行逻辑结构，以定位反映连续操作特性的指令级数据流节点集合，将串行逻辑结构提炼成线程级数据流节点；最后，识别融合了线程级数据流节点的数据流图中具有数据局部性的数据局部性片段，将每一个具有数据局部性片段的指令级数据流节点或者线程级数据流节点抽象为一个程序块级数据流节点。

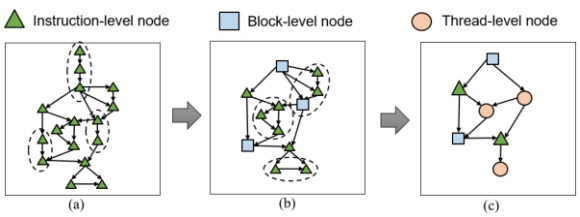


图21 异质数据流图融合示意

异质数据流图构造完成之后，接下来需要将异质数据流图映射到异质数据流抽象机中，异质数据流抽象机描述了异质数据流图在硬件层面的调度、分配和执行，具体结构如图22所示。异质数据流抽象机由计算节点（node）组成，计算节点之间以互联网络连接，每个计算节点由芯片（chip）组成，芯片以高速开关或总线互联，芯片上的核簇（cluster）以片上网络互连，每个核簇包含多个核心，核心分为计算单元（Compute Unit，CU）和调度单元（Scheduling Unit，SU），CU为结构简单的计算核，用于完成计算任务，SU为结构复杂的调度核，用于管理一个核簇内的所有硬件资源，以及将处于就绪状态的异质数据流节点依据调度策略调度给处于空闲状态的CU执行；芯片中的核簇包括一个指令级核簇和若干个线程级核簇，指令级核簇和线程级核簇中包含一个SU和若干个CU，其中，指令级核簇中的CU比线程级核簇中的CU数量多，指令级核簇中的SU比线程级核簇中的SU功能复杂。

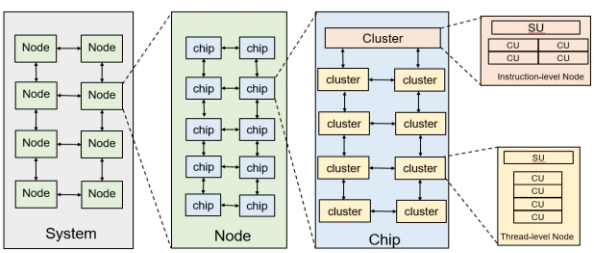


图22 异质数据流抽象机

异质数据流图到抽象机的映射规则如下。

首先需要对异质数据流图进行子图划分，确保各子图之间计算量负载均衡同时子图之间的数据通信达到最小，然后把数据流图中的每一个子图映射到异质数据流图抽象机中的一个计算节点上（node）。进一步细化，将一个子图中的一个程序块级数据流节点映射到一个chip上，把若干指令级数据流节点和线程级数据流节点也映射到一个chip上。再进一步细化，一个chip由一个指令级cluster和若干线程级cluster构成，一个线程级数据流节点与一个线程级cluster相对应，若干个指令级数据流节点与一个指令级cluster相对应。

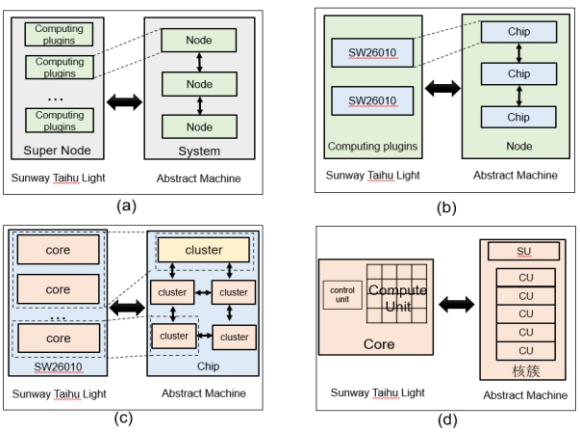


图23 抽象机模型到神威·太湖之光映射机制

最后还需要将抽象机模型映射到具体硬件进行执行，抽象机模型到神威·太湖之光超级计算机的映射机制如图23所示。神威·太湖之光超级计算机由160个超节点构成，每个超节点与1个异质数据流图抽象机相对应；每个超节点包含32块运算插件，每个运算插件和异质数据流图抽象机中的一个计算节点相对应；每个运算插件由8个神威26010处理器组成，每一个神威26010处理器和异质数据流图抽象机中的一个chip相对应；每个神威26010处理器包含260个计算核，一个计算核和异质数据流图抽象机中的一个线程级核簇相对应，若干个计算核连接后和异质数据流图抽象机中的一个指令级核簇相对应；其中，每个计算核由MPE和CPE组成，MPE和异质数据流图抽象机中的SU相对应，CPE和异质数据流图抽象机中的CU相对应。

5.3 面向检索增强生成系统的异构存内计算加速

检索增强型生成（RAG）系统通常由两个基本阶段组成：检索和生成。检索阶段由于其随机且不规则的内存访问模式，导致带宽利用率低。同时，生成阶段也受到内存带宽限制的制约，这是由于其涉及大量的通用矩阵向量乘法（GEMV）操作。这两个阶段共同导致了RAG 系统中的内存瓶颈。

为了解决这一问题，设计了HeterRAG，这是一种异构 PIM 系统，通过结合基于 HBM 的 PIM 和基于 DIMM 的 PIM 的优势来加速 RAG。在生成阶段，遵循当前基于 HBM 的 PIM 加速设计用于 LLM 推理，以利用 HBM 的高带宽和低功耗。对于内存需求更高的检索阶段，使用基于 DIMM 的 PIM，以利用 DIMM 的大容量和低成本。基于 HBM 的 PIM 和基于 DIMM 的 PIM 被设计为独立的设备，可以分别进行扩展和优化，以更好地适应各种 RAG 系统工作负载。

支持独立扩展两类PIM设备，增强系统灵活性与可扩展性。在过程中HeterRAG 在生成阶段使用基于 HBM 的 PIM 来满足带宽需求，在检索阶段使用基于 DIMM 的 PIM 来满足内存容量要求。为了进一步提高性能，HeterRAG 融合了三种软硬件协同优化技术：局部性感知检索、局部性感知生成和细粒度并行流水线。

如图24所示，主机接收用户查询，将其转化为向量，并通过互连网络广播至所有AccelDIMM设备。检索操作被卸载至AccelDIMM设备，各

设备独立执行近似最近邻搜索（ANNS），寻找与查询最相似的top-*k*向量。主机随后聚合所有设备的搜索结果，排序并确定最终的top-k最近邻向量。主机将这些向量ID映射至对应文档文本，将其与用户查询结合，并编码为张量。这些张量被发送至AccelHBM设备进行生成。AccelHBM 设备处理张量以生成令牌序列并返回主机。在迭代式RAG中，主机会派生新查询向量并重复检索–生成流程，直至满足终止条件（如达到最大迭代次数或置信度阈值）。最终，主机将令牌序列解码为自然语言文本，作为答案返回用户。

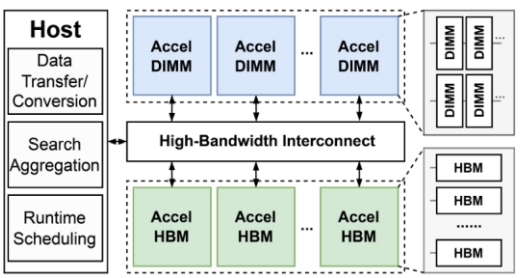


图24 HeterRAG架构

其中，AccelDIMM设备设计用于支持完整的近似最近邻搜索（ANNS）操作。邻居获取（Neighbor fetching）主要涉及内存访问和结果过滤，但在内存中维护顶点访问记录的已访问列表（visited list）会引入显著的硬件面积开销，使其不适合内存卸载。类似地，队列更新（queue updating）需集中聚合计算结果，同样不宜卸载。因此，内存卸载仅适用于距离计算操作。

AccelHBM架构则是为RAG生成阶段服务的基于HBM的PIM。主要包含：顶层处理模块：包含矩阵单元（Matrix Unit）、向量单元（Vector Unit）等组件，用于执行通用矩阵乘法（GEMM）及其他运算；Bank级处理模块：指存内计算单元（In-Memory Computation Unit），专用于执行通用矩阵–向量乘法（GEMV）运算。

在实验过程中，每个文档缓存KV张量（键–值张量）会消耗大量内存，因为同一文档在不同序列中可能需存储多份副本。例如序列{D1,D2,D3}和{D1,D4,D3}需重复存储D3的KV张量。虽然前缀树技术可通过共享公共前缀序列的KV缓存来优化，但对于长序列且前缀重叠有限的情况，该方法效能显著下降。此外，当文档出现在新序列中时，即使其条目已缓存，仍需重新计算KV。

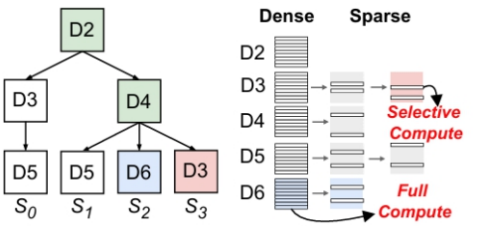


图25 使用位置感知生成的新文档序列的处理流程

为此提出创新解决方案：前缀树+选择性计算，在保持生成质量的同时提升缓存效率。如图25所示：系统处理新序列{D2,D4,D6}时：前缀树匹配到{D2,D4}：加载其稠密KV与稀疏KV，通过选择性替换构建新KV（稀疏KV覆盖稠密KV部分）；D6无匹配路径：需在预填充阶段计算KV；当序列{D2,D4,D3}到达时：D3虽无前缀树路径，但其稠密KV已缓存，因此仅重算重要词元生成稀疏KV并缓存。方案采取了缓存淘汰机制：空间不足时按LRU策略逐出稀疏KV条目；若文档所有稀疏KV被移除，则完全释放其缓存空间。

通过实验验证，相比于其他方案，HeterRAG对比CPU-GPU异构方案达13.72倍，对比NaiveHBM方案达19.47倍，对比OnlyDIMM方案达4.07倍；HeterRAG实现了极低的单次RAG请求端到端延迟；HeterRAG相比所有基线方案均实现能耗降低：较CPU-GPU异构方案降低49.8%，较NaiveHBM方案降低17.0%，较OnlyDIMM方案降低72.3%

6 总结

本小组工作主要立足于大数据高效处理相关的软硬件系统层面的核心问题，从体系结构、编译运行时系统和应用支撑三个不同层面开展了一系列研究工作，尤其是在内存计算、图计算、大模型推理和数据流体系结构等方面有较强的研究特色。相关研究工作获2025年教育部自然科学一等奖，并获Graph Challenge 2025（图计算领域最具影响力的国际赛事之一）全球总冠军，在其他各类专业竞赛和排名中也取得了优异的成绩。后续，我小组将继续围绕上述领域进一步深入展开研究。

附成果列表论文

[1] Yukang Dong, Ziyuan Shen, Wenbin Jiang, Zhenghang Liu, Ye Xu, Bingyi He, Ran Zheng, and Hai Jin, Bridging the Gap between Unstructured SpMM and Structured Sparse Tensor Cores, In Proceedings of The International Conference for High Performance Computing, Networking, Storage, and Analysis (SC), 2025.

[2] Kexin Li, Wenkan Huang, Qinggang Wang, Long Zheng, Xiaofei Liao, Hai Jin, Jingling Xue, Diff-MoE: Efficient Batched MoE Inference with Priority-Driven Differential Expert Caching, In Proceddings of The International Conference for High Performance Computing, Networking, Storage and Analysis (SC), 2025.

[3] Wenju Zhao, Pengcheng Yao, Dan Chen, Long Zheng, Xiaofei Liao, Qinggang Wang, Shaobo Ma, Yu Li, Haifeng Liu, Wenjing Xiao, Yufei Sun, Bing Zhu, Hai Jin, and Jingling Xue, MeHyper: Accelerating Hypergraph Neural Networks by Exploring Implicit Dataflows, In Proceedings of the IEEE International Symposium on High-Performance Computer Architecture (HPCA), 2025.

[4] Haiheng He, Haifeng Liu, Long Zheng, Yu Huang, Xinyang Shen, Wenkan Huang, Shuaihu Cao, Xiaofei Liao, Hai Jin, and Jingling Xue, MetaHG: Enhancing HGNN Systems Leveraging Advanced Metapath Graph Abstraction, In Proceedings of the Twentieth European Conference on Computer Systems (EuroSys), 2025

[5] Chaoqiang Liu, Haifeng Liu, Dan Chen, Yu Huang, Yi Zhang, Wenjing Xiao, Xiaofei Liao, and Hai Jin, HeterRAG: Heterogeneous Processing-in-Memory Acceleration for Retrieval-augmented Generation, In proceedings of the Annual International Symposium on Computer Architecture (ISCA), 2025.

[6] Chaoqiang Liu, Dan Chen, Yu Huang, Wenjing Xiao, Haifeng Liu, Yi Zhang, Huize Li, Xiaofei Liao, and Hai Jin, SeIM: In-Memory Acceleration for Approximate Nearest Neighbor Search, In proceedings of the ACM/IEEE Design Automation Conference (DAC), 2025.

[7] Zhuohui Duan, Hao Feng, Haikun Liu, Xiaofei Liao, Hai Jin, Bangyu Li, AegonKV: A High Bandwidth, Low Tail Latency, and Low Storage Cost KV-Separated LSM Store with SmartSSD-based GC Offloading, In proceedings of the 23rd USENIX Conference on File and Storage Technologies (FAST), 2025.

[8] Haodi Lu, Haikun Liu, Yujian Zhang, Zhuohui Duan, Xiaofei Liao, Hai Jin and Yu Zhang, Fast Distributed Transactions for RDMA-based Disaggregated Memory, In Proceedings of 2025 USENIX Annual Technical Conference (ATC）, 2025.

[9] Jianjun Zhao, Haikun Liu, Shuhao Zhang, Haodi Lu，Yancan Mao，Zhuohui Duan, Xiaofei Liao, and Hai Jin,Towards High-Performance Transactional Stateful Serverless Workflows with Affinity-Aware Leasing，In Proceedings of 2025 USENIX Annual Technical Conference (ATC)，2025

[10] Bing Tian, Haikun Liu, Yuhang Tang, Shihai Xiao, Zhuohui Dian, Xiaofei Liao, Hai Jin, Xuecang Zhang, Junhua Zhu, and Yu Zhang, Towards High-throughput and Low-latency Billion-scale Vector Search via CPU/GPU Collaboration Filtering and Re-ranking，In proceedings of the 23rd USENIX Conference on File and Storage Technologies （FAST），2025.

[11] Jin Zhao, Qian Wang, Ligang He, Yu Zhang, Sheng Di, Bingsheng He, Xinlei Wang, Hui Yu, Hao Qi, Longlong Lin, Linchen Yu, Xiaofei Liao, Hai Jin, TempGraph: An Efficient Chain-driven Temporal Graph Computing Framework on the GPU, In Proceedings of the 30th ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2025.

[12] Jin Zhao, Yu Zhang, Jun Huang, Weihang Yin, Hui

Yu, Hao Qi, Zixiao Wang, Longlong Lin, Xiaofei Liao, Hai Jin, A Data-Centric Hardware Accelerator for Efficient Adaptive Radix Tree, In Proceedings of the 62nd IEEE/ACM Design Automation Conference (DAC), 2025

[13] Yutao Fu, Zhongtian Long, Yu Zhang, Zirui He, Jin Zhao, Qiyuan Niu, Zixiao Wang, and Hai Jin, PairGraph: An Efficient Search-space-aware Accelerator for High-performance Concurrent Pairwise Queries, In Proceedings of the 62nd Annual Design Automation Conference (DAC), 2025

[14] Yukang Dong, Wenbin Jiang, Xinhai Shen, Haihong Guo, Zhiyuan Shao, Hai Jin, BRP-SpMM: Block-Row Partition Based Sparse Matrix Multiplication with Tensor and CUDA Cores, In Proceedings of 39th IEEE International Parallel & Distributed Processing Symposium (IPDPS), 2025.

[15] Jianrong Yan, Wenbin Jiang, Dongao He, Suyang Wen, Yang Li, Hai Jin, Zhiyuan Shao, RT-GNN: Accelerating Sparse Graph Neural Networks by Tensor-CUDA Kernel Fusion, ACM Transactions on Architecture and Code Optimization, 2025.

[16] Yi Zhang, Xiaomeng Yi, Yu Huang, Jingrui Yuan, Chuangyi Gui, Dan Chen, Long Zheng, Jianhui Yue, Xiaofei Liao, Hai Jin, and Jingling Xue, Cheetah: Accelerating Dynamic Graph Mining with Grouping Updates, ACM Transactions on Architecture and Code Optimization, 2025.

[17] Long Zheng, Bing Zhu, Pengcheng Yao, Yuhang Zhou, Chengao Pan, Wenju Zhao, Xiaofei Liao, Hai Jin, Jingling Xue, A Priority-Aware Hardware/Software Co-design for High-Throughput Graph Processing Acceleration, ACM Transactions on Architecture and Code Optimization, 2025.

[18] Haikun Liu, Bing Tian, Zhuohui Duan, Xiaofei Liao, and Yu Zhang, A SmartSSD-based Near Data Processing Architecture for Scalable Billion-point Approximate Nearest Neighbor Search, ACM Transactions on Storage, 2025

[19] Yingshuai Dong, Chencheng Ye, Haikun Liu, Liting Tang, Xiaofei Liao, Hai Jin, Cheng Chen, Yanjiang Li, Yi Wang, DTAP: Accelerating strongly-typed programs with data type-aware hardware prefetching, ACM Transactions on Architecture and Code Optimization, 2025.

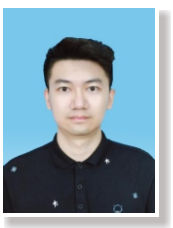
[20] Jianjun Zhao, Yancan Mao, Zhonghao Yang, Haikun Liu, Shuhao Zhang, Scalable Transactional Stream Processing on Multicore Processors, IEEE Transactions on Knowledge and Data Engineering, 2025.

[21] Jin Zhao, Yu Zhang, Donghao He, Qikun Li, Weihang Yin, Hui Yu, Hao Qi, Xiaofei Liao, Hai Jin, Haikun Liu, Linchen Yu, Zhang Zhan, An Efficient ReRAM-based Accelerator for Asynchronous Iterative Graph Processing, ACM Transactions on Architecture and Code Optimization, 2025.

[22] Fubing Mao, Zihan Xie, Longyu Nie, Yu Zhang, Haikun Liu, Xiaofei Liao, Hai Jin, Wei Zhang, Yapu Guo, Jingkang Liu, CEGraph: Cache-Efficient Management for Streaming Graph Processing, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2025.

[23] Yukang Dong, Fanxing Pan, Yi Gui, Wenbin Jiang, Yao Wan, Ran Zheng, and Hai Jin. Comprehensive Architecture Search for Deep Graph Neural Networks, IEEE Transactions on Big Data, 2025.

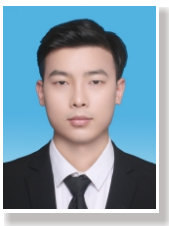
[24] Fubing Mao, Xu Liu, Yu Zhang, Haikun Liu, Xiaofei Liao, Hai Jin, Wei Zhang, Jian Zhou, Yufei Wu, Longyu Nie, Yapu Guo, Zihan Jiang, and Jingkang Liu, PMGraph: Accelerating Concurrent Graph Queries over Streaming Graphs, ACM Transactions on Architecture and Code Optimization, December 2024.



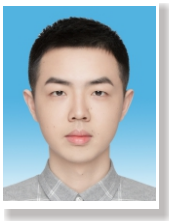
董雨康
博士研究生
研究方向：体系结构与系统软件、图计算、稀疏计算
Email：ykdong@hust.edu.cn

赵建军
博士研究生
研究方向：系统软件、数据库和流处理系统设计
Email：curry_zhao@hust.edu.cn

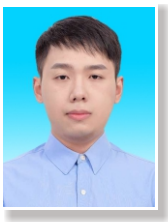
段卓辉
博士后
研究方向：内存计算
Email：zhduan@hust.edu.cn



赵　进
副教授
研究方向：图计算系统软件和体系结构
Email：zjin@hust.edu.cn



黄　禹
副教授
研究方向：图计算和体系结构
Email：yuh@hust.edu.cn



姚鹏程
讲　师
研究方向：图计算和体系结构
Email：pcyao@hust.edu.cn

王庆刚
副教授
研究方向：可重构计算机体系结构
Email：qgwang@hust.edu.cn

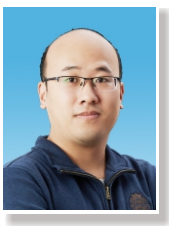
李钦宾
教　授
研究方向：机器学习系统、大模型智能体
Email：liqinbin3@gmail.com

张书豪
教　授
研究方向：大模型推理系统的设计与优化
Email：shuhao_zhang@hust.edu.

毛伏兵
讲　师
研究方向：系统软件和体系结构、集成电路物理设计
Email：fbmao@hust.edu.cn

叶晨成
副教授
研究方向：新型内存编程模型，软硬件协同设计
Email：yecc@hust.edu.cn

郑　龙
教　授
研究方向：可重构计算机体系结构及其运行环境
Email：longzh@hust.edu.cn



张　宇
教　授
研究方向：体系结构与系统软件、高性能图计算
Email：zhyu@hust.edu.cn

刘海坤
教　授
研究方向：内存计算
Email：hkliu@hust.edu.cn

邵志远
教　授
研究方向：体系结构和系统软件
Email：zyshao@hust.edu.cn

蒋文斌
教　授
研究方向：体系结构、图计算、深度学习系统
Email：wenbinjiang@hust.edu.cn

廖小飞
教　授
研究方向：系统软件和体系结构
Email：xfliao@hust.edu.cn

分布式系统组典型成果介绍

黄　卓、余庚花、罗瑞坤、戴小海、黄　航、张晓今、杜冰倩、
王　雄、姚德中、顾　琳、肖　江、余　辰、何　强、吴　松

关键词：分布式系统，区块链，边缘计算，
算网融合

1 介绍

分布式系统组长期致力于云计算、虚拟化、数据中心、区块链、边缘计算、算力网络等领域研究。基于本组所承担的“智能算力网络关键技术体系研究及验证”科技创新2030“新一代人工智能”重大项目，以及“服务器无感框架和协同调度”、“高并发可扩展区块链存储的基础理论和方法研究”、“硬件可适配高效率智能计算支撑环境”等国家重点研发计划项目/课题，以及“面向分布式异构计算虚拟化技术与软件”、“面向边缘计算环境的分布式数据去重方法研究”、“面向大规模分布式图神经网络高效训练的资源管理机制”等国家自然科学基金项目和一批企业合作项目，本组在四个方面开展了主要研究工作：在分布式系统软件方面针对服务无感计算的运行时优化、质量保障和镜像仓库管理三个方向展开研究；在区块链方面，针对智能合约执行加速、动态查询索引优化、及低延迟拜占庭共识等方向展开研究；在边缘计算方面，针对多模态数据融合、动态任务保障及高效联邦学习等方向展开研究；在分布式算网融合研究方面，针对算力网络资源管理和调度等展开研究。

1）在分布式系统软件方面，分布式系统软件研究内容聚焦于服务无感计算的运行时优化、质量保障和镜像仓库管理三个方向。针对服务器无感计算的运行时，提出了一种基于语言级虚拟化技术的高效数据传输技术，提升运行时的运行效率。针对有状态服务器无感知应用，提出了一种在线迁移技术，可以在秒级宕机时间下完成服务的迁移。针对分布式镜像仓库中存储粒度和拉取时延的权衡问题，提出MIS多粒度镜像存储规划策略，实现了存储空间高效利用与镜像拉取时延的大幅降低。

2）在区块链研究方面，主要聚焦于图式智能合约执行加速、动态查询索引优化、及低延迟拜占庭共识等。设计了分支逻辑感知的图式细粒度预执行机制，通过两级分支预测与基于检查点的快速路径执行，解决了图式区块链在复杂分支逻辑下预执行准确性低且低效的难题。提出了高效可验证的动态查询索引系统，结合强化学习优化索引选择及轻量化验证结构，显著提升查询效率并降低存储开销。设计了基于乐观路径的图式拜占庭协议，通过引入乐观路径机制，降低了图式共识延迟。提出了适应混合网络的多值拜占庭共识协议，通过引入异步二值共识，提升了多值共识协议网络的自适应能力。

3）在边缘计算研究方面，主要聚焦于复杂场景下的精准感知、多模态数据融合、动态任务保障及高效联邦学习等核心方向。针对高分辨率卫星影像中建筑物尺度差异大、边界模糊的挑战，提出基于不确定性图的分层注意力网络系统，通过不确定性聚焦变换精准强化关键区域建模，显著提升复杂场景下建筑物分割精度；针对智能制造中工人行为识别受单一传感器局限的问题，研制多模态神经网络，融合视觉时空特征与动作轨迹建模，有效解决遮挡、

动作相似等复杂工况下的识别难题。针对边缘智能系统的高并发排队时延与设备移动迁移问题，提出计算任务保障机制，通过差异化早退出策略动态平衡排队与迁移开销，保障任务执行效率；针对边缘设备算力受限下的联邦学习扩展性瓶颈，研发高效分割式联邦学习系统，通过数据并行与特征优化显著降低存储开销并提升模型收敛速度。此外，还研制了分布式稀疏学习框架、异构矩阵乘法协同方法及移动端推理加速方案等，在通信、计算与推理效率上取得突破，构建了覆盖感知、融合、任务保障与分布式优化的关键技术体系。

4）在分布式算网融合研究方面，聚焦算力网络资源管理与调度，针对复杂环境中的深度模型训练挑战，提出多项创新方案：设计弹性流水线训练框架，通过动态建模迭代时间与模型陈旧度，自适应调整分区并支持参数换出与迁移；研发MoE训练加速系统，通过异步预取热门专家参数和全局调度优化，降低通信开销与计算闲置；提出流水线化纵向联邦学习框架，交织模型更新与统计信息交换，减少通信开销；开发动态图神经网络训练框架，通过图式拓扑调度优化批次依赖，提升训练速度与模型精度；融合深度强化学习与联邦聚合，通过DDPG算法实现聚合权重精细控制与客户端选择；此外，揭示隐私保护LLM推理中隐私与效用的权衡关系，为隐私保护与模型效用的平衡提供理论指导。

2 分布式系统软件

分布式系统软件研究内容聚焦于面向用户自定义函数的高效WebAssembly沙箱运行时、增强服务弹性的有状态应用快速迁移以及多粒度分布式镜像仓库策略三个方向。针对WASM执行用户自定义函数过程中存在的数据传输开销问题，提出WAF运行时环境，通过将数据布局调整前移至编译阶段并利用共享内存消除数据拷贝，显著降低了UDF执行开销。针对Kubernetes中有状态Pod实时迁移的挑战，提出KubeSPT方案，通过控制网络状态同步、采用Hot Data and Lazy-Restore内存恢复方法以及解耦迁移操作，大幅减少了服务迁移的停机时间。对于分布式镜像仓库中存储粒度和拉取时延的权衡问题，提出MIS多粒度镜像存储规划策略，通过协同决策存储粒度与镜像分布，并设计具有近似比保证的优化算法，实现了存储空间高效利用与镜像拉取时延的大幅降低。

2.1 面向面向用户自定义函数的高效WebAssembly 沙箱运行时

用户自定义函数（User-Defined Functions，UDF）长期以来一直是扩展数据管理系统功能的标准方法。随着WebAssembly（WASM）的出现，UDF的依赖项（如语言运行时和库）可以被编译为一个WASM模块，并通过实例化来执行UDF。这种方式带来了几个关键优势：1）开发者可以使用自己熟悉的编程语言编写UDF，而不再受限于数据库引擎原生支持的语言；2）UDF的依赖被封装在WASM模块内部，避免了因主机上依赖冲突带来的错误风险；3）具备良好的跨平台兼容性，可在不同的数据库引擎、操作系统和体系结构间无缝运行。

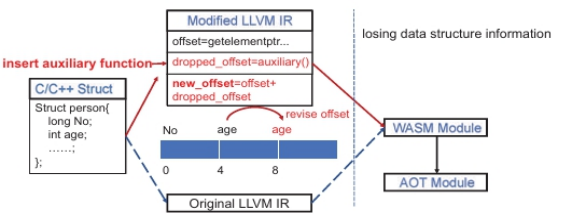


图1 WAF辅助函数

然而分析发现，WASM模块执行UDF的过程中会引入显著的数据传输开销。这一过程涉及数据库引擎与WASM运行时之间的数据布局调整与数据拷贝，严重影响了性能表现。为了解决这一问题，提出了WAF，一个基于WASM

的UDF执行环境，如图1所示。WAF将数据布局调整从执行阶段前移至编译阶段，并通过共享内存消除了数据拷贝。实验结果表明，WAF能将WASM-based UDF的执行开销降低3.1倍，相较于基于容器的方式提速18.1倍。

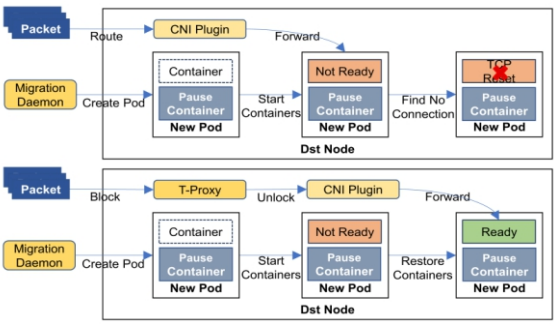


图2 KubeSPT的架构

2.2 增强服务弹性的有状态应用快速迁移

容器编排系统（如 Kubernetes）简化了容器化应用程序的部署。随着越来越多的应用程序在 Kubernetes 中部署，由于系统升级、节点故障和负载均衡优化等原因，重调度（将正在运行的应用迁移到不同的节点）的需求日益增加。实时迁移是应用重调度的理想手段。然而，由于 Kubernetes将 Pod 视为无状态的，因此实现运行有状态服务的 Pod 的实时迁移颇具挑战性。因此，提出了 KubeSPT 以实现重调度场景下有状态 Pod 的实时迁移，如图2所示。首先，通过控制数据流来同步 Pod 和内部容器的网络状态，并实现快速服务重定向。其次，引入了一种Hot Data and Lazy-Restore方法来进行内存恢复。最后，将 Pod 迁移操作与其他 Kubernetes 操作解耦，以确保与实时迁移兼容。实验结果表明，与当前的重调度方法相比，KubeSPT 将停机时间减少了 86%-93%。

2.3 多粒度分布式镜像仓库策略

为了应对日益增长且随时间变化的服务请求，基于容器的微服务作为一种有前景的服务提供范式逐渐兴起。尽管容器相对轻量，但其开销并非可以忽略。海量的容器镜像存储与高并发的镜像拉取导致镜像仓库存储利用率受限并显著延长镜像拉取时延。为应对这一挑战，现有工作提出了分布式镜像仓库的概念，以缓解集中式仓库的存储压力并提升镜像拉取效率。然而常用镜像中存在大量文件重复，采取不同的存储粒度时，与文件级相比，层级存储平均需要多出约350.17%的存储空间；而采用文件级存储虽然可以有效去重，但由于需要将文件首先重建为镜像层才能进行拉取，会带来约11.84%的额外时延。

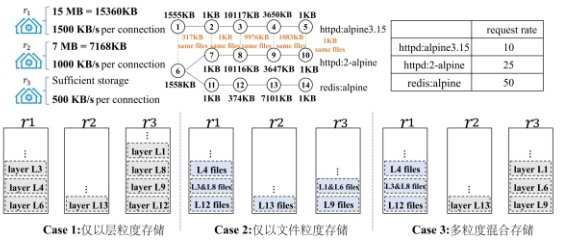


图3 常见镜像的分布式多粒度存储样例

为此，提出了面向分布式镜像仓库的多粒度镜像存储规划策略MIS，如图3所示。MIS进行两方面的协同决策：一方面，按镜像仓库选择仓库的存储粒度（层级或文件级）。另一方面，在给定仓库容量、带宽与连接数上，联合决定具体哪些层/文件分布到哪些仓库，并据此跨仓并行调度镜像分层拉取，从而在最大化利用存储空间的基础上，显著降低拉取时延。将该问题形式化为非线性混合整数规划，并进一步设计了具有近似比保证的随机舍入的算法实现放置优化。基于真实踪迹的实验表明，相比现有最先进方法，MIS能够在最大化利用存储空间的情况下，平均降低26.43%的镜像拉取时延。

3 区块链

区块链研究内容聚焦于图式智能合约执行

加速、动态查询索引优化、低延迟拜占庭共识及混合网络多值共识协议等。针对图式区块链预执行在复杂分支逻辑下准确性低且低效的问题，提出了分支逻辑感知的图式细粒度预执行机制 Seer，通过两级分支预测与基于检查点的快速路径执行，实现高效预执行结果复用。针对动态负载下区块链查询索引构建耗时、存储开销大的问题，提出了高效可验证的动态查询索引系统 FlexIM，结合强化学习优化索引选择及轻量化验证结构，提升查询效率并显著降低存储开销。针对现有图式 BFT 共识高延迟的瓶颈，提出了基于乐观路径的图式拜占庭协议 Remora，在乐观情况下实现与传统非 DAG 协议相当的3δ延迟，同时保持高吞吐与容错能力。针对多值共识协议缺乏对良好网络的适应能力问题，提出了适应混合网络的多值拜占庭共识协议 Pako，引入快速路径与异步二值共识保障一致性与活性。

3.1 分支逻辑感知的图式细粒度预执行机制

基于有向无环图并行拓扑的图式区块链通过高并发性提升了系统吞吐量，已成为业界的研究热点。现有的图式区块链系统在处理复杂智能合约时仍面临显著挑战，在交易执行阶段预执行技术虽能在关键路径外减少 I/O 和计算成本，但导致预执行与实际执行状态不一致，尤其在处理含多个状态相关分支的合约时，易引发预执行路径偏差。图式区块链预执行机制在面向复杂分支逻辑时准确性低且低效的问题，严重制约了图式区块链的性能。

提出了一种新型区块链智能合约执行引擎 Seer，如图4所示。通过预测所有与状态变量相关的分支方向提升了预执行结果的可复用性，从而增强预执行对实际执行的加速效用，在实际区块链负载下实现高效的预执行结果复用与事务加速。使用真实以太坊负载进行了实验评估，实验结果表明，与原生以太坊及现有预执行方案Forerunner和MTPU相比，Seer 在执行阶段平均提供 27.7 倍的事务级加速和 20.6 倍的总体加速，显著优于现有区块链执行加速解决方案。

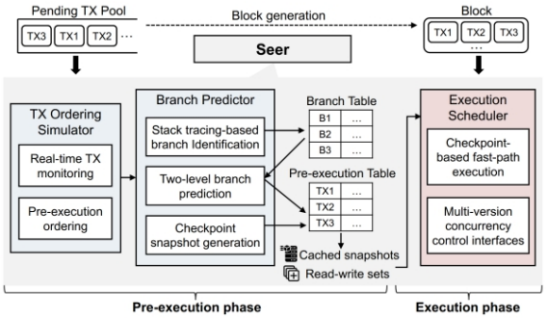


图4 执行引擎Seer的预执行机制

3.2 高效可验证的区块链动态查询索引方法

基于区块链的查询凭借可追溯性和数据溯源能力被广泛应用，但现有基于索引的查询方法仅在静态负载（查询属性或类型固定）下高效，面对动态负载时，因索引构建时间过长、存储消耗过大，难以构建高效索引，这一问题严重制约了区块链在动态查询场景中的实用性。

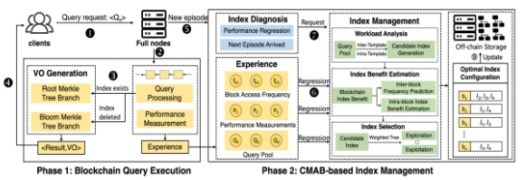


图5 FlexIM动态索引管理架构

为此，提出了一种高效且可验证的区块链动态查询索引管理系统 FlexIM，如图5所示。FlexIM能够在保证查询可验证性的同时显著提升效率，并在实际区块链负载下实现动态索引优化与低存储开销。具体而言，FlexIM 通过挖掘区块链的数据分布和块访问频率特征，利用强化学习技术在动态负载下优化选择索引；设计分层候选索引生成方法压缩索引选择空间，结合加权树平衡探索与利用以降低选择开销。

此外，针对索引验证的存储开销问题，利用Root Merkle Tree（RMT）和Bloom Filter Merkle Tree（BMT）增强可验证性，实现低存储开销的查询结果验证。

使用真实比特币数据集进行了实验评估，结果表明，与国际前沿的区块链查询机制vChain + 相比，FlexIM在平均查询速度上提升26.5%，同时存储消耗减少94.2%，显著优于现有区块链索引管理方案。

3.3 基于乐观路径的低延迟图式拜占庭协议

按照网络环境假设来划分，BFT共识协议可以划分为同步、半同步和异步共识。由于同步和半同步网络环境对网络攻击的抗性较为脆弱，最近的研究集中在提高异步BFT共识的效率上。为提示共识吞吐率，协议设计中引入了有向无环图（Directed Acyclic Graph, DAG）结构。然而，现有图式协议也伴随着高延迟的缺点。举例而言，在DAG-Rider这篇开创性工作之中，它提出了一个良好的延迟指标10δ（δ代表实际网络延迟），后续的图式BFT共识均努力减少协议的延迟，最新的工作GradedDAG实现了4δ的良好延迟，但对比传统的非DAG结构共识的最佳延迟，如PBFT的3δ，仍有提升空间。

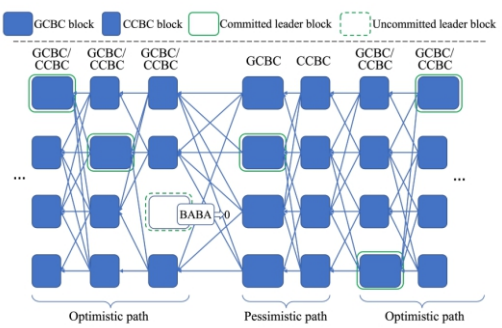
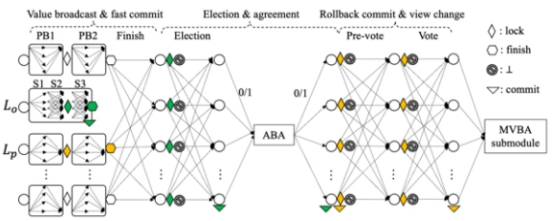


图6 Remora共识流程图

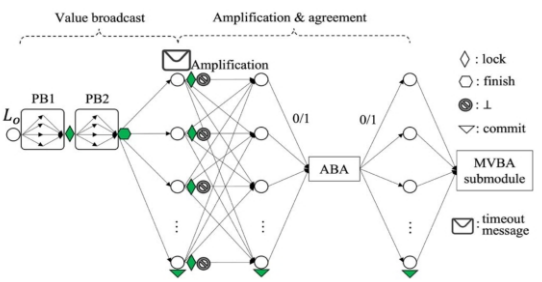
对此，提出了Remora，一种基于DAG框架的BFT协议并实现了3δ的延迟，如图6所示。Remora与传统的DAG共识相比，其核心在于将乐观路径融入到共识设计中。在乐观情况下，协议通过乐观路径达成共识可以实现3δ的最佳延迟；而若情况不佳，Remora也可通过悲观路径正常的选举当前轮次的leader来完成数据提交，并迅速切换回乐观路径继续共识。实验结果显示，在良好无故障的环境中，Remora吞吐率可分别达到GradedDAG和Tusk的1.24倍和2.25倍，并可降低37.9%和28.5%的延迟；而在有恶意节点作恶的环境中，Remora也表现出优秀的性能。

3.4 适应混合网络的多值拜占庭共识协议

为解决半同步共识在网络不稳定时丧失活性的问题，异步共识协议近年来得到了广泛关注。大量异步协议采用多值拜占庭共识（Multi-valued Byzantine Agreement，MVBA）作为核心组件，其在极端网络环境下同时保证了一致性与活性。然而，现有各类多值共识协议普遍缺乏对良好网络的适应能力，具体表现为其在共识性能上与半同步共识协议存在显著差距。



(a) 基于互广播路径的 Pako1 协议



(b) 基于单广播路径的 Pako2 协议

图7 Pako协议示意图

针对该问题，提出了一类新型多值共识协议Pako，如图7所示。该协议引入了提交预选主

节点区块的快速路径，节点直接提交携带聚合签名的该区块，减少了良好网络下的共识轮次。同时，协议利用异步二值共识（Asynchronous Binary Agreement，ABA）使所有节点一致选择快速路径或回退路径，保证协议的安全性及恶意攻击情况下的活性。通过权衡共识延迟与消息复杂度，提出了 Pako 的两种变体：Pako1 和 Pako2。其中，Pako1 在 O(n2) 的消息复杂度下实现了最优 3 轮的共识延迟，Pako2 则在最优情况下将消息复杂度降低为 O(n)，并实现了 5 轮的共识延迟。实验结果表明，Pako1 和 Pako2 高效适应了兼具半同步与异步特征的混合网络。具体而言，与当前最先进的多值共识协议 sMVBA 相比，Pako1 和 Pako2 可分别将延迟降低 51% 和 32%。

4 边缘计算

在边缘计算研究方面，主要聚焦于复杂场景下的精准感知、多模态数据融合、动态任务保障、高效联邦学习及分布式优化等核心方向。针对高分辨率卫星影像中建筑物尺度差异大、边界模糊的挑战，提出基于不确定性图的分层注意力网络系统，显著提升了复杂场景下建筑物分割的精度与稳定性；针对智能制造中工人行为识别受单一传感器局限的问题，提出基于摄像头与IMU传感器的多模态神经网络，增强了识别准确性与鲁棒性。

针对边缘智能系统中高并发推理导致的排队时延与设备移动触发的计算迁移问题，提出基于高排队时延和计算迁移的计算任务保障机制，保障了任务执行效率；针对边缘设备算力受限下联邦学习模型训练的扩展性瓶颈，提出基于数据并行的高效分割式联邦学习系统，显著降低了存储开销并提升了模型收敛速度与精度；此外，还研制了高效分布式稀疏相似性学习框架、异构稀疏矩阵乘法协同方法、移动端ViT推理加速方法及异构模型联邦学习框架，分别在通信成本、计算吞吐、移动推理效率及异构设备协同训练上取得突破。

4.1 城市规划中基于不确定性图的分层注意力网络系统识别建筑物信息

随着遥感成像技术的不断进步，获取到的卫星图像在分辨率和细节表现力方面都有了显著提升。然而，这也给利用高分辨率影像进行建筑物分割也带来了新的挑战：一方面，城市中的建筑物往往尺度差异极大，从宏观的高层建筑到微小的附属设施均需被精确识别；另一方面，许多建筑物边缘呈现近似直线的几何特征，这种规律在视觉特征上既容易与背景混淆，又使得边界区域像素存在较高的不确定性。

传统基于人工特征的方法受限于手工特征表达能力有限，难以捕捉复杂场景下的多尺度信息，往往导致分割结果粗糙、不稳定。尽管近年来深度学习方法凭借强大的高维特征提取能力显著提升了分割性能，但依然存在不足：当前主流的注意力机制多以全局空间或通道维度为对象生成权重图，这种“均匀化”的建模方式容易忽视真正困难的区域，例如建筑物的边界像素、低置信度区域或小型目标。在这些关键部位缺乏精细化建模，会直接造成边缘模糊、轮廓断裂或小目标漏检，从而降低整体分割精度。

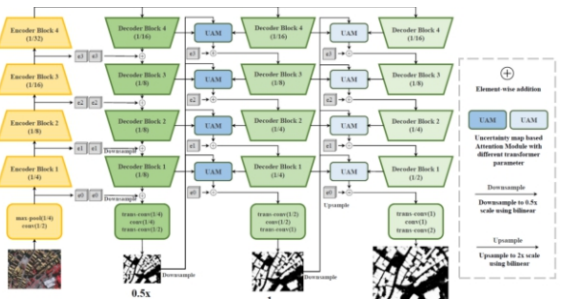


图8 基于不确定性图的建筑图像分割分层注意力网络系统

为了解决上述问题，基于不确定性图的分层注意力网络系统，如图8所示。与传统注意力机制普遍均匀作用于整个空间不同，基于不确

定性图的分层注意力网络将有限的计算与学习能力集中在最容易出错的部分，提高分割的针对性与有效性。其核心思路是：首先，利用分割网络的 Sigmoid 输出生成不确定性图，通过像素值的分布显式标识出边界区域和小目标区域中难以判别的低置信度像素。接着，设计了不确定性聚焦变换（Uncertainty Focal Transformer, UFT），将不确定性图进一步转化为注意力图，从而在特征空间中对这些“模糊”区域进行重点建模与强化。

为了验证所提出方法的有效性，在公开建筑物分割数据集上进行了实验，并将其与 主流模型进行对比。实验结果表明，基于不确定性图的分层注意力机制在关键指标上均取得了更优表现，能够在复杂建筑物分割任务中有效提升边界识别的精度与整体分割质量。

4.2 智能制造场景下基于摄像头与 IMU 传感器的多模态神经网络识别工人行为

在工业 4.0 时代，智能工厂逐渐普及，但由于小批量、个性化生产订单的仍然存在，完全依赖自动化生产线的成本与灵活性难以满足实际需求，工人依旧在装配、检测等环节中发挥不可替代的作用。与此同时，工人的失误或不规范操作不仅可能造成产品质量问题，还会带来安全隐患和生产效率的下降，因此对工人活动进行实时、准确的识别与监测，已成为智能工厂管理中的关键任务。现有的工人活动识别方法通常依赖单一传感器，例如基于摄像头的视觉识别或基于 IMU 的动作感知。虽然这些方法在标准场景下能够取得一定成效，但在实际工业环境中往往存在复杂挑战：视觉传感器容易受到遮挡影响，导致关键动作丢失；IMU 传感器则容易在动作相似或个体差异较大时发生混淆。此外，对于细微且不规则的动作，例如测量、微调等操作，单一模态传感器的识别能力明显不足。

而随着工业物联网和传感器技术的快速发展，工厂环境中可用的数据源日益多样化，多模态数据融合逐渐成为解决该问题的重要方向。通过将摄像头采集的视觉数据与 IMU 提供的动作数据进行互补性融合，不仅能够缓解单模态在遮挡、动作相似和个体差异等方面的局限，还能显著提升识别结果的准确性与稳定性。因此，如何设计高效的多模态数据神经网络来整合不同传感器的数据，成为智能制造场景的重要问题。

针对上述问题，提出了一种基于摄像头与 IMU 传感器的多模态神经网络，如图9所示。该方案包括三个核心环节：1）视觉特征提取:利用 3D 卷积神经网络提取工人视频数据的时空特征；2）动作特征提取：通过 长短期记忆网络网络从 IMU 传感器数据中学习长期依赖关系，捕获运动轨迹信息；3）多模态融合：在特征级实现通道堆叠与卷积关联，充分挖掘视觉与运动数据之间的互补性。

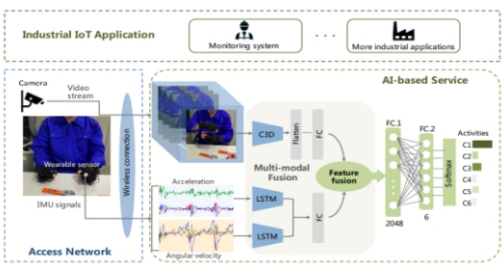
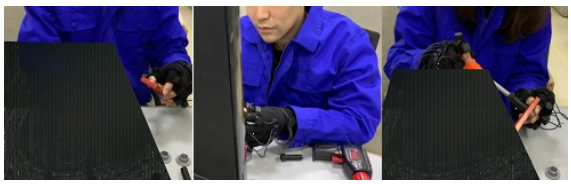


图9 基于摄像头和IMU传感器的智能工厂活动识别多模态神经网络

为了验证所提出的方案的性能，分别开展了针对视频数据识别以及IMU数据识别的实验。实验结果表明，多模态方法在工人的活动识别场景下是更有效的。

同时，为了进一步验证模型在复杂工况下的性能，构造了三类智能制造典型场景（如图10 所示），并分别在摄像头遮挡、任务不同但动作相似以及动作细微且不规律的情况下进行了实验验证。实验结果说明提出的模型能有效利用特征融合策略，提高了单一种类传感器的

识别准确度和精度。



(a) 视觉遮挡情况

(b) 任务不同但是动作类似情况

(c) 动作细微且不规律情况

图10 工业场景中影响活动识别的三种情况

4.3 边缘智能下基于高排队时延和计算迁移的计算任务保障机制

在边缘智能系统中，终端设备与边缘服务器协同运行，以便为用户提供低时延的 AI 服务。然而，随着深度神经网络规模和复杂度的不断提升，模型推理的计算开销和冗余计算显著增加，使得边缘服务器在高并发场景下容易出现任务堆积与排队时延攀升的现象，直接影响服务的响应延迟。同时，在实际部署中，终端设备往往处于移动环境中，这进一步拉长了任务完成的整体时延。

现有的优化研究主要集中在模型压缩、模型划分以及资源调度等方向，尽管能够在一定程度上缓解计算压力，却大多假设运行环境相对静态，难以应对任务排队与计算迁移交织产生的动态延迟问题。因此，如何在这种高度动态且资源受限的边缘智能场景下，综合考虑排队延迟与迁移开销，并在保障模型精度的同时确保任务能够按时完成，成为亟需解决的关键问题。

为了解决上述问题，提出了基于高排队时延和计算迁移的计算任务保障机制，以保障计算任务能够按时完成，从而保障服务质量。首先，将计算任务分为新到达的计算任务和部分完成的计算任务两类。针对这两类任务特点，提出了一种基于高排队时延与计算迁移的计算任务保障机制，核心思路是引入模型早退出技术，如图11所示。其核心思想是：对于新到达的计算任务，设计基于排队时延的早退出机制，通过优先队列与迭代策略为任务选择合适的退出点，降低排队导致的超时风险；对于迁移中的部分完成任务，设计基于任务完成状态的早退出机制，动态调整退出点，避免因迁移开销导致任务失败。

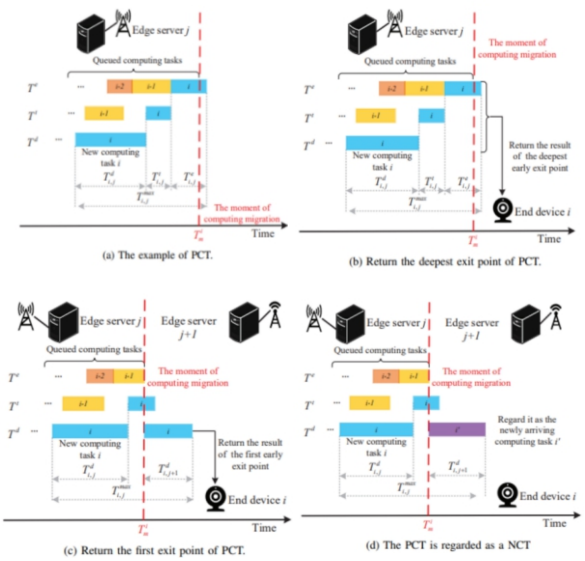


图11 部分完成的计算任务以及早退出点调整后的效果

为了验证所提出的系统机制的性能，设计并开展了一系列的实验。实验结果表示，所设计的机制在保证较高计算任务完成率的同时，能够有效优化任务执行时延，为边缘计算环境中的计算任务执行提供了支持。

4.4 基于数据并行的边缘联邦学习系统

近年来，联邦学习（Federated Learning, FL）

作为一种隐私保护的分布式机器学习框架，受到了广泛关注。然而，随着深度学习模型规模不断扩大，FL 在边缘设备上的训练受到算力与存储限制的严重制约。为缓解这一问题，研究者们提出了将分割学习（Split Learning, SL）引入联邦学习，从而将大规模模型划分为客户端和服务端两个部分，以降低客户端的计算压力。然而，现有的分割式联邦学习系统在服务端需要为每个客户端维护独立的服务端子模型，既导致了巨大的 GPU 计算开销，又造成了存储资源的指数级增长，严重限制了实际应用中的可扩展性。

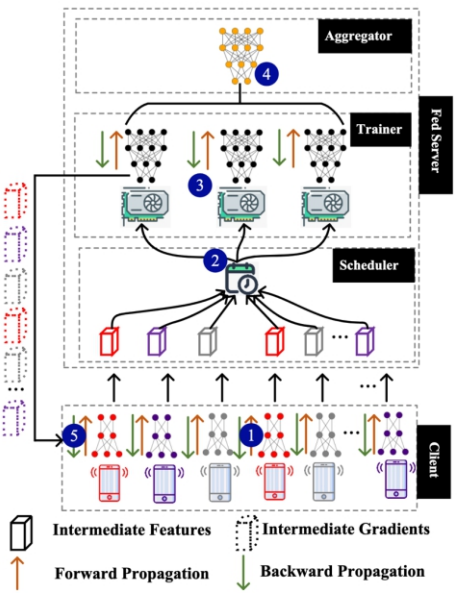


图12 Hourglass系统架构

针对上述问题，提出了一种高效的分割式联邦学习系统——Hourglass，如图12所示。其核心思想是利用数据并行（Data Parallelism）来降低服务端计算与存储开销，并通过新的特征调度策略提升模型收敛速度和精度。Hourglass在设计上具有三个显著特点：首先，在模型维护方面，不同于现有方法为每个客户端分配独立的服务端模型分区，Hourglass在单GPU场景下仅维护一个共享模型，在多GPU场景下则维护与GPU数量一致的模型分区；其次，在特征调度上，Hourglass提出“Dissimilar Feature First (DFF)”方法，将差异化的中间特征优先分配到同一GPU进行训练，有效避免模型陷入局部最优并加速收敛；最后，在特征聚类上，Hourglass 引入局部敏感哈希（Locality-Sensitive Hashing, LSH）代替传统的k-means算法。

实验结果表明，Hourglass在四个公开数据集和五类主流模型上的实验中，相较于现有分割式联邦学习方法，能够实现最高35.2倍的收敛加速，并带来最高9.28%的精度提升。此外，在单GPU条件下，Hourglass不仅显著降低了开销，还展现出更强的知识融合能力，使得全局模型的收敛精度优于传统SplitFed方案。该研究成果首次系统性地解决了分割式联邦学习在计算和存储上的扩展性瓶颈，推动了联邦学习在资源受限的边缘环境下的落地应用。

4.5 高效分布式稀疏相对相似性学习

相似性度量是信息检索、推荐系统与模式识别中的核心环节，但在分布式场景下，传统方法常受限于计算与通信成本。现有小批量随机梯度下降（Stochastic Gradient Descent, SGD）在并行训练中虽可加速，但对大批量的依赖带来内存与收敛效率的矛盾。为此，设计了一种新的 高效分布式稀疏相对相似性学习（Efficient Distributed Sparse Relative Similarity Learning, EDSRSL）框架，如图13所示。通过在分布式环境中引入稀疏建模与局部小批量SGD策略，有效缓解了大规模并行训练中通信与计算的双重瓶颈。

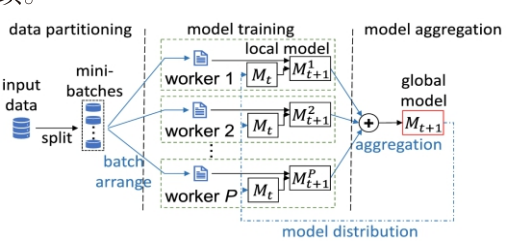


图13 EDSRSL框架训练工作流

一方面，稀疏约束显著降低了高维模型的存储与计算开销；另一方面，局部更新与延迟聚合减少了频繁的通信需求，从而提升整体效率。同时，批次重复策略提高了数据利用率，使得单次通信带来更多有效更新。理论分析表明，该方法能够在保证最优收敛率的同时实现近似线性加速。在多个真实大规模数据集上的实验验证了该框架的优势：EDSRSL不仅在精度上可与已有方法媲美，甚至在部分任务上超过传统的OASIS和SORS算法，更在通信与时间成本上展现出巨大优势，最高可减少90%以上通信负担，并实现超过5倍的加速。进一步的实验还显示，改进版本AdaEDSRSL在大规模稀疏数据集上表现尤为突出，通过引入自适应正则项在收敛速度和模型稀疏性上取得更好平衡。

4.6 通过异构协作和自适应面板加速大规模稀疏矩阵乘法

稀疏通用矩阵–矩阵乘法（SpGEMM）是许多应用（如代数多重网格法、图形处理和深度学习）的基础组件。然而，在GPU上计算高维、大规模稀疏矩阵乘法的延迟阻碍了这些应用的发展。一种有效方法是异构核心协同计算，但该方法必须解决三个方面的问题：（1）不规则的非零元素导致负载不平衡和不规则的内存访问；（2）不同核心的计算延迟差异降低了计算并行度；（3）不同核心间的临时数据传输引入了额外的延迟开销。

针对上述问题，提出了基于自适应面板的大规模稀疏通用矩阵乘法异构协同计算方法ApSpGEMM，如图14所示。ApSpGEMM通过一种四阶段协同计算框架加速大规模稀疏矩阵乘法：首先进行轻量级矩阵预分析以提取非零元素分布特征；接着基于稀疏性规则对矩阵行重排序并自适应地分割为稠密/稀疏面板；随后在GPU核内针对不同面板类型（SpGEMM/SpMM/DGEMM）分别优化线程分配与内存访问，实现负载均衡；最后引入核心亲和性度量，将面板动态分配给CPU或GPU计算，并通过异步传输策略重叠计算与数据通信，显著降低异构协同开销。

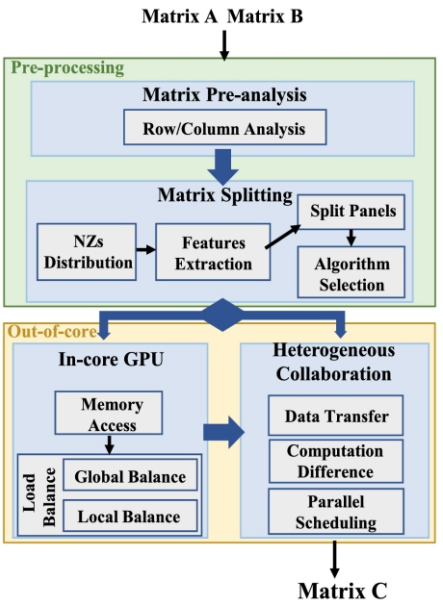


图14 ApSpGEMM工作流

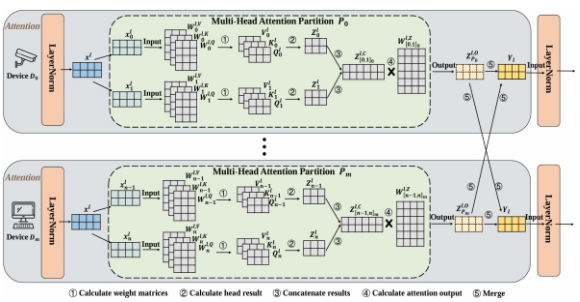
实验结果表明，ApSpGEMM在不同稀疏结构的大规模矩阵乘法任务中均表现出卓越性能。在GPU单设备环境下，相比cuSPARSE、AC-SpGEMM等先进方案，其计算吞吐量（GFlops）平均提升最高达2.31倍，峰值性能达到197.54 GFlops；在启用CPU-GPU异构协同计算后，通过自适应面板分配与异步传输优化，进一步将大规模矩阵乘法的性能相较于单一GPU提升2.25至7.21倍，有效克服了内存限制与跨设备通信瓶颈，显著加速了十亿级非零元素稀疏矩阵的运算效率。

4.7 通过自适应切分和卸载加速移动设备上ViT推理

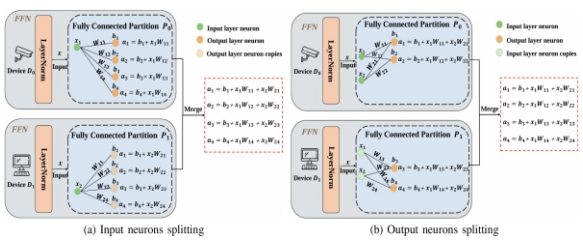
随着Vision Transformer（ViT）在计算机视觉任务中不断刷新性能纪录，其高精度表现已超越卷积神经网络（CNN）。然而，ViT依赖自

注意力机制，计算复杂度和参数规模远大于CNN，导致其在资源受限的移动设备上部署和推理困难。现有研究尝试通过模型压缩（剪枝、量化、蒸馏等）或硬件加速（GPU、FPGA、DSP等）来降低开销，但这些方法往往要在精度与延迟之间权衡，或者受限于设备算力，难以满足移动端实时应用和隐私保护的需求。

为解决这一挑战，提出了SPViT，一种基于自适应切分与协同卸载的推理加速方法，如图15所示。其核心思想是在边缘环境中利用多台可用设备（如手机、平板、摄像头等）的计算资源，实现ViT推理的并行化和低延迟执行，而非完全依赖单一设备或远程云端。ViT推理延迟的瓶颈主要集中在多头自注意力（MSA）层和前馈网络（FFN）层，其中矩阵乘法操作占据近80%的计算开销。为此，SPViT提出了两种细粒度的层内切分方式：Head-Width Splitting用于将MSA的多头划分到不同设备并行执行；Neuron-Width Splitting 用于将FFN层的神经元划分到设备执行。通过这种方式，ViT的计算负担可以均衡分配到多台设备。



(a) Head-Width Splitting



(b) Neuron-Width Splitting

图15 层内切分方式示意图

将SPViT部署在真实硬件平台（包括Manifold 2-G和Raspberry Pi 4B）上，并测试了DeiT、Swin Transformer和DaViT等多种主流ViT模型。结果表明，SPViT在不损失模型精度的情况下，相比单设备推理可实现2.2×至3.3×的加速效果，同时显著降低了计算和通信延迟；结合自适应卸载与异步推理机制，SPViT在动态带宽和设备异构环境中依然保持稳定高效的性能，验证了其即插即用的通用性与实用价值。

4.8 通过低秩分解的高效异构模型联邦学习

在现有联邦学习（Federated Learning, Fl）研究范式中，一个核心假设是所有本地模型共享相同的网络架构和大小。然而，对于具有不同硬件资源的设备（如智能手机、云服务器等），这一假设变得不切实际。不同客户端具有不同的计算能力和通信带宽。系统异构性会导致训练效率下降，尤其是掉队者的存在可能拖慢整体训练进程。直接要求所有客户端训练同一模型架构存在明显弊端。

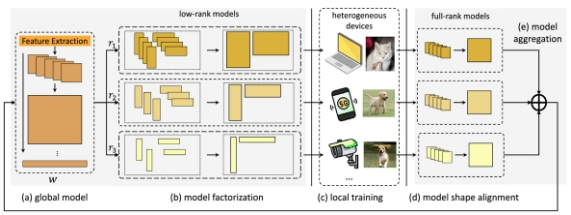


图16 FedHM框架

为了解决这一问题，提出了FedHM，一种采用低秩分解的异构模型联邦学习框架，如图16所示。服务器端将全秩大模型分解为多个低秩模型，并根据客户端硬件资源不同分发给各自设备。各客户端在本地仅训练适配自己能力的低秩模型。训练完成后，服务器将这些模型恢复为统一的全秩结构，并聚合为全局模型。该机制不仅能够支持不同计算能力设备的参与，还显著降低通信成本。

实验结果表明，FedHM 在多个基准数据集和不同的联邦学习场景下均优于现有方法。在保证全局模型精度的同时，有效降低了通信与计算开销。与传统同构方法相比，FedHM 能够支持不同规模和复杂度的异构模型协同训练，并在异构环境下保持稳定收敛。同时，小模型在与大模型共同训练时性能得到提升，体现了知识迁移效应，整体验证了该框架在效率、准确率和鲁棒性上的综合优势。

5 分布式算网融合

在分布式算网融合方面，聚焦算力网络资源管理与调度，针对复杂环境中的深度模型训练挑战，提出多项创新方案：设计弹性流水线训练框架DynPipe，通过动态建模迭代时间与模型陈旧度，自适应调整分区并支持参数换出与流水线迁移，应对GPU共享集群和云Spot实例中的任务抢占与网络波动；研发MoE训练加速系统PopFetcher，基于专家热度预测与通信–计算重叠机制，通过异步预取热门专家参数和全局调度优化，降低通信开销与计算闲置；提出流水线化纵向联邦学习框架BS-VFL，交织模型更新与统计信息交换，在保证精度同时减少通信开销；开发动态图神经网络训练框架PipeTGL，通过DAG调度优化批次依赖关系，减少流水线气泡，提升训练速度与模型准确率；融合深度强化学习与联邦聚合，设计FedAA框架，通过DDPG算法实现聚合权重精细控制与客户端选择，平衡模型鲁棒性与公平性；此外，揭示隐私保护LLM推理中隐私泄露与效用损失间的权衡关系，为隐私保护与模型效用的平衡提供理论指导。这些成果构建了覆盖训练、推理、通信与隐私保护的分布式算网融合关键技术体系。

5.1 具有干扰适应性的端到端的分布式深度神经网络训练框架

大规模神经网络训练通常需要在分布式环境中进行，然而当前主流分布式训练方法中，数据并行受GPU内存限制，模型并行因串行依赖导致资源利用率低。虽然流水线并行通过异步调度可部分缓解空泡时间，但仍面临静态分区策略，难以适应动态计算环境（如GPU共享集群、云Spot实例）中的任务抢占、节点性能波动和网络不稳定的问题。同时，为提高吞吐而采用的细粒度分区会加剧模型陈旧度和内存消耗，损害收敛效率。因此，如何在保证模型精度的同时，优化通信与计算资源的利用，提高训练效率，是亟待解决的重要问题。

为此，提出了一种面向动态计算环境的弹性流水线训练框架DynPipe，从系统与收敛性能的协同角度实现端到端优化，如图17所示。DynPipe通过精确建模多阶段异步流水线的迭代时间和模型陈旧度对收敛的影响，在分区过程中均衡硬件效率与统计效率，并依据运行时指标（如前向/反向时间、GPU状态、网络带宽）构建轻量级随机森林模型，以非侵入方式评估外部任务干扰、动态调整模型分区。DynPipe支持从输入到输出层渐进式分段将暂存参数换出至主机内存，减少显存占用并重叠计算和通信，支持流水线化迁移操作，优先传输训练流量，实现无中断的模型重部署与负载平衡。实验表明，DynPipe在真实数据集上优于现有系统，达到相同水平模型精度的训练收敛速度提了1.5至3.4倍。

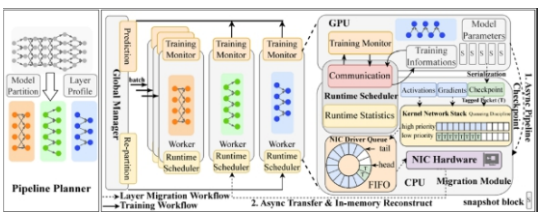


图17 动态神经网络训练自适应切割

5.2 基于热度统计专家预取的混合专家模型训练加速方案

在大模型遵循扩展定律持续增大规模的

背景下，模型算力需求已远超硬件摩尔定律的增长速度，导致训练资源不足。混合专家模型（Mixture-of-Experts，MoE）以稀疏激活的方式实现了万亿规模下的次线性计算需求增长，显著提升了模型容量与计算效率，但也引入了明显的系统瓶颈。具体而言，MoE训练过程中专家的激活高度依赖输入数据，导致工作负载波动剧烈、难以预测；同时，为支持稀疏激活必须引入专家并行，而其中两次All-to-All通信成为分布式MoE模型训练中的关键瓶颈。该通信模式不仅带来显著的延迟和带宽压力，还会引发负载不均衡，拖累整体训练效率，尤其在大规模集群中问题更为突出。

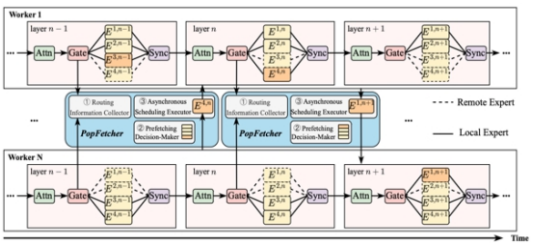


图18 热度统计MoE训练加速系统

为系统应对上述挑战，提出了一套基于实时热度预测与通信–计算重叠机制的综合优化方案PopFetcher，如图18所示。该系统通过滑动窗口动态追踪专家被选择的频率分布和层间关联特征，准确预测不同专家的实时“热度”，识别出高需求专家。在此基础上，系统利用非MoE层执行期间的计算间隙和空闲网络带宽，以异步预取的方式提前将热门专家参数分发至对应设备，大幅压缩后续All-to-All通信中需传输的 token 数量，同时结合CPU内存缓存机制避免参数的重复传输。为进一步提升系统效率，PopFetcher 还构建了端到端的延迟优化模型，通过全局调度和剪枝策略自动推导最优预取方案，并采用反向传播阶段的通信优先级调度机制，通过交替执行 All-to-All 和 All-Reduce 操作，有效隐藏通信延迟、缓解链路阻塞。实验表明，该系统在真实GPU集群环境中显著降低了MoE训练中的通信开销和计算闲置，相比现有方案最高减少94.5%的训练时间。

5.3 基于有界模型时效的流水线化纵向联邦学习

纵向联邦学习(VFL)支持各方之间的隐私保护协作，通过融合其地理分布数据特征来训练全局模型。当前的VFL系统编排训练过程通常遵循同步并行风格，需要频繁的参与方与服务器的统计信息交换，虽然同步VFL 易于部署并且可能具有最佳性能，但由于统计数据通常通过广域网(WAN)传输，因此训练时间可能会大大延长。相对于现有的同步VFL方案往往存在过多的通信开销，异步方案可能会引入严重的模型过时，从而潜在地降低学习的准确性。

为此提出了一种新的基于运行时感知的异步VFL框架BS-VFL，如图19所示。BS-VFL用于流水线化模型更新和统计数据交换，以实现远程通信与本地计算的重叠。该框架具有有限的模型过时性，它将训练计算和统计数据传输交织在一起，在不牺牲模型精度的情况下大大减少了通信开销。严格表征了BS-VFL的收敛误差，并证明了BS-VFL可以达到与理论上最优同步VFL相当的性能。

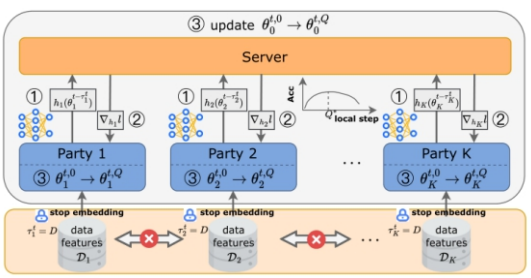


图19 BS-VFL系统架构图

5.4 动态图神经网络分布式高效训练系统优化

近年来，基于记忆的动态图神经网络（Memory-based Temporal Graph Neutral Network）

在动态图的学习任务中表现出了优越的性能，这归功于其具有的一种独特结构：用于收集每个节点历史信息的记忆模块（memory module）。然而，记忆模块在训练的不同批次之间存在依赖性，现有的分布式训练方法均会破坏这一依赖关系。论文通过实验发现，使用陈旧的记忆信息会带来模型准确率降低、收敛所需的轮次数增加等问题，这给动态图神经网络在分布式场景下的训练带来了新的挑战。因此，提出了一种基于流水线方法的分布式动态图神经网络训练框架——PipeTGL，如图20所示。PipeTGL通过将模型训练过程整理为有向无环图（Directed Acyclic Graph，DAG）来寻找训练批次间与批次内存在依赖关系的部分，并在此基础上采用更有效的调度方法减少流水线中由于通信与计算依赖产生的气泡。实验结果表明，相较于目前最先进的两种分布式动态图神经网络训练框架GNNFlow和DistTGL，PipeTGL将训练速度提升至前两者的1.27到4.74倍，且达到了更高的模型准确率。

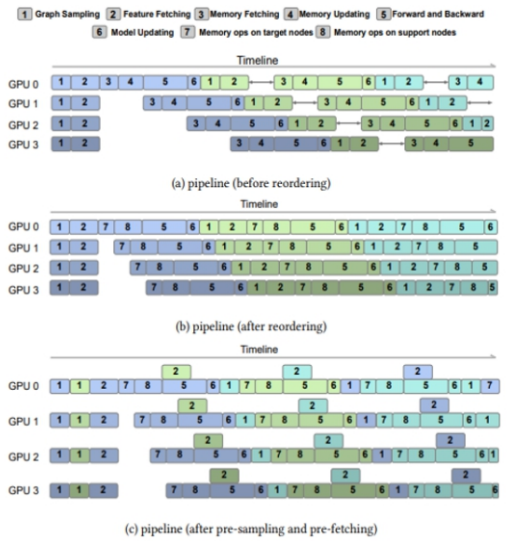


图20 优化前后动态图神经网络分布式执行策略对比

5.5 基于深度强化学习的公平鲁棒联邦聚合算法

联邦学习作为一种保护数据隐私的分布式机器学习范式，面临着两个核心挑战：数据异构性导致的全局模型性能偏向以及对抗攻击对模型完整性的威胁。尽管现有的个性化联邦学习方法如Ditto、lp-proj能够缓解数据异构问题，但这些方法主要聚焦于客户端层面的优化，忽略了服务器聚合阶段的脆弱性，且缺乏同时兼顾鲁棒性与公平性的集成解决方案。

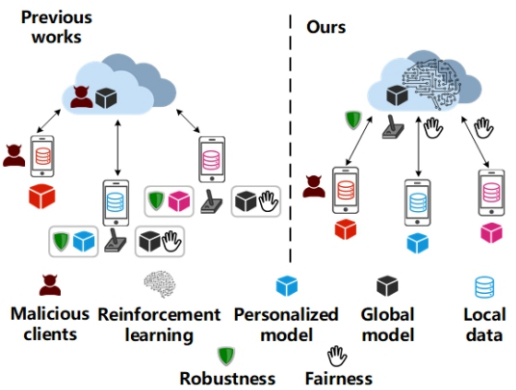


图21 联邦学习鲁棒性公平性权衡模式对比

针对上述问题，提出了FedAA（Federated Adaptive Aggregation）框架，如图21所示。FedAA创新性地将深度强化学习与联邦学习深度融合。该框架采用DDPG算法实现对聚合权重的连续精细控制，设计了基于模型参数欧氏距离的客户端选择算法来筛选可信客户端，并在服务器端构建公平验证集作为奖励信号指导优化。通过将联邦学习的每轮通信建模为马尔可夫决策过程，FedAA实现了“客户端选择–权重优化–奖励反馈”的闭环自适应聚合机制。

实验结果表明，FedAA在多个数据集和攻击场景下的鲁棒性均优于现有方法，同时在公平性指标上与最先进方法相当，成功在服务器层面平衡了模型的鲁棒性与公平性。通过调节参与聚合的客户端比例，该方法能在恶意客户端风险与模型泛化能力间找到最

优权衡点。尽管存在对强攻击的容忍度限制和DRL非凸优化的局限性，FedAA为联邦学习中的鲁棒性与公平性协同优化提供了新的解决思路。

5.6 LLM推理的隐私保护与效用权衡关系研究

随着ChatGPT、PaLM等LLM的广泛应用，用户在与这些模型交互时往往需要提供包含敏感个人信息的提示词，这引发了严重的隐私泄露风险。当前虽然存在一些基于随机化的隐私保护方法，但这些方法在保护隐私的同时不可避免地会损失模型的实用性。因此，如何在隐私保护和模型效用之间找到平衡成为一个关键问题。

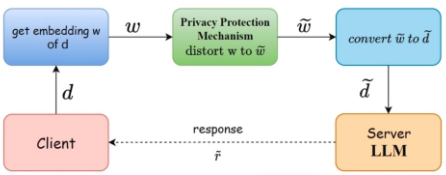


图22 隐私保护的LLM推理示意图

对此，建立了隐私保护LLM推理的理论框架，并提出了LLM推理的“无免费午餐定理”（No Free Lunch Theorem），如图22所示。首先形式化定义了隐私泄露和效用损失的概念，将隐私保护LLM推理问题建模为约束优化问题。随后通过严格的数学推导，证明了隐私泄露和效用损失的加权和存在一个非零的下界，这个下界取决于使用的隐私保护算法和隐私保护强度。通过在CNN/Daily Mail数据集上的实验验证了该理论，使用InferDPT算法展示了随着隐私预算的变化，隐私泄露和效用损失确实存在明显的权衡关系。

理论及实验结论表明，在隐私保护的LLM推理中存在不可避免的隐私–效用权衡。该“无免费午餐定理”揭示了隐私保护机制的根本局限性，即当隐私预算过于严格时，必然会导致效用损失。提醒研究者在追求隐私保护时必须考虑其对模型性能的影响，并在实际应用中寻求合理的平衡点。

总结

分布式系统小组的典型成果涵盖分布式系统软件、区块链技术、边缘计算及分布式算网融合四大方向，在用户自定义函数运行时优化、图式智能合约加速、边缘智能系统和流水线训练框架等方向形成显著特色，产出一系列代表性成果。小组在持续深耕服务器无感计算、容器存储管理等优势领域的同时，积极拓展边缘智能与算力网络等新兴方向，在WASM数据传输优化、复杂场景精准感知、多模态行为识别、联邦学习扩展等关键技术取得突破，构建了覆盖弹性服务、区块链加速、边缘智能与算网协同的系统性技术体系。

附成果列表论文

[1] Zhuo Huang, Hao Fan, Junhui Peng, Qi Wu, Song Wu, Chen Yu, Hai Jin, Qiming Liu, Wei Yang, and Shuo Yu, WAF: An Efficient WebAssembly-Based Execution Environment for User-Defined Functions, In Proceedings of the IEEE 41st International Conference on Data Engineering, 2025, pp. 1966-1980.

[2] Hansheng Zhang, Song Wu, Hao Fan, Zhuo Huang, Weibin Xue, Chen Yu, Shadi Ibrahim, and Hai Jin, KubeSPT: Stateful Pod Teleportation for Service Resilience with Live Migration, IEEE Transactions on Services Computing, 2025, 18(3): pp. 1500-1514.

[3] Zhengyi Yuan, Xiong Wang, Yuntao Nie, Yufei Tao, Yuqing Li, Zhiyuan Shao, Xiaofei Liao, Bo Li, and Hai Jin, DynPipe: Towards Dynamic End-to-End Pipeline Parallelism for Interference-Aware DNN Training, IEEE Transactions on Parallel and Distributed Systems.

[4] Junyi Zhang, Chuanhu Ma, Xiong Wang, Yuntao Nie, Yuqing Li, Xiaofei Liao, Bo Li, and Hai Jin, PopFetcher: Towards Accelerated Mixture-of-

Experts Training Via Popularity Based Expert-Wise Prefetch, In Proceedings of the USENIX Annual Technical Conference, 2025, pp. 1053-1069.

[5] Xiong Wang, Yi Zhang, Yuxin Chen, Yuqing Li, Chuanhu Ma, Bo Li, and Hai Jin. Runtime-Aware Pipeline for Vertical Federated Learning with Bounded Model Staleness. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2025, pp. 1539-1550.

[6] Jun Liu, Bingqian Du, Ziyue Luo, Sitian Lu, Qiankun Zhang, and Hai Jin, PipeTGL: (Near) Zero Bubble Memory-Based Temporal Graph Neural Network Training via Pipeline Optimization, In Proceedings of the VLDB Endowment, 2025, 18(8), 2722-2734.

[7] Liezhuo Zhang, Xianwei Lv, Chen Yu, Jiang Xiao, Kai Liu, and Hai Jin, UHA: An Intelligent Uncertainty Map Based Hierarchical Attention Network System for Building Segmentation, Transactions on Network Science and Engineering, 2024.

[8] Yujue Wang, Xin Niu, Xianwei Lv, and Chen Yu, FFCI: A Camera and IMU Sensors Based Multi-modal Neural Network for Activity Recognition in Smart Factory, IEEE Transactions on Consumer Electronics, 2024.

[9] Xin Niu, Xianwei Lv, Wang Chen, Chen Yu, and Hai Jin, Computing Tasks Saving Schemes through Early Exit in Edge Intelligence-Assisted Systems, IEEE Transactions on Computers, 2024.

[10] Dezhong Yao, Sanmu Li, Zhiwei Wang, Peilin Zhao, Gang Wu, Chen Yu, and Hai Jin, Efficient Distributed Sparse Relative Similarity Learning, ACM Transactions on Knowledge Discovery from Data, 2025, 19(3), pp. 1-27.

[11] Dezhong Yao, Sifan Zhao, Tongtong Liu, Gang Wu, and Hai Jin, ApSpGEMM: Accelerating Large-scale SpGEMM with Heterogeneous Collaboration and Adaptive Panel, ACM Transactions on Architecture and Code Optimization, 2025, 22(1), pp. 1-23.

[12] Sifan Zhao, Tongtong Liu, Hai Jin, and Dezhong Yao, SPViT: Accelerate Vision Transformer Inference on Mobile Devices via Adaptive Splitting and Offloading, IEEE Transactions on Mobile Computing, 2025.

[13] Dezhong Yao, Wanning Pan, Yuexin Shi, Michael J. O'Neill, Yutong Dai, Yao Wan, Peilin Zhao, Hai Jin, and Lichao Sun, Fedhm: Efficient federated learning for heterogeneous models via low-rank factorization, Artificial Intelligence, 2025, 344(104333).

[14] Ziyou Si, Lin Gu, Yunzhuo Ju, Deze Zeng, Hai Jin, Collaborative Multi-Granularity Distributed Registry Planning for Fast Container Image Pulling, Frontier of Computer Science, 2024.

[15] Jialuo He, Wei Chen, and Xiaojin Zhang, FedAA: A Reinforcement Learning Perspective on Adaptive Aggregation for Fair and Robust Federated Learning, In Proceedings of the AAAI Conference on Artificial Intelligence 2025, 39 (16), pp. 17085-93.

[16] Xiaojin Zhang, Yahao Pang, Yan Kang, Wei Chen, Lixin Fan, Hai Jin, and Qiang Yang, No Free Lunch Theorem for Privacy-Preserving LLM Inference, In Artificial Intelligence, 2025, vol. 341, pp. 104293

[17] Shijie Zhang, Ru Cheng, Xinpeng Liu, Jiang Xiao, Hai Jin, and Bo Li, Seer: Accelerating Blockchain Transaction Execution by Fine-Grained Branch Prediction. In Proceedings of the VLDB Endowment, 2025, 18(3), pp. 822-835.

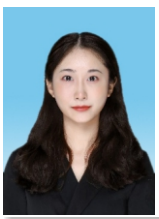
[18] Binhong Li, Licheng Lin, Shijie Zhang, Jianliang Xu, Jiang Xiao, Bo Li, and Hai Jin, FlexIM: Efficient and Verifiable Index Management in Blockchain, IEEE Transactions on Knowledge and Data Engineering, 2025.

[19] Xiaohai Dai, Zhengxuan Guo, Jiang Xiao, Guanxiong Wang, Yifei Liang, Chen Yu, and Hai Jin, Pako: Multi-Valued Byzantine Agreement Comparable to Partially-Synchronous BFT, IEEE Transactions on Computers, 2025.

[20] Xiaohai Dai, Wei Li, Guanxiong Wang, Jiang Xiao, Haoyang Chen, Shufei Li, Albert Y Zomaya, and Hai Jin, Remora: A low-latency dag-based bft through optimistic paths, IEEE Transactions on Computers, 2025.

[21] Qiang He, Kaibin Wang, Zeqian Dong, Liang Yuan, Feifei Chen, Hai Jin, Yun Yang, Hourglass: Enabling Efficient Split Federated Learning with Data Parallelism, In Proceedings of the Twentieth European Conference on Computer Systems, 2025, pp. 1317-1333.

黄　卓
博士后
研究方向：服务器无感知计算、运行时系统
Email：huangzhuo@hust.edu.cn



余庚花
博士后
研究方向：边缘计算
Email：2024510414@hust.edu.cn

罗瑞坤
博士后
研究方向：边缘计算、数据存储
Email：rkluo@hust.edu.cn

戴小海
讲　师
研究方向：区块链、分布式共识协议
Email：xhdai@hust.edu.cn

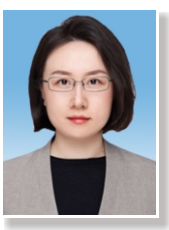
黄　航
讲　师
研究方向：容器与虚拟化、云原生大模型系统
Email：hanghuang@hust.edu.cn

张晓今
讲　师
研究方向：可信机器学习、理论计算机
Email：xiaojinzhang@hust.edu.cn

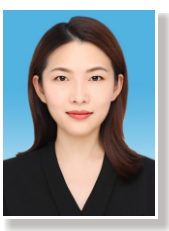
杜冰倩
讲　师
研究方向：分布式机器学习系统及算法
Email：bqdu@hust.edu.cn

王　雄
副教授
研究方向：分布式学习、云/边缘计算、网络分析
E-mail：xiongwang@hust.edu.cn

姚德中
副教授
研究方向：边缘计算、分布式机器学习
Email：dyao@hust.edu.cn



顾　琳
教　授
研究方向：网络功能虚拟化、软件定义网络、云计算
Email：lingu@hust.edu.cn



肖　江
教　授
研究方向：分布式计算与系统、数据治理与分析、无线网络与移动计算
E-mail：jiangxiao@hust.edu.cn

余　辰
教　授
研究方向：边缘计算
Email：yuchen@hust.edu.cn

何　强
教　授
研究方向：边缘智能
Email：hqiang@hust.edu.cn

吴　松
教　授
研究方向：云计算、虚拟化
Email：wusong@hust.edu.cn

网络空间安全组典型成果介绍

邹德清、徐　鹏、文　明、胡胜山、王虹飞、李　珍、袁　斌、李　志、吴月明

关键词： 代码大模型，云数据安全，移动安全，人工智能安全，芯片安全，安全测试，全同态加密

1 介绍

在软件应用安全方面，提出了面向Java反序列化漏洞挖掘的检测新方法。在Java应用中，由类方法连续调用形成的Gadget Chain可能引发反序列化漏洞，带来严重安全威胁。现有检测方法往往依赖不精确的调用图，难以处理反序列化、动态代理和反射等复杂特性，导致结果不完整且不准确。为此，研究提出了新型检测工具 Flash，通过反序列化驱动的调用图构建实现更精准的检测。具体而言，Flash首先进行可控性分析，判断变量是否可通过反序列化恢复；随后采用混合分派策略：对可控变量使用更全面但精度较低的方法（如类层次分析、代理分派），否则则采用更精确的技术（如指针分析）。Flash还通过处理涉及可控变量的反射来恢复缺失的调用边，并过滤掉不相关的边，以提高准确率和效率。对30个应用程序的评估表明，Flash在有限的开销下，实现了比现有技术更高的召回率（假阴性率降低了30.8%）和准确率（假阳性率降低了25.9%）。此外，Flash共检测到90个新的Gadget Chain。

在漏洞检测模型鲁棒性方面，深度学习模型的核心挑战在于，它们倾向于学习代码表层句法特征与漏洞标签之间的“伪相关性”，而非真正导致漏洞的因果特征。这种对伪特征的依赖，使得模型极易受到对抗性攻击的欺骗；攻击者仅需对代码进行不影响其语义功能的微小修改，就能成功规避检测。为解决这一问题，利用因果推断的原理来消除伪相关性，从而提升模型的鲁棒性。该CausalCode框架通过两大技术实现：首先，采用一种“因果数据增强”方法，通过模拟因果干预来生成干扰伪相关的样本；其次，利用正则化技术引导模型学习“不变性表示”，强制模型关注因果特征，使得原始代码与其干预样本在表征空间中的距离最小化。通过这种方式，CausalCode能够显著提升模型区分因果与伪特征的能力，从而大幅增强其在面对对抗性攻击时的鲁棒性，效果优于当前主流的对抗性训练等防御方法。实验结果显示，CausalCode可将SOTA对抗攻击攻击成功率从95.23%大幅降低至39.61%。

在漏洞修复方面，系统化地研究了现有的自动化漏洞修复方法、工具与评估体系，提出了首个面向C/C++程序的漏洞修复基准数据集Vul4C，包含144个真实漏洞及其漏洞触发信息与补丁。首先系统化的分析了现有自动化漏洞修复方法和工具，并按照提出的自动化漏洞修复工作流程的三个核心步骤：漏洞分析、补丁生成与补丁验证展开深入分析；为探究现有自动化漏洞修复方法的效果，通过收集了真实世界的可验证漏洞以其补丁，构建了首个面向C/C++程序的漏洞修复基准数据集Vul4C，并通过Vul4C评估了7个C/C++ AVR工具和2个Java AVR工具，实验发现基于语义的修复方法在生成高质量补丁方面优于基于学习的方法，并指出当前学习型方法缺乏严谨的评估机制。论文还提出了未来研究方向，包括改进漏洞分析技术、融合多方法优势、利用大语言模型（LLM）等。

在软件包恶意代码检测方面，提出了一种新型恶意NPM软件包检测器（MalPacDetector），

它利用大语言模型自动动态生成特征（而非依赖专家手动定义）。随着恶意行为的发展，MalPacDetector 可以自动、动态地更新特征。为了评估MalPacDetector与现有检测器的有效性，构建了一个新的NPM软件包数据集，该数据集克服了现有数据集的缺陷（例如样本数量少、恶意代码片段重复率高）。实验结果表明，MalPacDetector以1.3%的误报率和7.5%的漏报率的性能优于现有检测器。特别值得注意的是，MalPacDetector检测出39个此前未知的恶意软件包，并已得到NPM安全团队的确认。

在安卓恶意软件对抗攻击防御方面，因安卓的开源特性其已成为主流操作系统，却也沦为恶意软件主要攻击目标。为此，研究人员开发的基于机器学习的安卓恶意软件检测器，虽在识别上成效显著，但正受对抗样本严重威胁，这类经细微修改的样本可保留恶意功能并避检，部分工具甚至能将主流检测器效率降至1%。针对此，提出新型防御机制 HagDe：通过对样本沿梯度上升方向施加迭代扰动，借助对抗样本对扰动的高敏感性，观察微小扰动后损失函数的异常增幅来区分对抗与正常样本。基于 1.5 万个样本、15 种攻击模式的实验显示，HagDe 在 AdvDroidZero、BagAmmo 攻击下防御有效性分别达 88.5%、90.7%，较现有 KD_BU、LID 方法分别提升 32.45%、11.28%。

在网络协议安全测试方面，提出了一种基于有限状态机引导的模糊测试方法。具体而言，针对当前网络协议模糊测试工具在探索协议实现内部状态空间方面的不足，通过有限状态机技术对协议实现进行全方位建模，以更深入地理解其系统行为和状态转换规律。在此基础上，将学习到的状态机模型用于引导模糊测试过程，显著提升了测试的覆盖率和漏洞发现能力。以广泛使用的TLS协议为研究对象，实现了原型系统SNETFuzzer，并在多个开源TLS 实现上进行了实验验证。结果表明，SNETFuzzer 在代码覆盖率、路径探索和状态转移数量等关键指标上均优于现有主流工具AFLNET，并成功发现了多个安全漏洞，其中包括两个未公开的新漏洞，证明了所提方法的有效性和实用性。

在芯片安全方面，物理不可克隆函数（PUF）是最重要的硬件安全原语之一，被广泛地应用于密钥生成、设备认证等安全场景。虽然PUF得益于其固有的不可克隆性与不可预测性，能抵御多种物理攻击，但是其却易受基于机器学习的建模攻击。多路选线器PUF（MPUF）是一种复杂PUF设计，具备更强的抗建模攻击能力。提出了一种基于克罗内克积的对MPUF建模攻击框架，通过实现更高效、精确、稳定的攻击，为验证新的MPUF设计提供了有力支撑。

在人工智能安全方面，对抗样本是最具威胁性的攻击手段之一。在目标检测领域，尽管对抗攻击的研究已经取得了一定进展，但现有的攻击方法大多依赖于特定的目标检测器结构，如非最大抑制（NMS）和区域提议网络（RPN），这限制了它们的可扩展性。此外，大多数针对目标检测器的对抗攻击方法源自于图像级别的分类任务攻击，这导致了在非关键对象（如背景）上的冗余计算和干扰，从而降低了攻击效率。因此，如何设计一种模型不可知的高效攻击方法，以全面评估目标检测器的漏洞，仍然是一个具有挑战性且未解决的问题。针对于此，提出了NumbOD，这是一种全新的空间–频率融合攻击方法，旨在通过直接利用目标检测器输出的特征来生成对抗样本，而不依赖于其内部结构。通过大量的理论分析和实验验证，分析并解释了目标检测器在对抗攻击下的行为，并提出了一系列提高攻击效率和效果的方法。此外，近期研究侧重于针对特定目标检测器结构的攻击方法。NumbOD则能同时有效欺骗多种目标检测器。通过采用空间协调偏差攻击和关键频率干扰攻击，NumbOD可以生成具有高隐蔽性和高攻击性的对抗样本，

其不仅能有效欺骗现有的各种目标检测器，还能在保持图像视觉质量的同时，显著降低目标检测器的检测精度。

全同态加密（FHE）软硬件协同加速方面，全同态加密是一种新型的密码学技术，可以允许数据在加密状态下进行计算，对密文的运算结果等价于对明文的计算。针对基于FHE的应用中密钥切换（KeySwtich）与自举（Bootstrapping）操作计算开销高昂的问题，提出了一种基于GPU的全同态加密软硬件协同加速系统Athena。在KeySwitch加速方面：Athena针对KeySwtich的三个关键组件，设计了流水线高效的快速数论变换（NTT）、重用中间数据的基转换（EBConv）、缓解内存瓶颈内积（IP）的GPU Kernel函数进行加速。在Bootstrapping加速方面：（1）Athena首先针对Bootstrapping中的C2S/S2C阶段的明–密文矩阵乘法设计了基于Triple Hoisting的算法，大幅降低了C2S/S2C阶段中冗余的NTT、EBConv操作，减少了明文矩阵的存储空间，进一步地，设计了Triple Hoisitng大步小步计算的专用Kernel函数，大幅降低了中间数据的访存压力；（2）Athena同时针对Bootstrapping的EvalMod阶段的Chebyshev非线性函数拟合方法设计了一种高效计算方法，首先将Chebyshev多项式拆分成递归方式计算，再将递归计算方式拆分为平衡二叉树，采用树层间并行方式大幅提升了EvalMod的计算效率。实验结果表明，Athena在KeySwitch与Bootstrapping加速方面与现有最好的工作相比分别提升了1.9-4.5倍与1.8-4.7倍。

2 成果介绍

2.1 基于反序列化引导调用图构建的精准Gadget Chain挖掘

Java反序列化漏洞（JDV）指攻击者精心设计输入数据，在应用反序列化后触发一系列类方法调用，最终到达程序中的危险函数。JDV是一种常见且众所周知的漏洞类型，2024年CWE Top 10漏洞榜单凸显了反序列化漏洞的普遍性，其中“不可信数据反序列化”被列为关键问题。因此，检测和缓解JDV仍然是应用程序安全领域一个重要且相关的课题。Gadget Chain指从反序列化相关方法（即source）开始执行到危险方法（即sink）的方法调用链，是检测和缓解JDV的关键。然而，现有方法仍面临一个关键问题：其所依赖的调用图并不精确。为此，研究者提出一种基于反序列化驱动的指针分析算法（受需求驱动的指针分析启发），更高效地构建更加精确的调用图，从而提升Gadget Chain的检测能力。

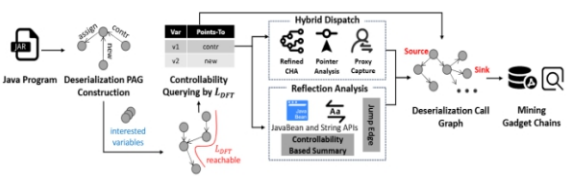


图1 Flash的工作流程

Flash的工作流程如图1所示。首先，Flash对感兴趣的变量执行反序列化驱动的可控性分析（DDCA），以确定变量是否可以通过反序列化恢复。然后，Flash利用可控性结果继续执行一系列下游任务（例如，混合分派和基于可控性的反射分析），从而尽可能构建更精确的反序列化调用图。在混合分派中，Flash根据接收变量的可控性来在调用点解析被调用者。如果接收变量在反序列化过程中可被攻击者控制，Flash会使用类层次分析法来解析被调用者。否则，Flash会利用指针分析根据接收者可能引用的对象来解析被调用者。且Flash基于静态分析将动态代理行为建模

为方法分派过程，主要通过研究整理的调用点特征识别可能触发代理行为的调用点，并将代理跳转边合并到调用图中。对于反射处理，Flash采用基于可控性的方法，分析可控字符串和JavaBean属性以识别潜在的反射目标，并添加反射跳转边。最终，基于构建的反序列化调用图Flash会进行Gadget Chain的检测。

实验结果表明，Flash成功识别了总共90个以前未发现的Gadget Chain，且相比于基准方法对已知链有更高的覆盖率。具体来说，Flash将误报率降低了30.8%，误报率降低了25.9%。而通过分析新发现的漏洞利用工具链，Flash发现了5种新的漏洞利用方法，这些方法可以绕过现有补丁，与之前已知的CVE相比风险更高。此外，使用研究所提出的混合调度策略，24.5%的调用点可以以更高的精度解析，4.8%的调用点可能触发动态代理，这凸显了Flash关键设计的益处。

2.2 基于因果学习提升漏洞检测模型鲁棒性方法

在自动化的软件漏洞检测中，深度学习（DL）模型正扮演着日益重要的角色。然而，这些模型表现出严重的脆弱性，极易受到对抗性攻击。攻击者通过对源代码进行微小的、不影响语义的改动，就能使模型的准确率从90%以上急剧下降到10%以下。这种脆弱性的根源在于，模型未能学习到决定程序行为的“因果特征”，反而依赖于代码表面的“伪相关性”。例如，在漏洞检测中，模型可能错误地将特定的变量名与漏洞关联，而不是捕获真正导致漏洞的不安全API使用或错误处理逻辑。现有用于提升鲁棒性的方法，如对抗性训练或对比学习，在消除这些伪相关性方面效果有限，因此亟需一种能从根本上区分并消除伪特征影响的全新技术，以确保代码模型的可靠性与安全性。

针对上述挑战，提出了如图2所示的CausalCode新型因果学习框架，旨在通过因果推断原理增强漏洞检测模型的鲁棒性。CausalCode的核心机制在于主动消除伪相关性并学习不变性特征表示。具体而言，该框架首先通过“因果数据增强”技术，利用因果分析中的do-operator生成“干预样本”。这些样本在保留原始代码功能的同时，破坏了代码风格等伪特征与漏洞预测结果间的虚假关联。随后，CausalCode利用一种专门设计的正则化策略，在模型训练过程中最小化原始样本与其干预样本在表示空间的距离，从而引导模型专注于学习独立于伪特征的、具有不变性的因果表示。

CausalCode在主流的CodeBERT、GraphCodeBERT及StarCoder2等代码模型上，针对漏洞（缺陷）检测任务进行了全面测试，并成功验证了其在抵御多种高级对抗性攻击方面

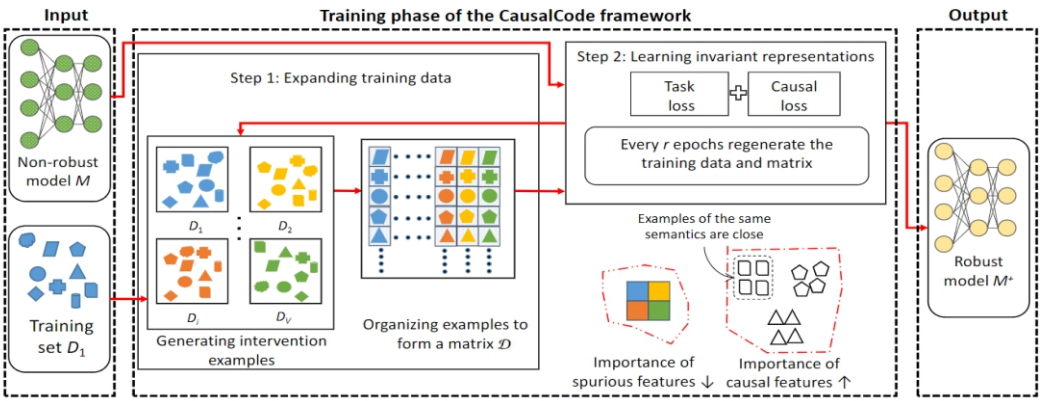


图2 CausalCode的工作流程

的有效性。实验结果表明，在针对StarCoder2模型的CARROT-I攻击下，CausalCode能将攻击成功率从95.23%大幅降低至39.61%；而在CodeBERT模型上，也能将攻击成功率从79.09%降至33.76%。此外，实验还表明，CausalCode在学习代码不变性表示方面的能力也超越了现有的先进防御方法，其增强后的模型能够更紧密地聚类语义等价的程序。通过这些成果，CausalCode为提升深度学习漏洞检测模型的鲁棒性和安全性提供了强有力的支持。

2.3 自动漏洞修复方法

随着软件复杂性的不断提升，漏洞数量持续增长，手动修复漏洞既耗时又依赖专家经验，因此自动化漏洞修复（Automated Vulnerability Repair, AVR）技术显得尤为重要。尽管自动化程序修复（Automated Program Repair, APR）已有较多研究，但其往往不适用于安全漏洞的修复，因为漏洞修复需考虑安全属性、漏洞触发条件等特殊因素。现有AVR研究分散，缺乏系统化总结和统一评估基准，尤其是针对C/C++程序的AVR工具评估存在数据集不统一、评估方法不一致等问题。

为了解决上述问题，首次系统化的分解了AVR的工作流程（如图3），将其分为三个步骤：漏洞分析、补丁生成和补丁验证。通过对现有AVR的方法和工具设计按照提出的工作流进行分析，指出了当前漏洞分析方法难以精确定位漏洞语句，模板型补丁生成方法虽对特定漏洞有效但泛化能力差，而静态验证方法因规则有限和路径爆炸问题误报率高等问题。对于评估方法不一致的问题，构建了首个C/C++漏洞修复基准数据集Vul4C，涵盖23个软件中的144个漏洞，涉及19种CWE类型，每个漏洞均提供漏洞触发输入、补丁，部分提供了测试用例。基于Vul4C和已有的Vul4J数据集，对7个C/C++ AVR工具和2个Java AVR工具进行了深入评估。

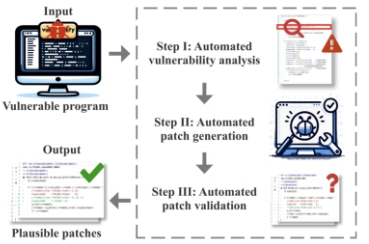


图3 自动化漏洞修复工作流程

实验结果表明，基于语义的AVR工具（如VulnFix）在补丁编译率和测试通过率上显著优于基于学习的方法（如VRepair、VulRepair）。语义方法能生成更多可编译且通过测试的补丁，但其有效性受限于程序分析的精度和可扩展性。学习型方法虽能生成补丁，但由于缺乏完整的验证流程和上下文匹配机制，实际修复效果较差。此外，模板型方法在Java漏洞修复中表现也不佳。总体来看，当前AVR工具在漏洞定位、补丁生成和验证方面仍存在显著不足，未来应加强漏洞分析技术、融合多方法优势、引入LLM等新技术，并设计更有效的自动化验证机制。

2.4 基于LLM特征生成的恶意NPM包检测器

针对NPM的攻击日益增多，已经激发了对恶意NPM包检测器的研究。现有的检测器遵循两种方法：程序分析与机器学习。程序分析方法主要通过模式匹配、克隆检测和动态分析来检测恶意软件包，这种方法通常使用规则来检测恶意软件包，并且由于这些规则相对通用而导致高误报率。机器学习方法使用专家定义的特征来训练检测器，这种方法的有效性取决于特征的质量。然而，高质量的特征往往是定义繁琐，难以定义（例如，当攻击利用混淆技术时），并且主观地定义（即，不同的专家可以定义不同的特征）。

针对上述问题，提出了恶意NPM包检测器MalPacDetector（如图4所示），使得LLM能够关注恶意NPM代码片段，从而自动识别NPM恶意代码片段并生成特征（即不需要人类专家定义特

征），并且MalPacDetector能够随着恶意行为的演变自动动态更新特征。其次，提出了一组基准数据集MalnpmDB来评估恶意NPM包检测器。MalnpmDB包含3258个恶意软件包，其中包括7种攻击类型，相比之下，现有数据集集中于单一类型的攻击，并且具有高重复率的恶意代码片段。

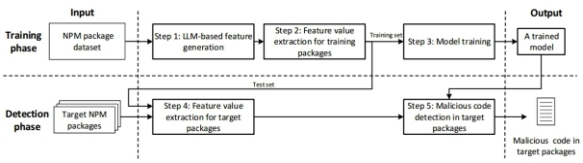


图4 MalPacDetector工作流程

使用MalnpmDB对MalPacDetector和现有的检测器进行了性能评估，实验结果表明，MalPacDetector的准确率为98.0%，误报率为1.3%，漏报率为7.5%，F1测度为95.2%。这意味着与最先进的基于机器学习的检测器相比，精度提高了2.9%，F1测量值提高了3.8%。特别的是，MalPacDetector检测到39个以前未知的恶意软件包，这些软件包已被NPM安全团队确认。

2.5 基于持续攻击的安卓恶意软件对抗样本检测

安卓系统因开源特性成为主流移动操作系统，但也频繁遭受恶意软件攻击，基于机器学习的安卓恶意软件检测器（AMD）（如Drebin、MaMaDroid）虽在理想场景下表现出色，却易受对抗样本（AEs）威胁——这类经过细微修改的恶意样本能规避检测且保留恶意功能。当前先进的对抗样本生成工具（如AdvDroidZero、BagAmmo）可将主流AMD的检测效能降至1%，而现有防御方法（如特征压缩）多针对图像领域设计，因安卓恶意软件特征的离散性，在该领域防御效果有限，亟需更具韧性的AMD防御机制。

研究提出名为HagDe的新型防御框架，核心思路是利用对抗样本对扰动的高敏感性来区分其与正常样本。如图5所示，该框架分三阶段实现：首先训练替代模型，模拟目标AMD的分类逻辑以获取梯度信息（解决传统AMD无法计算梯度的问题）；其次进行多阶段迭代扰动，沿梯度上升方向（对抗样本生成的逆方向）对样本特征施加微小扰动，记录每次扰动后的损失值序列，因对抗样本更接近决策边界，其损失值增长会远快于正常样本；最后训练检测分类器，以多阶段扰动得到的损失特征为输入，实现对对抗样本的识别，在样本进入AMD分类前完成对抗样本过滤。

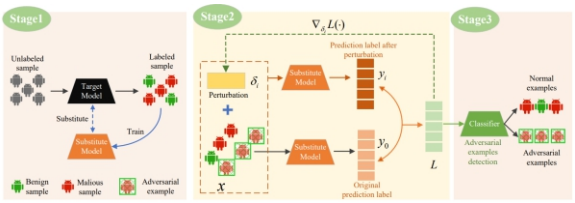


图5 HagDe工作流程

在包含15000个安卓应用样本和15种攻击模式的实验中，HagDe表现优异：针对AdvDroidZero和BagAmmo这两种主流攻击，防御效能分别达到88.5%和90.7%，较当前最新防御方法KD_BU和LID分别提升32.45%和11.28%；在F1分数上，HagDe在15种攻击模式中14种排名第一，平均F1分数较KD_BU和LID分别高38.7%和10.3%；效率方面，借助GPU加速，HagDe在高维特征（如MaP、Drebin）的训练和推理中耗时显著降低，且能有效增强现有AMD的抗对抗能力——当对抗样本比例从10%增至90%时，AMD的F1分数提升幅度达0.79%至474.4%。

2.6 基于有限状态机引导的网络协议模糊测试

网络协议作为网络通信的核心组成部分，具有使用范围广的特点，其安全性直接关系到整个网络系统的可靠性与安全性。然而，由于协议实现的复杂性和状态空间的多样性，传统的模糊测试方法往往难以全面覆盖其内部状态，导致许多深层漏洞无法被有效发现。现有模糊测试工具在生成大量的随机、无效或异常数据方面表现出色，但其在状态的探索、理解、模拟实现等能力的表现上仍然比较有限，缺乏对协议行为的系统性建模与引导。

通过引入有限状态机（FSM）学习技术，构建了TLS协议的精确状态机模型，能够全面捕捉协议握手过程中的状态转换和异常行为，解决决当前主流网络协议模糊测试工具对协议实现内部状态空间探索不充分的问题。如图6所示，该系统采用Angluin's L*算法作为核心学习算法，通过“成员查询”(membership query)和“等价查询”(equivalence query)逐步构建出反映真实协议行为的有限状态机。在此基础上，SNETFuzzer利用生成的状态机信息提取多条状态转移路径，并据此构造有效的测试种子，引导模糊测试过程更深入地探索协议状态空间。

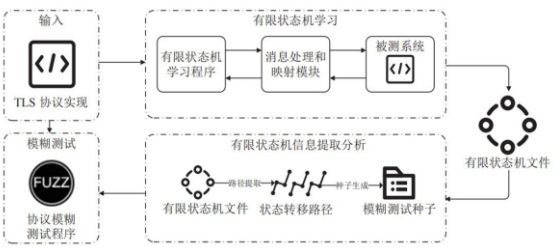


图6 SNETFuzzer系统框架

实验部分在OpenSSL、GnuTLS、MatrixSSL等5个主流TLS实现上进行了对比测试。结果显示，SNETFuzzer在程序覆盖率、边覆盖数和路径发现等方面均显著优于AFLNET，并成功在MatrixSSL中发现两个未记录的内存泄漏漏洞（CWE-762和CWE-415），在OpenSSL中复现了一个已知的空指针解引用漏洞（CVE-2014-3569）。这些成果不仅验证了该方法的有效性，也体现了其在复杂协议漏洞挖掘方面的潜力。

2.7 基于克罗内克积的MPUF高效建模设计

PUF是利用芯片半导体内部物理特性构建的轻量级单向函数，将输入（也称挑战）映射为输出（也称响应）。PUF将芯片中不可避免的随机制程差异作为熵源，使其映射关系独一无二且不可预测。而每个PUF的挑战相应对数据，便可作为其独特的指纹。仲裁器PUF（APUF）具备优秀的可靠性，是最被广泛使用的PUF设计之一，其结构如图7所示。然而，APUF易受基于机器学习的建模攻击。

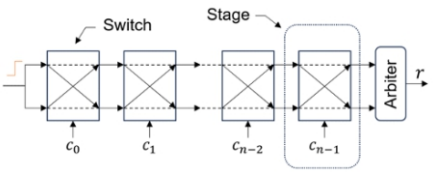


图7 仲裁器PUF结构示意图

MPUF是一种基于APUF的复杂PUF设计，其引入了基于多路选线器的非线性逻辑，具备更强的抗建模攻击能力。而MPUF的变体cMPUF以及rMPUF，则进一步增强了对各种攻击手段的抵抗能力，其结构如图8所示。

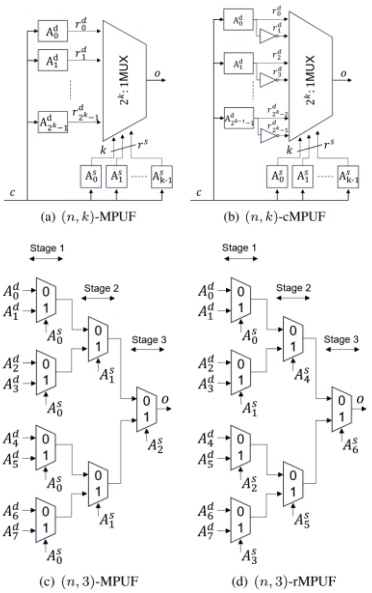


图8 MPUF及其变体结构示意图

现有的对MPUF的攻击存在精度差、成功率低、运行效率低等问题。其原因在于未能对选线器逻辑进行精确的建模，仅通过简单线性计算对选择逻辑进行近似，从而造成攻击精度差的问题。此外，该建模方法扩展性差。其仅能通过层层堆叠的方式实现对更大规模的MPUF的建模，从而使得该方案运行效率较为低下。为实现对MPUF的高效建模攻击，首先提出了对选线器逻辑的精简建模方案，不仅是对选线器

逻辑的精确建模，而且在噪声环境下能够更具鲁棒性。此外，利用克罗内克矩阵乘法对其进行了扩展，使得其能高效地扩展至更大的规模，不仅能够用于攻击更大规模的MPUF，也能够攻击MPUF的变体cMPUF以及rMPUF。实验结果表明，提出的攻击框架在多个维度都具有显著更优的表现：更高的精度（建模精度可达99.34%）；更低的数据需求（数据需求量减少50%以上）；更高的运行效率（攻击速度提升20至42倍）；更优的稳定性。

2.8 针对目标检测器的通用对抗样本实现方法

目标检测器（Object Detectors，ODs）在众多复杂场景中，如自动驾驶、安防监控等领域取得了显著的成果。然而，深度神经网络（DNNs）的脆弱性逐渐暴露，对抗性攻击作为一种通过在输入数据中添加精心设计的微小扰动来诱导模型产生错误输出的攻击方式，对目标检测器的安全性和可靠性构成了严重威胁。尽管在图像分类任务中对抗性攻击的研究已经取得了广泛进展，但目标检测任务由于其包含分类和回归两个子任务，面临着更大的挑战，相关研究相对较少。为了全面评估目标检测器的脆弱性，提出了一种新颖的、模型不可知的对抗性攻击方法NumbOD，旨在通过扰乱图像中的目标检测，使目标检测器无法正常检测到任何目标，从而为提高目标检测系统的安全性提供理论依据和技术支持。

NumbOD的核心思想是通过综合考虑空间域和频率域的特征，对目标检测器进行高效且隐蔽的攻击。具体而言，该方法首先设计了一种双轨攻击目标选择策略，独立地从分类和回归子任务中选择高质量的边界框作为攻击目标，从而确保攻击的针对性和有效性。在空间域中，NumbOD通过添加噪声来诱导预测边界框的位置偏差和分类结果的误分类，从而欺骗目标检测器，使其对目标的定位和识别产生错误。同时，在频率域中，利用离散小波变换（DWT）分解图像，专注于干扰图像的高频分量，这些分量对深度神经网络的语义纹理信息更为敏感，从而增强攻击效率，使攻击更具隐蔽性。NumbOD不依赖于目标检测器的内部结构，能够对不同架构的目标检测器进行有效攻击，具有广泛的适用性和强大的泛化能力。NumbOD架构设计如图9所示。

在九种目标检测器和两个数据集上的广泛实验表明，NumbOD能够实现强大的攻击性能，显著降低目标检测器的平均精度（mAP），并且具有很高的隐蔽性，生成的对抗性样本在视觉上难以与正常样本区分。即使在面对图像预处理（如亮度调整和溅泼效果）、模型剪枝、微调和对抗性训练等防御措施时，NumbOD依然能够保持有效的攻击性能，显示出良好的鲁棒性。这一成果不仅为研究目标检测器的脆弱性提供了新的视角

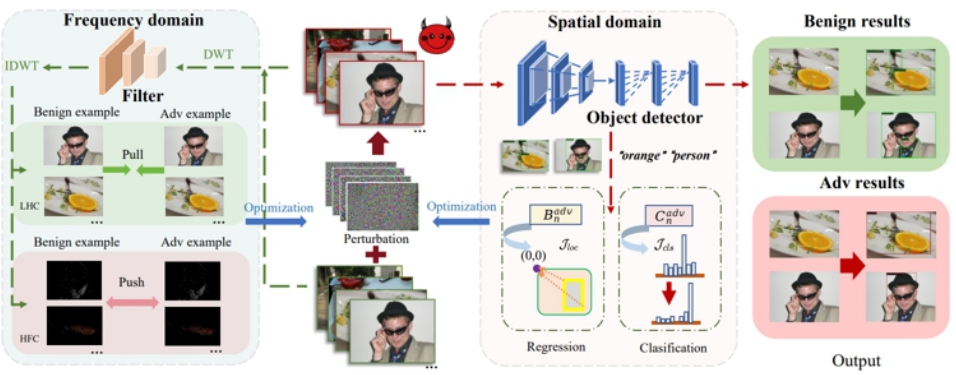


图9 NumbOD架构示意图

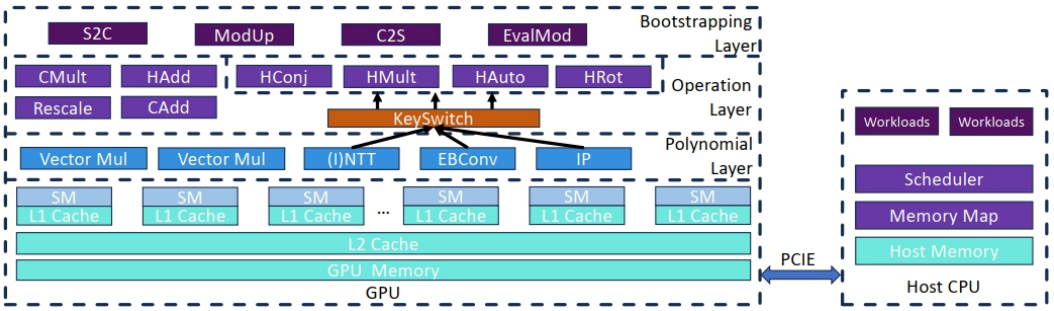


图10 基于GPU的全同态加密软硬协同加速系统Athena

和方法，也为设计更加安全可靠的目标检测系统提供了重要的参考和借鉴。

2.9 全同态加密软硬件协同加速系统

全同态加密是一种新型的密码学技术，可以允许数据在加密状态下进行计算，对密文的运算结果等价于对明文的计算，其在云计算、隐私外包计算等领域有着规范应用。

针对基于FHE的应用中密钥切换（KeySwtich）与自举（Bootstrapping）操作计算开销高昂的问题，提出了一种基于GPU的全同态加密软硬件协同加速系统Athena，系统架构图如图10所示。

在KeySwitch加速方面，Athena实现了最新的第三代KLSS密钥切换算法，并针对KeySwtich的三个关键组件设计专用的GPU Kernel函数加速：1）设计了流水线高效的快速数论变换（NTT）Kernel，通过构造Block、Warp、Thread三个层级的NTT流水线，有效消除了NTT计算过程中的流水线停顿；2）设计了中间数据重用的基转换（EBConv）Kernel函数，通过在Register File中重用EBConv的中间计算数据，大幅提升了EBConv的访存效率；3）缓解了内积（IP）的内存瓶颈，通过高效的数据拆分与任务分配，缓解了内积操作的访存瓶颈。

在Bootstrapping加速方面，1）Athena首先针对Bootstrapping中的C2S/S2C阶段的明–密文矩阵乘法设计了基于Triple Hoisting的算法，大幅降低了C2S/S2C阶段中冗余的NTT、EBConv操作，减少了明文矩阵的存储空间，进一步地，设计了Triple Hoisitng大步小步计算的专用Kernel函数，大幅降低了中间数据的访存压力；2）Athena同时针对Bootstrapping的EvalMod阶段的Chebyshev 非线性函数拟合方法设计了一种高效计算方法，首先将Chebyshev多项式拆分成递归方式计算，再将递归计算方式拆分为平衡二叉树，采用树层间并行方式大幅提升了EvalMod的计算效率。

实验结果表明，Athena对KeySwitch的优化实现成功提升了同态乘法与同态旋转操作的性能，与现有最好的GPU、FPGA加速工作相比分别提升了1.9-4.5倍，在Bootstrapping加速方面与现有最好的工作相比提升了1.8-4.7倍。

附成果列表论文

[1] Yiheng Zhang, Ming Wen, Shunjie Liu, Dongjie He, Hai Jin, Precise and Effective Gadget Chain Mining through Deserialization Guided Call Graph Construction，In Proceedings of the 34th USENIX Security Symposium. 2025: 2947-2964.

[2] Junyao Ye, Zhen Li, Xi Tang, Deqing Zou, Shouhuai Xu, Weizhong Qiang, Hai Jin, A Causal Learning Framework for Enhancing Robustness of Source Code Models, In Proceedings of the 2025 ACM International Conference on the Foundations of Software Engineering, 2025: 2641-2664.

[3] Yiwei Hu, Zhen Li, Kedie Shu, Shenghua Guan, Deqing Zou, Shouhuai Xu, Bin Yuan, Hai Jin, SoK: Automated Vulnerability Repair: Methods, Tools, and Assessments, In Proceedings of the 34th

USENIX Security Symposium, 2025: 4421-4440.
[4] Yinyuan Zhang, Cuiying Gao, Yueming Wu, Shihan Dou, Cong Wu, Ying Zhang, Wei Yuan, Yang Liu, Fighting Fire with Fire: Continuous Attack for Adversarial Android Malware Detection, In Proceedings of the 34th USENIX Security Symposium. 2025: 4897-4916.
[5] Zhou Ziqi, Bowen Li, Yufei Song, Zhifei Yu, Shengshan Hu, Wei Wan, Leo Yu Zhang, Dezhong Yao, and Hai Jin, Numbod: A spatial-frequency fusion attack against object detectors, In Proceedings of the AAAI Conference on Artificial Intelligence. 2025: 1201-1209.
[6] Yifan Yang, Kexin Zhang, Peng Xu, Zhaojun Lu, Wei Wang, Weiqi Wang, Kaitai Liang, Athena: Accelerating KeySwitch and Bootstrapping for Fully Homomorphic Encryption on CUDA GPU, In Proceedings of the 30th European Symposium on Research in Computer Security. 2025: 115-126.
[7] Jian Wang, Zhen Li, Jixiang Qu, Deqing Zou, Shouhuai Xu, Ziteng Xu, Zhenwei Wang, Hai Jin, MalPacDetector: An LLM-Based Malicious NPM Package Detector, IEEE Transactions on Information Forensics and Security, vol. 20, pp. 6279-6291, 2025.
[8] Hongfei Wang, Caixue Wan, and Hai Jin, Efficient Modeling Attack on Multiplexer PUFs via Kronecker Matrix Multiplication, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 44, no. 8, pp. 2883-2896, 2025.
[9] 袁斌，任家俊，陈群锦明，张驰，邹德清，金海，基于有限状态机引导的网络协议模糊测试方法，《软件学报》，36(8):3726-3743, 2025.

邹德清
教　授
研究方向：系统安全、网络攻防
Email：deqingzou@hust.edu.cn

徐　鹏
教　授
研究方向：公钥密码学、基于身份密码学、格密码学、云安全
Email：xupeng@hust.edu.cn

胡胜山
副研究员
研究方向：人工智能安全、应用密码学
Email：hushengshan@hust.edu.cn

文　明
副教授
研究方向：代码分析、缺陷漏洞的检测与修复
Email：mwenaa@hust.edu.cn

王虹飞
副研究员
研究方向：人工智能芯片、芯片安全、EDA
Email：hongfei@hust.edu.cn

袁　斌
副教授
研究方向：云安全、软件定义网络安全、物联网安全
Email：yuanbin@hust.edu.cn

李　珍
副教授
研究方向：软件安全、人工智能安全
Email：zh_li@hust.edu.cn

李　志
副教授
研究方向：云原生安全、云系统内生安全
Email：lizhi16@hust.edu.cn

吴月明
副教授
研究方向：软件安全、供应链安全
Email：yuemingwu@hust.edu.cn

大数据组典型成果介绍

石宣化、陈汉华、华强胜、丁晓锋、陆 枫、黄 宏、张 腾、万 瑶

关键词： 分布式算法，机密计算，流计算，代码大模型，图神经网络，医疗大数据

1 介绍

大数据在医疗、金融、军事等领域存在已有时日，但其体量大、类型多、价值密度低、处理速度快等特性给传统算法和系统带来了新挑战，如何实现大数据价值最大化是亟待解决的问题。研究小组旨在研究面向新型体系结构的大数据系统关键技术，以大数据基础理论为基础，研究大数据系统的一系列科学问题，例如大数据处理和大数据分析等，并在此基础上研发了一系列大数据典型应用项目，为大数据的广泛高效应用提供技术支持。基于本组所承担的“零知识证明硬件加速技术”国家重点研发计划课题和“基于社会影响力的异质时序网络表示学习”等国家自然科学基金项目以及湖北省“尖刀”技术攻关项目“AI大模型关键技术与系统”等项目，对不同领域大数据问题进行深入研究，从理论、算法以及系统等多个维度探讨了多个典型大数据领域数据处理问题的解决思路：

2 大数据理论研究

2.1 基于全同态加密的低延迟四维分布式矩阵乘法算法

全同态加密（Fully Homomorphic Encryption, FHE）允许在密态数据上直接进行计算，是实现隐私计算的最理想方案。目前学术界、工业界普遍认为基于FHE的密态计算和等价明文计算间存在5个数量级的性能差距。矩阵乘法作为最底层核心算子被广泛应用于隐私保护的机器学习中，而现有FHE方案在处理大规模矩阵乘法时面临计算和内存开销过大的挑战，导致单机难以高效完成任务。

为此，提出了FHE4DMM，一种基于全同态加密的低延迟四维分布式矩阵乘法算法[1]。首先，引入了适配FHE打包技术、安全参数自适应的“块中间布局”表示，并给出了最优块布局生成算法。基于块布局中间表示，设计了联合优化通信和同态计算的分布式算法。最后，通过同态计算任务中的数据复用优化，完全消除冗余的耗时同态数据移动。实验结果表明，FHE4DMM在处理大规模同态矩阵乘法时，相比最优分布式算法获得最高16.62倍加速比，并且所有测试规模下保持线性加速比。此外，FHE4DMM在扩展的手写数字图片EMNIST 数据集和十分类彩色图像CIFAR-10数据集上的安全外包推理任务中，相比现有最优算法，分别实现了最高3.54倍和4.22倍的加速。

2.2 面向非独立同分布数据中差分隐私联邦学习的收敛性分析和自适应优化

联邦学习作为一种新兴的分布式学习原型，能够保证众多客户端在云服务器的协调下共同学习联合模型，而无需共享本地数据。通过保证数据本地化，联邦学习在隐私和通信效率方面优于传统的集中式学习框架。然而，当联邦学习面对众多客户端时，客户端之间不可避免地会出现非独立同分布数据。虽然联邦学习可通过在本地保留训练数据来促进隐私安全，但如果本地数据包含敏感或私人信息，联邦学习应拥有强大的隐私保护能力，以保证云端或恶意第三方无法根据客户端之间的模型更新共享准确地恢复此类信息。但现有的对手模型表明，在共享各自的模型更新时会出现隐私隐患。然而，由于联邦学习中深度神经网络的模型规模通常较

大，且与附加噪声的尺度正相关，难免会在输出中添加过多的噪声，进而导致模型性能严重下降，最终对实现差分隐私联邦学习算法的隐私–实用性平衡带来前所未有的挑战。

为解决上述两个重大挑战，提出差分隐私算法DPNFL来保护客户端隐私。该算法将高斯噪声注入到本地梯度中，利用差分隐私的后处理特性，同时实现中间模型更新和最终本地模型的差分隐私保证。为了应对非独立同分布数据，在非独立同分布联邦学习设置下采用无放回部分客户端采样，以减轻异构客户端的影响。为了进一步提高算法性能，还提出一种名为AdDPNFL的自适应版本的DPNFL，该算法在服务器端采用自适应优化，同时缓解非独立同分布数据和差分隐私噪声对模型实用性的影响。首先提出一种新颖的DPNFL算法来共同解决联邦学习中的差分隐私和非独立同分布问题。与现有的少量工作相比，方法主要在差分隐私分析技术和客户端采样方法上有所不同。具体而言，创新性地在联邦学习场景下采用截断集中差分隐私技术。与传统的差分隐私和近期的瑞丽差分隐私不同，截断集中差分隐私技术能够更严格地记录端到端的隐私损耗，因此对于相同级别的差分隐私，仅需要更少的噪声注入。此外，与广泛采用的完全客户端采样和多项式分布采样不同，所使用的无放回部分客户端采样不仅比完全客户端采样更为实用，而且避免了多项式分布采样中存在的采样方差问题，该问题对非独立同分布联邦学习的收敛性分析形成阻碍。除上述方法创新外，在理论分析中还利用两种指标分别衡量联邦学习问题中目标函数为强凸和非凸时的非独立同分布程度，并给出严格的收敛上界。为了进一步提高算法的性能，还提出一种名为AdDPNFL自适应优化算法，该算法在服务器端采用自适应优化，既减轻了非独立同分布数据的影响，又有助于提高差分隐私噪声下的算法准确度。类似地，通过理论分析，同样给出了该算法下目标函数为强凸和非凸时的严格收敛上界。

方法在MNIST和Fashion-MNIST数据集上对强凸目标函数下的理论结果进行性能评估。为实现非独立同分布数据，将数据分布在所有客户端中并保证每个客户端都包含7位数字或服装样本。为研究数据非平衡性的影响，进一步调整客户端之间的样本数量，即在非平衡情况下，客户端之间的样本总量满足幂律分布；而在平衡数据集情况下，每个客户端分配相同数量的样本。类似地，对Fashion-MNIST数据集执行上述操作以验证算法通用性。

另一方面，采用SVHN和CIFAR10数据集来验证非凸目标函数下的理论结果。根据上述描述，同样对这两个数据集进行数据分配以获取非独立同分布数据。在平衡和非平衡SVHN数据集下的实验结果进行代表性说明，对于非凸情况，在平衡SVHN和非平衡SVHN数据集上和深度神经网络模型下进行实验。非隐私AdDPNFL在SVHN数据集上的测试准确率为86.74%，展示了自适应服务器聚合方法的优势。此外， SVHN数据集下的观察发现几乎适用于MNIST 数据集下的情况。但由于SVHN中的训练数据格式比MNIST中的更为复杂，因此可以观察到，在平衡数据下，SVHN中的变化比MNIST中的变化较为波动。因为同样的原因，非平衡SVHN数据集下的学习曲线比平衡SVHN下的学习曲线波动更大。此外，值得说明的是，在较大的模型结构和复杂数据集下，所提算法仍然比DPSCAFFOLD的算法性能更好。DPSCAFFOLD算法在进行大规模深度学习时，只对每一轮的梯度漂移进行近似恢复，而不能将其根除，因此通常会出现退化现象。这种残差会随着训练的进行而累积增长，因此导致较慢的收敛速度和较差的算法性能。但对于DPNFL而言，由于没有引入额外的控制变量来应对非独立同分布问题，算法性能对模型和数据集的反应较小。

2.3 HamOS：基于汉密尔顿蒙特卡洛的分布外样本生成和检测方法

分布外检测是构建可信可靠机器学习系统的关键环节。尽管深度学习模型在各类实际任务中表现突出，但面对非训练数据分布的输入时，往往会做出过度自信的错误预测，这在自动驾驶、医学影像分析等安全关键领域可能引发严重风险。现有分布外检测方法主要分为后处理方法与正则化方法两类。后处理方法通过设计鲁棒评分函数（如最大 softmax 概率、马氏距离等）区分分布内与分布外样本，但受限于固定模型架构，性能存在瓶颈；正则化方法则在训练阶段引入额外约束，通常结合辅助分布外数据提升模型辨别能力，整体表现优于后处理方法。其中，异常暴露类方法通过引入自然分布外数据增强检测效果，但严重依赖高质量自然异常样本的获取，在许多特定领域中难以实现；而虚拟异常样本合成方法虽可避免对自然分布外数据的依赖，却因采用单一高斯采样策略或依赖复杂参数化生成模型，存在合成样本多样性不足、计算成本高的问题。

针对上述挑战，研究小组提出了汉密尔顿蒙特卡洛异常样本合成（Hamiltonian Monte Carlo Outlier Synthesis, HamOS）框架，将异常样本合成过程建模为基于马尔可夫链的采样过程，仅依赖 ID 数据即可生成多样且具有代表性的虚拟异常样本，有效让模型学习潜在分布外场景，同时凭借近 100% 的采样接受率保证生成过程的高效性。该框架的核心工作流程如图1所示，主要包含特征嵌入、分类分支与潜在嵌入三大模块，具体实现分为异常样本合成与模型训练两个关键阶段。框架采用双分支结构，骨干网络负责特征提取：（1）全连接（FC）分支保留原始分布内数据的分类性能；（2）投影头将特征嵌入转换至低维单位超球面空间，基于汉密尔顿蒙特卡洛算法与创新的分布外似然程度估计实现异常样本显式生成。通过分布内对比损失与分布外辨别损失优化超球面空间，增强分布内与分布外样本的区分度。

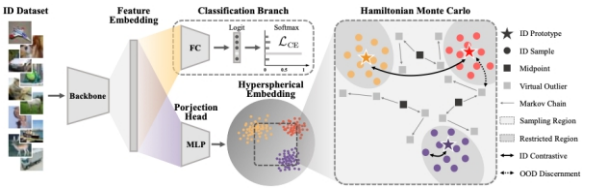


图1 HamOS方法的异常值采样工作流程

异常样本合成的核心是通过汉密尔顿蒙特卡洛算法在超球面特征空间中生成具有多样分布外特征的样本，关键步骤包括分布外似然程度估计、汉密尔顿采样与错误样本过滤。分布外似然程度评估机制在超球面空间中评估样本来自未知类别的概率水平，通过计算候选数据点相对于分布内特征嵌入的平均KNN距离来估计分布外似然概率，即$P^{oo D}(z; Z_c)=|z-z_{c(k)}|_2$。在生成过程中，HamOS选择具有较高分布外概率似然较高的数据点作为虚拟异常值，确保生成的异常值位于分布内类别簇之间，同时具有多样性和代表性。将分布外概率的负对数定义为势能函数$U^{oo D}(z; Z_u, Z_u)=-log\ P^{oo D}(z; Z_u, Z_u)$，结合动量变量 q 构建汉密尔顿量$H(z,q)=U^{oo D}(z)+\frac{2}{2}|q|_2^2$。采用球面汉密尔顿蒙特卡洛的蛙跳离散化更新位置与动量，确保新样本始终位于单位超球面。采样初始点设为两个分布内类别聚类中心的归一化中点，保证从低分布外概率区域逐步向高分布外概率区域迭代。引入基于冯–米塞斯–费希尔核的核密度估计方法计算分布内概率，设置硬边界阈值用于过滤位于分布内聚类内的错误样本。硬边界阈值定义为初始中点分布内概率对数，在采样过程中仅接受分布内概率低于该阈值的样本点。最后，通过结合分布内对比损失和分布外判别损失，训练模型在进行原始分类任务的同时，提升对分布外概率的检测性能。

为了评估HamOS的有效性，研究小组分别在小规模数据集CIFAR10和CIFAR100，以及大规模数据集ImageNet-1K上与多种先进基线方法进

行对比实验。实验结果表明，HamOS的分布外检测性能显著优于现有方法，对于关键指标FPR95，在上述三个数据集上分别提升35.63%，9.20%，以及2.98%。广泛的消融实验也验证了HamOS方法在不同实验设定下的稳健性和泛化性。

3 大数据处理技术

3.1 基于性能剖析的大模型算子内并行训练系统

大模型训练需要多层次的并行方法，在算子内并行方面，并行方案的搜索空间会随着算子数量增长而呈现出指数级的增长。

现有框架依赖静态代价模型对搜索空间内的方案进行评估和搜索，但静态模型无法准确捕捉编译器优化和底层运行时库的性能变化，这导致预测性能与真实性能偏离严重。针对算子内并行搜索空间大，现有方法依赖静态模型预测不准的问题，该研究的目标是在指数增长的搜索空间中提取出具有代表性的模型片段，显著减少需要评估的方案数量。通过对这些代表性片段进行实际性能剖析，使用真实运行性能作为代价模型来评估和搜索最优并行方案。

总体来说，该研究将巨大的并行方案搜索空间转换为了一个代表性程序的性能剖析空间。该研究首先通过元素级的数据依赖分析，识别计算图中保持并行性的数据流结构，将子图内多个算子聚合为粗粒度的并行单元，从而大幅缩减搜索空间，如图2所示。接着，该研究利用大模型在结构上的重复性，在计算图中提取出有限数量的独特模型片段，对这些代表性片段进行编译和运行，获取其在不同并行策略下的真实执行开销。随后，基于这些片段的真实执行开销，组合构建全局执行代价模型，并搜索全局最优并行方案。这种方法既保证了搜索效率，也提升了性能预测的准确性，如图3所示。

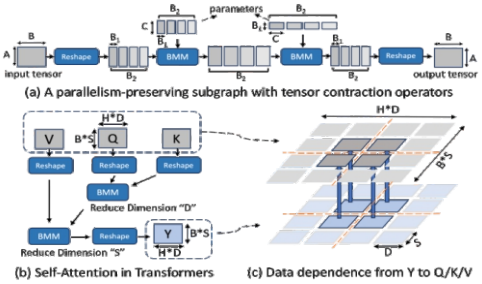


图2 元素级数据依赖分析，聚合并行性一致算子

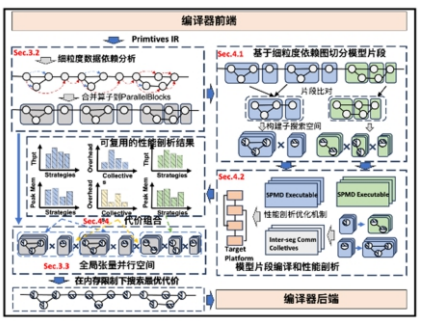


图3 算子内并行方案自动生成

该研究在两个A100节点和一个V100节点上进行了测试，相对领先的Alpa系统获得了显著的性能提升（如图4所示），同时，基于3个大模型的效率实验表明所提出的算法可以将搜索时间控制在15分钟以内。

3.2 基于GPU的椭圆曲线密码学高吞吐量框架gECC

椭圆曲线密码学（ECC）因较RSA具备更小密钥尺寸和更低计算复杂度，在区块链、安全多方计算和数据库加密等领域广泛应用。然而，ECC的核心操作（如点加、点乘）依赖高开销的模运算，尤其是模逆运算耗时可达模乘运算的500倍。现有GPU加速方案存在三大局限：缺乏批处理能力，仅优化单次ECC操作，无法利用

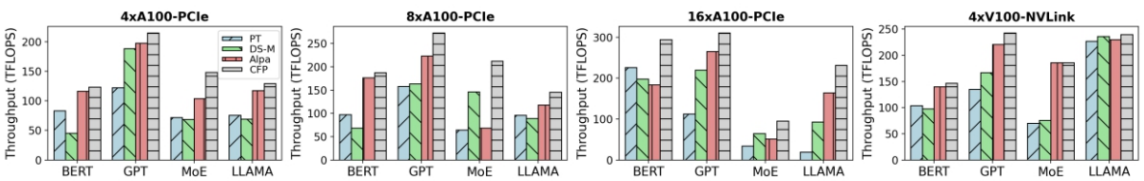


图4 实验结果

GPU大规模并行性；内存访问效率低，批量计算时中间数据（如百MB级）超出GPU共享内存容量；模运算指令级瓶颈，未针对GPU微架构优化底层指令流，整数乘加指令利用率低。

研究提出gECC框架（图5），旨在构建首个支持批量ECC操作的GPU优化框架。批量处理时，借鉴图处理系统的的Gather-Apply-Scatter（GAS）模型，减少并行蒙哥马利技巧中模逆运算的开销；设计GPU友好的内存管理，解决批处理时中间数据爆炸问题；从微架构层精简模乘运算中乘加指令的数量。

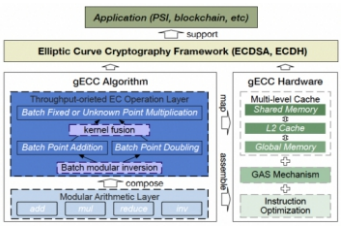
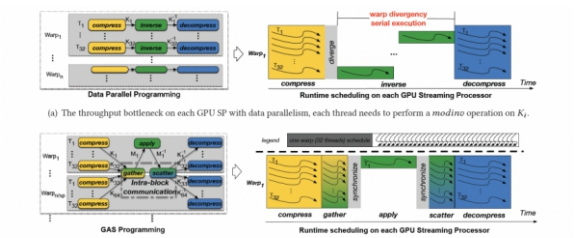


图5 gECC框架

gECC采用蒙哥马利技巧批量处理ECC点加、点乘操作。如图6所示，将N次模逆运算合并为1次模逆运算和3N次模乘运算，当批量规模N>20时显著优于现有研究方案。图6(a)所示，数据并行时每个线程处理一次模逆运算，模逆运算的算法含分支语句，会导致GPU线程束分化从而影响性能。gECC采用GAS模型（图6(b)）解决多个模逆运算并行化瓶颈，通过累乘的方式合并（gather）一个GPU block内部的多个线程的结果，仅使用一个线程对累乘结果执行（apply）一次模逆运算，再将模逆运算结果分发（scatter）到每个线程，再通过累乘得到每个线程的模逆运算结果。



(a) The throughput bottleneck on each GPU SP with data parallelism, each thread needs to perform a *modinv* operation on K_i.

(b) gECC's runtime execution on each GPU SP with GAS mechanism. After *compress step*, threads in consecutive n/sp warps use shared memory for intra-block communication and tree-based cooperative reduction to get the accumulated products M_j, j from 1 to *sp*.

Fig. 3. Different mechanism of batch modular inversion on GPU and corresponding runtime execution. The total input is N and the number of programmer-defined warps is *n*. T_i represent thread id, i from 1 to 32*n*. Each thread processes the inverse of $N/32n$ inputs.

图6 GAS模型处理多个并行模逆运算

gECC重构数据布局为列优先存储，使GPU线程束可聚合访问大整数。同时gECC采用多级缓存与内存管理（图7），根据GPU Ampere架构L2持久化缓存特性，将gather步骤产生的中间结果缓存到L2 cache上，在apply阶段中从对应的L2 cache读取从而提高中间数据的访问效率。同时gECC通过重计算减少45%内存使用。

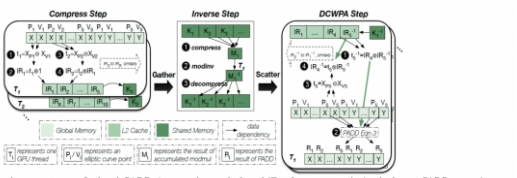


Fig. 4. Multi-level cache management for batch PADD. Assume that each thread (T_i, *i* from 1 to $32n/sp$) calculates 5 PADD operations (e.g., $R_1 = P_1 \bullet V_1$). The figure shows the memory access of the calculation within a GPU block. *compress* and *DCWPA* show the calculation of each thread. ❶ and ❷ in *inverse step* use intra-block communication to obtain M_i and K_i^{-1}, respectively.

图7 多级缓存管理设计

最后通过对现有方案的微指令架构的分析，得到整数乘加指令是模乘运算性能瓶颈。所以通过谓词寄存器传递进位信息，减少整数乘加冗余指令。针对中国标准SM2曲线，利用素数的特殊形式，以更高吞吐的整数加法指令替代乘加指令。

gECC通过批处理架构、内存层级优化及指令级重构，成为首个支持高吞吐ECC的GPU框架，在数字签名算法（ECDSA）签名生成和签名验证上相对目前最快GPU方案分别有4.18倍和5.56倍性能提升。在ECDH密钥交换有4.94倍提升。在模乘运算有最高1.72倍性能提升，推动高性能隐私计算落地。代码已在GitHub开源（https://github.com/CGCL-codes/gECC）。

3.3 基于FPGA的流式数据专用硬件加速器

面向流式大数据实时处理中面临的高吞吐、低延迟计算挑战，设计了一种基于FPGA的硬件加速架构。首先采用可动态调整的循环逻辑长度结构，优化计算资源分配，显著降低数据处理时延；其次结合高压缩存储策略，大幅减少内存占用与数据读写开销，提升系统能效；此外引入多通道并行流水线机制，充分发挥FPGA硬件并行性，实现数据吞吐量的数量级提升。为实现执行算法的细粒度并行化，采

用三重优化策略。首先，对评分矩阵中元素间的依赖关系进行深入分析，以挖掘并行计算潜力。其次，采用2比特变量记录评分矩阵计算过程中的回溯路径，降低FPGA的存储资源开销。最后，在任务级并行方面，将多个延伸数据整合为单一数据包进行处理，有效减少主机与FPGA 之间的通信开销，提升系统效率。实验验证表明（图8），基于FPGA的流式数据专用硬件加速器在实际流式数据处理场景中，相比GPU方案在性能上展现出显著优势，尤其适用于需要实时响应的高频数据流处理场景。

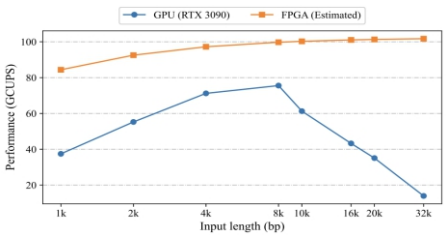


图8 超关系抽取模型的整体工作流程

4 大数据分析和应用

4.1 WebCode2M：真实场景网页设计到代码生成数据集

网页设计到代码（Design-to-Code）任务是前端自动化开发的重要方向，但现有多模态大模型在该任务上的性能受到数据不足的严重制约。以往的数据集要么规模过小（如Design2Code仅484个样本），要么主要为合成数据（如WebSight），无法覆盖真实网页的复杂性。为此，研究团队提出并构建了WebCode2M，这是首个大规模真实网页设计到代码生成的数据集，涵盖256万条样本，每个样本包含网页截图、对应 HTML/CSS代码以及布局树信息。

如图9所示，在数据构建中，研究团队基于Common Crawl采集网页，经过代码净化、截图渲染和神经网络评分器的多层过滤，确保数据质量；最终通过布局树提取形成三元组数据（图像—代码—布局）。与合成数据相比，WebCode2M 的样本规模更大（平均 3.1 万 Token，158 个标签，DOM 深度 13），显著提升了结构多样性与真实度。

为了验证有效性，研究团队基于ViT设计了基线模型 WebCoder，并提出新的结构化指标TreeBLEU，在真实场景下相比 WebSight 和Design2Code基线显著提升结构召回率和视觉相似度。同时，WebCode2M还对GPT-4o、Gemini 等通用MLLM进行了系统性基准测试，结果表明该数据集能显著提升模型在复杂网页生成中的表现。WebCode2M已在HuggingFace平台公开，下载量逾万次，成为设计转代码领域的重要国际基准。

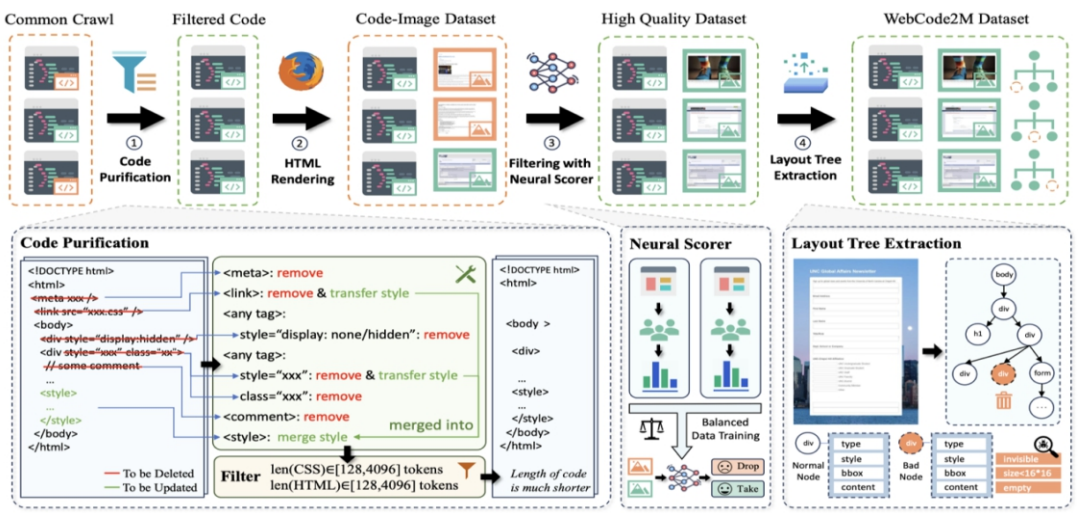


图9 数据集构造过程

4.2 LaTCoder：基于Layout–as–Thought的网页设计转代码方法

尽管多模态大模型（MLLMs）在Design-to-Code任务中展现出潜力，但其布局保持能力不足，容易在复杂网页生成中丢失部分结构信息（例如GPT-4V在真实网页案例中将横向排列误生成为纵向）。研究团队受到Chain-of-Thought (CoT) 推理启发，提出了 Layout-as-Thought (LaT) 策略，并基于此设计了新方法LaTCoder，如图10所示。

LaTCoder首先通过高效算法将网页设计划分为布局感知的图像块，结合OCR保证文字完整性；随后利用CoT提示逐块生成 HTML/CSS 代码；最后通过绝对定位与MLLM组装双策略结合动态选择机制进行代码拼装，从而最大化保留设计布局信息。此外，团队构建了更具挑战性的CC-HARD基准（平均 DOM 深度16，标签数274），作为复杂布局下的测试平台。

为了验证所提出方法的有效性，在Design2Code-HARD与CC-HARD上的实验结果表明，LaTCoder在 DeepSeek-VL2、Gemini、GPT-4o等多种骨干MLLM上均取得显著提升，例如在GPT-4o上，TreeBLEU提升60%，MAE降低43.23%，视觉得分提升2.56%。人工评测显示，超过60% 的标注者更偏好LaTCoder生成的网页，尤其在复杂布局场景中表现优越。该工作不仅提出了新的方法论，还建立了新的数据基准，推动了复杂UI场景下的智能代码生成研究。

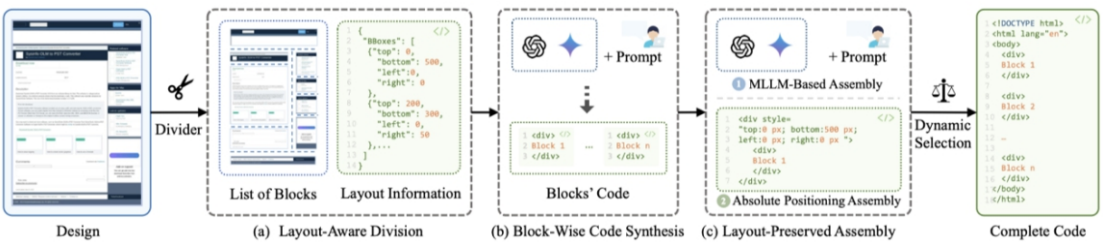


图10 LatCoder方法架构图

4.3 Nester：数据流引导的神经–符号融合类型推断模型

在真实软件开发中，类型推断对保障代码可靠性和提升开发效率至关重要，但受制于 模型规模与本地部署约束，现有大模型难以直接应用。研究团队提出Nester，这是首个数据流引导的神经–符号融合方法，在不增加模型规模的情况下显著提升类型推断能力，如图11所示。

Nester将类型推断任务分解为基于数据流与控制流的子任务，并编码为模块化程序，逐步执行表达式求值、分支分析等动作；同时结合静态类型检查与语言模型，推理潜在的类型信息，从而实现更精细的推断过程。

在ManyTypes4Py数据集上的实验显示，

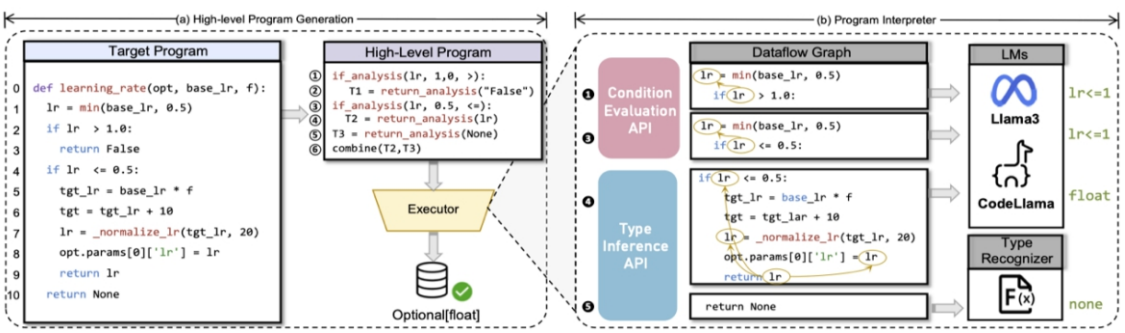


图11 Nester方法架构图

Nester的Top-1精确率达到70.7%，比HiTyper和TypeGen分别提升18.3%和3.6%；在复杂类型（如Optional、Union）预测上分别达到51.0%和16.7%，相较TypeGen 提升28.3%和5.8%。该成果验证了神经网络与符号化方法结合的优势，为代码智能在轻量化、本地化部署中的应用开辟了新方向，也为可信AI在软件工程中的实践提供了新思路。

4.4 一种基于节点插入的图神经网络公平性攻击

尽管图神经网络已经在处理图结构数据中展现出了非常强大的实力，越来越多的研究发现，图神经网络（GNN）在面临对抗攻击时其实非常脆弱。然而，大多数已有的攻击方式都仅关注于模型的预测是否准确，而忽略了GNN 公平性的鲁棒性和脆弱性。这自然而然地引出了一个问题：“GNN的公平性是否也容易遭受对抗攻击呢?”

事实上，不同于传统的攻击方法，公平性攻击要求在不影响模型准确性的前提下对模型的公平性指标进行影响。为实现这一目的并尽可能符合实际应用场景，文章提出了一种基于节点插入的公平性攻击方法 — NIFA。具体而言，NIFA仅需通过向原始图中插入新的虚假节点即可完成攻击，而无需更改原始图中已经存在的节点和连边。这种攻击方式可以大大降低攻击者的攻击难度和成本。例如，在社交网络中，攻击者仅需创建若干僵尸账户即可进行攻击，而不再需要修改真实用户的个人资料和好友列表。

基于节点插入的公平性攻击主要面临两大挑战：1. 如何进行节点插入？作为攻击的第一步，节点插入的位置非常重要。但节点插入是一个非连续的动作行为，而由于动作空间较大，使用强化学习等优化方式将带来极大的时间开销。2. 在插入节点后如何确定虚假节点的特征？与真实节点相同，虚假节点也会参与到消息传播过程当中，如何设计虚假节点的特征也将影响最终的公平性攻击效果。针对上述挑战，文章提出了一个灰盒毒化攻击模型NIFA，结构如图所示。具体而言，如图12所示，NIFA 首先提出了两个节点插入原则 —— 不确定性最大化原则和同质率增加原则，这两个原则的提出帮助NIFA能够以较高地效率完成虚假节点的插入，并有效影响模型最终的公平性。其次，在完成节点插入之后，NIFA设计了多种目标函数用于优化虚假节点的特征，进一步从特征层面上确保了攻击的有效性。

文章在Pokec-z, Pokec-n和DBLP三个真实数据集上，对GCN，APPNP，SGC，GraphSAGE，FairGNN，FairVGNN和FairSIN等经典GNN模型和公平性GNN模型进行了毒化攻击。实验表明NIFA仅需要1%的节点插入量，即可在不影响模型性能的前提下大幅恶化GNN模型的公平性指标。

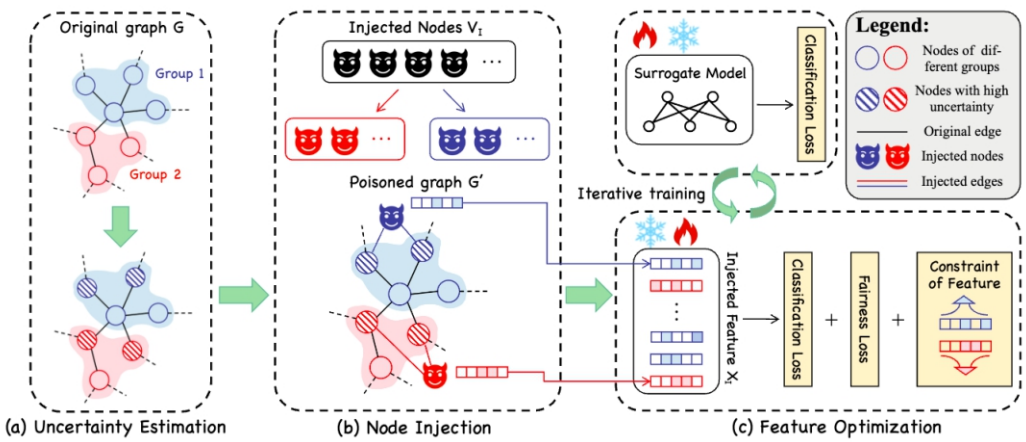


图12 NIFA的总体架构

4.5 图神经网络混合公平性优化框架

图神经网络（Graph Neural Networks, GNNs）在挖掘图结构数据方面展现出了强大的能力。然而，传统的 GNN 往往会面临各种公平性问题。例如，当涉及到具有不同敏感属性（如性别或种族）的节点时，模型可能会产生带有偏见的预测；又或者在面对度数差异较大的节点时，预测性能存在显著差别。现有研究大多聚焦于解决某一种特定的公平性问题，但现实中，GNN 模型往往会同时面临多种不公平，仅仅解决单一问题仍可能让模型处于不公平的状态。

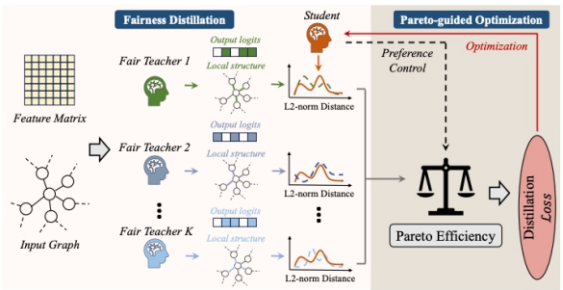


图13 LibraGNN的总体架构

研究人员关注的是如何在 GNN 中同时实现多种公平性，将其称为混合公平（Hybrid Fairness）。为此，提出了一种新颖的 GNN 框架——LibraGNN，框架示意图如图13所示。具体来说，针对不同公平性问题的成因不同这一特点，首先采用了多教师知识蒸馏（multi-teacher knowledge distillation）的训练框架，将多种公平性的学习范式成功统一在一起，避免了过于复杂的模型结构设计，并使得整个优化框架拥有比较好的可扩展性。考虑到不同的公平性指标之间可能存在潜在的矛盾关系，为了在不同公平性之间取得更好的平衡，进一步将多教师知识蒸馏转化为一个多目标优化问题，并进一步引入帕累托最优（Pareto efficiency）来指导优化。具体而言，帕累托最优是指在一个多目标权衡的系统中，如果要让某个目标变得更好，就必然会牺牲另一个目标；此时的状态被称为帕累托最优。最后，设计了一个可控的偏好向量，用来帮助 LibraGNN 在多种公平性之间灵活调节偏好权重，从而实现可控的混合公平性定义。

在Pokec，Weibo等三个真实数据集上进行了大量的实验，实验结果表明LibraGNN可以在不牺牲模型准确率的前提下大幅提升模型的混合公平性，基于GCN模型的LibraGNN分别在三个数据集上提升混合公平性指标55.10%，64.54%，和69.18%，远超于现有的图神经网络和其他公平性算法。

4.6 一种基于正交性的时间序列分布外分类方法

传统的时间序列分类工作严重依赖于训练数据和测试数据属于同一分布的假设，在分析分布外（Out-of-distribution，OOD）数据时，它们的性能往往会大幅下降。不幸的是，分布变化在现实生活中很普遍，比如广泛存在于医疗领域的脑电信号，采集设备和个体特征的变化可能导致分布变化，使得一般方法无法推广到未知设备和人群。实际上，不同的个体和设备可以看作是不同的域，这些域之间存在着分布偏移，也被称为域偏移。

因此，越来越多研究人员关注于OOD时间序列分类。通常，他们将时间序列映射为域无关以及域特定的特征。前一个特征在各个域保持不变，在下游任务中起着至关重要的作用，而后一个特征表明它们在每个域都有所不同。这些方法通常设计下游任务来最小化这两个特征之间的相似性，并利用域无关的特征进行OOD泛化。然而，以前的工作只关注在任务级别优化域无关和域特定的特性。因此，它们生成的两个特征仍然表现出高度的相似性，表明域无关特征中仍然存在着特定于域的信息，并且仍然受到分布偏移的影响。

为了解决这个问题，提出了一种简单而有效的方法，称为不变时间序列表示（Invariant Time Series Representation，ITSR），旨在将时间序列分解为相互正交的不变特征和相关特征，其中不变特征在不同域的下游任务中起着至关重要

的作用，相关特征则暗示着域的变化。如图14所示，ITSR首先利用编码器从时间序列中提取信息，然后将其分解为不变特征和相关特征。不变特征不受分布移位的影响，并能决定相应时间序列的标签。另一方面，相关特征表示随分布变化而变化的特征。ITSR的核心概念是通过正交性确保不变特征和相关特征之间的低相似性。为此，ITSR维持两组正交轴(称为不变轴和相关轴)，并将时间序列投影到这两组轴上，得到不变特征和相关特征，从理论上保证了它们的低相似性。随后，利用这两个特征分别对标签和域进行分类，并利用交叉熵损失函数来优化模型。

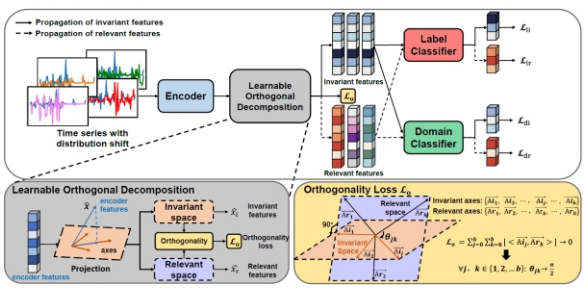


图14 ITSR的总体架构

选取了四个真实的时间序列数据集，每个数据集都包含多个域，各个域之前存在分布偏移，以评估ITSR的表现。在四个真实数据集上的结果表明ITSR通过维持不变特征和相关特征之间的正交性提高了泛化能力，并且在性能方面优于最先进的方法约5%。

4.7 一种自监督的基于补丁匹配的图像拼接方法

无创的胶囊内窥镜技术为患者带来了极大的便利，但也对医生的阅片工作提出了新的挑战，尤其是在复杂的胃肠道环境中，海量的碎片化图像使得图像拼接与定位变得困难，增加了精准检测病灶区域的难度。然而，创新性的自监督补丁级匹配算法为这一难题提供了有效的解决方案。通过该算法，胶囊内镜图像的高效拼接不仅变得可行，而且极大提高了处理纹理较弱、视角变化大、拍摄角度复杂的图像的精度。这一技术突破填补了传统内镜无法实现精准拼接与定位的空白，帮助医生更全面、清晰地观察胃肠道区域，极大提升了胃肠道疾病筛查的效率。凭借这一算法，医生能够在短时间内作出更准确的诊断和治疗决策，从而推动无创内镜技术在临床中的更广泛应用，为患者提供更高效的医疗服务。

研究背景源自胶囊内窥镜（MCCE）的局限性。MCCE是一种用于胃肠道检查的非侵入性设备，具备无痛和避免交叉感染的优势。然而，由于其摄像头受胃肠蠕动控制，MCCE难以有效捕捉医生关注的特定区域（即感兴趣区域，ROI），往往只能拍摄到大量碎片化、视角不固定的图像。由于这些图像存在弱纹理、视角变化大、近距离拍摄等问题，导致传统的图像匹配方法难以处理。

研究提出了一种自监督的基于补丁匹配的图像拼接方法（S2P-Matching，如图15所示），专门用于处理胶囊内窥镜图像的拼接问题。通过利用增强的对比学习与Transformer网络，该方法能够自动提取和匹配胃肠道内的图像片段，提升了图像拼接的准确性和成功率。这一技术的应用将有助于医生通过更广阔的视野来观察病变区域，进而提升早期发现和诊断胃肠道疾病的能力。

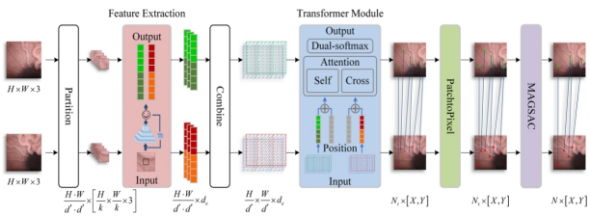


图15 S2P-Matching方法的处理流程图

研究首先解决了图像匹配的难题。MCCE拍摄的图像通常是分散且纹理较弱的，传统的特征描述符方法（如SIFT和ORB）难以应对这些图像的匹配问题。针对此，S2P-Matching通过仿真胶囊内镜的拍摄行为，基于仿射变换增强式的

生成了仿真图像数据集。其次，S2P-Matching引入改进的自监督对比学习方法，采用双分支编码器来提取局部特征。这些特征用于训练Transformer模型，以进行补丁级别的图像匹配，最终通过Patch-to-Pixel方法细化为像素级匹配。

论文使用胶囊内镜图像来评估其提出的S2P-Matching方法的性能，进行了一系列实验来验证其与其他图像匹配算法（如CAPS、ASIFT、DeepMatching、R2D2、SuperPoint等）的效果对比。实验结果表明，S2P-Matching在所有实验类型（弱纹理、近距离变换、大角度旋转）中均表现出最高的NCM（正确匹配点数）和SR（成功率）分数，平均NCM为311，平均SR为81.7%。与传统算法相比，其匹配准确率明显提升。在胶囊内镜图像的连续帧拼接中，S2P-Matching的拼接效果最为自然，能够有效应对图像弱纹理和旋转等难题。与其他算法相比，该方法生成的匹配对最多，拼接结果的纹理连接自然，无明显的错位和缩放问题。

4.8 面向时序数据的深度学习模型显著性解释方法研究

在疾病早期，病灶往往“藏得很深”，也就是疾病早期的病灶往往与正常组织极为相似，即使是资深医生也难以凭肉眼从医学影像中准确识别出异常之处。这为早期诊断带来了极大挑战。传统医学影像分类模型（如ResNet、VGG等）在面对细粒度图像差异时表现不佳。尤其是在疾病的早期阶段，病灶区域可能只有轻微的模糊、阴影或结构变化，很容易被忽略。

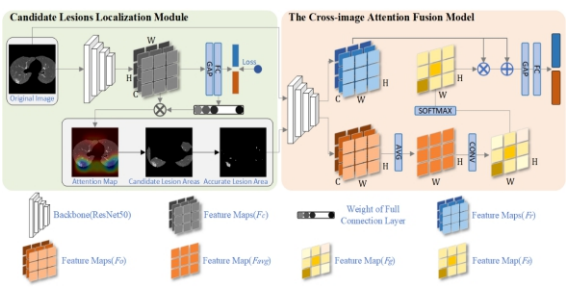
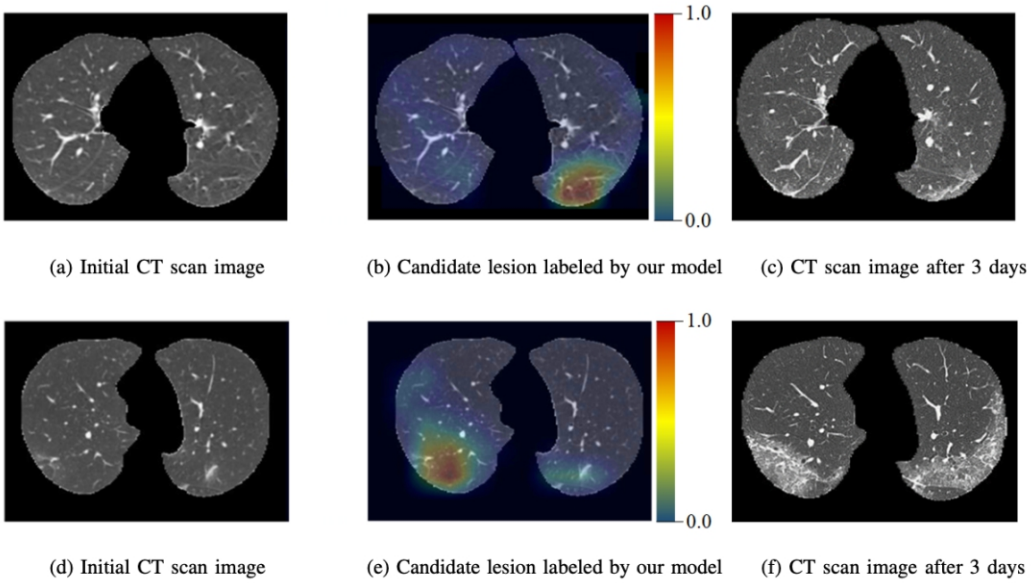


图16 研究提出的精细病灶分类框架

研究提出了一种面向早期辅助诊断的精细病灶分类框架（Fine-Grained Lesion Classification Framework），创新性地结合了注意力机制与细粒度视觉分类（FGVC）技术，显著提升了医学图像中早期病灶的识别与分类准确率。如图16，框架由两个核心模块组成，分别是“候选病灶定位模块（Candidate Lesions Localization Modeule）”与“跨图像注意力融合模型（The Cross-image Attention Fusion Model）”。

病灶定位模块（Candidate Lesion Localization Module）的目标是从整体医学图像中自动识别出具有判别性的“候选病灶区域”，即图像中可能蕴含病变线索的关键区域。为此，作者引入了类激活图（Class Activation Map, CAM）的机制，通过分析神经网络中高层特征图的响应强度，判断哪些图像区域对分类决策贡献最大。跨图像注意力融合模型（Cross-Image Attention Fusion Model）用于模拟临床医生“先看病灶，再看环境”的认知路径。具体做法是：先对提取到的病灶区域进行特征编码，生成一张“空间注意力图（Spatial Attention Map）”，这张图明确指出在整幅原始图像中，哪些位置与病灶特征高度相关。随后，该注意力图被用于对原始图像的全局特征图进行加权处理，通过”点乘运算强化关键区域特征表达“，同时保持空间位置不变，保留病灶周边组织的信息。最后，模型将原始图像的原始特征图与强化后的特征图在通道维度上拼接（concatenate），由后续全连接层自动学习其重要性权重，实现融合后更具判别力的图像表示。这一机制不仅提升了模型对“细粒度异常区域”的关注能力，还有效融合了病灶与上下文之间的语义联系，体现出仿照医生“以点带面”“局部—整体—局部”诊断思维的智能特征。

通过上述两个模块的联合优化，模型不仅能够自动识别出早期、微小、形态复杂的病灶区域，还能够“借助注意力机制提升分类性能与可解释性”，为AI在医学辅助诊断中的可信



(a) Initial CT scan image (b) Candidate lesion labeled by our model (c) CT scan image after 3 days

(d) Initial CT scan image (e) Candidate lesion labeled by our model (f) CT scan image after 3 days

图17 研究提出的精细病灶分类框架识别出了新冠早期在CT影像中的表现，该表现肉眼很难辨识

应用打下坚实基础。

实验结果表明，对于COVID-19分类任务（轻症+普通型），准确率提升最高达35.50%，F1分数提升36.61%，AUC达到0.8803，显著优于ResNet、ViT、CBAM等模型。在图17所示的例子中，第一次CT检查中的病灶区域很难被人眼所发现。的方法标注出的区域在第二次检查中发展成明显病灶。在ONFH早期检测任务中，由于X光图像在I期常无明显骨质塌陷或坏死特征，传统模型识别难度极大。而该方法凭借病灶定位与注意力融合机制，能够在视野中突出微小病变区域，最终实现了0.9179的AUC和0.8468的F1-score。尤其值得注意的是，方法能识别出其他模型容易忽视的双侧轻度病灶，验证了其在早期无症状阶段具备潜在的临床辅助诊断价值。

附成果列表论文

[1] Jie Cheng, Lifu Hu, Wei Xu, Hanhua Chen, Tian Xia, Hardware Acceleration of Minimap2 Genomic Sequence Alignment Algorithm, In Proceedings of the 53rd International Conference on Parallel Processing (ICPP), Gotland, Sweden, pp. 887-897, August 12-15, 2024.

[2] Yi Gui, Zhen Li, Yao Wan, Yemin Shi, Hongyu Zhang, Yi Su, Bohua Chen, Dongping Chen, Siyuan Wu, Xing Zhou, Wenbin Jiang, Hai Jin, Xiangliang Zhang, WebCode2M: A Real-World Dataset for Code Generation from Webpage Designs, In Proceedings of The Web Conference (WWW), 2025.

[3] Yi Gui, Zhen Li, Zhongyi Zhang, Guohao Wang, Tianpeng Lv, Gaoyang Jiang, Yi Liu, Dongping Chen, Yao Wan, Hongyu Zhang, Wenbin Jiang, Xuanhua Shi, Hai Jin, LaTCoder: Converting Webpage Design to Code with Layout-as-Thought, In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2025.

[4] Gen Li, Yao Wan, Hongyu Zhang, Zhou Zhao, Wenbin Jiang, Xuanhua Shi, Hai Jin, Zheng Wang, Dataflow-Guided Neuro-Symbolic Language Models for Type Inference, In Proceedings of the 42nd International Conference on Machine Learning, 2025.

[5] Zihan Luo, Hong Huang, Yongkang Zhou, Jiping Zhang, Nuo Chen and Hai Jin, Are Your Models Still Fair? Fairness Attacks on Graph Neural Networks via Node Injections, In Proceedings of the 38th Conference on Neural Information Processing Systems (NeurIPS), December 10-15, 2024, Vancouver, Canada.

[6] Zihan Luo, Hong Huang, Jianxun Lian, Xiran Song, and Hai Jin, Towards Controllable Hybrid Fairness in Graph Neural Networks, In Proceedings of the

31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), 2025.

[7] Ruize Shi, Hong Huang, Kehan Yin, Wei Zhou, Hai Jin, Orthogonality Matters: Invariant Time Series Representation for Out-of-distribution Classification, In Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining（KDD）, August 25-29, 2024, Barcelona, Spain, pp. 2674-2685.

[8] Yi Chen, Qiang-Sheng Hua, Zixiao Hong, Lin Zhu, Hai Jin, FHE4DMM: A Low-Latency Distributed Matrix Multiplication with Fully Homomorphic Encryption, IEEE Trans. Parallel Distributed Syst., 2025, 36(4): 645-658

[9] Chen Lin, Ding Xiaofeng, Bao Zhifeng, Zhou Pan, Jin Hai, Differentially Private Federated Learning on non-iid Data: Convergence Analysis and Adaptive Optimization, IEEE Transactions on Knowledge and Data Engineering, 2024, 36(9): 4567-4581.

[10] Qian Xiong, Weiliang Ma, Xuanhua Shi, Yongluan Zhou, Hai Jin, Kaiyi Huang, Haozhou Wang, and Zhengru Wang, GECC: A GPU-based high-throughput framework for Elliptic Curve Cryptography, ACM Transactions Architecture Code Optimization. Just Accepted (May 2025). https://doi.org/10.1145/3736176

[11] Feng Lu, Dao Zhou, Haoyang Chen, Shuai Liu, Xianliang Ling, Lei Zhu, Tingting Gong, Bin Sheng, Xiaofei Liao, Hai Jin, Ping Li, David Dagan Feng, S2P-Matching: Self-supervised Patch-based Matching Using Transformer for Capsule Endoscopic Images Stitching, IEEE Transactions on Biomedical Engineering, https://doi.org./10.1109/TBME.2024.3462502.

[12] Feng Lu, Wei Li, Canyu Li, Shuai Liu, Dong Wu, Minghao Fang, Xiaojing Zou, Mi Li, Ran Zheng, Yufei Ren, Xiaofei Liao, Hai Jin, and Albert Y. Zomaya, Fine-grained Lesion Classification Framework for Early Auxiliary Diagnosis, IEEE-ACM Transactions on Computational Biology and Bioinformatics. 2024, 21(4):971-982.

石宣化
教　授
研究方向：大数据处理、异构计算
Email：xhshi@ hust.edu.cn

陈汉华
教　授
研究方向：大数据处理系统、分布式处理系统
Email：chen@hust.edu.cn

华强胜
研究员
研究方向：并行与分布式计算理论与算法
Email：qshua@hust.edu.cn

丁晓锋
教　授
研究方向：大数据管理系统、隐私保护、深度学习
Email：xfding@hust.edu.cn

陆　枫
副教授
研究方向：智慧医疗、分布式计算
Email：lufeng@hust.edu.cn

黄　宏
副教授
研究方向：网络表示学习、数据挖掘
Email：honghuang@hust.edu.cn

张　腾
副教授
研究方向：机器学习
Email：tengzhang@hust.edu.cn

万　瑶
副教授
研究方向：代码大数据
Email：wanyao@hust.edu.cn

当AI开始写算法——AlphaEvolve

（史瑞泽 https://blog.csdn.net/lei967809/article/details/148033914）

引言

在人工智能以惊人速度持续演进的今天，新模型、新技术层出不穷，不断颠覆人类对于计算与创造的既有认知，重塑着各行各业的运作逻辑与创新路径。尤其是在算法与优化领域，AI 的力量正在从辅助工具逐步演变为主动探索者与开拓者。2025 年 5 月 14 日，谷歌旗下的 DeepMind 团队正式发布了全新一代编程 AI 智能体——AlphaEvolve□ 这一面向算法发现与优化的智能系统，在发布瞬间便引爆全球科技与学术圈，成为媒体、研究机构及产业界共同关注的焦点。

AlphaEvolve 的诞生，标志着 AI 在“自主进化与创造”这一方向上的一次质的飞跃。它基于 Gemini 架构所驱动的进化式代码生成引擎，能够在无人干预的情况下不断尝试、优化与迭代算法，从而在极短时间内实现突破性成果。例如，在计算机科学中至关重要的矩阵乘法领域，它将传统 4×4 矩阵乘法的运算次数成功压缩至仅 48 次，刷新了长期以来被视为瓶颈的性能极限；在高维几何领域，AlphaEvolve 还攻克了困扰数学界长达 300 年的“密接数”难题，这一成就不仅震撼了数学研究社群，也为工程与数据处理带来了前所未有的可能性。

值得注意的是，AlphaEvolve 的能力并不局限于理论推导或数学计算，它具备从算法设计到硬件架构乃至芯片级实现的全栈优化潜力，为跨学科协作与产业落地打开了全新的想象空间。这一切不仅是技术突破的展示，更是新研究范式的开端。

AlphaEvolve 技术剖析

1. 核心技术融合

AlphaEvolve 并非单一技术的产物，而是深度学习模型、自动化验证机制和进化算法框架深度融合的结晶。它的底层依托 Gemini 大模型体系结构开展工作，其中 Gemini Flash 能够快速对大量数据进行洞察，在广泛的数据海洋中迅速捕捉有用信息，为算法生成提供丰富的素材与方向；Gemini Pro 则凭借强大的深度挖掘能力，深入剖析潜在规律，从复杂的数据关系里提炼出关键要点，助力生成更具深度和价值的算法。二者相互配合，犹如一个既有广度视野又有深度洞察力的智慧大脑，为算法生成奠定坚实基础。

2. 自动化评估与自进化机制

为了确保生成的算法既可靠又高效，AlphaEvolve 引入了“自动化考官”系统。这个系统运用深度强化学习技术以及大规模数据验证手段，对生成的算法进行全方位考核。通过设定一系列严格的评估指标，如算法的准确性、执行效率、资源消耗等，从多个维度衡量算法质量。在此基础上，借鉴“进化论”中的“优胜劣汰”思想，构建起一个结构化反馈循环。每一轮生成的算法都会在这个循环中接受评估，表现优秀的算法被保留并用于启发下一轮算法生成，而表现不佳的则被淘汰。如此一来，经过多轮迭代，每一代算法在性能、鲁棒性等方面都逐步超越前一代，不断趋近最优解。

AlphaEvolve 应用成果展示

1. 数学领域重大突破

在数学研究的广阔天地中，AlphaEvolve 展现出惊人实力。研究团队将其应用于分析、组合学、几何、数论等多个方向的超 50 个开放性难题研究中。令人瞩目的是，在约 20% 的问题上，它成功找到了超越人类现有认知的全新最优解。以困扰数学家们长达 300 多年的“接吻数问题”为例，该问题主要探讨在 n 维空间中，与一个公共单位球接触的最大非重叠球数量。AlphaEvolve 在 11 维空间中，创新性地发现了一种由 593 个外球组成的全新配置，为这一古老难题的研究开拓了新方向，建立了新的下限。此外，在矩阵乘法算法这一计算机科学基础问题上，它也取得显著进展。针对 4x4 复值矩阵乘法，它设计出仅需 48 次标量乘法的全新算法，成功超越 1969 年发布的 Strassen 算法，为该领域发展注入新活力。

2. 助力谷歌数据中心效率飞升

在实际业务应用方面，AlphaEvolve 为谷歌数据中心的高效运转立下汗马功劳。它为谷歌大规模集群管理系统 Borg 量身定制了一种简单却极为有效的启发式调度算法。该算法投入使用一年多来，稳定且持续地为谷歌全球计算资源实现了 0.7% 的回收。这一成果看似比例不高，但考虑到谷歌庞大的数据中心规模，意味着在任何时刻，相同的计算资源能够额外完成大量任务，极大提升了资源利用率，降低运营成本。

3. 推动芯片设计优化升级

在芯片设计领域，AlphaEvolve 同样发挥重要作用。它运用 Verilog 语言，对矩阵乘法关键算术电路提出巧妙重写方案，精准删除不必要的位。经过严格的功能正确性验证后，这一方案顺利被集成到谷歌下一代定制 AI 加速器张量处理单元（TPU）中。通过这种方式，AlphaEvolve 有效促进 AI 工程师与硬件工程师之间的紧密合作，大幅缩短未来专用芯片的设计周期，提升芯片性能与竞争力。

4. 加速AI模型训练进程

对于 AI 模型训练而言，AlphaEvolve 通过巧妙将大型矩阵乘法运算合理划分为多个更易处理的子问题，成功将 Gemini 架构中的关键内核运算速度提升 23%，进而使得 Gemini 模型整体训练时间缩短 1%。不要小看这 1% 的时间节省，对于动则需要长时间、高成本运算资源投入的大模型训练来说，这不仅意味着时间成本的降低，还能让研究人员在相同时间内开展更多次实验与优化，极大加速 AI 研究创新进程。同时，它还能对底层 GPU 指令进行优化，在 Transformer - based AI 模型的 FlashAttention 内核实现中实现高达 32.5% 的提速，助力开发者快速定位性能瓶颈，优化代码库。

总结与展望

AlphaEvolve 的诞生，无疑是 AI 发展历程中的一座重要里程碑。它突破传统算法设计过度依赖专家经验与手动调优的局限，赋予 AI 系统强大的“自我进化”能力，开启了算法自动化、智能化设计与优化的崭新时代。从解决复杂数学难题，到提升数据中心效率、优化芯片设计以及加速 AI 模型训练，其应用成果已在多个关键领域展现出巨大价值与潜力。展望未来，随着技术不断成熟与完善，凭借自身强大的通用性，AlphaEvolve 有望在材料科学、药物发现、可持续发展等更多领域大显身手，为解决全球性挑战提供创新解决方案，推动人类社会加速迈向智能化发展新阶段。

从“存”到“算”：大模型推理的内存卸载与计算卸载

（严 鑫 https://www.toutiao.com/article/7546265293899792931/?log_from=213a963defe8a_1757002187930）

一、引言

大语言模型（LLM）在自然语言处理、对话式AI、图像/视频生成、文档等领域释放了巨大潜能。但伴随能力而来的，是参数规模的持续膨胀：如GPT-4、Gemini、Llama-3、PaLM、Bloom、OPT等模型已达数百亿至数千亿参数且仍在增长。这使单卡GPU难以同时容纳模型参数与中间状态（如KV缓存与激活），即便最新高端GPU（H100，最高94GB HBM）在推理时也难以满足。传统思路是采用多GPU与模型并行，但其成本高、运维复杂（例如OPT-175B推理至少需5张H100，约$150,000）——在成本敏感的推理场景中不可取。

更关键的是，即便勉强“塞得下”，推理过程也会暴露两类核心瓶颈：一方面，庞大的参数与KV缓存不断挤压GPU显存，导致显存不足与频繁的数据搬运；另一方面，prefill与decode两个阶段在算力与带宽占比上差异极大，decode阶段往往成为延迟与吞吐的瓶颈。围绕这两个矛盾，研究者逐渐提出了两条互补路径：其一是内存卸载（Memory Offload），通过将权重、激活与KV Cache分层存放在GPU、CPU内存或CXL/SSD等多级介质中，缓解显存压力并提升可支持的上下文长度与批量；其二是计算卸载（Compute Offload），利用CPU新指令集（如AMX）与协同调度，把部分算子迁移到CPU端执行，从而削减GPU压力、改善尾延迟并提升整体吞吐。

二、内存卸载

在大模型推理中，显存不足始终是最直接的约束。模型参数、激活与KV缓存随着模型规模、上下文长度与对话轮数呈指数式增长，往往超过单卡GPU的物理容量。单纯依赖多GPU虽能缓解，但带来成本高昂、调度复杂、能耗过大的问题。因此，研究者提出了内存卸载思路，即将不同状态（参数、激活、KV）分层放置在GPU HBM、CPU DRAM、CXL大内存甚至SSD中，通过高效的数据通道与调度算法实现按需迁移。

典型工作如FlexGen将模型权重与KV分层放在GPU/CPU/SSD，结合低比特压缩和I/O调度，使得单卡也能运行超大模型；Mooncake、Pensieve等则进一步将KV缓存视作一等公民，利用CPU/SSD作为远端层，支持长上下文和多轮对话时的缓存复用与分层淘汰；FlashGen与FastServe在此基础上引入多级缓存与请求级调度，缓解历史放大与尾延迟问题。更前沿的研究如LIA则结合CXL扩展GPU外挂大容量内存，把参数/激活合理分布在不同介质上，使单卡GPU的最大batch规模显著提升。

总体来看，内存卸载的目标是解决“放哪儿”的问题：通过分层存储、压缩与复用，让有限的GPU显存不再成为瓶颈。其关键挑战在于跨设备带宽有限，如何降低传输开销、如何选择合适的分层策略，成为决定性能的核心。

三、计算卸载

除了存储，算力利用效率也是大模型推理

中的重要瓶颈。推理过程中的不同阶段与算子具有极不均衡的特性：prefill阶段算力密集，而decode阶段往往受限于带宽与访存；FFN层通常算力主导，而注意力打分层在长上下文下则更接近内存主导。这种异构特性启发了计算卸载思路，即让GPU与CPU协同执行，不同算子按算/存比（ops/byte）动态分流。

早期的卸载框架（如FlexGen）仅在CPU上执行最轻量的注意力打分层，但由于传统CPU吞吐远低于GPU，实际加速有限。随着Intel AMX等矩阵扩展的引入，CPU端算力获得数量级提升，使卸载更大算子成为可能。典型代表如LIA，通过分析不同batch/序列长度下各子层的算/存比，动态决定哪些次层由GPU执行、哪些迁移到CPU，从而在小批延迟优先和大批吞吐优先两种场景下均可获益。另一些工作则探索更细粒度的拆分：PowerInfer利用激活的幂律分布，将“热神经元”留在GPU，“冷神经元”迁至CPU；LLM–Mesh则在serverless多租环境下，按token级调度将合适的decode请求分配给CPU，以提升整体SLO。

因此，计算卸载要解决的是“谁来算”的问题：利用CPU–GPU协同和代价感知调度，动态平衡延迟、吞吐与能效。其难点在于跨设备调度与同步开销，如何避免频繁切换带来的额外传输，以及如何充分发挥CPU的矩阵指令优势。

四、未来发展趋势

未来大模型推理中的内存卸载与计算卸载将呈现出融合与智能化的发展趋势。内存卸载方面，CXL等新型互连将成为常态，GPU HBM与CPU DRAM、CXL大内存、SSD形成分层架构，KV缓存不再是临时副产物，而会被视作系统的一等公民，通过智能缓存、会话态复用与按需重建实现高效管理，解决长上下文与多轮对话的存储扩展问题。计算卸载方面，随着AMX等矩阵扩展在CPU上普及，卸载将走向更细粒度和动态化，不仅限于轻量子层，而是根据算/存比与负载特征灵活调度GPU与CPU，形成真正的协同算力池。两者的融合趋势则体现在统一调度：未来系统将以SLA为核心目标，综合首字延迟、逐字延迟与吞吐，自动决定“搬状态”还是“搬计算”。这种代价感知与策略自适应的框架，将让单卡甚至边缘设备也能以低成本承载大规模推理，推动大模型应用更广泛落地。

参考文献：

[1] Y. Sheng, C. Zhang, et al. FlexGen: High-throughput Generative Inference of Large Language Models with a Single GPU. Proceedings of the 40th International Conference on Machine Learning (ICML), 2023.

[2] Z. Kim, H. Kim, et al. LIA: A Single-GPU LLM Inference Acceleration with Cooperative AMX-Enabled CPU-GPU Computation and CXL Offloading. Proceedings of the 52nd International Symposium on Computer Architecture (ISCA), 2025.

[3] Y. Wang, J. Lee, et al. Mooncake: Decoupled Prefill and Decode for Efficient Long-Context LLM Serving. Proceedings of the 23rd USENIX Conference on File and Storage Technologies (FAST), 2025.

[4] H. Chen, L. Zhang, et al. Pensieve: Session-Aware KV Cache Management for Multi-turn LLM Serving. Proceedings of the 20th European Conference on Computer Systems (EuroSys), 2025.

[5] J. He, K. Li, et al. FastServe: Efficient Token-Level Preemption and State Offloading for LLM Inference. arXiv preprint arXiv:2402.12345, 2024.

[6] X. Liu, T. Zhang, et al. FlashGen: Multi-level Caching and Scheduling for Multi-round LLM Inference. Proceedings of the 30th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), 2025.

[7] W. Deng, H. Liu, et al. PowerInfer: Fast LLM Inference on Consumer GPUs through Power-Law Activation Offloading. Proceedings of the 29th ACM Symposium on Operating Systems Principles (SOSP), 2024.

[8] S. Park, J. Kim, et al. LLM-Mesh: Serverless Multi-Tenant Scheduling for Token-Level LLM Decoding. arXiv preprint arXiv:2503.06789, 2025.

第一届先进计算技术与系统论坛暨2025年实验室暑期年会顺利召开

燕　燕

2025年8月1日，第一届先进计算技术与系统论坛暨2025年实验室暑期年会在武汉召开，特邀嘉宾学者和毕业生代表及实验室400余名师生参加了本次年会。

实验室主任金海教授作了简要致辞，北京大学梁云教授作了题为“敏捷芯片设计前端工具”的报告，湖南大学唐卓教授作了题为“分布式调度驱动的超算互联网算力原生技术与实践”的报告，讯飞研究院王士进教授作了题为“分处理器芯片密码失效”的报告，清华大学吕勇强教授作了题为“处理器芯片密码失效”的报告，华为计算技术高级专家肖雄作了题为“计算技术未来关键挑战展望”的报告。五位专家的特邀报告内容新颖，理论与实践相结合，让在场的师生都受益匪浅。

接着实验室的毕业生代表从不同角度做了四场精彩的职场中所学与所用相结合的分享报告，分别是：2018届博士王新猴作了题为“一个普通师兄的“非成功学”职场分享”的报告，2014届硕士陈明作了题为“第四代大数据计算架构—通用增量计算”的报告，2007届硕士刘伟作了题为“从学术前沿到资管前沿–金融科技的需求、实践与挑战”的报告，2004届硕士周润松作了题为“第三方软件测试解读与工作心得分享”的报告。针对毕业生代表的各种无私分享，与会师生纷纷踊跃发言、主动提问，对其中的理论、方法和技术实践展开了热烈的探讨。

最后四个方向的学生代表报告了过去一年来的科研进展、成果情况和下一步工作计划。金海教授对相关工作给予了积极的指导，其他老师也针对各研发组的工作提出了宝贵的意见和建议。

燕　燕
负责事务：实验室宣传、科研项目管理等
Email：yany@hust.edu.cn

实验室举行2025级新生见面会

燕　燕

9月10日下午，实验室在东五楼210学术报告厅举行了2025级研究生新生见面会，实验室部分教师和100余名2025级博硕士新生参加了此次见面会，网安部分未能到现场的同学通过腾讯会议也全程线上参加。

实验室主任金海教授对新生的到来表示热烈的欢迎，介绍了实验室“先做人，后做事”的室训，希望同学们尽快适应在实验室的科研生活，鼓励大家多与老师交流沟通，多向师兄师姐请教分享，勉励新生们追求卓越、夯实基础，在科研道路上踏实奋进、刻苦钻研，为计算机领域的发展贡献青春力量。随后，余辰教授向新生们介绍了实验室概况、师资力量、研究方向和科研计划等情况。

余辰教授还向大家介绍了实验室各项规章制度、各阶段管理等情况，重点强调了实验室的学术道德规范、考勤管理等制度要求，详细介绍了实验室对论文投稿的要求及发表高水平论文的各种奖励。并以师长的身份对新生同学开启研究生的学习生活提出了建议，对同学们提出的相关问题一一作答。

青春不负韶华，学子勇往直前！祝愿实验室2025级研究生在新的开始，铸就新的辉煌！在新的起点上，勇往直前，展现无限的潜力。

燕　燕
负责事务：实验室宣传、科研项目管理等
Email：yany@hust.edu.cn

Maat: Analyzing and Optimizing Overcharge on Blockchain Storage

张浩杰　推荐

文章“Maat: Analyzing and Optimizing Overcharge on Blockchain Storage”是发表在2025年USENIX FAST上的一个工作，其主要聚焦于区块链存储过度收费问题。

区块链采用交易费用机制TFM向用户收取交易费来补偿网络中验证者的计算资源（包括存储、计算和带宽）消耗。在交易费的计算公式“交易费=Gas消耗量 × Gas price”中，Gas price为用户设定每单位Gas的支付价格，反映市场对区块链资源的供需关系（主观）；Gas cost是区块链系统对每类操作设定的固定资源成本（客观），Gas消耗量是交易执行过程中所有操作的Gas cost总和。理想的Gas cost应当贴合实际资源消耗，但本文的研究发现，现行实现中对存储资源的收费常常高于实际消耗。

现有致力于优化区块链交易费的工作中：EIP-4844设计非持久化blobs存储结构以降低临时数据成本；SuperStack优化智能合约代码以减少操作码执行量，但通用性有限；EIP-2929根据存储数据是否已缓存调整Gas cost，区分磁盘费用（首次访问，高Gas cost）和内存费用（重复访问，低Gas cost），但仅覆盖同一交易内的缓存场景，未涉及跨交易或跨区块的存储访问优化。

为减轻处理交易的磁盘I/O，以太坊在节点中实现了四种缓存机制：（1）CoW缓存以写时复制方式缓存当前区块的账户和变量；（2）SSAS缓存维护最近128个区块的账户和变量以减少跨块访问的磁盘I/O；（3）MPT缓存记录MPT节点以加速世界状态的查验与更新；（4）字节码缓存记录字节码哈希和字节码的映射以加速合约账户字节码性能。但本文对多个以太坊客户端（包括geth、erigon、nethermind、besu）的实现分析发现TFM的费用计算普遍存在缓存命中却被错误计算为磁盘费用的情况，具体有三种：

（1）块内缓存错误计费：在对同一区块内同一对象非首次访问，虽命中CoW缓存却仍被当成磁盘访问计费；

（2）跨块缓存错误计费：在对最近128个区块内同一对象非首次访问，虽命中SSAS缓存却仍被当成磁盘访问计费；

（3）重复合约部署：部署链上已存在的合约只需引用对应物理字节码副本，但仍收取部署全部字节码的磁盘费用。

Overcharging issues		Ethereum	BSC
Issue#1	Fraction of transactions	43.7%	63.1%
	Fraction of fees	25.6%	23.7%
Issue#2	Fraction of transactions	42.4%	62.0%
	Fraction of fees	12.9%	15.3%
Issue#3	Fraction of transactions	0.5%	0.3%
	Fraction of fees	3.5%	2.7%
Total	Fraction of transactions	70.4%	92.8%
	Fraction of fees	42.0%	41.7%

图1　以太坊和BSC中过度收费问题的影响

文章针对区块链的过度收费问题设计了Maat，将存储操作的工作负载与相应的Gas cost对齐。其遵循三个设计原则：共识（优化后各节点的Gas cost一致）、公平（Gas cost与实际工作负载一致）、效率（对性能、操作的负面影响低）。

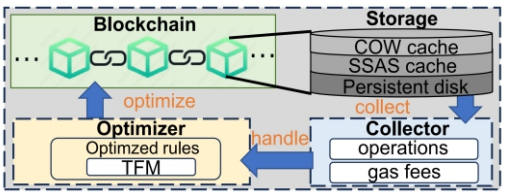


图2 Maat架构

Maat的结构包含三个实体：带有其存储的区块链客户端（执行交易时监控存储操作实时优化对应的Gas cost）、收集器（监控客户端的存储操作与对应的Gas cost）、优化器（根据四条优化规则处理存储操作和Gas cost）。该系统实施三项关键技术：

（1）细粒度数据收集：区别于opcode级的粗粒度数据收集，细粒度数据收集对存储操作的高级语义（如账户加载、字节码存储），按内存加载/存储和磁盘加载/存储四个维度划分Gas cost，兼顾细节与效率。

（2）共识导向优化：通过四条形式化规则（O1/O2针对块内缓存错误计费、O3针对跨块缓存错误计费、O4针对重复合约部署）解决三类过度收费问题，所有规则基于区块链确定性数据结构（区块编号、交易顺序、状态访问位置）触发，确保不同节点对规则是否适用和如何调整费用的判定完全一致。

（3）资源预分配：预分配缓存资源（如128个区块的访问数据，约230MiB内存），强制异构节点（不同硬件/客户端）维持相同缓存策略，确保内存访问和磁盘访问判定逻辑统一。

本文通过四个研究问题评估Maat：

RQ1：优化过度收费问题

在以太坊的100万个区块（2023年8月至2024年1月）范围内，Maat优化实现了32%的交易费降低，优化效果是基线方案（EIP-2929）的近三倍。在不同交易类型中，合约调用和创建交易的优化率分别达39.55%和34.7%，即使是存储操作较少的以太币转账也实现12.43%优化。性能开销方面，仅增加1.4%的区块处理时间和5.6%的内存消耗，验证了其高效性与实用性。

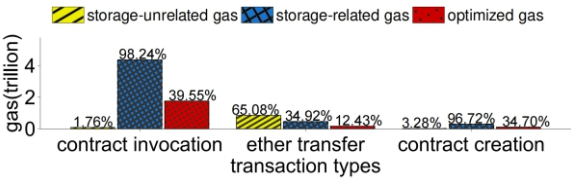
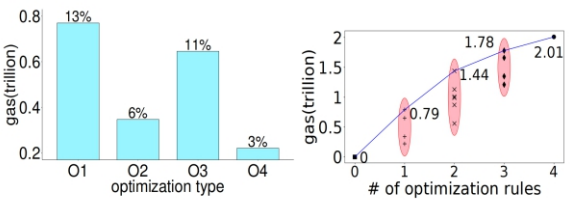


图3 三种类型交易的优化效果

RQ2：不同优化规则的影响

所有优化规则及其组合在缓解过度收费问题上均表现出有效性。O1/O2（块内重复读写）是优化主力，优化率合计24%（O1占13%，O2 占6%），解决块内高频存储访问重复收费；O3（128区块缓存复用）优化率11%，针对跨块缓存复用场景；O4（重复合约部署）优化率3%，但因合约部署费用高，仍节省3.5%费用。同时启用四条规则时总优化率33%（费用优化32%）。

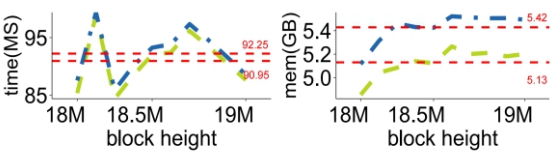


(a) four optimization rules (b) 16 combinations

图4 不同优化规则的优化后Gas cost

RQ3：性能开销

在时间开销上，区块处理时间仅增加1.4%；在空间开销上，内存消耗仅增加5.6%。Maat的低性能开销确保在优化交易费用的同时，不会影响区块链节点的正常运行效率。



(a) time overhead (b) space overhead

图5 Maat的时间和空间开销

Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention

董雨康　推荐

“Native Sparse Attention: Hardware-Aligned and Natively Trainable Sparse Attention”是自然语言处理领域的国际顶级会议ACL 2025录用的一篇文章，该会议于2025年7月在奥地利维也纳举行。此篇论文获得了Best paper，作者团队来自 DeepSeek-AI、北京大学和华盛顿大学等。

论文针对大语言模型在长上下文处理中的效率和能耗挑战，提出了一种新的注意力机制NSA。随着模型规模不断增大和上下文窗口持续延伸，标准的全量注意力机制在计算和存储上的二次方复杂度成为主要瓶颈。在 64k token 长度的推理中，注意力计算占总延迟的 70%–80%，严重制约了大模型的性能与能效。虽然已有研究尝试通过稀疏化来降低复杂度，但大多方法只在推理阶段生效，无法在训练中发挥作用，导致模型精度下降和能耗难以降低。因此，如何在保证性能的同时实现真正可训练、硬件高效的稀疏注意力，成为亟待解决的问题。

NSA 的核心思路是构建一种分层稀疏框架，如图1所示，通过压缩、选择和滑动窗口三种机制来同时捕捉全局语义和局部依赖。压缩机制将连续的 token 聚合为块级表示，在保持整体语义的同时显著减少计算量；选择机制基于块的重要性分数挑选出最关键的部分，确保模型保留核心细节；滑动窗口机制保留最近的一段 token，从而保证短程依赖和局部上下文信息不被稀疏化忽略。这三种机制通过门控融合为最终输出提供支持，使得 NSA 既能覆盖全局，又能兼顾局部。与许多方法逐 token 的稀疏化不同，NSA 强调块式处理，更符合现代 GPU 的访存模式和 Tensor Core 的计算特性，有助于实现高效的硬件利用率。

接上页

RQ4：可扩展性

Maat的可扩展性得益于区块链对以太坊代码仓库的复用，其核心验证结果如下：在BSC的100万区块（含1.68亿笔交易）中成功迁移并部署，实现31%的交易费降低（每周节省154万美元）；进一步扩展至50个复用以太坊TFM的区块链（如Polygon、Optimism），无需额外代码适配即可直接应用，优化规则在异构节点环境中保持一致，验证了其对共享存储架构区块链的普适性。

本文识别区块链存储中块内连续访问、跨块缓存复用及重复合约部署三类过度收费问题，提出Maat方案实现费用与存储负载的动态匹配。通过细粒度数据收集、共识导向优化规则及资源预分配技术，在以太坊及50+区块链的部署验证，大幅降低交易费用（以太坊32%、BSC31%），提升了区块链系统的经济性与可扩展性。该工作已促成相关社区提案EIP-7863。

张浩杰
2025级硕士研究生
研究方向：区块链技术
Email：hodge@hust.edu.cn

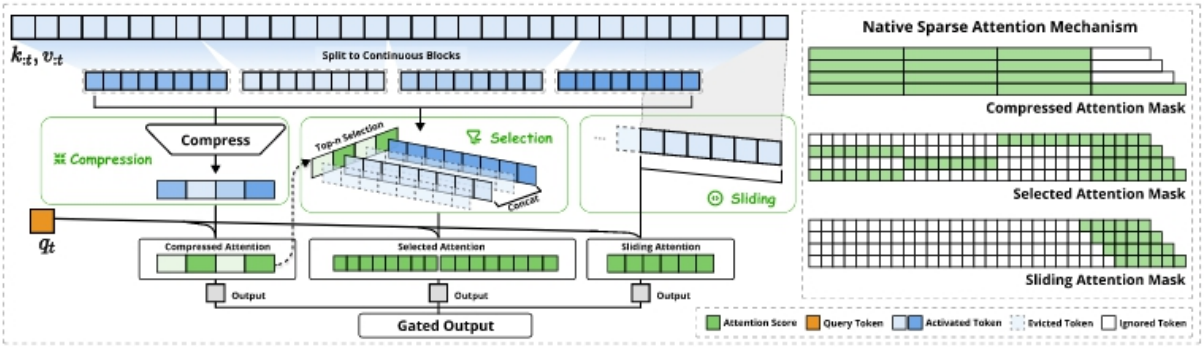


图1 架构图（Query 输入经过三条分支：Compression、Selection、Sliding Window，最终在 Gate 模块处融合输出。绿色区域代表被计算的部分，白色区域代表被跳过的部分）

在实现上，NSA 设计了硬件友好的内核，基于 Triton 框架优化稀疏注意力的计算。它采用组内查询加载和共享 KV 缓存的方式来减少冗余访问，并通过精细的调度策略平衡算强度与访存压力。在训练和预填充阶段，NSA 通过提升算强度发挥 GPU 的计算潜力；在自回归解码阶段，它通过减少 KV 加载降低带宽瓶颈，从而在不同计算阶段都能显著提高效率。实验结果表明，NSA 在 A100 GPU 上相比全量注意力，在 64k token 的序列下前向计算最高加速 9 倍，反向传播加速 6 倍，解码阶段则达到 11.6 倍的加速。随着序列长度的增长，这些性能优势更加突出，同时能耗也显著降低。

在模型能力方面，作者在 27B 参数的 Transformer 模型上进行了大规模预训练和评估。结果显示，NSA 在通用基准如 MMLU、GSM8K 和 DROP 上整体表现不逊于全量注意力，甚至在多数任务中略有提升。这说明，虽然注意力计算被大幅稀疏化，模型仍然保持甚至增强了泛化与推理能力。在长上下文测试中，NSA 在 64k Needle-in-a-Haystack 检索任务上实现了 100% 的准确率，在 LongBench 上的平均分比全量注意力提升 0.032，比最佳稀疏方法提升 0.046，特别是在多跳问答和代码理解等复杂任务中表现最为突出。进一步在链式推理任务上的实验也验证了 NSA 的优势，在 AIME24 数学推理基准中，稀疏版本 NSA-R 明显优于全量注意力基线，且在 16k token 的长推理链条件下依然保持较高准确率，说明其在复杂逻辑推理中具有稳定性。

总体来看，NSA 的贡献体现在三个方面。第一，它首次提出了可原生训练的稀疏注意力机制，突破了过去方法只能在推理阶段使用的局限，使得模型在训练和推理的全生命周期都能受益。第二，它通过分层稀疏策略和硬件对齐优化，在性能和能耗上取得了兼顾，既降低了计算复杂度，又在实际 GPU 上实现了接近理论值的加速。第三，它在长上下文和复杂推理等任务上展现出优于全量注意力的性能，为未来大模型在超长上下文条件下的高效训练与推理提供了切实可行的路径。NSA 在理论与实践层面都对稀疏注意力的发展做出了重要贡献，展示了降低复杂度与提升性能并行不悖的可能性，为后续研究开辟了新的方向。

董雨康
2024级博士研究生
研究方向：体系结构与系统软件、图计算、稀疏计算
Email：ykdong@hust.edu.cn

科研是一场慢跑

宋熙然

硕士研究生的三年时光已然定格与封装，翻阅记忆的纸张，字里行间写着的是实验室里的平凡日常、喜悦瞬间、和辛苦过往。回望过去，人们总是能够清楚地看到自身的改变，细数有收获的经历，记住遗憾的故事。而眺望未来的时候，我们却经常迷茫于自己将要成为什么样的人。此刻，我站在硕士生涯的终点，得以回望这场三年的科研生活慢跑，于是分享一些浅薄的经历和简单的故事，希望能够给大家对于未来的思考提供一些参照。

放下纠结，敢于出发

刚开始进行科研工作时，我们可能在各种地方感到困惑和纠结，包括如何阅读总结论文、如何做好PPT、如何代码实现、如何设计进行实验、如何写论文等等。我们内心想要做出完美的东西，害怕做得不地道、不专业，于是常常陷入长时间的犹豫、等待、和拖延，这样的纠结将大大降低我们向前推进的速度，不利于出成果。

实际上，科研工作刚刚起步出发的时候，我觉得不必告诉自己要做得十分完美，重要的是放下纠结，敢于出发，记住科研是一场长时间的慢跑，而不是一次性的考试。我认为应该敢于快速做出一个粗糙的半成品，不怕别人嫌丑，不断寻求导师和同学的反馈，快速迭代改进，反复打磨，这样直至最终收敛到接近完美的状态。

明确目标，扎实前行

在科研的道路上，我们发现前行的岔路纷繁复杂，其中可能有很多坑，如果思路不明确或者考虑不够全面的话就容易踩进坑里，例如，论文的逻辑大有问题，或者做一些无意义的实验。这时便需要我们在科研慢跑的途中保持清醒，明确目标，扎实前进。

为了做到这些，我想一是需要完全理清楚论文思路的每一个环节，了解了现有工作之后，需要想清楚自己工作的明确目标，明确每一步的逻辑，多和老师同学讨论。二是需要充分调研前人的结论，而不是匆忙地开始写代码、跑实验。对相关方向的已有论文进行充分的调研，总结出一些实验结论，这样能够避免自己踩到别人可能已经踩过的坑里。

调节心态，保养身体

日复一日的科研工作常常让人身心疲惫，在这场慢跑中，维护良好的心态和健康的身体非常重要。对于心态调整，我推荐直接搜索一些相关的帮助视频或文章，例如TED演讲等，帮助自己缓解焦虑，摆脱拖延。我曾认为这些视频和文章只是空洞的鸡汤，但实际上真的能够改进自己的认知。

对于身体健康方面，我的作息还算规律，偶尔会到操场上长跑锻炼。饮食方面，我建议大家不要过多地喝咖啡和茶，它们会提取和消耗人的内力。我自己经常磨咖啡、冲咖啡，这使得我亲身体验了咖啡喝多了会感到精力被耗尽。

三年时光，转瞬即逝，实在需要好好珍惜！以上是我浅薄的经历和简单的故事，希望能够帮到大家！

宋熙然

2024届硕士毕业生

研究方向：数据挖掘，图表示学习

Email：xiransong@outlook.com