



中文核心期刊 中国科技核心期刊 中国核心学术期刊
第三届国家期刊奖百种重点期刊 信息通信领域产学研合作特色期刊

ISSN 1009-6868
CN 34-1228/TN

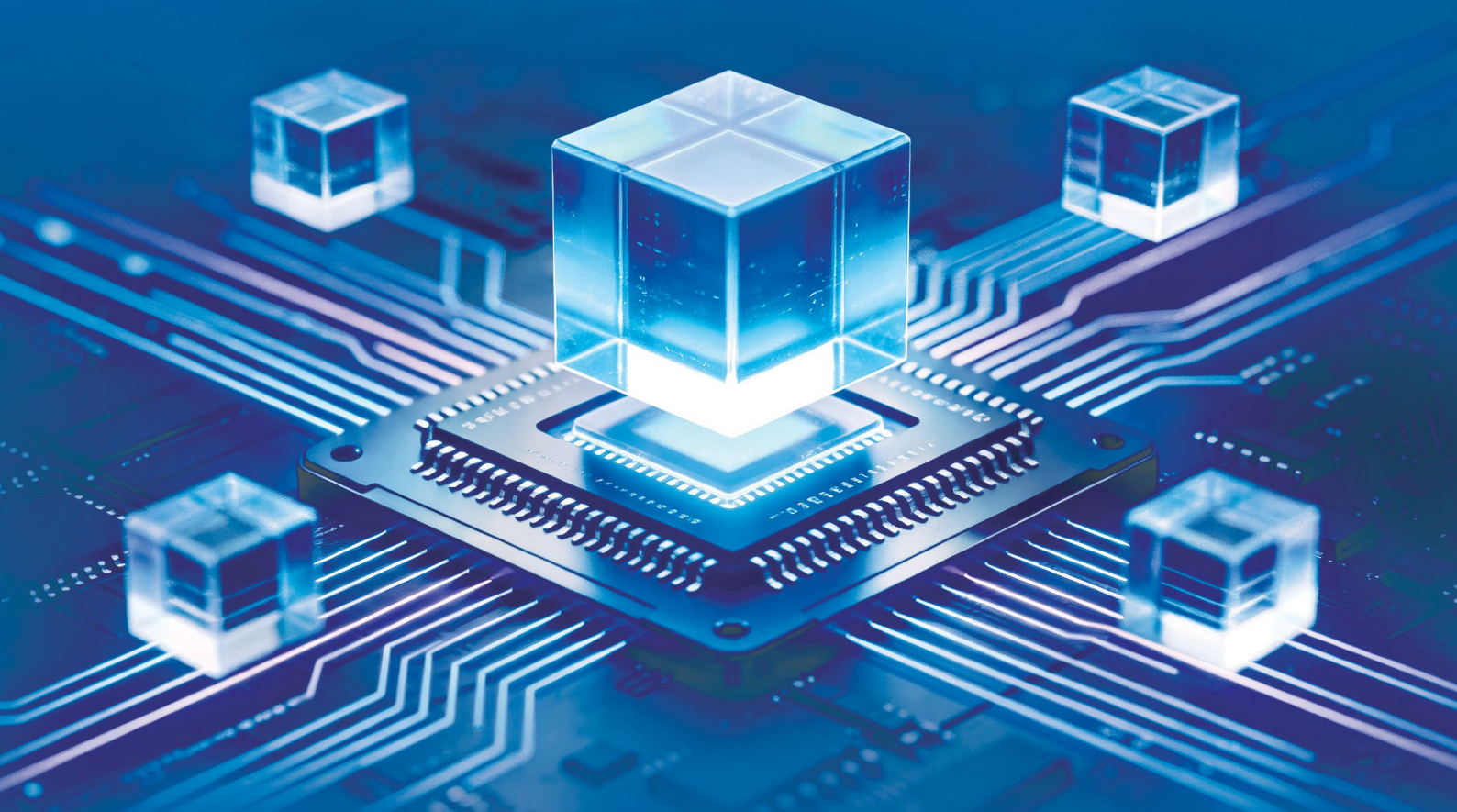
中兴通讯技术

ZTE TECHNOLOGY JOURNAL

<http://tech.zte.com.cn>

第 32 卷 · 总第 190 期 · 2026 年 8 月 · 第 4 期

专题：大模型推理中的存算技术



ISSN 1009-6868



《中兴通讯技术》第10届编辑委员会

顾问 侯为贵(中兴通讯股份有限公司创始人) 钟义信(北京邮电大学教授)
糜正琨(南京邮电大学教授) 李自学(中兴通讯股份有限公司前董事长)

主任 陆建华(中国科学院院士)

副主任 方 榕(中兴通讯股份有限公司董事长) 李建东(西安电子科技大学教授)

编 委

敖 立	中国信息通信研究院副院长	唐万斌	电子科技大学教授
陈建平	上海交通大学教授	唐雄燕	中国联通研究院副院长、首席科学家
陈前斌	重庆邮电大学教授、副校长	陶小峰	北京邮电大学教授
段晓东	中国移动研究院副院长	汪烈军	新疆大学教授、副校长
方 榕	中兴通讯股份有限公司董事长	王 翔	中兴通讯股份有限公司高级副总裁
高新波	西安电子科技大学教授、校长	王文博	雄安空天信息研究院院长
葛建华	西安电子科技大学教授	王文东	北京邮电大学教授
管海兵	上海交通大学教授、副校长	王喜瑜	中兴通讯股份有限公司执行副总裁
郭 庆	哈尔滨工业大学教授	王耀南	中国工程院院士、湖南大学教授
洪 伟	中国科学院院士、东南大学教授	王志勤	中国信息通信研究院副院长
江 涛	华中科技大学教授	卫 国	中国科学技术大学教授
蒋林涛	中国信息通信研究院科技委主任	邬贺铨	中国工程院院士
金 石	东南大学教授、副校长	吴春明	浙江大学教授
李尔平	中国工程院外籍院士、浙江大学教授	向际鹰	中兴通讯股份有限公司首席科学家
李红滨	北京大学教授	肖 甫	南京邮电大学教授、副校长
李厚强	中国科学技术大学教授	解冲锋	中国电信新一代信息通信专业首席专家
李建东	西安电子科技大学教授	徐安士	北京大学教授
李乐民	中国工程院院士、电子科技大学教授	徐子阳	中兴通讯股份有限公司总裁
李融林	华南理工大学教授	续合元	中国信息通信研究院首席专家
林晓东	中兴通讯股份有限公司副总裁	薛向阳	复旦大学教授
刘 健	中兴通讯股份有限公司高级副总裁	杨义先	北京邮电大学教授
刘建伟	北京航空航天大学教授	易芝玲	中国移动研究院首席科学家
隆克平	北京科技大学教授	张 杰	内蒙古工业大学教授、学术副校长
卢光跃	西安邮电大学教授、校长	张 平	中国工程院院士、北京邮电大学教授
陆建华	中国科学院院士、清华大学教授	张 卫	复旦大学教授
马建国	中原工学院教授、学术副校长	张宏科	中国工程院院士、北京交通大学教授
毛军发	中国科学院院士、深圳大学教授	张钦宇	哈尔滨工业大学(深圳)教授、副校长
孟洛明	北京邮电大学教授	张云勇	中国联通网络信息安全部总经理
尼玛扎西	中国工程院院士、西藏大学教授	赵慧玲	工业和信息化部信息通信科技委常委
石光明	鹏城实验室副主任	郑纬民	中国工程院院士、清华大学教授
史振威	内蒙古大学教授	钟章队	北京交通大学教授
苏 森	重庆邮电大学教授、校长	周 亮	南京邮电大学教授、副校长
孙知信	南京邮电大学教授	朱近康	中国科学技术大学教授
谈振辉	北京交通大学教授	祝宁华	中国科学院院士、南开大学教授
唐 宏	中国电信 IP 专业首席专家		

目次

中兴通讯技术 (ZHONGXING TONGXUN JISHU)
第 32 卷 总第 190 期 2026 年 8 月 第 4 期

中文核心期刊 中国科技核心期刊 第三届国家期刊奖百种重点期刊 信息通信领域产学研合作特色期刊 中国知网、万方数据、重庆维普等数据库收录期刊 1995 年创刊

热点专题 ▶

大模型推理中的存算技术

01 专题导读 郑纬民, 陆游游

03 BidKV: 显存受限推理的KV缓存回收机制 陈彦博, 王明琪, 张书豪

09 存储层级感知的大模型前缀键值缓存系统 何水兵, 瞿皓阳, 党政

17 面向智能体时代大语言模型推理的新型存算解聚合架构 熊先奎, 王程晨, 蔡文豪

23 新架构大模型推理的存算特征演变与近存计算适配技术研究 庄凤云, 孙菲, 钱岭

32 面向大模型推理的存算协同理论综述 汪俊峰, 刘璇, 李亚奇, 石亮

41 大规模向量存储系统研究综述 郭昊, 谢旻晖, 陆游游

47 端侧具身智能系统: 挑战与展望 徐梦炜, 徐大亮

56 城域光网络演进与fgOTN技术的应用 王光全, 董嫻

63 存算一体——重塑芯片架构新势力 刘新阳, 李婷宇, 巨鸣

72 网络运行安全完整性等级确定方法研究 路晔绵, 郑梦青, 吴荣春, 张治兵, 王晓颖

79 AM-WiGeR: 一种基于注意力机制增强残差网络的Wi-Fi手势识别方法 李业深, 姚可欣, 熊轲

88 B5G 高铁网络演进方案研究 孙普

08 中兴通讯技术杂志社成功举办第31次编委会议暨2026年通信热点技术研讨会

名家视点 ▶

企业视界 ▶

技术广角 ▶

综合信息 ▶

《中兴通讯技术》2026 年热点专题名称及策划人

1. 6G 关键技术的标准化: Day-1 与未来 中国移动研究院首席科学家 易芝玲	3. 智算网络 工信部信息通信科技委常委 赵慧玲	5. 星地太赫兹高速传输技术 中国科学院院士、东南大学教授 洪伟 电子科技大学教授 唐万斌 东南大学教授 郝张成
2. 广域立体覆盖低空通信技术 东南大学副校长 金石 东南大学教授 刘凡	4. 大模型推理中的存算技术 中国工程院院士、清华大学教授 郑纬民 清华大学副教授 陆游游	6. 智能多天线技术 北京交通大学副校长 艾渤 北京交通大学教授 章嘉懿

MAIN CONTENTS

ZTE TECHNOLOGY JOURNAL
Vol. 32 No. 4 Aug. 2026

Special Topic ►	Compute–Storage Technologies for Large Model Inference
	01 Editorial Zheng Weimin, Lu Youyou
	03 BidKV: KV Cache Reclamation for Memory–Constrained Inference Chen Yanbo, Wang Mingqi, Zhang Shuhao
	09 Storage–Hierarchy–Aware Prefix KV Cache System for Large Language Models He Shuibing, Qu Haoyang, Dang Zheng
	17 A Novel Storage–Compute Disaggregated Architecture for Large Language Model Inference in the Agent Era Xiong Xiankui, Wang Chengchen, Cai Wenhao
	23 Evolution of Memory–Compute Characteristics in New–Architecture LLM Inference and Near– Memory Computing Adaptation Technologies Zhuang Fengyun, Sun Fei, Qian Ling
	32 A Survey on Compute–Storage Co–Optimization Theory for Large Model Inference Wang Junfeng, Liu Xuan, Li Yaqi, Shi Liang
	41 Survey on Large–Scale Vector Storage Systems Guo Hao, Xie Minhui, Lu Youyou
	47 On–Device Embodied Intelligence System: Challenges and Opportunities Xu Mengwei, Xu Daliang
Expert View ►	56 Evolution of Metro Optical Networks and Application of fgOTN Technology Wang Guangquan, Dong Shan
Enterprise View ►	63 CIM/PIM/PNM: An Emerging Force Reshaping Chip Architecture Liu Xinyang, Li Tingyu, Ju Ming
Research Papers ►	72 Research on Method for Determination of Network Operation Safety Integrity Level Lu Yemian, Zheng Mengqing, Wu Rongchun, Zhang Zhibing, Wang Xiaoying
	79 AM–WiGeR: A Wi–Fi Gesture Recognition Method Based on Attention Mechanism Enhanced Residual Network Li Yeshe, Yao Kexin, Xiong Ke
	88 Research on Evolution of B5G High–Speed Railway Networks Sun Pu

期刊基本参数: CN 34–1228/TN*1995*b*16*94*zh*P*¥20.00*6500*13*2026–08

敬告读者	本刊享有所有发表文章的版权, 包括英文版、电子版、网络版和优先数字出版版权, 所支付的稿酬已经包含上述各版本的费用。 未经本刊许可, 不得以任何形式全文转载本刊内容; 如部分引用本刊内容, 须注明该内容出自本刊。
------	---

大模型推理中的存算技术 专题导读



专题策划人



郑纬民



陆游游

随着大模型的落地应用，大模型推理服务的部署规模持续扩大，对人工智能基础设施提出了更高的要求。智能体的多轮迭代、长程任务执行等特点，推动上下文长度进一步增加，这对系统的计算与存储能力提出了新的挑战。本期专题以“大模型推理中的存算技术”为主题，聚焦大模型推理的存算技术，邀请该领域的专家学者撰写了7篇文章。这些文章从键值（KV）Cache的复用与存储管理、近存计算架构与存算协同、大规模向量管理以及大模型推理在端侧具身场景下的应用等角度，对相关技术进行了深入讨论。本专题旨在梳理大模型推理场景中的存算技术研究进展，促进中国大模型推理系统方向的学术交流和思想碰撞，助力相关技术的创新与工程落地。

《BidKV：显存受限推理的KV缓存回收机制》提出了一种代价感知的显存KV回收机制BidKV，通过对可释放容量与抢占扰动进行效用信号编码，选择回收单位代价收益高的对象。该机制以轻量适配器接入了虚拟大型语言模型（vLLM）和SGLang主流框架，并在英伟达和昇腾等加速卡上验证了效果。

《存储层级感知的大模型前缀键值缓存系统》提出了一种面向显存—内存—固态硬盘（SSD）多层存储架构的大模型前缀键值存储系统PCache，介绍了访存代价感知的重要KV选择、页面级连续KV加载、动态KV重排和变序计算等技术，并通过实验验证了其能够在保证推理精度稳定的同时降低首Token时延。

《面向智能体时代大语言模型推理的新型存算解聚合架构》提出了一种基于光互联的以内存为中心的新型存算解聚合架构，以实现计算和存储资源的独立伸缩、池化共享和按需协同。该架构通过光互联技术连接计算侧和存储侧资源，利用了光互联低延迟、高带宽和可扩展能力，介绍了存储侧多层级管理和近存计算的技术，并通过仿真验证了架构的有效性。

《新架构大模型推理的存算特征演变与近存计算适配技术研究》分析了大模型架构中多头潜在注意力（MLA）和混合专家（MoE）等机制对硬件存算的影响。文章借助Roofline模型分别对MLA和MoE机制在算术与访存强度上进行了分析，讨论了已有近存计算的不足，并展望了MLA与MoE组合架构下的分层近存计算框架。

《面向大模型推理的存算协同理论综述》对大模型推理中的计算与存储之间的协同关系进行了综述，从以算换存、

以存换算和KV存储路径这3个角度展开，具体包括输入/输出（I/O）重组、压缩、裁剪与重算等以算换存技术，KV布局重用和推测解码等以存换算技术，以及多路径选择等KV存储路径技术，并展望了未来的存算协同技术。

《大规模向量存储系统研究综述》对大规模向量存储技术进行了综述，概述了内存图索引与聚类索引机制，分析了大规模向量场景下向量化压缩、固态硬盘上索引、异构存算硬件加速等技术，讨论了向量检索与大模型检索增强的协同技术，总结并展望了大规模向量存储技术的未来发展。

《端侧具身智能系统：挑战与展望》对端侧具身智能的算法、系统和芯片优化进行了综述，介绍了端侧智能在步骤压缩、权重与激活精度量化、扩散特征缓存、分块异步等通用算法方面的优化，以及视觉—语言—动作模型（VLA）和世界—动作模型（WAM）两种模型的特定算法优化，对比了当前端侧具身推理系统的优劣，并展望了端侧智能系统的未来发展。

本期作者来自知名高校和企业，这些作者或参与国家科技重大专项、国家重点研发计划，或参与国家实验室、国家人工智能学院的智算集群相关工作，或负责企业前沿技术与系统的研发工作，均具有一线研发经验，从大模型系统的存

储管理、存算系统、向量存储和端侧具身智能等方面介绍了最新的研究成果。希望本期专题能为读者提供技术参考，也感谢所有作者和审稿专家对本期专题的支持！

策 划 人 简 介

郑伟民，清华大学计算机系教授、博士生导师，中国工程院院士，曾任中国计算机学会第十届理事长；长期从事并行/分布处理、大规模数据存储系统领域的科研与教学工作；曾获国家科技进步奖一等奖1次、国家科技进步奖二等奖2次、国家发明奖二等奖1次，2016年获何梁何利基金科学与技术进步奖，2016年获ACM戈登·贝尔奖，长期讲授计算机系统结构课程，2008年该课程被评为国家级精品课程；发表论文530余篇，拥有著作10部。

陆游游，清华大学计算机系长聘副教授、博士生导师，清华大学（计算机系）—软通计算机人工智能系统联合研究中心主任，青年北京学者，FAST、SOSP、OSDI、USENIX ATC、EuroSys等国际会议组织委员会或程序委员会委员；主要研究方向为存储与人工智能系统；曾获国家优秀青年科学基金项目资助，作为项目负责人主持了新一代人工智能国家科技重大专项项目、国家重点研发计划项目、国家自然科学基金重点项目等；6次获得最佳国际论文奖项，参与研制的高性能文件系统SuperFS部署于鹏城云脑II，2023年—2025年该系统多次蝉联世界超算存储IO500榜单第一名；发表论文80余篇。

BidKV: 显存受限推理的KV缓存回收机制



BidKV: KV Cache Reclamation for Memory-Constrained Inference

陈彦博/Chen Yanbo, 王明琪/Wang Mingqi,
张书豪/Zhang Shuhao

(华中科技大学, 中国 武汉 430074)
(Huazhong University of Science and Technology, Wuhan 430074, China)

DOI: 10.12142/ZTETJ.202604002

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260831.0932.002>

网络出版日期: 2026-08-31

收稿日期: 2026-06-01

摘要: 在线大模型推理中, 键值 (KV) 缓存是制约并发服务的关键显存资源。在显存压力下, 顺序回收或容量贪心回收易抢占高重算代价请求, 导致释放的显存空间被重复预填充占用, 从而显著拉长首Token延迟 (TTFT)。为此, 提出BidKV, 将KV回收建模为存算权衡问题, 依据可释放容量与扰动代价选择单位代价收益最高的请求进行回收。BidKV不改变推理结果, 并复用虚拟大语言模型 (vLLM) 与SGLang的原生抢占路径。ShareGPT实验结果表明, BidKV将vLLM的P95的TTFT由6 769 ms降至631 ms, 300 ms服务等级目标 (SLO) 达成率由68.4%提升至83.2%; 在SGLang上, P95的TTFT较原生最近最少使用 (LRU) 降低94.6%, SLO达成率提升38.8个百分点。在双卡昇腾910B2 (TP=2) 实验中, BidKV在3.8请求/秒负载下实现了631.8 ms的TTFT P95与72.0%的SLO达成率。上述结果验证了BidKV的有效性、跨框架可移植性与多卡适用性, 表明显存受限推理应从容量驱动转向存算协同调度。

关键词: 大语言模型推理; 大语言模型服务; KV缓存; 请求调度; 受害请求选择

Abstract: In online large language model (LLM) inference, the key-value (KV) cache is a critical memory resource that constrains concurrent serving capacity. Under memory pressure, sequential or capacity-greedy eviction policies tend to preempt requests with high recomputation costs, causing the freed memory to be repeatedly occupied by redundant prefill operations and consequently amplifying time-to-first-token (TTFT) latency. To address this issue, BidKV is proposed, which formulates KV cache eviction as a memory-computation trade-off problem. BidKV selects the request with the highest unit-cost benefit for eviction based on its releasable capacity and the corresponding perturbation cost. BidKV does not alter the inference results and reuses the native preemption paths of virtual large language model (vLLM) and SGLang. Experimental results on the ShareGPT dataset demonstrate that BidKV reduces the P95 TTFT latency of vLLM from 6 769 ms to 631 ms and improves the 300 ms service-level objective (SLO) attainment rate from 68.4% to 83.2%. On SGLang, BidKV reduces the P95 TTFT latency by 94.6% compared to the native least-recently-used (LRU) policy and improves the SLO attainment rate by 38.8 percentage points. In a two-card Ascend 910B2 (TP=2) setup, BidKV achieves a P95 TTFT of 631.8 ms and an SLO attainment rate of 72.0% under a workload of 3.8 requests per second. These results validate the effectiveness, cross-framework portability, and multi-GPU applicability of BidKV, demonstrating that memory-constrained inference should shift from capacity-driven to memory-computation co-scheduled strategies.

Keywords: LLM inference; LLM serving; KV cache; request scheduling; victim selection

引用格式: 陈彦博, 王明琪, 张书豪. BidKV: 显存受限推理的KV缓存回收机制 [J]. 中兴通讯技术, 2026, 32(4): 3-8. DOI: 10.12142/ZTETJ.202604002

Citation: Chen Y B, Wang M Q, Zhang S H. BidKV: KV cache reclamation for memory-constrained inference [J]. ZTE technology journal, 2026, 32(4): 3-8. DOI: 10.12142/ZTETJ.202604002

大语言模型 (LLM) 的在线推理需同时满足高吞吐与低交互时延的双重目标。在实际服务中, 请求在提示长度、生成长度及到达时间上存在显著差异, 且在预填充与解码阶段异步推进。随着并发度与上下文长度的增加, 用于保

存历史词元键值向量的键值 (KV) 缓存逐渐成为主要的运行时显存资源^[1]。以Llama-3.1-8B-Instruct为例, 在bf16精度下每个词元的KV缓存空间约为128 KiB, 一个包含4 096个词元的请求将占用约512 MiB显存。在多请求并发场景下, 可用KV容量极易达到上限。

当KV缓存空间不足时, 推理引擎通常会抢占部分正在

基金项目: 新一代人工智能国家科技重大专项 (2025ZD0123804)

运行的请求，释放其占用的KV缓存，待后续通过重计算路径恢复执行。该机制虽不改变最终输出结果，但会丢弃已完成的中间计算。若被抢占的请求接近生成完成，其恢复过程将消耗大量预填充资源，并重新扩张KV缓存，甚至可能进一步触发新一轮的抢占。由此，释放出的显存并不一定真正用于服务等待中的请求，反而可能被重复的预填充操作所消耗，导致首Token延迟（TTFT）升高，服务等级目标（SLO）达成率下降。

因此，KV回收的关键问题在于选择回收哪一个运行请求，而非是否释放显存。对于请求，其可释放的KV容量记为 r_i ，回收代价记为 C_i 。当系统存在 N 个词元的KV空间缺口时，理想的调度目标是在释放总量不低于 N 前提下，使被回收请求集合的总代价尽可能低。这一目标体现了显存受限推理中的存算权衡：单纯追求释放容量可能抢占高重算代价的请求，而仅关注请求顺序则可能遗漏高收益、低代价的回收对象。

现有工作尚未显式求解跨请求的回收选择问题。虚拟大语言模型（vLLM）主要采用后进先出（LIFO）抢占策略^[2]；Orca关注连续批处理优化^[3]；Sarathi-Serve与DistServe分别针对分块预填充与阶段解耦进行改进^[4-5]，均未对不同运行请求的回收代价加以区分。H2O与StreamingLLM则聚焦于请求内部的KV缓存压缩^[6-7]，缺乏跨请求的全局选择机制。

本文提出代价感知KV回收机制BidKV，将可释放容量与抢占扰动编码为效用信号，在活跃请求间选择单位代价收益最高的对象进行回收。主要贡献包括：构建竞价接口与在线效用排序方法；以轻量适配器形式接入vLLM与SGLang^[8]的原生抢占路径；基于真实对话负载验证了BidKV在TTFT与SLO指标上的改善效果。

1 竞价驱动的效用调度框架

1.1 总体架构

BidKV包含竞价表示、竞价生成、效用排序和运行时适配4个模块，分别负责统一接口、估计回收收益与代价、比较候选请求并执行抢占操作。该分层设计将代价估计与对象选择解耦，便于替换不同的估计器，并支持迁移至不同的推理引擎。

图1展示了BidKV效用调度的总体流程。系统包含两个异步过程：后台排序与压力触发回收。后台排序周期性收集运行请求的KV占用、完成进度及历史抢占状态，计算各请求的回收效用，并将优先级写入缓存表。压力触发回收在每个调度周期检测KV利用率；当利用率达到设定阈值时，运

行时适配器根据缓存优先级调整运行队列，使最高效用请求进入框架原生抢占路径；当KV利用率低于阈值时，系统不干预框架的默认调度行为。

1.2 基于最短作业优先的准入优化

BidKV在准入侧采用最短作业优先（SJF）策略，按提示长度由短到长对等待队列进行重排。提示长度在请求到达时即可获得，以此低成本近似预填充开销，减少短请求被长请求阻塞的情况，从而改善TTFT及SLO达成率。

SJF决定等待请求“谁先进入”，效用排序决定KV压力下“谁先退出”：前者缓解长请求阻塞，后者减少低效抢占，二者共同优化准入响应与重算代价。

1.3 竞价信号与扰动估计

每个运行请求对应一个竞价对象 b 。竞价对象包含请求标识、被抢占后可释放的KV词元数 r 、扰动估计 δ ，以及可选的置信度与策略元数据。其中， r 表示回收收益， δ 表示中断该请求可能引入的系统扰动，包括已完成计算的丢弃、请求重新排队及重复抢占等代价。扰动估计越小，说明回收该请求对系统造成的影响越低。

当前实现使用请求的生命周期特征对扰动代价进行估计：

$$\delta = 1 + w_c c + w_p P \quad (1),$$

其中， c 表示相对于请求配置最大生成长度的归一化解码进度， P 表示此前被抢占次数， w_c 与 w_p 分别为完成度权重与重复抢占惩罚权重，默认取值为0.5与0.3。在线阶段无法预知实际输出长度，因此 c 并非真实的完成比例，而是已投入解码计算量的低成本代理。BidKV仅将其作为软惩罚项，并结合可释放KV容量与历史抢占状态进行相对排序；完成度项用于降低对高进度请求的回收倾向，重复抢占项则有助于减少饥饿风险。第3.5节将进一步检验权重的敏感性。

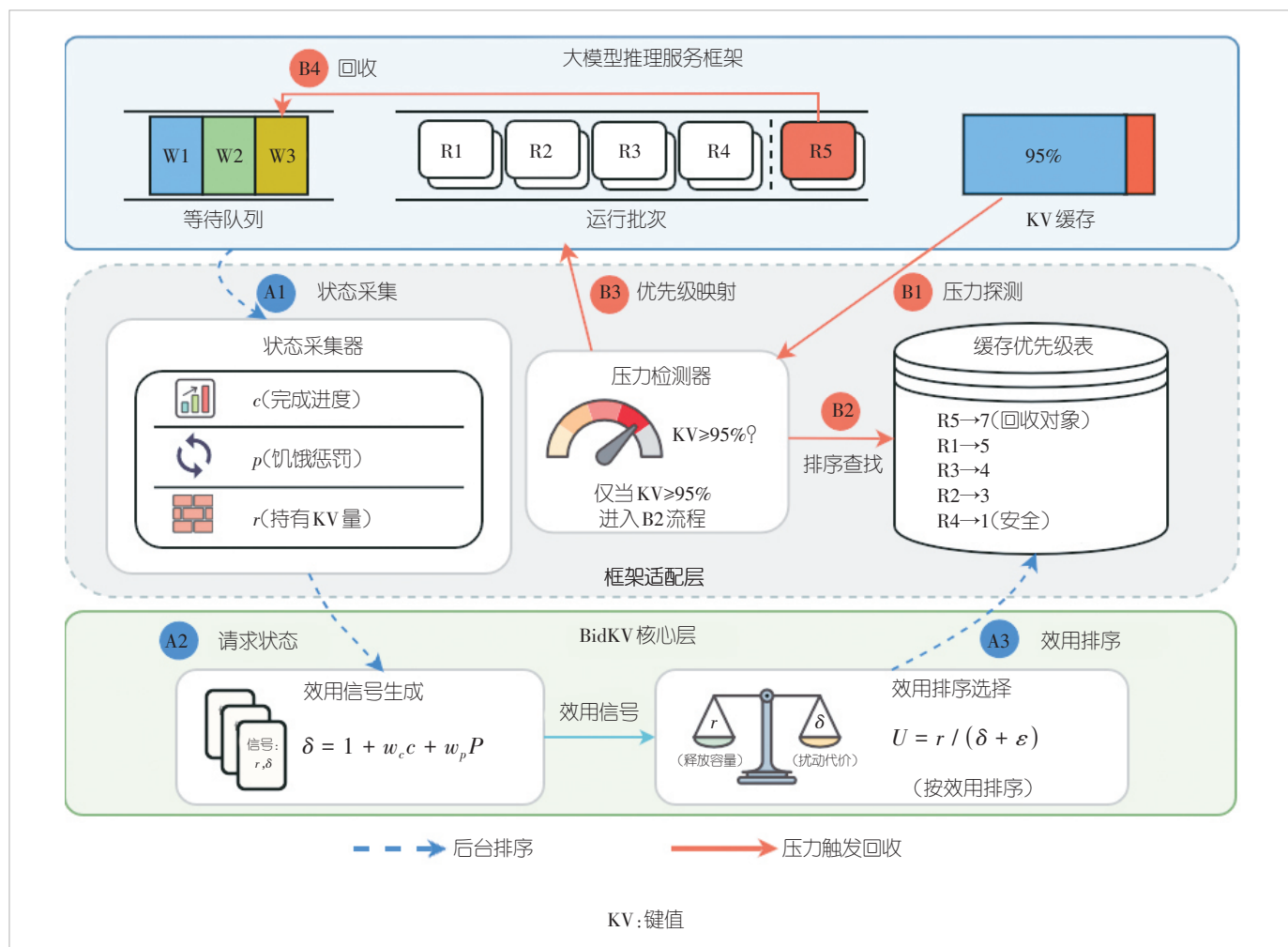
1.4 基于效用的在线选择

BidKV依据下式计算请求竞价效用：

$$U(b) = r / (\delta + \varepsilon) \quad (2),$$

其中， $\varepsilon = 10^{-3}$ 用于避免分母为零。该效用表示单位估计扰动代价所能释放的KV容量。效用越高，说明该请求能够以较低代价释放较多的KV空间，因此更适合作为优先回收对象。

每轮后台排序按效用 U 由高到低生成优先级序列；KV压力发生时，依次回收高效用请求，直至满足容量需求。运



行请求数为 B 时, 排序复杂度为 $O(B \log B)$, 通常低于一次解码步骤的开销。该排序同时利用 KV 占用与生命周期状态, 优先选择释放容量大、重算代价低的请求。

1.5 运行时适配接口

轻量运行时适配器负责读取 KV 容量、拦截压力事件、调用原生抢占、更新请求状态, 并在完成后清理竞价信息。

BidKV 仅决定“回收哪个请求”, KV 释放、重新排队与恢复执行仍由原框架完成, 因此不改变生成结果或底层内存路径。在 vLLM 中, BidKV 通过队列调整引导原生抢占; 在 SGLang 中, 则以私有后缀估计可释放容量。

2 跨框架轻量化适配机制

BidKV 以独立 Python 包的形式非侵入式接入推理引擎, 通过统一注册接口支持评分策略的替换, 并复用框架原生的

KV 管理与恢复路径。

在 vLLM 0.17.1 中, BidKV 通过插件系统在每次原生调度前执行回调。适配器根据缓存的效用排序调整运行队列, 使目标请求位于 vLLM LIFO 抢占路径的末端, 从而引导原生抢占机制选择最高效用的请求。实际的 KV 块释放、请求重新排队及恢复执行仍由 vLLM 完成, BidKV 不修改 KV 缓存协调器或内存块池。为避免主动抢占引入额外的重算开销, BidKV 仅在 KV 利用率超过 95% 且平均提示长度不高于 500 词元时启用运行队列重排; 低于该条件时, 系统保持框架的默认调度行为。请求完成后, 与该请求相关的效用对象及历史状态将被及时清理。

与 vLLM 不同, SGLang 的适配需要处理其独特的 KV 组织方式。SGLang 采用 RadixAttention 管理 KV 缓存, 允许多个请求共享前缀树中的 KV 节点。在此机制下, 抢占某个请求并不会释放其共享前缀, 仅能释放该请求独有的后缀部分;

共享前缀仍保留于缓存中，且在请求恢复时无须重新计算。因此，若直接以请求的总KV长度作为可释放容量，调度器将高估包含大量共享前缀的请求的回收收益。

为适配这一特性，SGLang适配器将效用公式中的可释放容量 r 替换为请求私有KV词元数 r_{priv} 。在每轮的调度中，适配器通过前缀树节点的引用计数识别共享KV的位置，统计各运行请求独有后缀的长度，并以此来估计真实可释放容量以及相应的重算影响。除该框架相关的调整外，SGLang与vLLM共享相同的效用生成以及效用排序逻辑。该实现不修改SGLang的核心执行与内存管理代码，表明代价感知KV回收可通过少量适配迁移至不同KV组织方式的推理框架中。

3 性能评估与可移植性验证

3.1 实验设置

单卡实验使用1张NVIDIA RTX A6000（48 GiB，CUDA 12.5）与bf16精度的Llama-3.1-8B-Instruct^[9]模型，主要平台为vLLM 0.17.1，KV容量固定为9 600词元。测试负载由从ShareGPT数据集中采样的1 000个混合长度请求构成^[10]，请求按泊松过程到达；每种配置运行3次，并报告测量结果的平均值。多卡补充实验使用2张昇腾910B2，配置张量并行度为TP=2，请求速率为3.8请求/秒（req/s）。

评估指标包括吞吐率、300 ms SLO达成率、TTFT的95分位值，以及每输出词元时间（TPOT）的95分位值。对比方法包括默认抢占策略（PE）、短作业优先抢占策略（PE-SJF）、静态随机回收（Static-Random）、最大容量优先回收（Largest-First）以及本文方法（BidKV）。

3.2 vLLM中等负载性能

表1给出了请求到达率为3.8 req/s时的实验结果。在该负载下，系统持续承受KV压力，但尚未完全饱和，因此能够较为清晰地反映不同回收对象选择策略之间的性能差异。

与默认抢占策略相比，本文方法将TTFT P95从6 769 ms降至631 ms，降幅超过90%，并将300 ms SLO达成率从

68.4%提升至83.2%，绝对值提升了14.8个百分点。PE-SJF的结果表明，仅调整等待队列顺序即可显著改善TTFT，但其TPOT P95升至153.5 ms，这说明了准入优化无法单独解决抢占后的重算干扰问题。本文方法则进一步在运行请求间进行效用排序，在保持较高吞吐率的同时，取得了最低的TTFT P95。

静态随机回收策略虽取得了最高的吞吐率与最高的SLO达成率，但其TTFT P95达到1 216 ms，约为本文方法的两倍，这说明积极抢占虽能提升整体处理速度，却可能加重部分请求的尾部等待。最大容量优先回收策略的TTFT P95为677 ms，但其SLO达成率比本文方法低3.3个百分点。原因在于，KV占用最大的请求通常已执行较长时间，回收后需付出较高的重算代价。本文方法同时兼顾容量收益与扰动代价，在吞吐率仅略低于默认策略的情况下，显著改善了面向用户的首Token响应性能。

3.3 SGLang可移植性验证

为验证该方法能否迁移至不同的推理框架，本文在相同混合负载与3.8 req/s到达率的条件下，将BidKV接入SGLang的RadixAttention后端，并与原生SGLang及随机回收适配器进行比较。

BidKV将SGLang的SLO达成率由54.4%提升至93.2%，TTFT P95由6 207 ms降至337 ms；与随机回收策略相比，亦取得了更高的SLO达成率、更低的TTFT及更高的吞吐率。表2为SGLang可移植性实验结果。上述结果表明，采用私有KV长度作为可释放容量估计，可有效避免因共享前缀导致的回收收益高估，且所设计的效用接口能够通过轻量适配

表2 SGLang可移植性实验结果

策略	300 ms SLO达成率/%	首Token延迟 P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间 P95/ms
原生方法	54.4	6 207	3.01	58.8
随机回收	92.3	386	3.69	70.1
本文方法	93.2	337	3.72	70.9

SLO:服务等级目标

表1 vLLM中等负载下的性能对比(请求速率为3.8 req/s)

策略	300 ms SLO达成率/%	首Token延迟 P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间 P95/ms
PE	68.4	6 769	3.48	97.0
PE-SJF	79.6	666	3.06	153.5
Static-Random	84.1	1 216	3.57	90.9
Largest-First	79.9	677	3.28	112.9
BidKV	83.2	631	3.42	107.2

PE:抢占策略 PE-SJF:短作业优先抢占策略 SLO:服务等级目标

迁移至不同推理框架。

3.4 双卡TP并行验证

为贴近多卡部署场景，本文在2张昇腾910B2上配置张量并行度（TP=2），并在3.8 req/s的请求速率下，比较了默认PE、PE-SJF、Static-Random、Largest-First以及本文方法（BidKV）的性能表现。

表3显示，在3.8 req/s的负载下，BidKV的TTFT P95为631.8 ms，SLO达成率为72.00%，吞吐率为1.806 req/s。与PE相比，BidKV的TTFT P95降低了94.2%，SLO达成率提升了25.20个百分点；与Static-Random相比，TTFT P95降低了70.1%，SLO达成率提升了7.78个百分点。PE-SJF的SLO达成率比BidKV高2.35个百分点，但BidKV的TTFT P95低27.4%；Largest-First的TTFT P95比BidKV低14.6 ms，但BidKV的SLO达成率高出5.70个百分点。上述结果表明，BidKV并非追求单项指标最优，而是在双卡TP并行场景下实现了尾部延迟与SLO达成率之间更好的平衡。

3.5 超参数敏感性分析

在与表1相同的负载条件下（3.8 req/s）进行单因素扫

表3 双卡昇腾910B2上TP=2的性能对比(请求速率为3.8 req/s)

策略	300 ms SLO达成率/%	首Token延迟P95/ms	吞吐率/(req·s ⁻¹)	单Token输出时间P95/ms
PE	46.80	10 945.3	1.768	208.4
PE-SJF	74.35	869.7	1.848	199.4
Static-Random	64.22	2 113.2	1.800	196.1
Largest-First	66.30	617.2	1.796	198.5
BidKV	72.00	631.8	1.806	198.8

PE: 抢占策略 PE-SJF: 短作业优先抢占策略 SLO: 服务等级目标

描实验，其余参数固定为默认值： w_c 分别取0.25、0.5、1.0和2.0， w_p 分别取0.1、0.3、0.6和1.0；各配置独立运行。

表4结果显示，吞吐率为3.34~3.43 req/s，SLO达成率为82.3%~84.2%，TTFT P95为600~659 ms，TPOT P95为100.7~112.0 ms。所有配置均显著优于默认抢占策略（SLO达成率68.4%，TTFT P95为6 769 ms）。

在较宽的扫描参数范围内，系统性能保持稳定，这表明基于相对效用的排序机制不依赖于某一组特定的最优权重。默认值 $w_c=0.5$ 和 $w_p=0.3$ 位于扫描区间的内部，能够在保护高进度请求与防止重复抢占之间取得良好平衡。该结果验证的是权重的鲁棒性，并非直接衡量模型对实际终止长度预测误差的容忍度。

4 结束语

本文中我们提出BidKV，以可释放KV容量与扰动代价构造效用函数，在跨请求范围内选择高收益、低代价的回收对象，并复用推理引擎的原生抢占与恢复路径。实验结果表明，BidKV能够显著降低vLLM与SGLang的尾部TTFT，提升SLO达成率，并保持具有竞争力的吞吐率；双卡TP=2的进一步实验验证了其在多卡场景下的适用性。当前采用的归一化解码进度无法精确反映因停止词元导致的提前终止，未来可结合输出长度预测或终止概率估计进行改进。目前，BidKV已集成至华中科技大学vLLM-HUST国产推理引擎工程链路，可为自主可控、低时延、高并发的推理服务提供轻量级调度支撑。

参考文献

- [1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [PP/OL]. arXiv(2023-08-02) [2026-06-16]. <http://arxiv.org/abs/1706.03762>

表4 BidKV超参数敏感性实验(请求速率为3.8 req/s)

参数配置	w_c	w_p	吞吐率/(req·s ⁻¹)	300 ms SLO达成率/%	首Token延迟P95/ms	单Token输出时间P95/ms
默认配置	0.5	0.3	3.39	82.3	633	109.6
$w_c=0.25$	0.25	0.3	3.34	82.8	631	112.0
$w_c=1.0$	1.0	0.3	3.43	84.2	600	100.7
$w_c=2.0$	2.0	0.3	3.35	83.1	659	108.2
$w_p=0.1$	0.5	0.1	3.40	83.5	636	104.3
$w_p=0.6$	0.5	0.6	3.36	83.2	654	105.4
$w_p=1.0$	0.5	1.0	3.42	83.9	640	102.4

注: 默认配置为 $w_c=0.5, w_p=0.3$; 各配置独立运行
SLO: 服务等级目标

- [2] Kwon W, Li Z, Zhuang S, et al. Efficient memory management for large language model serving with PagedAttention [PP/OL]. arXiv (2023-09-12)[2026-06-12]. <https://arxiv.org/abs/2309.06180>
- [3] Yu G I, Jeong J S, Kim G W, et al. Orca: a distributed serving system for Transformer-based generative models [C]// Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI '22). Carlsbad: USENIX Association, 2022: 521-538.
- [4] Agrawal A, Kedia N, Panwar A, et al. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve [PP/OL]. arXiv (2024-06-17)[2026-06-15]. <https://arxiv.org/abs/2403.02310>
- [5] Zhong Y, Liu S, Chen J, et al. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving [PP/OL]. arXiv(2024-06-06) [2026-06-15]. <https://arxiv.org/abs/2401.09670>
- [6] Zhang Z, Sheng Y, Zhou T, et al. H2O: heavy-hitter oracle for efficient generative inference of large language models [PP/OL]. arXiv(2023-12-18) [2026-06-14]. <https://arxiv.org/abs/2306.14048>
- [7] Xiao G, Tian Y, Chen B, et al. Efficient streaming language models with attention sinks [PP/OL]. arXiv(2024-04-07) [2026-06-13]. <https://arxiv.org/abs/2309.17453>
- [8] Zheng L, Yin L, Xie Z, et al. SGLang: efficient execution of structured language model programs[PP/OL]. arXiv(2024-06-06) [2026-06-13]. <https://arxiv.org/abs/2312.07104>
- [9] Dubey A, Jauhri A, Pandey A, et al. The Llama 3 herd of models [PP/OL]. arXiv(2024-11-23) [2026-06-10]. <https://arxiv.org/abs/2407.21783>
- [10] ShareGPT Team. ShareGPT: share your ChatGPT conversations [EB/OL]. [2026-06-10]. <https://sharegpt.com>

作者简介



陈彦博，华中科技大学在读硕士研究生；主要研究方向为大模型推理系统优化。



王明琪，华中科技大学在读硕士研究生；主要研究方向为大模型推理系统优化。



张书豪，华中科技大学教授；主要研究方向为大模型推理引擎、分布式推理、KV Cache与内存管理、服务调度、算子与内核优化；主持国家级及企业科研项目10余项；入选国家高层次人才项目；发表论文40余篇。

中兴通讯技术杂志社成功举办 第31次编委会议暨2026年通信热点技术研讨会

2026年8月1日—2日，中兴通讯技术杂志社第31次编委会暨通信热点技术研讨会在山西大同成功召开。本次盛会汇聚国内外高校、通信运营商、科研院所、中兴通讯等50家单位的130余位专家学者；陆建华、段宝岩、张宏科、张文军、陈勇5位院士，谈振辉教授、蒋林涛总工等资深编委，中兴通讯董事长方榕、CTO兼杂志社总编王喜瑜、高级副总裁王翔等公司领导出席了会议。

中兴通讯董事长方榕致欢迎辞。她表示，中兴通讯聚焦“连接+算力”双轮战略，持续研发投入，取得一系列成绩。在中兴通讯的创新之路上，两本学术期刊成为了联动产业界与学术界的关键桥梁。方榕董事长希望中兴通讯的两本期刊坚守学术初心，发挥期刊学术引领与产学研平台的协同优势，促进科技创新与产业创新深度融合，发挥编委优势，培养创新人才。

中兴通讯 CTO、杂志社总编王喜瑜在致辞中指出，中兴通讯将集中资源攻坚智算核心芯片，自主搭建国产化算力底座。杂志社将坚守“科技向善”办刊定位，打造高水平交流平台。

杂志社副主编卢丹作年度工作报告，系统复盘本年度期刊品质提升、行业影响力建设相关工作，介绍中英文双刊斩获的各类行业荣誉，同步发布下一年度重点工作规划。

会上，陆建华院士作特邀主旨学术报告，系统解读中国天地一体算力网络建设与发展，报告内容引发在场专家广泛关注。

本次会议同步开设中兴通讯未来发展高端论坛，参会院士、学界专家、企业高管围绕企业中长期战略布局、底层技术自主创新等核心议题充分交流，输出多项具备落地价值的建设性意见。

通信热点技术研讨环节，张宏科、艾渤、任奎、赵慧玲、向际鹰、敖立等14位海内外知名专家依次作专题分享，围绕人工智能通信、新一代无线信道、AI网络安全等前沿方向，分享最新科研成果与产业落地实践。

作为年度固定学术交流平台，中兴通讯技术杂志社编委会议打通学术界与产业界沟通壁垒，搭建高效、高质量的双向交流通道，持续为中国信息通信产业技术创新与发展注入动能。

存储层级感知的大模型前缀键值缓存系统



Storage-Hierarchy-Aware Prefix KV Cache System for Large Language Models

何水兵/He Shuibing, 瞿皓阳/Qu Haoyang, 党政/Dang Zheng

(浙江大学, 中国 杭州 310027)
(Zhejiang University, Hangzhou 310027, China)

DOI: 10.12142/ZTETJ.202604003

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260826.1410.002>

网络出版日期: 2026-08-26

收稿日期: 2026-05-25

摘要: 大语言模型服务中的前缀键值 (KV) 缓存容量需求不断增长, 逐渐超出图形处理器 (GPU) 显存和中央处理器 (CPU) 内存的承载能力。利用容量大、成本低的固态硬盘 (SSD) 扩展前缀缓存具有较大潜力, 但其延迟较高、随机读带宽较低, 直接应用会显著增加首字延迟 (TTFT), 难以满足实际部署需求。针对这一问题, 提出了存储层级感知的大模型前缀键值缓存系统 PCache。PCache 面向显存-内存-SSD 分层存储架构, 通过数据布局感知的重要 KV 选择、页面级连续 KV 加载、动态 KV 重排和变序计算等技术, 减少 SSD 访问与随机输入/输出 (I/O) 开销。相关实验验证了 PCache 可以在保持推理精度稳定的同时降低 TTFT。

关键词: 大语言模型; 前缀 KV 缓存; 分层存储; 首字延迟

Abstract: As the demand for prefix key-value (KV) cache capacity in large language model (LLM) services continues to grow, the required capacity has gradually exceeded what graphics processing unit (GPU) memory and central processing unit (CPU) memory can support. Although using high-capacity and low-cost solid state drives (SSDs) to expand prefix caches is promising, SSDs have high access latency and low random-read bandwidth, and directly introducing them can significantly increase time to first token (TTFT), making it difficult to meet practical deployment requirements. To address this issue, this paper proposes PCache, a storage-hierarchy-aware prefix key-value (KV) cache system for large language models. On a GPU memory-CPU memory-SSD hierarchy, PCache reduces SSD accesses and random input/output (I/O) overhead through data-layout-aware important KV selection, page-level continuous KV loading, dynamic KV reordering, and out-of-order computation. Experimental results show that PCache significantly reduces TTFT while maintaining stable inference accuracy, providing a technical reference for designing efficient prefix KV cache systems for long-prefix large language model serving.

Keywords: large language model; prefix KV cache; hierarchical storage; TTFT

引用格式: 何水兵, 瞿皓阳, 党政. 存储层级感知的大模型前缀键值缓存系统 [J]. 中兴通讯技术, 2026, 32(4): 9-18. DOI: 10.12142/ZTETJ.202604003

Citation: He S B, Qu H Y, Dang Z. Storage-hierarchy-aware prefix KV cache system for large language models [J]. ZTE technology journal, 2026, 32(4): 9-18. DOI: 10.12142/ZTETJ.202604003

以 ChatGPT 为代表的大语言模型 (LLM) 在自然语言理解、代码生成、文本摘要、智能问答等任务中展现出强大的泛化能力, 已成为人工智能应用领域的重要基础设施。随着模型服务从单轮问答扩展至多轮对话、检索增强生

成 (RAG) 和智能体等复杂场景, 输入侧的上下文长度不断增加, 不同请求之间也更容易出现共享前缀。因此, 如何减少长前缀带来的重复处理开销, 逐渐成为提升大模型推理效率的关键问题。

在实际服务中, 共享前缀通常源于系统提示词、用户身份和权限信息、输出格式约束、工具描述以及外部知识文档等。为避免重复计算这类前缀, 现有系统通常将其对应的键值张量缓存为前缀键值缓存 (Prefix KV Cache), 后续请求

基金项目: 浙江省自然科学基金重大项目 (LD24F020012); 国家自然科学基金区域创新发展联合基金项目 (U25A20423); 杭州城西科创大走廊专项发展资金项目 (K20241957)

若命中相同前缀,则直接复用缓存结果,从而减少预填充阶段的重复计算,并降低首字延迟(TTFT)。

目前,前缀键值缓存主要存储在图形处理器(GPU)显存和中央处理器(CPU)内存中,但随着缓存规模持续增长,二者在容量和成本方面的瓶颈逐渐凸显,难以支撑长前缀场景下的大规模缓存需求。相比之下,固态硬盘(SSD)具备大容量、低成本且易于部署等优势,可作为扩展前缀缓存的存储层级。然而,SSD的访问特性与前缀KV的加载需求并不匹配:SSD的访问延迟远高于显存和内存,细粒度、离散化的读取方式还会降低顺序读带宽,并且在页级管理机制下引发读放大问题,导致扩展后的缓存容量难以直接转化为TTFT收益。

针对上述问题,本文提出一种存储层级感知的大模型前缀键值缓存系统PCache。PCache面向显存-内存-SSD分层存储架构,对前缀键值数据的选择、加载、布局 and 计算过程进行协同优化:首先,通过感知数据布局的重要KV选择,在满足精度约束的同时减少高代价的SSD访问;其次,采用页面级连续KV加载模式,将离散访问转化为更适合SSD的顺序读取;再次,通过动态KV重排提高单次页加载的有效数据命中率,缓解读放大问题;最后,利用变序计算策略,避免重排后恢复原始顺序所带来的额外开销。通过上述设计,PCache能够在保持推理精度稳定的同时,降低前缀键值缓存的访问开销,从而提升推理性能。

1 大模型前缀缓存与分层存储

大模型推理通常通过键值缓存(KV Cache)减少生成阶段的重复计算;而在多会话推理场景下,进一步引入前缀缓存(Prefix Cache),以存储和复用不同请求中共享前缀对应的KV Cache,从而避免对该部分KV Cache的重复计算。随着可复用前缀规模持续增长,缓存系统需要借助分层存储扩展缓存容量,但不同存储介质的访问代价差异也会带来新的数据访问问题。

1.1 键值缓存

大模型每个Token的注意力输出仅依赖于其之前的Token。在自回归生成过程中,除新生成的Token外,历史Token对应的键值张量保持不变。基于这一特性,可将注意力计算中的键值张量缓存到显存中,后续生成步骤直接复用历史键值,仅计算当前Token的键值及注意力输出。这就是键值缓存^[1]。该策略将注意力层在生成阶段的重复计算转化为缓存访问,显著提升后续Token的生成速度。

1.2 前缀缓存

在大模型应用场景中,不同推理请求之间往往存在明显的上下文重叠。例如,多轮对话中的历史消息、插件调用中的系统提示词以及RAG任务中的共享文档片段都可能作为可复用前缀。前缀缓存技术将常用输入前缀对应的键值缓存预先计算并持久化存储,使后续匹配相同前缀的请求能够直接加载缓存状态,绕过重复的前缀计算过程,从而降低首字延迟,节省计算资源。随着智能体等大模型应用场景中可复用前缀的长度不断增长,现有基于内存的前缀缓存系统越来越难以满足应用的存储容量需求。

1.3 前缀缓存分层存储

为解决上述问题,业界开始探索将SSD进一步纳入存储层级,构建显存、内存、SSD三级前缀缓存存储架构。然而,显存、内存和SSD在容量、成本与访问延迟上存在明显差异:显存适于保存近期或高频访问的前缀键值,但容量有限;内存可作为显存之外的缓存层,容量更大但访问延迟更高;SSD能以较低成本保存大量前缀数据,但在随机访问和小粒度读取场景下面临较高延迟。因此,前缀键值缓存不再只是简单的存储扩容问题,而是需要在不同存储层级之间协调数据选择、加载粒度和物理布局,使容量扩展不会显著增加TTFT。

1.4 现有技术与发展趋势

现有前缀KV缓存系统主要从数据选择、缓存管理和存储硬件3个方面降低存储与访问开销。在数据选择方面, SnapKV^[2]根据注意力分布预测生成阶段的KV重要性,优先保留重要KV; IMPRESS^[3]面向多级前缀KV存储,根据KV重要性选择加载关键KV而非全部KV,以减少存储输入/输出(I/O)延迟; CacheGen^[4]识别并压缩非关键KV,降低缓存交换成本; CacheBlend^[5]面向RAG多轮请求,根据KV相似性选择性复用KV。上述系统能够降低逻辑读取量,但仍存在两点局限:其一,重要性评估侧重于模型贡献,缺乏对分层存储访问代价的建模;当重要KV位于SSD或分布较为离散时,仍可能产生频繁的高延迟访问;其二,Token级或KV条目级选择容易破坏SSD连续读取的特性,将数据访问转化为大量随机小读,从而降低有效读带宽。

在缓存管理方面, PagedAttention/vLLM^[6]将KV缓存组织为固定大小的块,通过虚拟页机制复用相同前缀对应的KV块,从而降低存储和计算开销; SGLang^[7]通过RadixAttention管理共享前缀,提高多请求间KV复用效率; AttentionStore^[8]将可复用KV直接卸载到SSD,并通过块级换入换出减少内

存占用；华为统一缓存管理器（UCM）^[9]将前缀KV替换为可唯一寻址的缓存块，通过块索引定位并加载已有KV。这些工作有效提高了前缀KV的缓存空间利用率，但未充分考虑SSD中KV数据的物理布局。当目标KV分散在多个SSD页面时，即使每页仅有少量KV被使用，系统仍需读取完整页面，造成读放大。

在存储硬件方面，NVIDIA^[10]在本地SSD与通用共享存储之间引入面向KV缓存的G3.5闪存层，通过高速网络和专用数据处理器实现跨节点KV缓存的高带宽共享；高带宽闪存（HBF）^[11]利用NAND堆叠与高带宽封装技术，在提供更大存储容量的同时，缩小闪存与高带宽内存之间的带宽差距。上述技术侧重于优化硬件性能，通过提升存储带宽来加速KV缓存访问，但尚未在数据组织层面减少无效I/O。综上，面向SSD扩展的前缀缓存系统需要从数据选择、读取粒度和物理布局3个层面进行协同优化。

2 存储层级感知的大模型前缀键值缓存系统

2.1 总体设计

针对上述问题，我们设计了PCache，一种感知存储层级的前缀键值缓存系统。PCache特别适用于多轮对话、知识库问答和智能体应用等通常存在大量重复上下文的大模型推理服务场景，其整体架构如图1所示。PCache基于显存-内存-SSD分层存储架构管理大模型前缀键值缓存，并围绕SSD访问瓶颈进行针对性布局优化。系统首先结合KV所在

存储层级、访问代价、物理页面分布及精度约束，选择需要取回的重要KV；随后采用页面级连续KV加载以减少随机读；同时根据历史访问模式动态重排KV布局，提高单次页面读取的有效数据密度；最后通过变序计算直接使用重排后的前缀KV，从而避免恢复原始顺序带来的额外开销。

2.2 访存代价感知的重要KV选择策略

PCache首先需要确定在多个存储层级中优先读取哪些重要的KV。为在减少SSD访问的同时保持推理精度，系统借助已有的Token贡献度评估结果作为精度约束，而不是将其作为唯一选择依据。我们在OPT-6.7B模型上，使用LM Evaluation Harness基准测试套件^[12]中的4个数据集，对不同KV选择方式进行了测试：一组采用注意力得分最高的 k 个Token（Top- k ），另一组随机选择Token（Rand），直至随机组的注意力得分总和超过Top- k 集合的总分。如图2所示，当保留Token数量较多时，Top- k 策略与Rand策略的推理精度相差无几；当保留Token数量较少时，Rand策略在部分任务中甚至优于Top- k 策略。该结果表明，在一定范围内，推理精度更依赖于被保留KV集合的总贡献度，而不完全取决于具体保留哪些Token。

基于上述结果，PCache在选择重要KV时同时考虑数据贡献度与访问代价。对于位于SSD上的高访问代价KV，系统可使用多个已驻留在显存或内存中的低访问代价KV进行替代，只要替代后的KV集合仍能满足相同的精度约束即可。这样既避免了单纯按重要性排序所引发的高延迟SSD访

问，也能优先利用分层缓存中已有的数据，从而减少对SSD的随机读取请求。

具体而言，PCache将候选Token按物理存储位置划分为显存集合 S_{GPU} 、内存集合 S_{CPU} 和SSD集合 S_{SSD} ，每个集合内的Token按贡献度降序排列。随后，系统结合各层访存代价设置不同的候选粒度：显存和内存中的Token可以按多个Token聚合为候选单元，SSD中的Token则以更严格的访存代价参与选择。最后，系统计算各候选单元的累计贡献度，并按单位访存代价收益进行贪心选择，直到满足当前请求的精度约束。其中，重要KV选择策略的效果与保留比例相关。当保留比

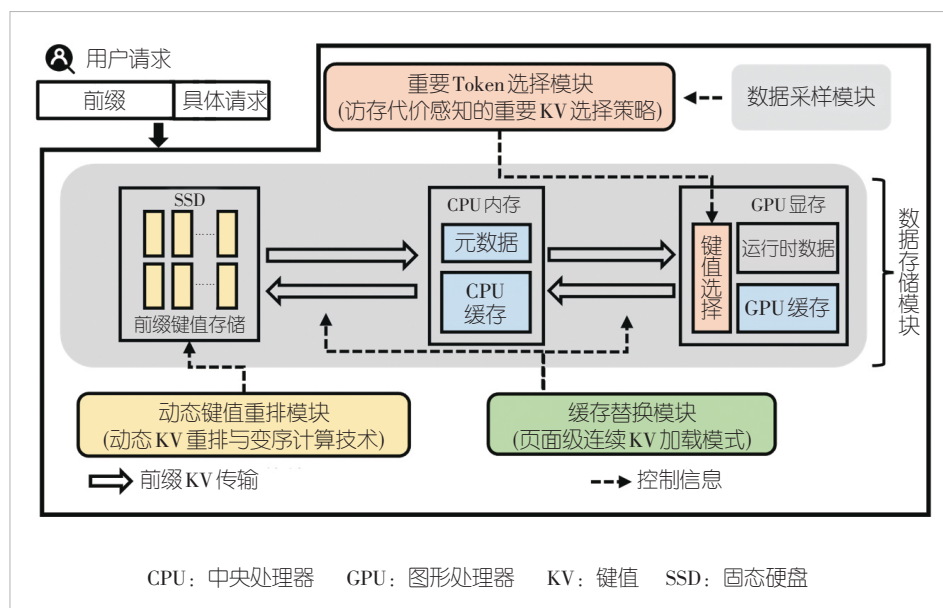


图1 PCache整体架构

例过低时，部分任务的推理精度可能出现波动。因此，PCache将根据离线精度测试设置贡献度阈值，在访问开销和模型精度之间取得平衡。

以图3为例，假设用户输入的前缀为“你是一个翻译，将用户的中文转为英文”。当从前缀缓存中选择相应的重要KV时，传统策略会直接选取贡献度最高的Token集合，如“中文”“英文”“转为”。若其中“转为”对应的KV位于SSD上，则会触发高延迟读取。而PCache的访问代价感知选择策略会检查显存和内存中是否存在若干低访问代价KV，能够共同提供相近的贡献度。若存在，系统则优先选取显存和内存中的KV。例如，图中将“转为”替换为“你是”和“翻译”，从而在不损失贡献度的前提下减少SSD访问，并缩短前缀数据的取回时间。

2.3 页面级连续KV加载模式

SSD更适合连续读取，而非大量小粒度的随机读取。若系统按Token粒度逐个取回前缀KV，即便最终读取的数据总量不大，也会产生大量离散访问，从而降低SSD的有效带宽。针对这一问题，PCache采用页面级连续KV加载模式：当系统判定需要访问某个页面中的目标KV时，不再仅提取页内少量Token，而是按页连续加载相关KV数据。如图4所示，页面级连续KV加载将原本分散的多次随机读合并为少量顺序读，从而充分利用SSD的连续读带宽。同时，页内一并载入的KV可能提供额外的上下文贡献，使系统在满足同等精度约束的前提下，减少所需访问的页面数量。

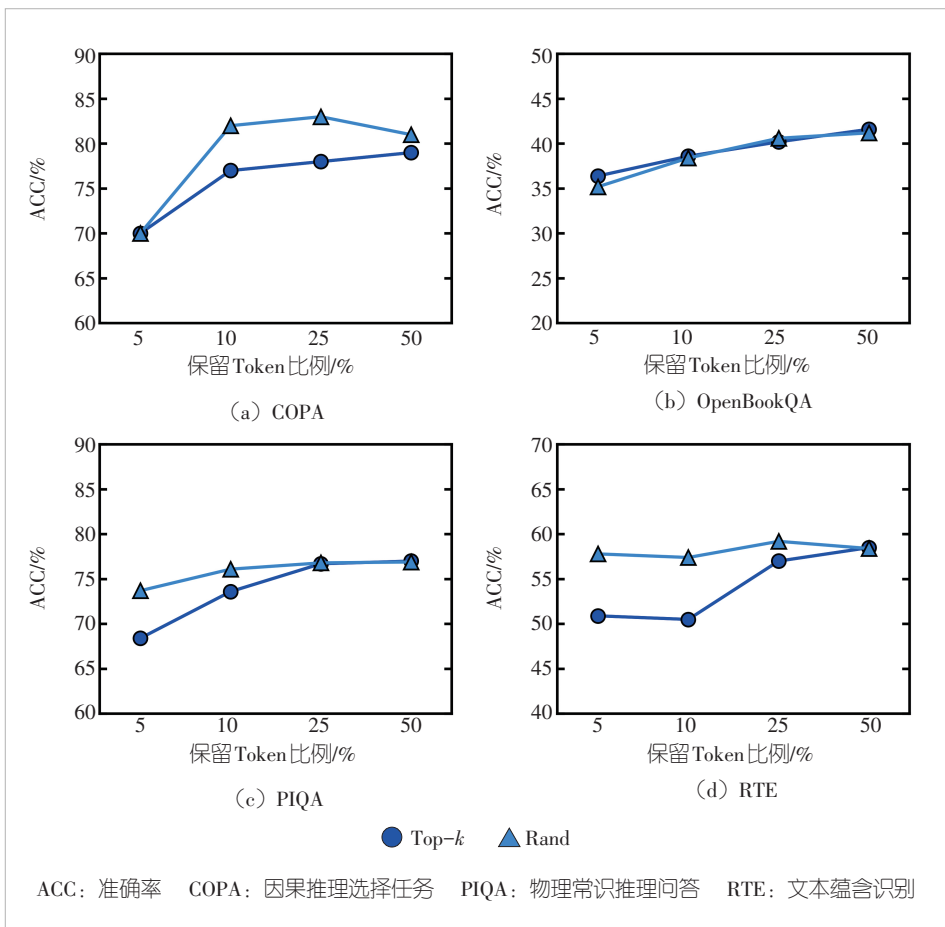


图2 不同KV选择策略精度对比

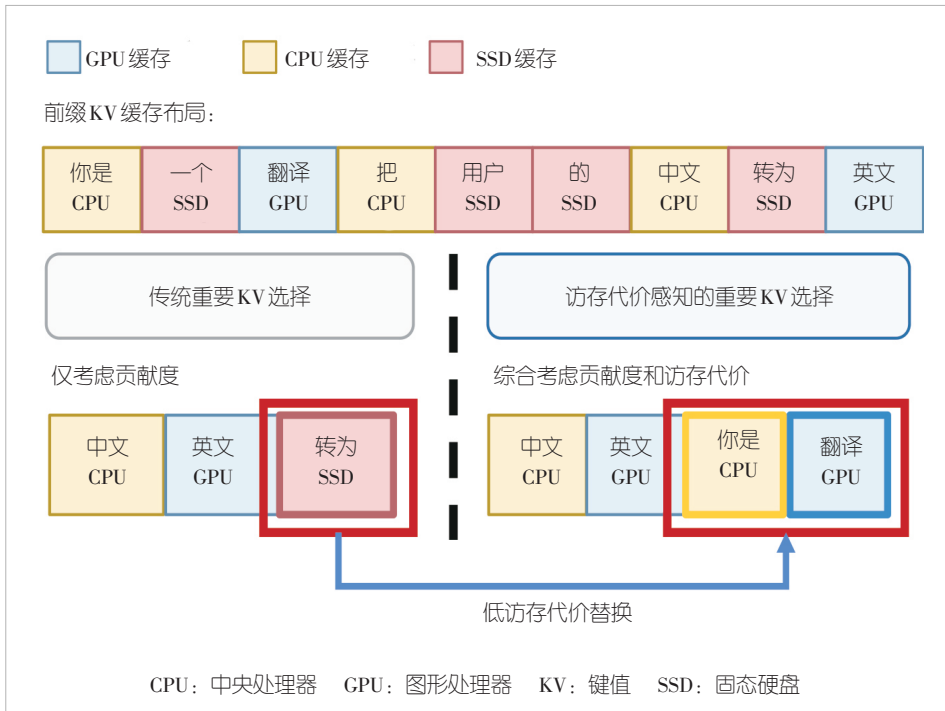


图3 访问代价感知的重要KV选择示例

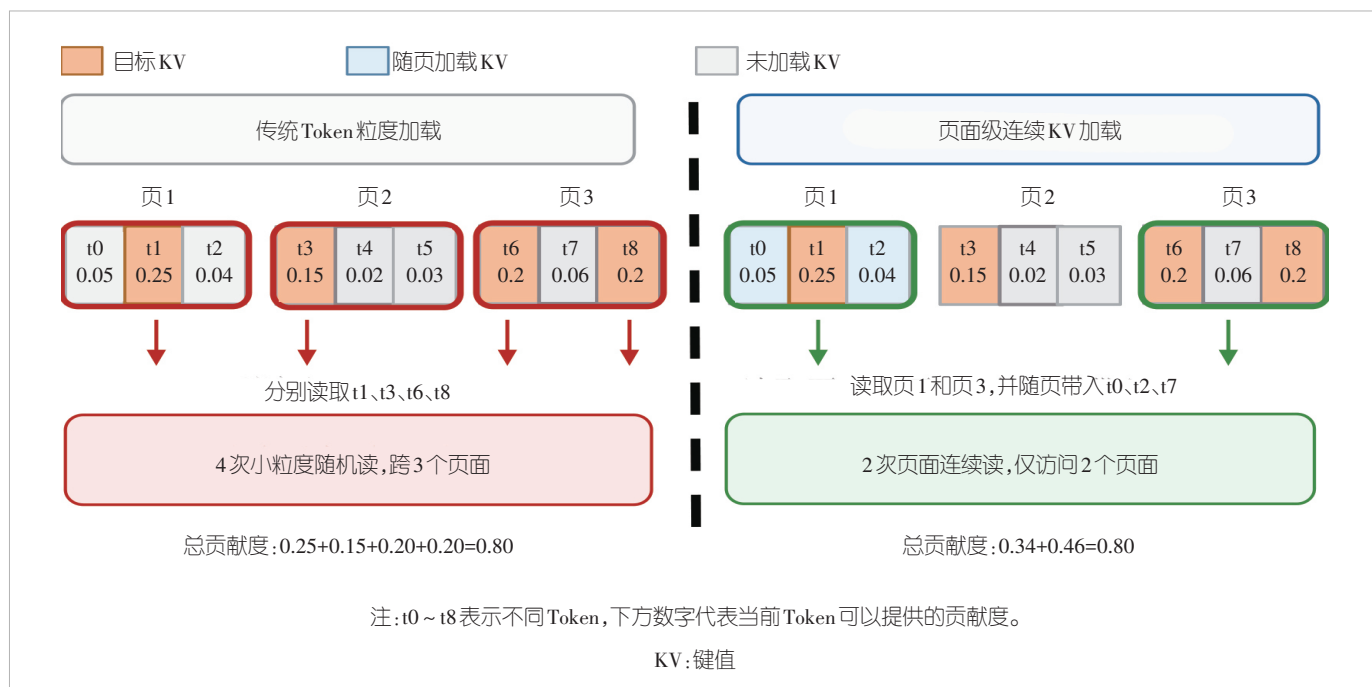


图4 页面级连续KV加载模式和传统模式对比

页面级连续KV加载会捎带部分非目标KV,从而引入额外的数据读取。但PCache通过减少随机I/O次数、提高SSD连续读带宽,同时让额外加载的数据参与后续计算或缓存复用,从而降低整体数据取回时间。

2.4 动态KV重排与变序计算技术

在长期运行过程中,某些前缀KV可能被频繁访问,但其在SSD中的物理位置常分散于多个页面,导致单次请求需访问大量页面,产生读放大。为解决这一问题,PCache基于近期请求中KV的选中频率,动态重排页面中KV数据的物理布局。重排过程将高频访问KV迁移并紧凑排列至少量物理页,将低频KV集中至尾部页面。重排后,单次页加载命中有效KV的概率提升,系统在相同精度约束下仅需访问更少页面,从而缓解读放大,降低SSD读取开销。如图5所示,经KV重排后,数据加载时仅需访问页1和页2,即可覆盖原本分散于3个页中的目标KV。

KV重排会带来额外的计算开销和SSD数据写入。为此,PCache采取自适应的异步KV重排方法。首先,系统通过可配置的请求阈值动态调整重排触发频率。例如,当检测到上下文长度增加导致单次重排数据量上升时,系统会提高阈值以降低重排频率,从而减少对SSD的反复写入。同时,对于系统提示词、任务模板等高复用前缀,其访问热点在一定时间窗口内相对稳定,历史重排结果可被后续多个请求复用,

从而摊薄数据迁移开销。当累计请求数量超过设定阈值后,KV重排在低峰期后台异步执行,避免干扰在线推理服务的性能。

此外,动态KV重排会改变前缀KV在物理存储中的排列顺序,因此系统需要额外维护元数据的一致性。常规做法是在元数据中维护原始位置到重排后位置的映射关系,并在数据从存储介质加载至显存后执行重排序,以恢复原始逻辑顺序。然而,这会引入额外的元数据存储、索引查询以及数据搬移开销。从注意力计算的角度来看,只要每个Token对应的KV保持配对,且位置编码信息已包含在缓存张量中,前缀部分参与注意力计算时并不依赖其物理读取顺序。模型推理只需保证当前请求的Token能够访问其之前的所有前缀KV,而无须按原始物理顺序排列。因此,PCache在推理时直接加载并使用重排后的前缀KV,无需恢复为原始Token顺序。变序计算简化了系统元数据维护,减少了加载后的数据搬移和随机写入,从而使动态KV重排能够以较低的运行开销获得布局优化的收益。

3 实验与结果

为评估PCache在分层存储场景下对推理精度和首字延迟的影响,本文基于OPT-6.7B、OPT-13B和OPT-30B 3个开源模型进行实验。测试平台配备单张NVIDIA A800 GPU (80 GB显存)、128 GB内存和2 TB Intel SSD,SSD连续读带

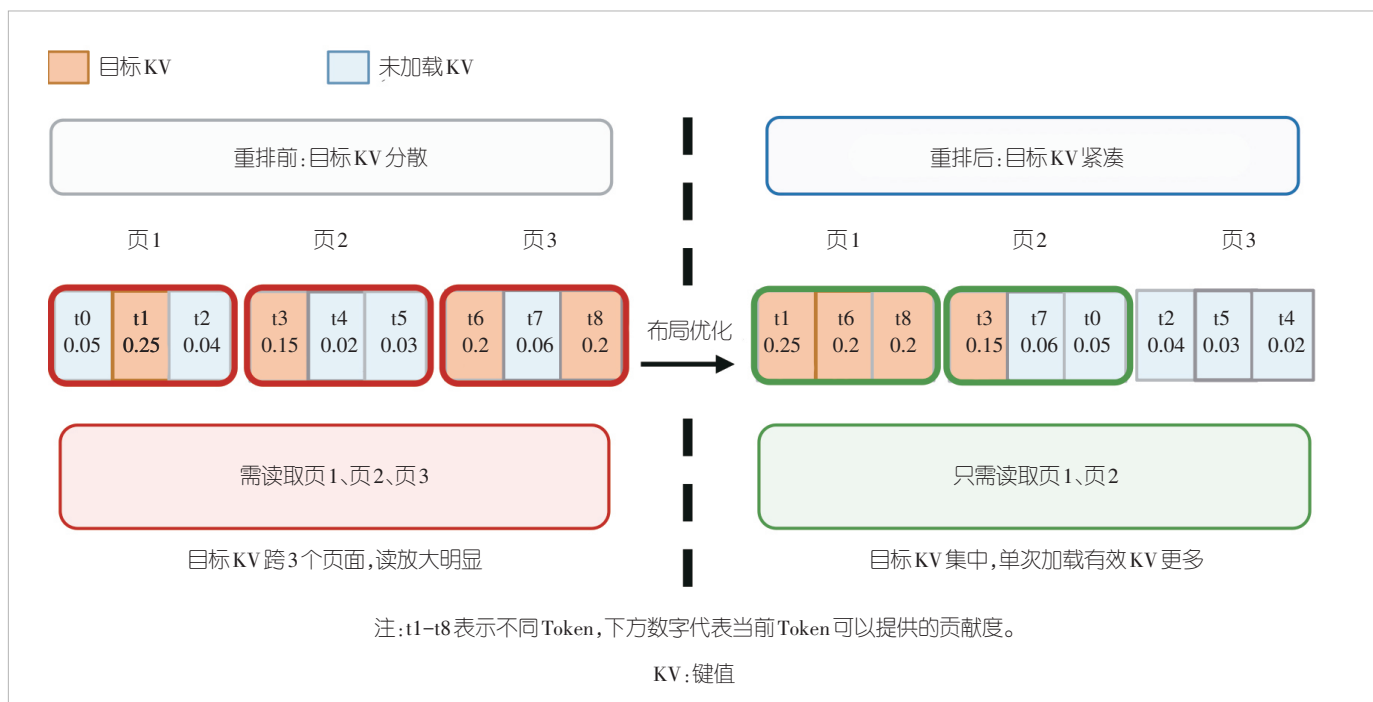


图5 KV重排前后对比

宽实测约为5 GB/s。实验中，分配10 GB显存和32 GB内存用于前缀缓存，其余前缀KV存储于SSD，构成GPU显存-CPU内存-SSD三级缓存架构。为模拟实际大模型服务中系统提示词等前缀在不同请求间重复使用的场景，实验选用LM Evaluation Harness基准测试套件^[12]中的PIQA、RTE、COPA和OpenBookQA共4个数据集，参照现有工作^[13]，在每个请求前添加2~10个few-shot示例作为共享前缀，并通过正态分布设定不同前缀的复用频率，以构建具有非均匀访问热度的前缀缓存负载。

本文将PCache与以下3个基准系统进行对比：1) 重计算(ReComp)：不使用前缀缓存，每个请求到达时重新计算全部前缀Token的键值，作为精度无损但延迟最高的基线；2) 注意力存储系统(AS)：基于FlexGen复现Attention-Store^[8]，采用全量异步存储与加载键值缓存。为保证公平性，本文为其配置了与PCache相同的GPU显存-CPU内存-SSD三级缓存架构，并禁用了其依赖先来先服务(FCFS)调度的特定预取优化；3) AS+重击者选择方法(H2O)：在AS基础上集成H2O^[14]，该基线先加载前缀的全量K向量以计算Token贡献度，再加载被选中的V向量以完成后续推理。

实验从模型推理精度和TTFT两个维度评估PCache性能，分别用于衡量分层缓存优化对模型输出质量和前缀KV取回延迟的影响。

3.1 模型推理精度

本节的测试遵循现有工作^[13-14]，采用回答正确率(ACC)作为核心评价指标，并在5%~50%范围内调整KV保留比例，以探究不同压缩比例下PCache的数据选择与加载机制是否会影响模型推理精度。

图6展示了各系统在不同KV保留比例下对模型推理精度的影响。由于重计算与AS在推理精度上无显著差异，且PCache主要改变数据调度与物理访问方式，并未修改模型计算逻辑，因此图6以ReComp代表重计算和AS的重合结果，并展示了AS+H2O与PCache的精度曲线。实验结果显示，PCache在所有测试模型和数据集上的推理精度与对比基线基本持平。与重计算基准及AS+H2O相比，PCache的精度损失均控制在1%以内。这表明，PCache的访存代价感知重要KV选择策略与页面级连续KV加载机制，能够在减少SSD访问开销的同时，有效保留模型输出所需的关键上下文信息。

3.2 模型推理延迟

为在精度与延迟之间取得合理平衡，基于图6的精度结果，延迟实验中将COPA数据集的KV保留比例设为50%，其余数据集设为25%。在此配置下，PCache带来的精度损失不足0.2%，处于实际应用可接受的范围内。图7展示了各系统的平均TTFT对比。PCache相对于重计算基线实现了

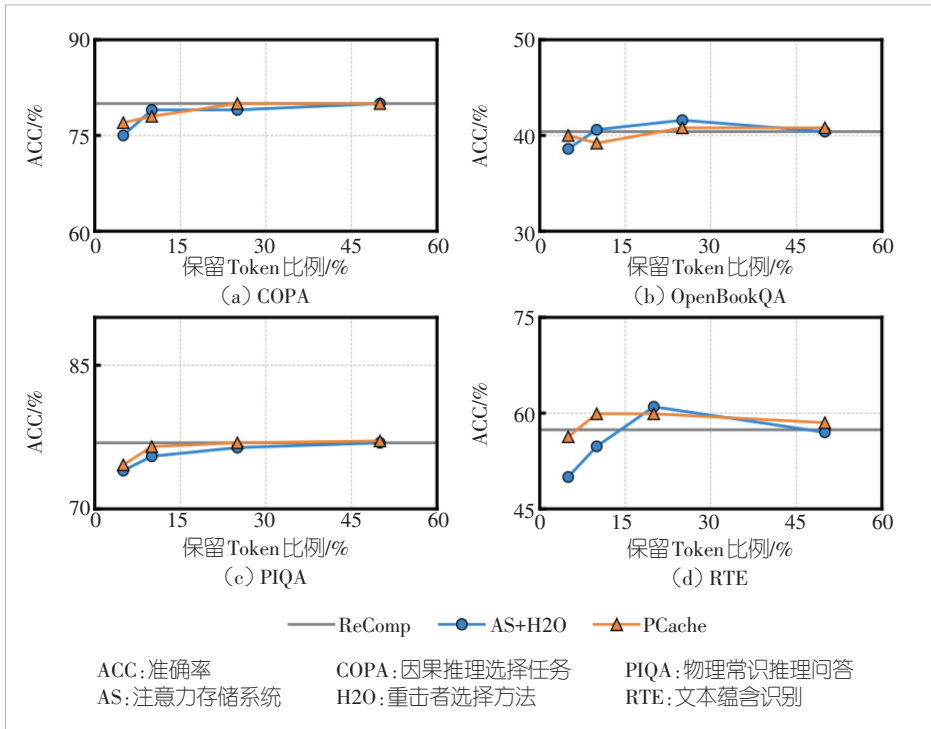


图6 不同数据集下模型推理精度对比

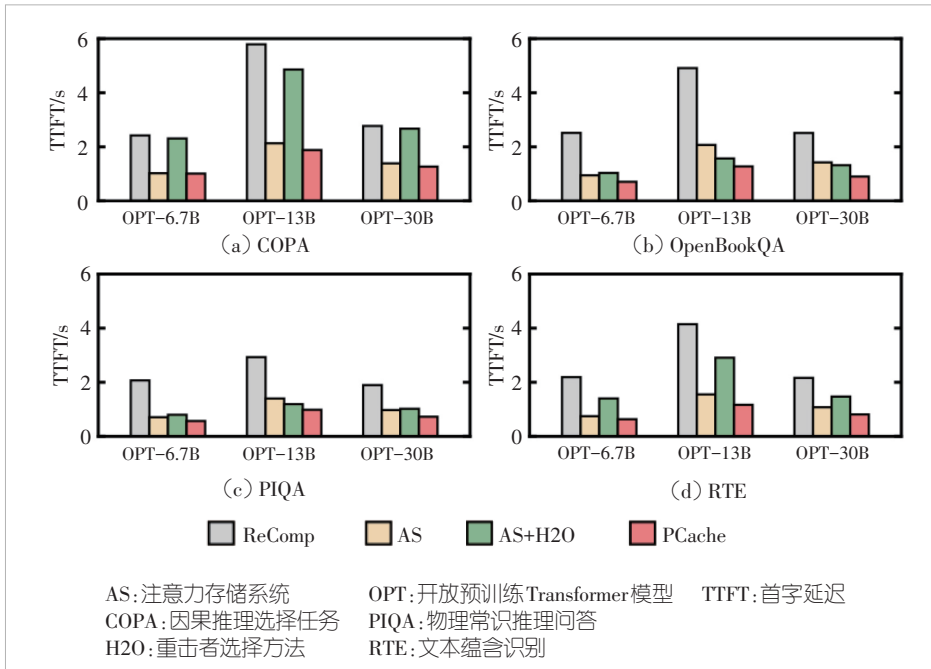


图7 不同数据集下模型推理TTFT对比

2.18 ~ 3.84 倍的 TTFT 加速。即便与每组对比中的最优基线相比, PCache 仍能获得最高约 46% 的额外性能提升。

4 结束语

随着大模型服务向长上下文和高复用场景的持续演进,

前缀 KV 缓存已成为影响系统成本与服务首字延迟的关键资源。本文提出存储层级感知的大模型前缀键值缓存系统 PCache, 通过重要 KV 选择、页面级连续 KV 加载、动态 KV 重排及变序计算等机制, 有效降低 SSD 扩展带来的随机访问开销。实验结果表明, PCache 能够在保持推理精度稳定的同时, 显著降低 TTFT 延迟。这说明, SSD 扩展前缀缓存的关键不仅在于增加存储容量, 还需要围绕访问粒度、物理布局与计算过程进行协同优化。

参考文献

- [1] Pope R, Douglas S, Chowdhery A, et al. Efficiently scaling transformer inference [J]. Proceedings of machine learning and systems, 2023, 5: 606-624
- [2] Li Y H, Huang Y B, Yang B W, et al. SnapKV: LLM knows what you are looking for before generation [C]// Proceedings of Advances in Neural Information Processing Systems 37. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2024: 22947-22970. DOI: 10.52202/079017-0722
- [3] Chen W, He S, Qu H, et al. IMPRESS: an importance-informed multi-tier prefix KV storage system for large language model inference [C]// Proceedings of the 23rd USENIX Conference on File and Storage Technologies. 2025: 187-201
- [4] Liu Y H, Li H C, Cheng Y H, et al. CacheGen: KV cache compression and streaming for fast large language model serving [C]// Proceedings of the ACM SIGCOMM 2024 Conference. ACM, 2024: 38-56. DOI: 10.1145/3651890.3672274
- [5] Yao J Y, Li H C, Liu Y H, et al. CacheBlend: fast large language model serving for RAG with cached knowledge fusion [C]// Proceedings of the Twentieth European Conference on Computer Systems. ACM, 2025: 94-109. DOI: 10.1145/3689031.3696098
- [6] Kwon W, Li Z H, Zhuang S Y, et al. Efficient memory management for large language model serving with PagedAttention [C]// Proceedings of the 29th Symposium on Operating Systems Principles. ACM, 2023: 611-626. DOI: 10.1145/3600006.3613165

- [7] Zheng L M, Yin L S, Xie Z Q, et al. SGLang: efficient execution of structured language model programs [PP/OL]. arXiv (2023-12-12) [2026-05-25]. <https://arxiv.org/abs/2312.07104>
- [8] Gao B, He Z, Sharma P, et al. Cost-efficient large language model serving for multi-turn conversations with CachedAttention [PP/OL]. arXiv (2024-03-23) [2026-05-25]. <https://arxiv.org/abs/2403.19708>
- [9] ModelEngine-Group. Unified cache management [EB/OL]. (2026-02-10) [2026-05-25]. <https://github.com/ModelEngine-Group/unified-cache-management>
- [10] Anschel M, Zilberstein E, Duer O, et al. Introducing NVIDIA BlueField-4-powered CMX context memory storage platform for the next frontier of AI [EB/OL]. (2026-03-26) [2026-05-25]. <https://developer.nvidia.com/blog/introducing-nvidia-bluefield-4-powered-inference-context-memory-storage-platform-for-the-next-frontier-of-ai>
- [11] Sandisk. High bandwidth flash (HBF) fact sheet [EB/OL]. [2026-05-25]. https://documents.sandisk.com/content/dam/asset-library/en_us/assets/public/sandisk/collateral/company/Sandisk-HBF-Fact-Sheet.pdf
- [12] Gao L, Tow J, Biderman S, et al. A framework for few-shot language model evaluation [EB/OL]. [2026-05-25]. <https://zenodo.org/records/12608602>
- [13] Lee W, Lee J, Seo J, et al. InfiniGen: efficient generative inference of large language models with dynamic KV cache management [C]//Proceedings of the 18th USENIX Symposium on Operating Systems Design and Implementation. 2024: 155-172
- [14] Zhang Z Y, Sheng Y, Zhou T Y, et al. H2O: heavy-hitter oracle for efficient generative inference of large language models [C]//Proceedings of Advances in Neural Information Processing Systems 36. Neural Information Processing Systems Foundation, Inc. (NeurIPS), 2023: 34661-34710. DOI: 10.52202/075280-1506

作者简介



何水兵，浙江大学计算机科学与技术学院教授、博士生导师；主要从事智能计算、存储系统、计算机体系结构等方面的研究；主持国家级和省部级项目 10 余项；获 2024 年浙江省科技进步奖一等奖、2023 年浙江省科技进步奖一等奖；发表论文 100 余篇。



瞿皓阳，浙江大学在读硕士研究生；主要研究方向为大语言模型推理优化。



党政，浙江大学在读博士研究生；主要研究方向为智能计算、异构内存管理等。

面向智能体时代大语言模型推理的新型存算解聚合架构



A Novel Storage-Compute Disaggregated Architecture for Large Language Model Inference in the Agent Era

熊先奎/Xiong Xiankui, 王程晨/Wang Chengchen,
蔡文豪/Cai Wenhao

(中兴通讯股份有限公司, 中国 深圳 518057)
(ZTE Corporation, Shenzhen 518057, China)

DOI: 10.12142/ZTETJ.202604004

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260901.0957.002>

网络出版日期: 2026-08-24

收稿日期: 2026-06-02

摘要: 针对传统大模型推理在长上下文场景及新兴智能体时代推理中所面临的键值 (KV) Cache 膨胀、本地显存容量受限及跨节点通信开销高等问题, 提出一种基于光输入/输出 (OIO) 拉远且以内存为中心的新型存算解聚合架构。该架构在物理层面实现计算与存储资源的解耦, 支持二者的独立伸缩、池化共享与按需协同; 在计算侧将 Prefill 与 Decode 阶段划分为异构计算簇, 在存储侧构建基于异质材料的全局内存池, 并借助近存计算单元对集合通信原语进行卸载。基于 8 个异构计算处理单元 (8XPU) 与 8 个内存处理单元 (8MPU) 全互联配置的系统级仿真分析表明, 该架构在长上下文及高并发推理场景下, 可有效降低首 Token 时延与逐 Token 时延, 减少数据搬运开销, 提升系统吞吐量, 并在满足同等服务水平协议 (SLA) 的条件下显著降低对计算资源的需求, 验证了其在智算基础设施中的潜在应用价值。

关键词: 智能体时代; 存算解聚合; 内存为中心; 光输入/输出; 近存计算

Abstract: To address the challenges of Key-Value (KV) cache explosion, limited local video memory capacity, and high inter-node communication overhead in long-context inference for traditional large language models as well as emerging agent-based reasoning scenarios, a novel compute-storage disaggregated architecture centered on optical I/O (OIO)-attached remote memory is proposed. At the physical level, this architecture decouples compute and storage resources, enabling independent scaling, pooled sharing, and on-demand collaborative orchestration. On the compute side, heterogeneous clusters are designated separately for the Prefill and Decode phases, while on the storage side, a global memory pool composed of heterogeneous materials is constructed, with near-memory processing units offloading collective communication primitives. System-level simulation results based on a fully interconnected configuration of 8 X processing units (8XPU) and 8 memory processing units (8MPU) demonstrate that the proposed architecture effectively reduces both time-to-first-token and per-token latency, minimizes data movement overhead, and improves system throughput in long-context and high-concurrency inference scenarios. Moreover, under equivalent service-level agreement (SLA) constraints, the required compute resources are significantly reduced, confirming its potential applicability in intelligent computing infrastructure.

Keywords: agent era; compute-storage disaggregation; memory-centric; optical I/O; near-memory computing

引用格式: 熊先奎, 王程晨, 蔡文豪. 面向智能体时代大语言模型推理的新型存算解聚合架构 [J]. 中兴通讯技术, 2026, 32(4): 17-22. DOI: 10.12142/ZTETJ.202604004

Citation: Xiong X K, Wang C C, Cai W H. A novel storage-compute disaggregated architecture for large language model inference in the agent era [J]. ZTE technology journal, 2026, 32(4): 17-22. DOI: 10.12142/ZTETJ.202604004

1 大语言模型推理现状与智能体时代的核心挑战

随着大语言模型 (LLM) 迈入规模化落地阶段, 智算基础设施的发展重心正从基础模型训练转向代理式人工智能 (Agentic AI) 驱动的推理服务^[1]。多模态应用的普及、长思维链能力的提升等, 使得 Agentic AI 应用中的核心瓶颈逐渐由算力限制转向内存约束。键值 (KV) Cache、记忆读写与状态管理所带来的访存开销, 已取代计算开销, 成为制约大规模智能体集群部署的关键因素。

1.1 大语言模型算法规模、上下文长度对内存容量的需求

随着 LLM 参数规模从 Llama-3 时期的 32B、DeepSeek V3 时期的 671B^[2], 快速扩展至 DeepSeek-V4 Pro^[3]、Kimi K2.5、Qwen3-Max 等万亿级规模, 基础设施面临的计算与存储压力日益严峻。长上下文指 LLM 在单次推理中处理超过 12.8 万超长文本的能力, 使模型能够一次性理解长篇书籍或完整代码库, 实现跨文档的精准推理与记忆。例如, 基于长上下文能力的推理阶段扩展范式由 OpenAI O1 模型首次提出, 并迅

速成为 LLM 的核心能力之一^[4]。进入智能体时代后，长程任务固有的复杂性、多智能体协作中的知识拓扑结构，以及多跳推理与证据链追溯等需求，对长上下文能力提出了更高要求^[5]。当前主流模型的上下文长度已达 25.6 万个（开源）至 200 万个（闭源）Token。然而，Transformer Decode-only 架构的计算与存储开销随上下文长度 N 呈 $O(N^2)$ 增长。尽管可通过量化、稀疏化、压缩等算法手段减轻负担，但长上下文带来的 KV Cache 容量需求依然可观。例如，具有混合注意力层的 Qwen3.5-235B 模型在 100 万 Token 长文本推理时，KV Cache 容量需求约为 90 GB^[6]；而 DeepSeek-V4-Pro 模型中，即便引入了压缩稀疏注意力（CSA）与重度压缩注意力（HCA）机制^[3]，其单个百万级 Token 上下文的 KV Cache 大小仍达 10.33 GB。KV Cache 正逐步取代模型权重，成为智算基础设施中主要的访存瓶颈。

未来智能体基础模型的重要能力“持续学习”和“可解释性”对存储系统提出了新要求。DeepSeek 提出的 Engram 架构将静态知识存储从高带宽内存（HBM）转移到主机内存（DRAM）中，在当前 Engram-27B 架构中，内存开销约为 5 GB^[7]，在 DeepSeek-V4 架构参数下，以 $\rho=0.7$ 估算，内存开销约为百 GB 级别；Google 提出的 Titan 和 Nest-Learning 架构^[8]，引入了推理过程中的可学习参数，考虑到参数读写和优化器存储开销，将使当前参数存储需求提升 3~5 倍。

1.2 智能体应用特征对内存带宽与访问模式的需求

Agentic AI 是一种能够自主运作、规划并执行复杂多步骤任务的智能系统。其核心特征在于从传统 LLM 的单一指令推理与被动响应模式，转向“自主决策、目标导向、持续学习”的主动式框架。Agentic AI 具备多步规划、工具调用及长期记忆等重要能力，单个任务可包含数百轮 LLM 请求^[5]，这使得推理负载从单次请求演变为连续的长链任务，模型存储开销与复用特征随之发生明显变化。

以 KV Cache 为例，在智能体典型工作流 ReAct 中，LLM 请求以“增量追加”的方式构建，每次执行动作或观察环境后，模型均在前文基础上继续推理。因此，多轮对话中的 KV Cache 命中率显著提升，典型复用概率约为 70%~84%^[9]。为支撑复杂记忆与长链推理能力，存储系统不仅需要大容量 HBM 以支持 Decode 计算，还需在前缀缓存命中时频繁从主机内存甚至固态硬盘（SSD）中读取 KV Cache，高频复用特征进一步强化了对智算基础设施存储容量与带宽的需求。

学术界和工业界已从推理框架、微架构及底层软件栈等多个层面对智算基础设施开展针对性优化。例如，vLLM 等主流推理框架围绕高效内存管理与低时延推理持续演进^[10]，

通过 PagedAttention、Speculative Decoding 等关键技术缓解显存压力与吞吐受限问题；PD 微架构设计将计算密集型的 Prefill 阶段与访存受限的 Decode 阶段解耦，从而提升系统推理效率；LMCache 等分层存储架构通过跨节点缓存迁移与持久化机制，有效降低 PD 场景下的通信开销与计算冗余。国产化异构算力平台亦从算子库、工具链等方面推进优化，着力解决框架兼容性、长尾应用场景及推理性能提升等问题。上述技术共同推动智算基础设施从传统一体化部署模式，向“阶段分离、存储分层、资源弹性调度”的新范式转变。

上述优化仅能解决特定场景下 LLM 推理的瓶颈问题。面向万亿级模型与 Agentic AI 场景，需应对百万级上下文处理、KV Cache 动态管理与高频复用等关键技术挑战，架构级创新将成为智算基础设施的核心能力。因此，未来 Agentic AI 计算集群的设计需从传统的以算力为中心，转向以内存与数据为中心的“Memory-centric”范式。

2 以内存为中心的存算解聚合架构发展现状

工业界围绕内存为中心的优化方向，已从共享内存编排、光互联扩展和内存池化等多个维度展开探索，以应对大模型推理中的内存墙问题。例如，微软 Feng Huang 通过多层共享内存、张量优化与近内存计算，缓解本地显存容量与跨图形处理器（GPU）通信压力；Celestial AI 借助光输入/输出（OIO）将计算芯片与更大规模的共享内存资源互联，试图在更大物理范围内提供接近本地 HBM 的访问性能；三星 CMM-D 则基于计算快速链接（CXL）总线提供面向数据中心的内存扩展与池化能力，以提高容量弹性与资源利用率。然而，上述方案仍主要聚焦于远端内存扩展、互联增强或通用池化能力，对大模型推理场景中计算负载特性、不同类型数据的访存需求，以及存储、通信与计算之间的协同优化，考虑仍不充分。

当前，业界对 OIO 技术的探索已从实验室验证迈入工程化落地阶段。作为一种先进的芯片级光互连技术，OIO 的核心在于利用光信号替代传统电信号进行片间数据传输，旨在将光互连功能直接集成至计算芯片内部，从而突破电互连在带宽密度、传输能耗和物理距离等方面的固有瓶颈。在产业界，主流芯片与半导体厂商正加速推进该技术的布局。2024 年，英特尔率先公开展示了其集成的光计算互连（OCI）芯片组原型；英伟达则明确将在下一代 Feynman 架构中大规模引入硅光子互连，以实现传统铜线互连的彻底替代。台积电研发的 COUPE（紧凑型通用光子引擎）技术采用 3D 异质光电集成方案，已于 2026 年实现规模化量产。在中国，以光联芯科为代表的本土企业亦积极推动 OIO 技术从研发验证

走向实用化部署，展现出良好的产业化前景。

为此，本文提出一种以内存为中心的新型存算解聚合架构。该架构借助 OIO 可将总线物理距离延展的特性，在物理层面将计算资源与存储资源彻底解耦，实现算力与存力的独立扩展、池化共享与按需协同。与现有方案不同，本架构不仅关注远端内存容量的扩展，更强调围绕推理流程构建分层访问语义与协同执行机制：在计算侧设置本地高速内存，并配备面向 Prefill 与 Decode 两类负载特征优化的异构计算簇；在存储侧构建由多个内存处理单元（MPU）节点组成的全局内存池，并结合异质存储材料形成兼顾性能与容量的层次化异构存储结构，从而实现对模型权重、KV Cache、元数据及过程数据的统一管理 with 高效支撑。

进一步地，本方案在 MPU 中引入近存计算能力，使部分数据处理与集合通信原语能够在靠近数据的位置完成，从而降低大模型推理过程中的数据搬移开销以及互联带宽压力，提升系统整体的数据访问效率与资源利用率。基于上述设计，所提出架构能够更好地适配智能体推理场景下的弹性部署需求，为该场景提供一种兼顾容量、带宽、时延与可伸缩性的新型架构支撑。

本方案面向智能体推理场景，主要特征如下：

- 1) 提出一种以内存为中心的存算解聚合架构，面向智能体场景设计，突破传统计算与存储强绑定的部署模式，为大规模智能体系统提供高灵活性的基础设施支撑。
- 2) 构建面向推理执行特征的异质融合内存与异构计算协同机制，提升系统对智能体推理不同阶段及不同负载类型的适配能力与执行效率。
- 3) 结合动态随机存取存储器（DRAM）与 Flash 构建层

次化大容量异质融合内存体系，在保障热点访问性能的同时，突破本地高速内存的容量瓶颈，增强对超大模型参数及过程数据的承载能力。

- 4) 在存储侧引入近存计算能力，使集合通信与部分数据处理能够在靠近数据的位置完成，从而提升系统级协同处理效率。

3 新型存算解聚合智算架构分析

如图 1 所示，该系统主要由本地内存与近端计算层、OIO 互联层以及全局内存与远端存储层 3 部分组成。其中，本地内存层采用字节级访问方式，容量为 GB 级别，主要用于存放指令、元表、元数据等高频访问的小规模数据，并为上层计算簇提供低时延的数据支撑。

近端计算资源根据大模型推理流程的不同阶段，划分为 Prefill 计算簇与 Decode 计算簇，分别适配高吞吐的上下文计算任务与低时延的自回归生成任务。该功能划分使系统能够针对不同负载特征进行差异化的资源配置，从而提升推理过程中的吞吐性能与响应效率。

中间的 OIO 互联层作为连接计算侧与存储侧的核心通道，承担大规模数据交换与远程访存请求的转发功能。相较于传统电互连方式，光互联具备更高带宽、更低时延及更强的扩展能力，能够有效支撑多计算节点与多存储节点之间的高速通信。同时，OIO 互联层为资源池化提供了基础，使不同计算簇可按需访问远端存储资源，从而实现跨节点的数据共享与协同处理。

在存储侧，全局内存层采用分块或 Tile 级访问方式，容量可扩展至百 TB 甚至 PB 级，由多个 MPU 存储节点构成，其

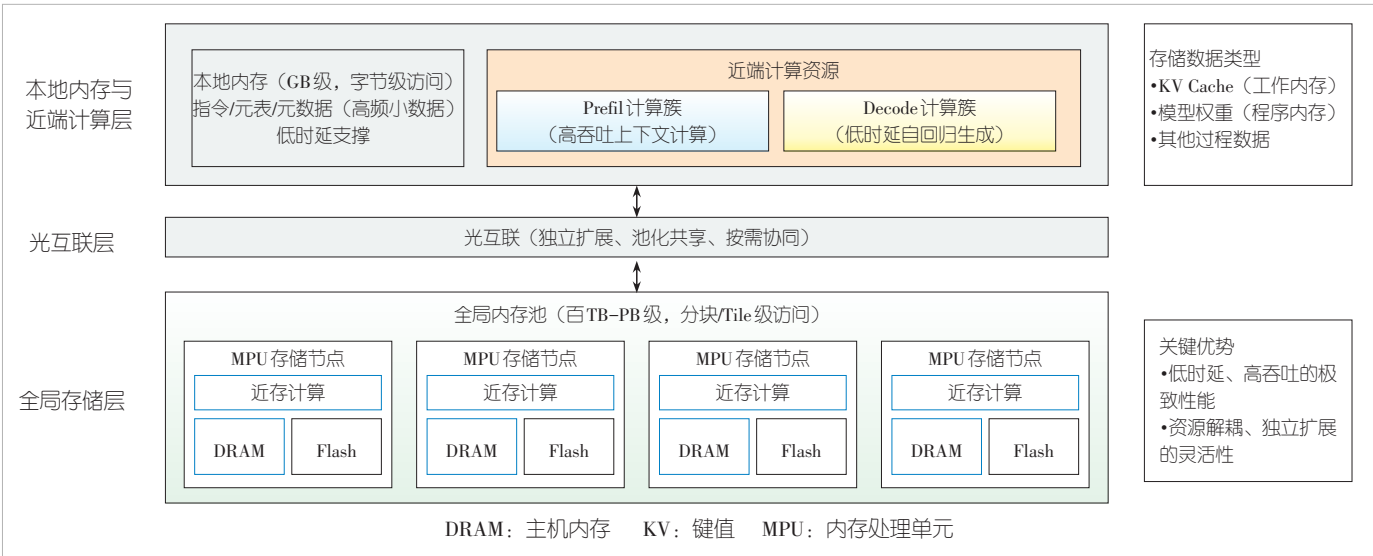


图1 新型存算解聚合架构

中 MPU 是一个定制化 ASIC。每个节点内部结合 DRAM 与 Flash 介质，形成兼顾访问性能与存储容量的层次化存储结构。MPU 还具备近存计算能力，通过自定义近存指令集，可在靠近数据的位置完成部分数据处理与集合通信原语操作，从而减少数据在存储侧与计算侧之间的往复搬移，降低互联带宽压力，并提升系统整体的数据访问与协同处理效率。按照功能划分，全局内存主要存放 3 类数据：一是作为工作内存的 KV Cache 数据，用于支撑推理过程中的上下文状态维护；二是作为程序内存的模型权重参数，用于支持模型计算执行；三是其他过程数据，包括中间激活、调度信息等。通过对不同类型数据的统一组织与分层管理，系统能高效地满足大模型推理过程中的数据访问需求。

该架构的关键优势在于，能够根据业务需求弹性扩展计算与存储资源，避免传统融合式部署中资源按固定比例绑定所带来的扩展受限与利用率不足问题。尤其在 Agent 场景下，模型参数、KV Cache 等大容量数据可集中部署于全局存储侧，而推理任务由不同计算簇动态承担，从而支撑多智能体任务下的弹性部署、高效共享与资源优化配置。

4 新型存算解聚合智算架构的性能分析

为评估本文提出的存算解聚合架构在智能体推理场景下的性能收益，我们通过整合专用加速器、高带宽异构池化内存等仿真库，构建了标准化交互模型，并实现了算网存联合仿真平台。本文以 8 张 H800 为基线配置（如表 1 所述），对比新型存算解聚合架构（如表 2 所述）与常规融合式 GPU 推理架构的性能差异。两种架构采用相同的计算侧资源配置，区别在于存算解聚合架构将模型权重、KV Cache 等大容量数据部署于远端全局内存，并借助 OIO 互联与 MPU 近存计算能力，降低计算侧的通信与显存压力。

新型存算解聚合架构采用 8 个异构计算处理单元（8XPU）+ 8 个内存处理单元（8MPU）的单节点配置，上层计算层由 8 个 XPU 组成，存储层由 8 个 MPU 组成，二者通过全互联的方式协同工作。单个 XPU-MPU 链路的基准带宽为 2 Tbit/s，系统聚合带宽为 2 TB/s。每个 MPU 配置 256 GB DRAM 与 1 TB Flash 存储器，形成总计 2 TB DRAM 与 8 TB Flash 的分层全局存储资源。基于 Qwen 与 Llama 3 模型，在两种配置下分别从首 Token 时延、逐 Token 时延、系统吞吐率、数据搬移量与资源需求等多个维度进行仿真分析。

图 2 给出了 Batch = 50 条件下，不同上下文长度下两种架构的首 Token 时延（TTFT）对比结果。可以看出，在短上下文场景下，两种架构的 TTFT 差异相对有限；而随着上下文长度持续增加，存算解聚合架构表现出更好的扩展性，二

者差距逐步拉大。以 3.2 万输入为例，Qwen235B 的 TTFT 相比常规融合架构下降 80.54%，Llama3 70B 下降 64.50%。这表明在长上下文推理场景下，存算解聚合架构通过将模型权重、KV Cache 及过程状态从本地设备侧解耦，并依托高带宽互联实现按需访问，能够有效缓解本地存储压力，从而显著降低首 Token 生成时延。

图 3 显示，在固定上下文长度为 4 000 的条件下，随着 batch 数目的增加，两种模型的 TPOT 均有所上升，但存算解聚合架构在所有测试点上均优于常规融合架构。其中，Qwen235B 在 batch=100 时 TPOT 由 0.041 3 s 降至 0.023 5 s，降幅达 43.1%；Llama3 70B 在 batch=10、50、100 时分别获得 62.1%、48.3% 和 54.0% 的时延优化。结果表明，存算解聚合架构能够有效降低解码阶段的单 Token 时延，并在高 batch 场景下展现出更优的性能优势与扩展能力。

图 4 显示，在固定 1 000 上下文和 batch=10 的条件下，Qwen 系列模型的吞吐率随参数量增大而下降，但存算解聚合架构在所有参数规模下均显著优于常规融合架构。具体而言，Qwen78B、Qwen130B 和 Qwen235B 的吞吐率分别提升约 138.6%、184.8% 和 138.5%。结果表明，存算解聚合设计能够有效提升大模型推理吞吐率，并在不同参数规模下保持稳定的性能收益。原因在于，传统融合式部署在面对更大规模模型时，往往需要依赖更细粒度的并行切分来适配本地高速内存容量，这不仅增加了跨设备通信与调度管理开销，也降低了系统的有效计算效率。本文架构则通过将模型参数、

表 1 仿真测试硬件基准

参数	规格
GPU 型号	NVIDIA H800
本地 HBM 容量	80 GB
本地 HBM 带宽	2 TB/s
单 GPU 聚合互联带宽	900 GB/s
节点间网络	400 Gbit/s InfiniBand

GPU：图形处理器 HBM：高带宽内存

表 2 新型存算解聚合架构配置参数

参数	规格
XPU 型号	NVIDIA H800
单 MPU 上行带宽(对应 8XPU)	2 Tbit/s(2 TB/s)可配
单内存节点 DRAM 容量	256 GB
单内存节点 FLASH 容量	1 TB
MPU 个数	8
近存计算支持功能	广播、聚合、累加等

DRAM：动态随机存取存储器 XPU：异构处理单元
MPU：内存处理单元

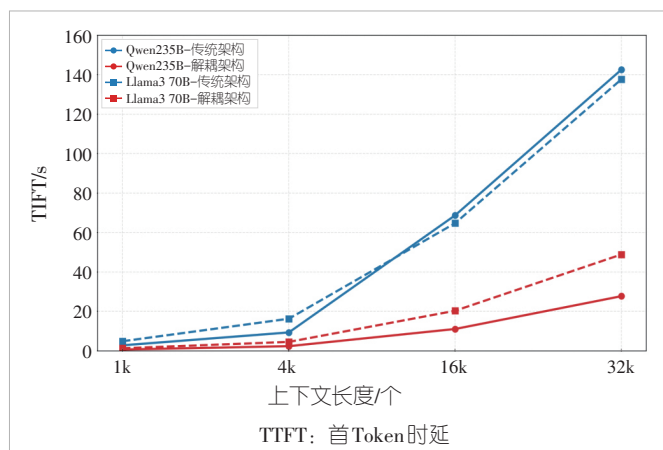


图2 不同上下文的TTFT

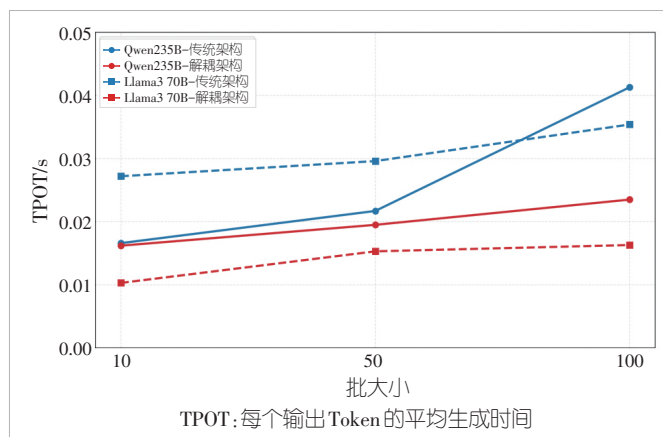


图3 不同batch的TPOT

KV Cache 及中间状态数据从计算侧剥离，统一存放于全局存储层，使计算节点的扩展不再主要受限于本地存储容量，而更多由实际推理算力需求决定。

图5比较了不同模型规模下单请求数据搬移量的差异。结果表明，本文架构在不同模型规模下的单请求数据搬移量均低于常规融合架构，且随着模型规模增大，这种优势愈加明显。本文架构的性能收益不仅源于远端存储的扩容能力，更来自对数据流动路径的系统性优化。通过将热点数据保留于DRAM层、冷数据驻留于Flash层，并在MPU侧完成部分数据处理与集合通信操作，本文架构将原本需频繁发生在计算层的跨设备搬移与同步，转化为更高效、更可控的远端访问与局部处理过程，从而降低了系统内部的数据搬移开销。

图6进一步给出了在满足同等服务质量约束（SLA）条件下，两种架构所需XPU数量的对比结果。可以看出，本文架构在不同模型规模下均需更少的XPU即可达到与常规融合架构相近的服务能力，且在405B等超大模型场景下这一优势尤为显著。该结果表明，在传统融合式部署中，计算资源的扩展不仅受算力需求驱动，也往往受限于本地存储容

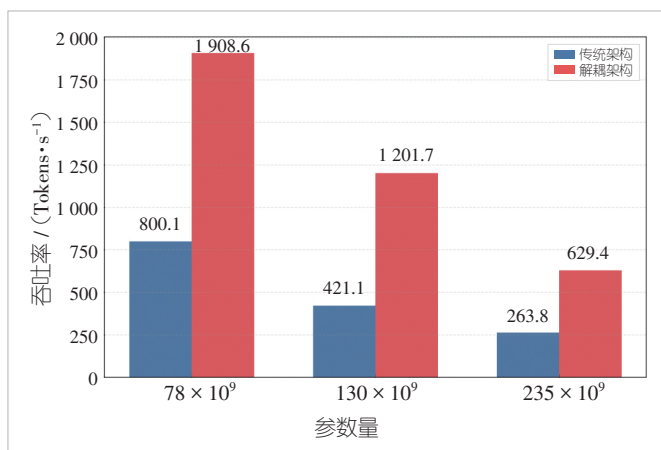


图4 不同参数数量下的吞吐量

量，因而需增加更多XPU以分摊模型参数与KV Cache的驻留压力。本文架构通过在存储层提供2 TB DRAM与8 TB Flash的分层全局内存，使计算资源扩展回归至“按算力需求扩展”的模式，从而在同等SLA约束下有效减少计算层资源需求，体现出更高的资源利用效率与部署弹性。

本文提出的存算解聚合架构在长上下文、大模型及高并发推理场景下，能够显著降低首Token与逐Token时延，减少系统内部数据搬移量，提升整体吞吐量，并在满足同等SLA条件下有效降低计算资源需求。上述结果表明，该架构能够有效突破传统融合式部署中由本地存储容量与跨设备通信共同造成的性能瓶颈，为面向大模型推理任务的弹性部署与高效共享提供了有效的体系结构支撑。

5 结束语

本文针对大模型长上下文及智能体推理场景下日益突出的内存瓶颈问题，提出了一种基于OIO互联拉远、以内存为中心的新型存算解聚合架构。该架构利用OIO互联低损耗、

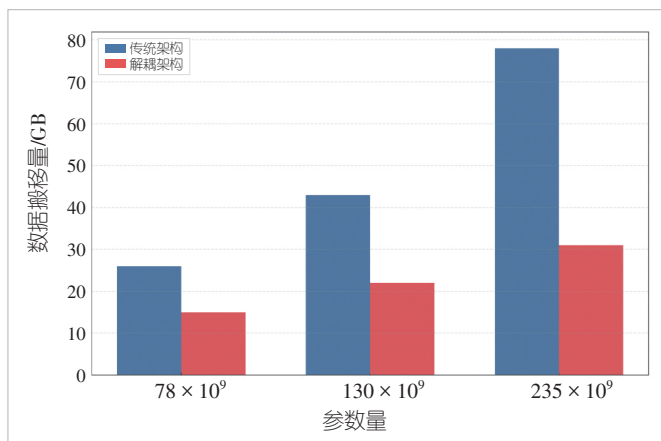


图5 不同模型规模下数据搬移量

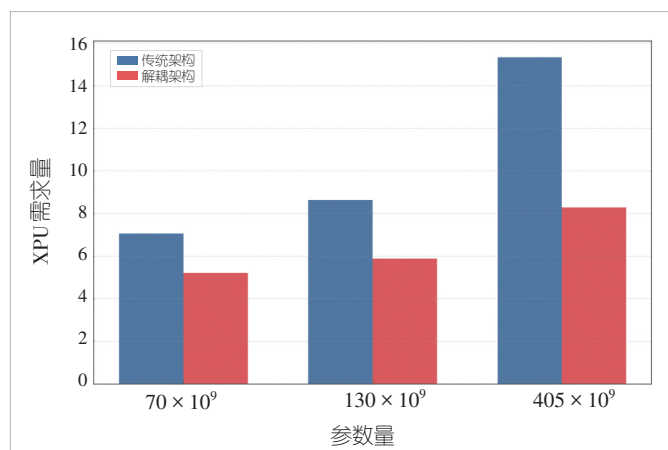


图6 服务质量约束条件下两种架构所需XPU数量

大带宽且与距离解耦的特性，将传统封装内受电互连严重制约的本地短距IO拉远，从根本上打破单芯片的IO密度墙，使计算资源与存储资源实现物理解耦、独立扩展与池化共享。在此基础上，构建了面向Prefill与Decode的异构计算组织，以及由MPU节点组成的DRAM-Flash分层全局异构内存，并引入近存计算能力，降低了推理过程中的数据搬移与互联压力。仿真表明，该架构通过IO拉远真正释放存算解耦的潜力，为大模型推理系统提供了一种高效能的新范式。

未来工作将围绕以下3个方面展开：一是结合原型系统实现，对远端访存、近存计算与光互联协同机制开展更贴近真实部署环境的验证；二是进一步研究面向模型权重、KV Cache及过程数据的自适应分层管理与动态迁移机制；三是面向国产异构算力平台，探索体系结构、运行时系统与推理框架之间的软硬件协同优化，以提升该架构在自主可控智算基础设施中的工程落地能力。

参考文献

- [1] Yao S Y. The second half [EB/OL]. (2025-04-10) [2025-06-05]. <https://ysmyth.github.io/The-Second-Half/>
- [2] DeepSeek-AI, Liu A X, Feng B, et al. DeepSeek-V3 technical report [PP/OL]. arXiv(2025-02-18) [2026-06-11]. <https://arxiv.org/abs/2412.19437>
- [3] DeepSeek-AI, Xu A X, Lin B C, et al. Deepseek-V4: towards highly efficient million-token context intelligence [PP/OL]. arXiv(2026-04-

- 26)[2026-06-14]. <https://arxiv.org/abs/2606.19348>
- [4] OpenAI. Learning to reason with LLMs [EB/OL]. (2024-09-12) [2026-06-14]. <https://openai.com/index/learning-to-reason-with-llms/>
- [5] Park G, Lee S, Park Y. Minimizing response latency in LLM-based agent systems: a comprehensive survey [J]. IEEE access, 2026, 14: 26140-26168. DOI: 10.1109/ACCESS.2026.3664226
- [6] Yang A, Li A F, Yang B S, et al. Qwen3 technical report [PP/OL]. arXiv(2025-05-14)[2026-06-18]. <https://arxiv.org/abs/2505.09388>
- [7] Cheng X, Tian R, Zeng W D, et al. Conditional memory via scalable lookup: a new axis of sparsity for large language models [PP/OL]. arXiv(2026-01-12)[2026-06-06]. <https://arxiv.org/abs/2601.07372>
- [8] Behrouz A, Razaviyayn M, Zhong P L, et al. Nested learning: the illusion of deep learning architectures [PP/OL]. arXiv(2025-12-31) [2026-06-13]. <https://arxiv.org/abs/2512.24695>
- [9] Ganjihal S R. Predictive multi-tier memory management for KV cache in large-scale GPU inference [PP/OL]. arXiv(2026-04-19) [2026-06-15]. <https://arxiv.org/abs/2604.26968>
- [10] Kwon W, Li Z H, Zhuang S Y, et al. Efficient memory management for large language model serving with PagedAttention [C]//Proceedings of the 29th Symposium on Operating Systems Principles. ACM, 2023: 611-626. DOI: 10.1145/3600006.3613165

作者简介



熊先奎，中兴通讯股份有限公司智算首席架构师及前瞻团队负责人；长期从事计算系统与体系结构、先进计算范式及异构计算加速器研究；主导了中兴通讯ATCA高级电信计算平台、服务器存储平台、智能网卡及AI加速器的系统设计；发表论文10余篇，拥有专利数十项。



王程晨，中兴通讯股份有限公司工程师；从事先进计算存储领域的相关工作；发表论文10余篇，申请专利9篇。



蔡文豪，中兴通讯股份有限公司技术预研工程师；主要研究方向包括深度学习算法、大语言模型、具身智能、模拟计算、无线通信系统等；发表论文10余篇，拥有专利6篇。

新架构大模型推理的存算特征演变与近存计算适配技术研究



Evolution of Memory-Compute Characteristics in New-Architecture LLM Inference and Near-Memory Computing Adaptation Technologies

庄风云/Zhuang Fengyun, 孙菲/Sun Fei, 钱岭/Qian Ling

(中移(苏州)软件技术有限公司, 中国 苏州 215004)
(China Mobile (Suzhou) Software Technology Co., Ltd., Suzhou 215004, China)

DOI: 10.12142/ZTETJ.202604005

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260827.1355.002>

网络出版日期: 2026-08-27

收稿日期: 2026-05-18

摘要: 多头潜在注意力 (MLA) 与混合专家 (MoE) 正在重塑大模型推理的存算特征: MLA 通过低秩键值 (KV) 联合压缩, 使权重吸收模式下的注意力算术强度由传统多头注意力 (MHA) 解码的约 1 Op/B 提升至数百 Op/B; MoE 则因实际批量下的专家权重访问瓶颈, 成为近存计算的新核心场景。系统梳理解码阶段算术强度的解析模型以及 “MHA→分组查询注意力 (GQA)→MLA” 的存算特征演变规律, 对比分析 MLA 双执行模式的算术强度差异与现有存内处理 (PIM) 设备的算力密度约束, 按临界批量模型对 4 类主流 AI 加速器进行解析估算与横向研判, 并结合 FP8/INT4 等低精度计算特征, 提出面向 MLA+MoE 组合架构的分层近存计算适配方向并给出框架性建议。

关键词: 多头潜在注意力; 混合专家模型; 存算一体; 近存计算; 算术强度

Abstract: Multi-head latent attention (MLA) and mixture of experts (MoE) are reshaping the memory-compute characteristics of large language model (LLM) inference. Through low-rank joint compression of key-value (KV) representations, MLA raises the arithmetic intensity of attention under the weight-absorption mode from approximately 1 Op/B for conventional multi-head attention (MHA) decoding to several hundred Op/B, whereas MoE introduces an expert-weight memory-access bottleneck at practical batch sizes and becomes a new target for near-memory computing (NMC). This paper systematically reviews analytical models of arithmetic intensity in the decoding stage and investigates the evolution of memory-compute characteristics along the path of MHA → grouped-query attention (GQA) → MLA. It further compares the arithmetic-intensity characteristics of the two MLA execution modes and analyzes them in conjunction with the compute-density constraints of existing processing-in-memory (PIM) devices. Based on a critical-batch-size model, analytical estimations and cross-platform evaluations are conducted on four representative AI accelerators. Building upon these observations, the paper incorporates the characteristics of FP8/INT4 low-precision computation, proposes a hierarchical near-memory computing adaptation framework for combined MLA+MoE architectures, and provides corresponding architectural recommendations.

Keywords: multi-head latent attention; mixture of experts; processing-in-memory; near-memory computing; arithmetic intensity

引用格式: 庄风云, 孙菲, 钱岭. 新架构大模型推理的存算特征演变与近存计算适配技术研究 [J]. 中兴通讯技术, 2026, 32(4): 23-31. DOI: 10.12142/ZTETJ.202604005

Citation: Zhuang F Y, Sun F, Qian L. Evolution of memory-compute characteristics in new-architecture LLM inference and near-memory computing adaptation technologies [J]. ZTE technology journal, 2026, 32(4): 23-31. DOI: 10.12142/ZTETJ.202604005

大语言模型的大规模部署使推理效率问题日益凸显。冯·诺依曼架构中, 处理器与存储器间的数据搬运已成为性能瓶颈, 即 “存储墙” 问题。大模型自回归解码阶段的算术强度 (AI) 仅约 1 Op/B, 较主流图形处理器 (GPU) 数十至数百 Op/B 的脊点低 2 个数量级以上, 导致计算资源严重闲置^[1]。存内处理 (PIM) 技术通过将计算单元部署于存储器内部或近存储位置, 可将有效内存带宽提升数倍至数十倍, 被视为突破存储墙的关键路径; 三星高带宽存储器

(HBM) -PIM^[2]与 SK Hynix AiM 等均以注意力加速为核心目标场景。与此同时, 以 OpenAI o1/o3 和 DeepSeek-R1 为代表的推理时扩展 (Test-Time Scaling) 范式通过延长思维链显著拉长解码序列, 使本就访存受限的解码阶段在总推理开销中的占比进一步扩大, 存储墙矛盾愈发尖锐。

然而, 以 DeepSeek-V2 和 DeepSeek-V3 为代表的新一代大模型架构包含两项关键创新。其一, 多头潜在注意力 (MLA) 通过低秩 KV 联合压缩, 将键值缓存 (KV Cache)

规模压缩至同等参数规模多头注意力（MHA）的约 1/60，并在权重吸收执行模式下使注意力层 AI 由约 1 Op/B 提升至数百 Op/B（提升约 2 个数量级），达到甚至超越主流 GPU 脊点^[3-4]。其二，混合专家（MoE）通过稀疏条件激活实现参数规模扩展，但在实际推理批次下引入了新的专家权重访存瓶颈。现有以 MHA 为设计基准的 PIM 方案面临根本性的适配挑战。

现有工作对标准 Transformer 负载在 PIM 系统上的映射已进行较系统的研究^[2,5]，但针对 MLA 双执行模式、长上下文压缩稀疏注意力、低精度 MoE 推理以及多级异构存储组合代价的统一分析仍不充分。本文系统梳理解码阶段算术强度的解析模型及“MHA→分组查询注意力（GQA）→MLA”的存算特征演变规律；对比分析 MLA 双执行模式对 PIM 适配的差异化影响，并指出现有 PIM 设备的算力密度约束；归纳 MoE 专家层临界批量的估算方法，在 4 类主流 AI 加速器的公开规格参数下进行解析对比，分析其作为近存计算核心目标的可行性；最后，提出面向“MLA+MoE”组合架构的分层近存计算适配框架。

1 大模型推理存算特征基础

1.1 算术强度与 Roofline 模型

AI 定义为单位内存访问量对应的运算量。假设某个算子的总运算量为 W （单位：Op）、片外访存量为 Q （单位：B），则

$$AI = \frac{W}{Q} \quad (1)。$$

Roofline 模型^[1]是评估模型在特定硬件上潜在性能的有效理论框架：假设设备峰值算力为 π （Op/s）、峰值内存带宽为 β （B/s），则算子可达性能：

$$P_{\text{attain}} = \min(\pi, \beta \cdot AI) \quad (2)。$$

两段上界的交点即设备“脊点”：

$$RP = \frac{\pi}{\beta} \quad (3)。$$

算子在硬件设备上的执行过程如图 1 所示，数据从片外存储器（双倍数据速率存储器或 HBM）传输至片上缓冲区，再由片上处理单元完成计算，最终将结果写回内存。因此，性能评估需要同时考虑内存访问能力和处理单元的计算能力：当 $AI < RP$ （脊点）时，即算子的访存流量较大而计算操作数相对较少，则该算子为访存密集型，性能受带宽约束；当 $AI = RP$ 时处于性能拐点；当 $AI > RP$ 时则为计算密集

型，性能受算力约束。主流 AI 芯片的脊点参数如表 1 所示。

近几年，存储技术快速演进。HBM4 接口宽度由 1 024 位扩展至 2 048 位；2026 年公开商用产品已达 11.7 Gbit/s 稳定速率（最高 13 Gbit/s），单堆栈峰值带宽达 3.3 TB/s，12 层容量覆盖 24~36 GB，并通过低电压硅通孔（TSV）和供电网络优化提升能效^[6]。带宽代际跃升会降低同一精度下的 GPU 脊点（即进入计算密集区的门槛降低），并缓解访存压力，但 HBM 的容量、封装成本、逻辑底座面积及热设计约束仍然存在；同时，低精度 Tensor Core 算力的增长可重新抬高精度匹配的脊点。因此，“带宽增长即可消除存储墙”的论断并不成立，仍需结合算子精度和实际数据流分析。

1.2 低精度计算与精度感知 Roofline 修正

低精度量化会同时改变片外存储器访问字节总量与硬件峰值运算吞吐。设量化后权重、KV Cache、激活值的单元有效字节数分别为 p_w 、 p_{kv} 、 p_a ，对应元素数分别为 N_w 、 N_{kv} 和 N_a ，量化缩放因子、零点及对齐填充等引入的元数据访存流量为 Q_{meta} ；在精度 q 下，硬件峰值算力和脊点分别为 π_q 与 RP_q ，则总片外访存字节 Q_q 、算术强度 AI_q 、硬件脊点 RP_q 满足：

$$Q_q = p_w N_w + p_{kv} N_{kv} + p_a N_a + Q_{\text{meta}}, \quad AI_q = \frac{W}{Q_q}, \quad RP_q = \frac{\pi_q}{\beta} \quad (4)。$$

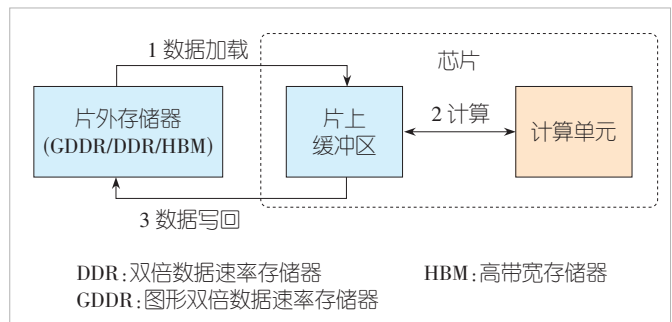


图1 算子执行过程

表1 主流AI加速器脊点

AI加速器	BF16/FP16理论算力/ TFLOPS	内存带宽/ (TB·s ⁻¹)	脊点/ (Op·B ⁻¹)
A100 SXM	312	2.00	156
H100 SXM	989	3.35	295
H20	148	4.00	37
Ascend 910B2	376	1.60	235

AI:算术强度 FP:浮点格式
BF:脑浮点格式 TFLOPS:每秒万亿次浮点运算

BF16/FP16、FP8/INT8 和 INT4/FP4 的理想单元数据载荷分别为 2 B、1 B 和 0.5 B。以 H100 SXM 为例，在不计结构化稀疏时，BF16 与 FP8 峰值算力分别约为 989 TFLOPS 和 1 979 TFLOPS；在 HBM 带宽 3.35 TB/s 条件下，硬件脊点由 BF16 下 295 Op/B 提升至 FP8 下 591 Op/B。当计算峰值与数据压缩比同步提高时，算术强度与硬件脊点近似同比右移，访存/计算属性未必改变。

Hopper 架构并不具备 INT4 原生矩阵计算能力，INT4 部署多采用仅权重量化方案，权重在计算单元内完成反量化后参与矩阵运算。TensorRT-LLM、NeMo 已提供 INT4-AWQ、FP8 等完整部署路径，但缩放因子读取、数据格式解包、离群值处理以及内核实现差异，都会削弱量化带来的理论性能收益。AMD MI300X 等主流 AI 加速器均原生支持 FP8 与 INT8 运算，低精度已经成为现代加速器的通用系统设计变量。

1.3 MHA 解码阶段的存算特征

标准 MHA 的自回归解码按照逐 Token 生成，假设缓存序列长度为 L 、注意力头数为 n_h 、头维度为 d_h 、模型层数为 l 、每元素字节数为 p ，则单请求每 Token 的 KV Cache 占用为：

$$m_{KV} = 2 \cdot l \cdot n_h \cdot d_h \cdot p \quad (5),$$

其中， n_{KV} 为 KV 头数，MHA 下 $n_{KV} = n_h$ ，FP16/BF16 精度下 $p=2$ ，FP8 下 $p=1$ ；总占用随批量 B 与序列长度线性增长。解码时每个请求的 KV Cache 为该请求私有，在不存在共享前缀缓存复用的情况下，批量维度难以摊薄 KV Cache 的读取开销；而前馈网络（FFN）权重为模型静态参数，可在批量样本间复用。每生成一个 Token，单层注意力核心运算（ QK^T 与 SV 操作）的运算量约为 $4n_h L d_h$ ，而须从显存读取的 KV 缓存约为 $2n_h L d_h \cdot p$ ，因此 MHA 的算术强度为：

$$AI_{MHA} \approx \frac{4n_h L d_h}{2n_h L d_h \cdot p} = \frac{2}{p} \quad (6).$$

也就是说，FP16/BF16 精度下 $AI_{MHA} \approx 1 \text{ Op/B}$ ，且与批量和上下文长度基本无关，与主流 AI 芯片脊点相差 2 个数量级以上，计算单元绝大部分时间在等待数据。这是三星 HBM-PIM^[5]、SK Hynix AiM 以及 LoL-PIM^[7] 等系统选择以注意力加速为核心设计目标的根本动机。

1.4 GQA 的改善与局限

分组查询注意力（GQA）^[8] 令每 G 个查询（Query）头共享同一组 KV 头，即 $n_{KV} = n_h / G$ ，KV Cache 规模降至 MHA 的 $1/G$ 。由于运算量不变而访存量按 G 倍缩减，解码的算术强

度相应提升为：

$$AI_{GQA} \approx \frac{2G}{p} \quad (7).$$

当 $G=8$ 且处于 FP16 精度时， $AI_{GQA} \approx 8 \text{ Op/B}$ 。以 LLaMA-3-70B（ $G=8$ ）为例，FP16 精度下每 Token 的 KV 存储量由 MHA 的约 2.6 MB 降至约 320 KB，但 AI 仍处于访存密集区，尚未突破存储墙约束的本质。

2 MLA 存算特征深度分析

2.1 MLA 架构原理与 KV Cache 压缩效果

MLA 由 DeepSeek-V2^[3] 首次提出，核心思想是通过下投影矩阵 W_{DKV} 将所有注意力头的 KV 联合压缩为低维潜变量 C_{KV} ，压缩维度 $d_c \ll n_h \cdot d_h$ ，推理时仅缓存 C_{KV} 及解耦的旋转位置编码分量（维度为 d_r ），按需通过上投影矩阵 W_{UK} 、 W_{UV} 还原完整 KV 矩阵。其每 Token 缓存占用为：

$$m_{MLA} = l \cdot (d_c + d_r) \cdot p \quad (8).$$

以 DeepSeek-V3 为例，其 $l=61$ ， $d_c=512$ ， $d_r=64$ ，在 BF16 精度下，每 Token 的 KV Cache 仅约 70 KB^[9]；按 DeepSeek-V3 等效配置估算（ $l=61$ ， $n_h=128$ ， $d_h=128$ ），MHA 每 Token 的 KV Cache 约为 4 MB，压缩比高达约 60 倍；采用 GQA（ $G=8$ ），每 Token 的 KV Cache 约为 500 KB，压缩比约 7 倍。表 2 从 KV Cache 规模与算术强度角度对各注意力机制进行横向比较。

2.2 MLA 双执行模式的算术强度量化

MLA 存在两种代数等价的执行路径，具有截然不同的存算特性，如图 2 所示。

Absorb（吸收）模式：利用矩阵结合律将上投影矩阵 W_{UK} 、 W_{UV} 分别与查询投影、输出投影预先融合（“矩阵吸收”），推理时直接在压缩潜变量空间内执行注意力运算，无须展开完整 KV。此时全部 n_h 个查询头共享读取同一份潜变量缓存 C_{KV} ，且该缓存同时承担 K 与 V 的双重角色，单层

表 2 不同注意力机制 KV Cache 规模与算术强度对比

注意力机制	单位 Token KV Cache	解码 AI/ (Op·B ⁻¹)	近存计算适用性
MHA	约 4 MB	约 1	高
GQA($G=8$)	约 500 KB	约 8	较高
MLA(Naive 模式)	约 70 KB	1 ~ 8	中
MLA(Absorb 模式)	约 70 KB	256 ~ 512	低

AI: 算术强度 MHA: 多头注意力
GQA: 分组查询注意力 MLA: 多头潜在注意力

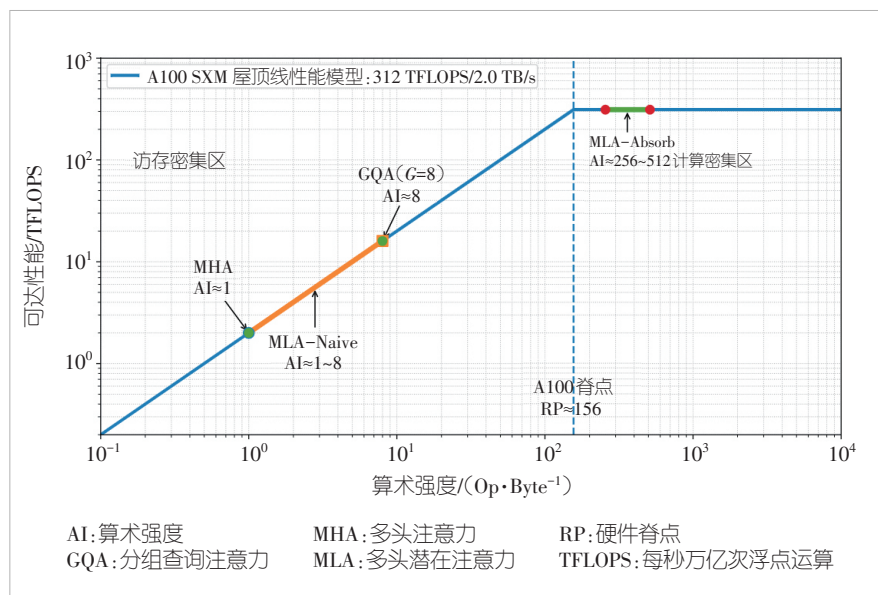


图2 不同注意力机制在Roofline模型中的算术强度分布 (A100 SXM)

核心运算量约为 $4n_h L_d c$ ，而潜变量缓存读取量仅为 $L_d p$ ，故：

$$AI_{\text{Absorb}} \approx \frac{4n_h L_d c}{L_d \cdot p} = \frac{4n_h}{p} \quad (9)$$

以 DeepSeek-V3 ($n_h=128$) 代入，在 BF16 精度下， $AI_{\text{Absorb}} \approx 256 \text{ Op/B}$ ，FP8 精度下约为 512 Op/B ，已达到或超越 A100 乃至 H100 的脊点。注意力层由此脱离深度访存密集区，进入计算密集区域或其边缘^[4]。

Naive (原始) 模式：运行时从 C_{KV} 在线展开完整 KV 后执行标准注意力运算。其访存特征近似于 GQA，小批量 ($B < 8$) 场景下 AI 处于约 $1 \sim 8 \text{ Op/B}$ 区间，呈访存密集特性，且随批量增大 AI 逐渐升高。

需要特别指出的是，Absorb 模式并非所有场景下的默认选项，在实际部署中两种模式常混合使用。TyphoonMLA^[10] 的工程实践表明，对于共享前缀 (Shared Prefix) 段，已有历史 Token 的 KV 可直接从 KV Cache 复用，此段采用 Naive 模式无须重新展开上投影矩阵，仅对新增 Token 段切换至 Absorb 模式以获得更高吞吐。此外，受出口管制影响，部分场景只能使用 H20 等低算力高带宽卡，其 MLA-Absorb 模式的绝对吞吐上限受算力瓶颈制约，而 Naive 模式配合高内存带宽的组合可能更具竞争力。因此，模式选择本质上是一个依赖硬件算力/带宽比的动态决策。

2.3 MLA 对近存计算设计的新挑战

MLA-Absorb 模式使注意力层脱离访存密集区，从根本

上动摇了 PIM 加速注意力的传统价值主张。与此同时，MLA 还带来 4 类新的硬件设计挑战：

1) PIM 算力密度制约 Absorb 模式执行：现有三星 HBM-PIM 和 SK Hynix AiM 等存内计算设备，其片内算力密度通常为 $1 \sim 10 \text{ TFLOPS}$ 量级，而 MLA-Absorb 模式要求执行规模较大的矩阵乘，例如，其在 A100 上饱和运行对应的算力需求约 312 TFLOPS ，与 PIM 片内算力相差约 2 个数量级。即使内存带宽充裕，PIM 设备也无法充分驱动 Absorb 模式所需的计算量。PIM 对 MLA-Absorb 注意力的加速价值从根本上受到算力约束，而非带宽约束。

2) 张量并行 (TP) 分片瓶颈： C_{KV} 为单个共享潜变量头，无法按头维度进行

TP 切分。多卡推理时每张卡必须冗余加载完整 KV Cache，且带宽开销随设备数量线性累积^[11]。

3) 混合执行的灵活性需求：在实际部署中，可能采用灵活的混合执行模式。例如，共享前缀段采用 Naive 模式复用已缓存 KV，新增词元段采用 Absorb 模式以获更高吞吐。混合执行模式下，近存硬件需支持两种模式的动态切换，对可编程性提出更高要求。

4) 新变种持续演进：分组查询潜在注意力 (GQLA)^[12] 将 MLA 与 GQA 融合，同一套训练权重可同时支持 MQA-Absorb 和 GQA 展开两类解码路径，分别适配 H100 等高计算/带宽比硬件和 H20 等低算力、高带宽硬件；DeepSeek-V3.2 则引入 DSA 稀疏注意力机制^[13]，通过设计闪电索引器模块和细粒度 Token 选择机制，将核心注意力复杂度由 $O(L^2)$ 降至 $O(Lk) (k \ll L)$ 。这些进展表明，注意力机制正沿“潜空间化压缩”和“细粒度稀疏化”方向演进，并逐步走向由硬件算力带宽比、上下文长度和推理批量共同决定执行路径的硬件自适应形态。

3 MoE 存算特征分析

3.1 MoE 推理机制与内存规模

MoE 通过稀疏条件激活实现参数规模扩展，每个输入 Token 经门控网络选择 Top-K 个专家子网络参与计算，其余专家权重在当前步骤不被访问。

以 DeepSeek-V3 为例，其总参数量为 6 710 亿，每 Token 激活参数约 370 亿；除前 3 个 dense FFN 层外，每个 MoE 层包

含1个共享专家与256个路由专家，每Token激活8个路由专家。虽然MoE降低了单Token计算量，但低时延推理通常仍要求专家权重常驻或快速可达，在BF16精度下约占1.3 TB，远超当前GPU单机HBM容量上限（例如单机8卡H100 80G总容量640 GB）^[9]，形成超出单机容量的结构性挑战。

3.2 MoE 专家访问的访存特征

1) 长尾分布规律：专家访问频率高度不均匀，遵循Zipf型长尾分布——约前20%的“热专家”承担约80%的激活请求，其余为低频“冷专家”^[14]。这一规律为热冷分层的近存计算策略提供了量化依据。

2) 批量大小与存算特性的动态关系：MoE专家层本质上是FFN，其专家权重可在多个Token间复用，因此算术强度随有效批量增大而提升。设总专家数为 n_e 、每Token激活专家数为 n_k ，批量 B 下平均每个专家处理 Bn_k/n_e 个Token；专家权重读取量恒定而有效运算量随路由Token数线性增长，故专家层算术强度近似为：

$$AI_{MoE} \approx \frac{2Bn_k}{p \cdot n_e} \quad (10)。$$

令 $AI_{MoE} = RP_{acc}$ （AI加速器的脊点），即可得使MoE专家层由访存密集转为计算密集的临界有效批量：

$$B_{th} = \frac{p}{2} \cdot RP_{acc} \cdot \frac{n_e}{n_k} \quad (11)。$$

在BF16/FP16精度下 $p=2$ ，公式（11）可简化为 $B_{th} = RP_{acc} \cdot \frac{n_e}{n_k}$ 。以A100（ $RP \approx 156$ Op/B）和DeepSeek-V3（ $n_e=256$ ， $n_k=8$ ， $n_e/n_k=32$ ）代入，则 $B_{th} \approx 4\ 896$ 。这意味着在绝大多数实际推理批量（ $B \leq 512$ ）下，MoE专家层均持续处于访存密集状态，是近存计算的天然目标。

表3基于各AI加速器的公开规格，对A100、H100、H20及Ascend 910B2共4种主流AI加速器的 B_{th} 进行解析估算，其 B_{th} 均远高于典型推理批量上限。因此，MoE专家层的访存密集特征并不会因加速器差异而消失，但其程度会随

硬件Roofline脊点、批量大小和路由分布而变化。

3.3 MLA+MoE 组合架构的存算解析

DeepSeek-V2/V3将MLA与MoE结合，形成异构的存算特征分布。在MLA-Absorb执行模式下，注意力层的算术强度显著提升，由传统意义上的访存密集逐步转向脊点附近或计算密集区域，对PIM的带宽缓解需求降低；与此同时，当前PIM架构受限于算力密度、数据复用能力和矩阵计算效率，也难以高效承载高算术强度矩阵乘。

相比之下，MoE专家层在典型批量下仍需频繁读取大规模专家权重，权重复用不足，持续呈现访存密集特征，因而成为新的近存计算核心目标。由此可见，大模型推理的存算瓶颈重心正在从传统注意力层向MoE专家层发生结构性迁移。

4 面向MLA/MoE的近存计算适配方向

4.1 现有PIM方案的局限性

当前主流PIM方案多以MHA/GQA解码访存瓶颈为设计基准（其核心负载特征为KV Cache密集访问），面临双重局限：一是MLA-Absorb等高算术强度执行模式的出现，使原有算术强度匹配关系不再成立；二是现有PIM计算单元的算力密度难以匹配GPU Tensor Core/AI Core上大规模通用矩阵乘法（GEMM）的计算需求。

例如，三星HBM-PIM和SK Hynix AiM分别在动态随机存取存储器（DRAM）存储体附近集成面向通用矩阵-向量乘法（GEMV）的乘加运算（MAC）单元或者AI引擎，其单器件或单芯片级算力通常处于约1 TFLOPS量级，而A100的BF16/FP16 Tensor Core稠密峰值为312 TFLOPS，二者相差约2.5个数量级。因此，传统PIM更适合MHA/GQA解码中的低算术强度KV读取与GEMV类操作，而难以直接高效承载MLA-Absorb类高算术强度GEMM负载。

面向长上下文MHA解码，LoL-PIM^[15]与L3^[16]分别探索了可扩展的DRAM-PIM系统与双列直插式内存模块

表3 主流AI加速器MoE专家层临界批量(基于DeepSeek-V3)

AI加速器	理论算力/TFLOPS	内存带宽/(TB·s ⁻¹)	脊点/(Op·B ⁻¹)	B_{th}	典型批量	MoE专家层状态
A100 SXM	312	2.0	156	4 896	≤512	访存密集
H100 SXM	989	3.35	295	9 440	≤512	访存密集
H20	148	4.0	37	1 184	≤512	访存密集
Ascend 910B2	376	1.6	235	7 520	≤512	访存密集

AI：算术强度 MoE：混合专家 TFLOPS：每秒万亿次浮点运算

(DIMM) PIM/GPU 协同架构; STARC^[9]进一步通过语义聚类将 KV 对映射到与 PIM Bank 结构对齐的连续存储区域, 在 KV Cache 规模为 1 024 词元时, 相比完整 KV 检索实现 54%~74% 的注意力层延迟降低和 45%~67% 的能耗降低。然而, 这些方案总体仍基于标准 K/V Cache 数据结构, 尚未针对 MLA 的低维潜变量缓存结构, 在布局、聚类与执行路径层面实现联合优化。

值得注意的是, SK Hynix 于 2025 年展示 AiMX 加速卡与 vLLM 框架协同的注意力卸载方案, 将访存受限的注意力计算卸载至 AiMX, 而将计算密集部分保留在 GPU, 印证了按算子存算特征进行异构分工的方向, 但其底层计算形态仍主要面向 GEMV/MAC, 对 MLA-Absorb 类 GEMM 负载的支撑能力仍受限于其 MAC/GEMV 导向的计算架构。综上, 现有方案的设计基准 (MHA/GQA) 与新架构 (MLA/MoE) 的负载特征之间仍存在不匹配, 构成共同挑战。

4.2 面向 MLA 的近存适配方向

面向 MLA 的近存适配思路需要从可重构计算、潜变量感知布局和多设备缓存协同 3 个层面进行重新设计。

1) 可重构计算单元支持双模式动态切换: MLA-Naive 模式在小批量解码下仍属访存密集型, 适合利用现有 HBM-PIM 的 GEMV/MAC 单元加速; 而 MLA-Absorb 模式算术强度显著提升, 更接近 GEMM 型计算, 对近存侧算力密度提出更高要求。因此, 可在 HBM 逻辑层或 3D 堆叠逻辑 Chiplet 中引入可重构矩阵计算单元, 支持 GEMV 与 GEMM 近存原语的动态切换。

2) 潜变量感知的 KV 存储布局优化: 传统 PIM 以完整注意力头为粒度分配 DRAM Bank, 而 MLA 的 C_{KV} 维度 d_c 远小于 $n_h \times d_h$, 建议以 d_c 为粒度重新设计 Bank 对齐策略, 使 C_{KV} 在物理上连续存储, 消除跨 Bank 冲突; 同时, 低维潜变量空间的语义聚类质量优于原始高维 KV 空间, STARC 类聚类方案在 MLA 布局下有望获得更高的预取命中率。

3) 多设备广播缓存缓解 TP 分片瓶颈: 针对 C_{KV} 单头结构的 TP 分片瓶颈, 可在近存层设置跨设备共享广播缓冲, 令多张加速卡共享同一 C_{KV} 副本, 避免各设备独立从 HBM 重复加载。在广播缓存命中率较高、跨设备访问延迟可控的理想条件下, 多卡 KV 带宽开销可由 $O(N_{\text{device}})$ 降至 $O(1)$ ^[11]。

4.3 面向 MoE 的近存计算适配方向

在典型在线推理场景下, 单个专家接收的有效 Token 数有限, 专家权重读取难以被批量充分摊销, 使 MoE 专家层持续呈现访存密集特征。因此, 面向 MoE 的近存计算适配

应围绕专家热冷分层、近存并行调度和路由感知预取 3 个方向展开。

1) 热冷专家分层近存计算: MoNDE^[14]提出将少量高频热专家常驻 GPU HBM, 将大量低频冷专家保留在宿主 DRAM 或扩展内存侧 (如计算快速互连 (CXL) 内存池), 由近数据处理 (NDP) 单元在近数据侧就地执行专家计算。该方法以规模较小的激活数据传输替代大规模冷专家参数在高速串行计算机扩展总线 (PCIe) 等链路上的往返搬运, 从而降低专家卸载开销。在典型批量配置下, 实测相对于 GPU 卸载框架, 编码器推理提速最高 7.5 倍, 解码器提速最高 3.7 倍, NDP 单元面积开销仅 3.0 mm²。

2) 3D 近存处理混合并行调度: HD-MoE^[17]通过混合键合将逻辑计算层与 DRAM 堆叠紧密集成; 设计了离线混合并行映射与在线动态调度策略, 融合张量并行 (TP) 与专家并行 (EP), 以应对专家激活的实时不均匀性。在典型 MoE 模型配置下, 实测相比纯 TP 方案提速 1.1~1.8 倍, 相比纯 EP 提速 1.1~1.5 倍。

3) 路由感知预取策略: 利用 MoE 路由器输出的 Top-K 概率分布预判下一步热专家集合, 提前将冷专家数据从主存预取至近存缓冲区。可与投机解码结合, 以草稿模型预测词元序列, 从而将专家预取窗口从 1 步延伸至多步, 提升命中率。

4.4 MLA+MoE 组合架构的分层近存框架

综合以上分析, 图 3 给出了面向 MLA+MoE 组合架构的分层近存计算框架建议。框架自上而下分为 4 层: GPU 核心层承担 MLA-Absorb 路径下的高算术强度 GEMM 及大批量 MoE 专家计算, 发挥 Tensor Core/AI Core 的高算力密度优势; HBM 逻辑近存层承担 MLA-Naive 小批量注意力的 GEMV 加速、 C_{KV} 跨卡广播缓存, 以及热专家预取缓冲, 并支持 Naive/Absorb 双模式的动态切换; 3D 堆叠 DRAM 近存层执行冷专家 NDP 就地计算、KV Cache Token 淘汰评分, 以及路由感知预取; 主机 DRAM/CXL 扩展层负责冷专家权重的大规模存储及长上下文 KV Cache 溢出扩展, 并可结合 DIMM-PIM 等协同架构实现近存侧 Token 选择、缓存压缩与预取管理。

上述 4 层存储体系并非简单叠加即可获得性能收益, 其实际部署还伴随着相应的硬件代价与软件开销。

1) 硬件集成代价: HBM 逻辑层集成矩阵单元将占用逻辑底座面积与功耗预算, 并与物理层、可靠性/可用性/可维护性 (RAS) 机制及内存控制器形成资源竞争; DRAM 工艺对高频逻辑的支持有限, 使得“带宽充裕而算力密度不足”的矛盾依然存在。3D 混合键合虽能提供更高的垂直带宽,

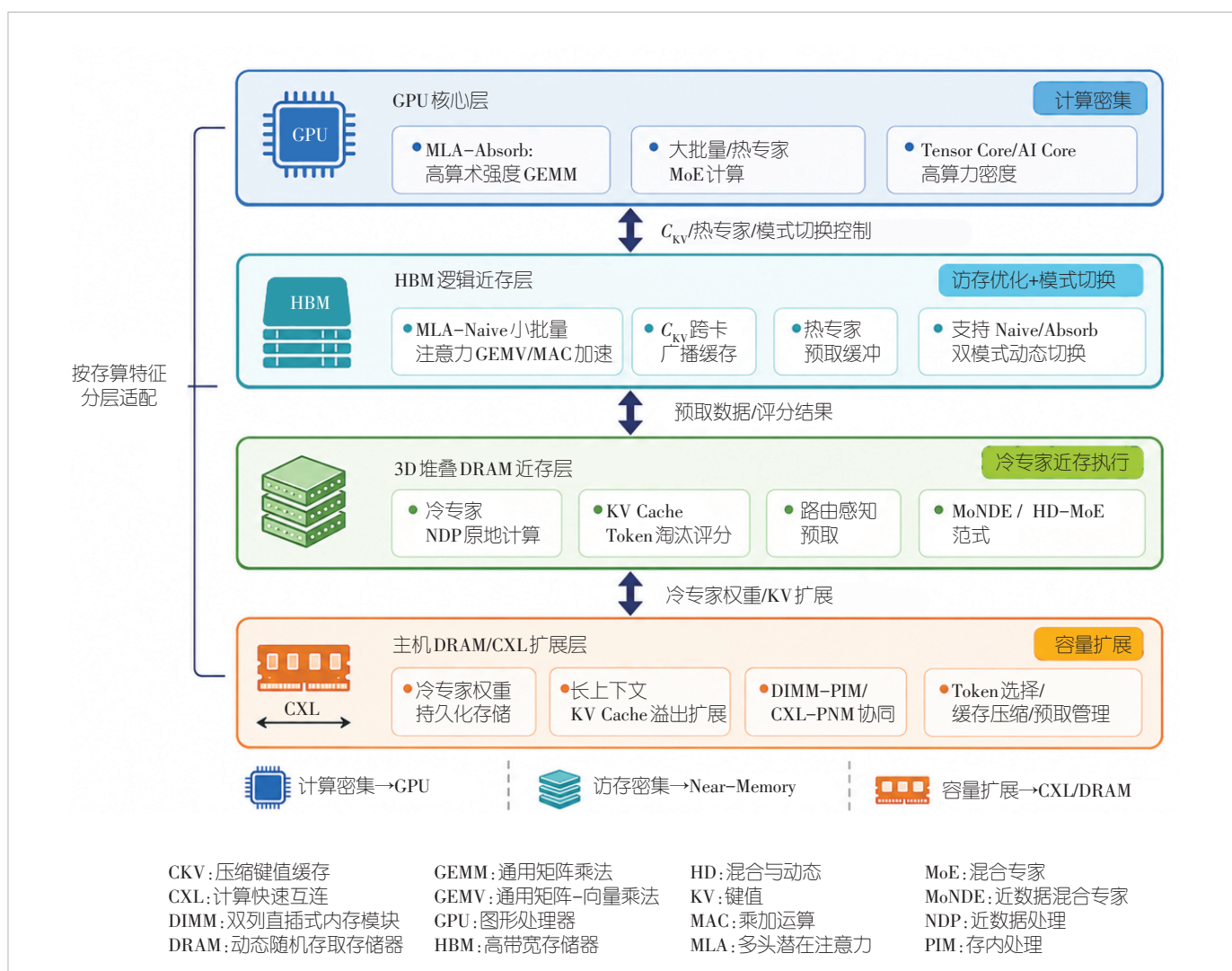


图3 面向MLA+MoE组合架构的分层近存计算框架

但也引入了堆叠良率、热密度、供电、测试及封装成本等多重工程挑战。

2) 跨介质带宽与延迟落差：HBM、3D DRAM、主机 DDR/CXL在带宽、粒度和访问时延上相差显著，若按Token细粒度同步访问慢层，协议开销与往返延迟可能抵消近存计算带来的性能收益。CXL提供一致性与内存语义，可支持内存池化和跨节点共享^[18]，但其链路带宽远低于HBM。对于KV Cache等难以压缩的数据，跨CXL传输时的数据压缩还会受到布局、编解码效率及有效带宽的多重限制，因此需与预测搬移、异步流水及访问热度分层等策略进行联合设计。

3) 软件对象化与调度开销：运行时需同时维护专家块和共享前缀等不同对象；每类对象的生命周期、页粒度、量化格式及一致性域各不相同。MoE路由还存在动态不均衡

性，预取过早会占用近存容量，预取过晚则暴露CXL/PCIe延迟。HD-MoE和ELMoE-3D的结果表明，离线映射必须与在线动态调度、缓存及投机执行进行联合设计。

4) 低精度与编译开销：需处理异常值、不同层精度、累加精度和跨层格式转换。编译器必须联合选择数据布局、量化分组、GEMV/GEMM内核与放置层级，否则反量化、转置和Padding流量会抵消带宽节省。

5 总结与展望

本文围绕大模型推理阶段的存算特征演变，系统分析了MLA与MoE架构对近存计算适配的影响。通过对传统MHA、GQA、MLA-Naive、MLA-Absorb以及MoE专家层的算术强度建模可以看出，新一代大模型推理已不再呈现单一的访存瓶颈，而是形成了注意力层、专家层与硬件算力带宽比共同

塑造的异构存算特征分布。

MLA-Absorb 模式通过权重吸收显著提高注意力层算术强度,使其由传统 MHA 解码阶段约 1 Op/B 的强访存密集区域,提升至数百 Op/B 量级,并达到甚至跨越主流 AI 加速器的 Roofline 脊点。这意味着,传统面向 MHA KV Cache 读取瓶颈设计的 PIM 注意力加速器,在 MLA-Absorb 场景下面临算术强度匹配失效与算力密度不足的双重挑战。相比之下,MLA-Naive 模式在小批量、长上下文场景下仍具有较强访存密集特征,因而仍存在近存加速空间。这表明,未来面向 MLA 的近存计算架构需要从以 GEMV 为核心的注意力加速,转向同时支持 Naive/Absorb 双路径的可重构计算体系。

MoE 架构虽然通过稀疏激活降低了单 Token 计算量,但其大规模专家权重需要常驻或快速可达,且在线解码场景下有效批量通常难以充分摊薄专家权重读取开销。解析估算表明,在 A100、H100 等典型 AI 加速器上,MoE 专家层由访存密集转向计算密集所需的临界批量普遍高于常见在线推理批量。因此,MoE 专家层在实际部署中仍持续表现为访存主导,并逐步成为近存计算新的核心优化对象。热冷专家分层、路由感知预取、3D 近存处理以及 TP/EP 混合并行调度等方法,为缓解专家权重搬移瓶颈提供了重要方向。

从发展趋势看,近存计算架构需要从传统注意力加速器逐步演进为面向 MLA 与 MoE 的异构支撑体系。一方面,应面向 MLA 低维潜变量缓存特征,设计潜变量感知的 Bank 对齐、连续存储、聚类预取和跨设备广播缓存机制;另一方面,应面向 MoE 专家访问的长尾分布,构建热专家 HBM 常驻、冷专家近存执行、路由预测预取,以及与 CXL/DRAM 扩展层协同的分层存储与计算体系。随着 3D/3.5D Chiplet、HBM 逻辑层计算、CXL 内存扩展以及 RRAM-CIM^[19]等新型存储与封装技术的发展,近存计算有望在 MoE 冷专家存储、长上下文 KV 管理和低频专家原地计算等场景中发挥更大价值。

以 2026 年上半年发布的 DeepSeek-V4^[20]为代表,新一代 MoE 模型进一步融合压缩稀疏注意力(CSA)、高度压缩注意力(HCA)与多词元预测(MTP)机制,将百万级超长上下文变为推理标配。该类架构并非简单消除 KV 缓存压力,而是将瓶颈进一步分化:长序列下原始 KV 被压缩块替代,有效降低原生 KV 容量,但引入压缩索引、块元数据、稀疏检索计算等新开销;同时万亿级总参 MoE 模型的专家集合进一步扩张,FP4 混合量化成为权重部署主流,Group-wise 缩放因子及路由索引的元数据访存占比持续抬升。这一演变预示存算架构将面临三重新的适配需求:一是近存单元不仅要处理原始 KV,还需要支持压缩 KV 块的原位检索、块筛选

与索引遍历,硬件需兼容稠密潜变量与稀疏压缩块两套缓存访问范式;二是针对 FP4 等超低位宽权重量化,近存通路需要原生兼容 Scale 元数据并行加载、快速解包,缓解元数据流量对理论带宽收益的侵蚀;三是 MTP 多令牌预测会改变单步 Token 的算子调用模式,专家层与注意力层的计算访存交织关系进一步复杂化,固定流水线的近存硬件容易出现利用率下降。与此同时,细粒度专家划分、共享专家复用等模型层优化,会改变专家访问的热-冷分布,热专家集合的动态漂移现象,对近存侧的专家预取及缓存置换策略提出更高要求。

与此同时,该方向仍面临软件栈与编译器支持不足、低精度量化误差控制、MoE 路由预测不确定性,以及压缩注意力索引检索的硬件加速缺失等问题。总体而言,大模型推理的存算瓶颈正在从传统 MHA 注意力层,迁移至 MLA 潜变量缓存、压缩式长上下文 KV 块访问与 MoE 专家权重访问共同主导的新格局。未来近存计算不应继续沿用单一的 MHA-PIM 设计范式,而应围绕“按算子存算特征分层适配”的原则,形成可重构矩阵计算、潜变量与压缩块双模式感知的存储布局、热冷专家分层执行、路由感知预取以及多设备缓存协同的一体化架构,适配新一代融合压缩注意力、MTP 与超低位宽 MoE 模型的推理负载。

参考文献

- [1] Yuan Z H, Shang Y Z, Zhou Y, et al. LLM inference unveiled: survey and roofline model insights [PP/OL]. arXiv (2024-02-26) [2026-05-25]. <https://arxiv.org/abs/2402.16363>
- [2] Lee S, Kang S H, Lee J, et al. Hardware architecture and software stack for PIM based on commercial DRAM technology: industrial product [C]//Proceedings of ACM/IEEE 48th Annual International Symposium on Computer Architecture (ISCA). IEEE, 2021: 43-56. DOI: 10.1109/ISCA52012.2021.00013
- [3] DeepSeek-AI, Liu A X, Feng B, et al. DeepSeek-V2: a strong, economical, and efficient mixture-of-experts language model [PP/OL]. arXiv (2024-05-07) [2026-05-25]. <https://arxiv.org/abs/2405.04434>
- [4] Geens R, Verhelst M. Hardware-centric analysis of DeepSeek's multi-head latent attention [J]. Electronics letters, 2025, 61: e70504. DOI: 10.1049/ell2.70504
- [5] Fan Z H, Gagnon G, Liu Z Y, et al. Sparse attention remapping with clustering for efficient LLM decoding on PIM [PP/OL]. arXiv (2025-05-09) [2026-05-25]. <https://arxiv.org/abs/2505.05772>
- [6] Kim E A. Samsung ships industry-first commercial HBM4 with ultimate performance for AI computing: Samsung electronics [EB/OL]. (2026-02-10) [2026-05-25]. <https://news.samsung.com/global/samsung-ships-industry-first-commercial-hbm4-with-ultimate-performance-for-ai-computing>
- [7] Kwon H, Koo K, Kim J, et al. PIMphony: overcoming bandwidth and capacity inefficiency in PIM-based long-context LLM inference system [C]//Proceedings of IEEE International

- Symposium on High Performance Computer Architecture (HPCA). IEEE, 2026: 1–21. DOI: 10.1109/HPCA68181.2026.11408592
- [8] Ainslie J, Lee-Thorp J, de Jong M, et al. GQA: training generalized multi-query transformer models from multi-head checkpoints [C]//Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: ACL, 2023: 4895–4901. DOI: 10.18653/v1/2023.emnlp-main.298
- [9] DeepSeek-AI, Liu A X, Feng B, et al. DeepSeek-V3 technical report [PP/OL]. arXiv (2024–12–27) [2026–05–25]. <https://arxiv.org/abs/2412.19437>
- [10] Yüzügüler A C, Çelik A, Zhuang J W, et al. TyphoonMLA: a mixed naïve-absorb MLA kernel for shared prefix [PP/OL]. arXiv (2025–09–25)[2026–05–25]. <https://arxiv.org/abs/2509.21081>
- [11] Zhao C G, Deng C Q, Ruan C, et al. Insights into DeepSeek-V3: scaling challenges and reflections on hardware for AI architectures [C]//Proceedings of the 52nd Annual International Symposium on Computer Architecture. ACM, 2025: 1731–1745. DOI: 10.1145/3695053.3731412
- [12] Meng F X. GQLA: group-query latent attention for hardware-adaptive large language model decoding [PP/OL]. arXiv (2026–05–13)[2026–05–25]. <https://arxiv.org/abs/2605.15250>
- [13] DeepSeek-AI, Liu A X, Mei A X, et al. DeepSeek-V3.2: pushing the frontier of open large language models [PP/OL]. arXiv (2025–12–02)[2026–05–25]. <https://arxiv.org/abs/2512.02556>
- [14] Kim T, Choi K, Cho Y, et al. MoNDE: mixture of near-data experts for large-scale sparse models [C]//Proceedings of the 61st ACM/IEEE Design Automation Conference. ACM, 2024: 1–6. DOI: 10.1145/3649329.3655951
- [15] Kwon H, Koo K, Kim J, et al. Lol-pim: long-context LLM decoding with scalable DRAM-PIM system [PP/OL]. arXiv (2024–12–28)[2026–05–25]. <https://arxiv.org/abs/2412.20166>
- [16] Liu Q, Chen L, Yang Y, et al. L3: DIMM-PIM integrated architecture and coordination for scalable long-context LLM inference [PP/OL]. arXiv (2025–04–24) [2026–05–25]. <https://arxiv.org/abs/2504.17584>
- [17] Huang H C, Zhong S Z, Zhang Z, et al. HD-MoE: hybrid and dynamic parallelism for mixture-of-expert LLMs with 3D near-memory processing [C]//Proceedings of IEEE/ACM International Conference on Computer Aided Design (ICCAD). IEEE, 2025: 1–9. DOI: 10.1109/ICCAD66269.2025.11240984
- [18] Natu M. Introducing the CXL 3. X specification [EB/OL]. [2026–05–25]. [https://computeexpresslink.org/wp-content/uploads/](https://computeexpresslink.org/wp-content/uploads/2025/02/CXL_Q1-2025-Webinar-Presentation_FINAL.pdf)

- 2025/02/CXL_Q1-2025-Webinar-Presentation_FINAL.pdf
- [19] 何斯琪, 穆琛, 陈迟晓. 基于存算一体集成芯片的大语言模型专用硬件架构 [J]. 中兴通讯技术, 2024, 30(2): 37–42. DOI: 10.12142/ZTETJ.202402006
- [20] DeepSeek-AI, Xu A Y, Lin B C, et al. DeepSeek-V4: towards highly efficient million-token context intelligence [PP/OL]. arXiv (2026–04–26)[2026–05–25]. <https://arxiv.org/abs/2606.19348>

作者简介



庄凤云, 中移(苏州)软件技术有限公司工程师; 主要研究方向为大模型训推加速及智能体技术。



孙菲, 中移(苏州)软件技术有限公司高级工程师; 主要研究方向为AI推理加速。



钱岭, 中移(苏州)软件技术有限公司副总工程师, 教授级高工, 国务院政府特殊津贴获得者, 电信行业“大国工匠”; 主要研究方向为量子计算应用算法、量子计算云服务创新、量子计算与传统计算的异构算力融合、多样化量子算力并网等; 主持并负责国资委未来产业量子计算任务、国家自然科学基金、中国移动集团重点研发等多项课题; 曾获中国通信学会自然科学奖一等奖、北京市自然科学奖二等奖; 发表论文20余篇, 获授权发明专利10余项。

面向大模型推理的存算协同理论综述



A Survey on Compute-Storage Co-Optimization Theory for Large Model Inference

汪俊峰/Wang Junfeng¹, 刘璇/Liu Xuan¹,
李亚奇/Li Yaqi^{1,2}, 石亮/Shi Liang^{1,2,3}

(1. 华东师范大学, 中国 上海 200062;

2. 上海创智学院, 中国 上海 200233;

3. 华中科技大学武汉光电国家研究中心, 中国 武汉 430074)

(1. East China Normal University, Shanghai 200062, China;

2. Shanghai Innovation Institute, Shanghai 200233, China;

3. Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan 430074, China)

DOI: 10.12142/ZTETJ.202604006

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260907.1604.002>

网络出版日期: 2026-08-28

收稿日期: 2026-05-16

摘要: 大模型推理在长上下文、高并发与多轮复用条件下, 受算力、存储容量与跨层数据移动共同约束。量化、缓存、卸载与阶段解耦常组合部署, 但压缩率、命中率、显存占用下降等局部指标改善, 并不等价于端到端关键路径缩短。从权重、键值 (KV) 缓存与激活的生命周期出发, 以关键路径净收益为判据, 系统梳理以算换存、以存换算与存算路径3类方法, 并归纳其适用条件、附加开销、质量约束与失效边界, 为面向负载的存算协同部署提供可复用的分析视角。认为未来应从生命周期控制、压缩表征、跨引擎接口与真实负载评测4个方面开展存算协同相关研究。

关键词: 大模型推理; 存算协同; 数据移动; KV缓存; 资源置换; 瓶颈迁移

Abstract: Under long context, high concurrency, and multi-turn reuse, large model inference is constrained by computation, storage capacity, and cross-tier data movement. Quantization, caching, offloading, and stage disaggregation are often combined, but improvements in compression ratio, hit rate, or memory footprint do not imply a shorter end-to-end critical path. Starting from the lifetimes of weights, key-value (KV) caches, and activations, and using critical-path net benefit as the evaluation criterion, this paper systematically reviews three categories of methods: computation for storage, storage for computation, and storage-computation pathways. It further summarizes their applicability, overheads, quality constraints, and failure modes, and provides a reusable analytical lens for workload-oriented compute-storage co-optimization. This paper holds that future research related to compute-storage co-optimization should be conducted from four aspects: lifecycle control, compressed representation, cross-engine interfaces, and real-workload evaluation.

Keywords: large model inference; compute-storage co-optimization; data movement; KV cache; resource exchange; bottleneck migration

引用格式: 汪俊峰, 刘璇, 李亚奇, 等. 面向大模型推理的存算协同理论综述 [J]. 中兴通讯技术, 2026, 32(4): 32-40. DOI: 10.12142/ZTETJ.202604006

Citation: Wang J F, Liu X, Li Y Q, et al. A survey on compute-storage co-optimization theory for large model inference [J]. ZTE technology journal, 2026, 32(4): 32-40. DOI: 10.12142/ZTETJ.202604006

Transformer 自回归推理需反复访问共享权重, 并为活跃序列保存随上下文增长的键值 (KV) 缓存^[1]。随着上下文长度、并发请求数及多轮复用次数的增加, 算力、高带宽内存 (HBM) 容量与跨层数据移动共同成为首词元时延

(TTFT)、逐词元时延 (TPOT) 及服务等级协议 (SLO) 达成的瓶颈。已有研究从服务系统与KV管理的角度进行了梳理^[2-3], 但压缩率、命中率、显存占用下降等局部指标的改善, 并不必然缩短端到端关键路径; 当量化、缓存、卸载与阶段解耦叠加使用时, 单项收益也不能直接累加。

本文以权重、KV与激活的生命周期为线索, 按3类资源置换组织讨论: 以算换存、以存换算与存算路径。评价准

基金项目: 华中科技大学武汉光电国家实验室开放课题计划项目 (2023 WNLOKF004)

则是关键路径净收益——仅当节省落在当前瓶颈且新增开销不高时，局部优化才转化为服务收益。图1展示了面向大模型推理的资源置换与瓶颈迁移统一决策框架。

1 数据对象与边界

推理侧的核心对象为权重、KV与激活：权重可跨请求共享，在解码（Decode）阶段易受内存带宽约束；KV在Prefill阶段生成，在Decode阶段主要被重复读取，同时持续追加新生成Token对应的KV状态，是长上下文、高并发场景下的主要容量压力来源；激活的生命周期较短，但长Prefill过程与大批量处理会抬升资源峰值。HBM需要同时承载上述对象及运行时工作区，因此压缩、缓存与跨层迁移技术，只有在削减当前主导开销，且新增的编解码、数据恢复、传输代价更低时，才能带来端到端收益。3类资源置换的主要边界如表1所示。

2 以算换存

以算换存通过增加或重组计算，减少需物化、驻留或实

际访问的数据量。其典型手段包括注意力输入/输出（I/O）重组、权重与KV的可计算压缩，以及词元裁剪与查询相关访问。该方法的收益取决于额外计算是否作用于当前带宽或流量瓶颈，且编解码开销、索引维护与质量损失是否可控。

2.1 I/O 重组

标准注意力机制会在片上存储与HBM之间多次读写中间结果。FlashAttention通过分块与在线Softmax避免物化完整注意力矩阵；FlashAttention-2优化了任务划分策略；FlashAttention-3借助异步流水线与低精度计算路径提升硬件利用率^[4-6]。FlashInfer进一步适配变长序列、分页机制与Decode内核^[7]。上述方法以更精细的计算组织来削减HBM写回开销，通常不改变计算所得的数学结果，可作为各类优化方案的基线。

I/O重组的优化收益具有负载依赖性。在短序列与小批量场景下，分块加载、数据同步及专用内核调度开销将占据主导地位。当推理瓶颈转移至权重读取、KV带宽或采样环

节时，持续降低注意力中间写回流量的效果对TPOT的优化作用将趋于有限。相关性能评价需同步考量实际HBM流量、算力利用率、内核耗时占比及端到端TTFT指标，不宜仅依赖注意力内核加速比开展单一维度评估。

长Prefill阶段亦可通过缩减参与计算的数据量降低物化开销。MInference依据注意力模式筛选稀疏计算块^[8]；Ring Attention通过分块通信将长序列分摊至多设备^[9]。前者依赖注意力模式预测，存在预测误判与输出质量下降的风险；后者能够保留更为完整的上下文，但会带来额外的通信与同步开销。当注意力具备稳定稀疏特性时，选择性计算方案更具优势；当单卡存储容量受限且设备互联条件充

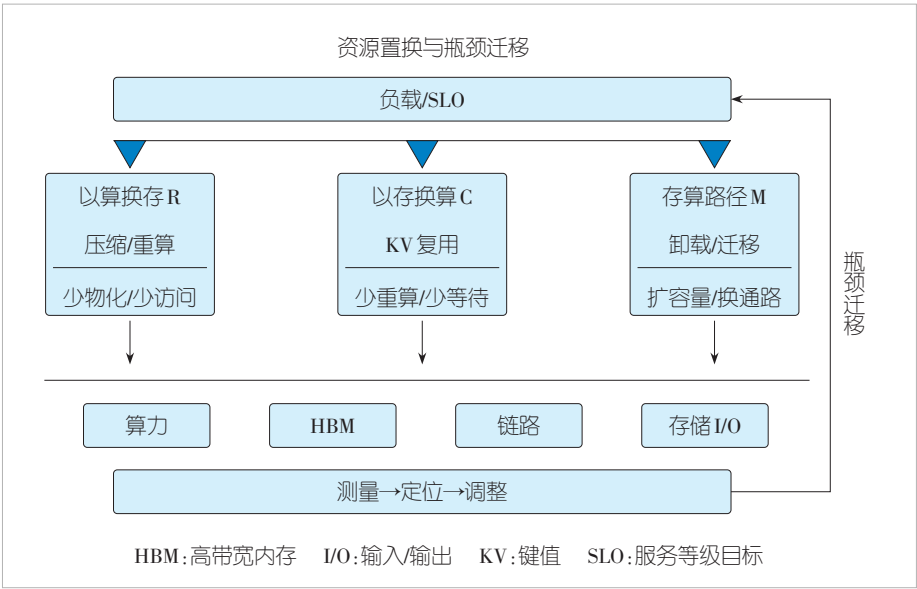


图1 面向大模型推理的资源置换与瓶颈迁移统一决策框架

表1 3类资源置换动作比较

维度	以算换存(R)	以存换算(C)	存算路径(M)
目标	减少物化、存储或读取	避免重复Prefill	扩展容量与共享范围
主要手段	I/O重组、压缩、裁剪与重算	KV分页、前缀/片段复用	卸载、预取与分布式KV
主要代价	编解码、索引与质量风险	驻留、恢复与一致性开销	I/O、链路争用与目录开销
适用条件	带宽或流量主导，附加计算可控	复用率高且恢复快于重算	跨层驻留收益大于迁移开销
正确性	布局无损，压缩或裁剪可能有损	精确且兼容的复用无损	完整迁移无损，选择性恢复有风险

I/O:输入/输出 KV:键值

足时,序列分摊方案更为适用。针对超长输入场景,还可将稀疏筛选与多设备分摊相结合:在单设备内部削减无效块计算,在多设备之间分摊有效块算力。该方案要求统一分块边界与通信调度策略,否则索引不一致将引发数据重复传输或者计算漏算问题。

2.2 权重与KV压缩

权重低比特表示可降低驻留容量与Decode阶段的读取量。SmoothQuant将激活离群值迁移至权重侧;AWQ依据激活统计信息保护重要通道^[10-11];PowerInfer根据激活稀疏性将热、冷神经元分配到不同处理器^[12]。需要区分模型文件大小与实际计算格式:仅当矩阵乘内核可直接消费低比特权重时,容量下降才较有可能转化为设备带宽的降低;若每层先恢复为FP16或BF16,则缩放元数据、反量化读写及工作区开销可能抵消收益。

KV的数值分布与权重不同。KIVI与KVQuant利用Key、Value及通道差异实现低比特表示^[13-14];MiniCache跨层合并相似状态;Palu以低秩投影缩减缓存维度^[15-16]。压缩后占用应计入缩放因子、异常值、残差窗口与临时缓冲。可直接参与计算的表示可同时降低容量与HBM读取;仅用于固态硬盘(SSD)或网络传输的格式,其收益条件为慢链路字节节省大于编码、传输、解码与临时写回开销。

以算换存的目标不是最小化位宽,而是使增加的计算作用于当前瓶颈。若HBM带宽占主导,应优先采用可直接读取的压缩布局;若高速串行计算机扩展总线(PCIe)或网络占主导,可将编码置于慢层并以流式恢复;若算力饱和,增加复杂编解码会延长TTFT。表示格式会影响后续缓存与卸载粒度,因此需与Block边界、预取接口及错误回退一并设计。

对于短生命周期的激活以及可由Prompt快速重建的KV,释放存储并在需要时重算,可能优于长期保留;而对于长上下文或跨轮次状态,重算会再次消耗Prefill并延长恢复时间。是否采用重算,需综合权衡对象大小、重建层数、当前图形处理器(GPU)排队情况与下次使用时间。局部可逆重算也可作为近似选择或缓存失效后的回退手段。需要强调的是,重算路径会与在线Decode争用同一GPU算力。在排队较长时,即使对象可由Prompt重建,优先恢复已有KV仍可能更有利于满足SLO。

2.3 KV裁剪与按需访问

词元选择将“保留全部历史”改为“保留或读取必要历史”。H2O与Scissorhands依据历史重要性驱逐非必要的Token或KV状态;StreamingLLM保留注意力汇点与滑动窗

口^[17-19];SnapKV、PyramidKV与ChunkKV分别依据观察窗口、层间预算与语义块选择位置^[20-22]。此类方法可同时降低容量与读取量,但永久删除后通常无法恢复。

Quest依据当前查询(Query)与页级统计选择本步需加载的KV页^[23];DuoAttention区分检索头与流式头并设置不同历史范围^[24];ShadowKV以低秩Key表示辅助稀疏恢复^[25]。选择粒度越细,潜在节省越大,索引、预测与不规则访存开销就越高。误删会造成不可逆质量损失,并可能在后续多步持续影响生成。

因此,I/O重组、数值压缩与信息裁剪应分别按语义等价、数值误差与信息删除进行评价。工程上可先建立无损布局基线,再验证可直接计算的压缩表示,最后在质量预算明确时引入动态选择。

此外,以算换存还受控制开销约束。稀疏模式识别、查询打分、词元重要性估计与格式转换均引入额外计算。当请求很短、模式变化快或批量很小时,控制开销难以被摊薄。在线服务宜把复杂分析置于Prefill或后台阶段,将Decode中的决策限制为少量可预测操作;离线长上下文任务则可接受更高分析成本以换取更大的访问削减。同一算法在两类负载上的结论不宜直接迁移。

3 以存换算

以存换算是指用存储容量换取计算时间,在保留历史KV后可避免重复Prefill与状态重建。请求内分页布局决定可接纳批量,跨请求前缀与片段复用决定能否缩短TTFT;但命中并不自动等价于端到端收益,还需权衡恢复开销、容量机会成本与兼容边界。

3.1 KV布局与复用

KV缓存以存储容量换取计算时间:保留历史Key与Value后,Decode无需重新处理已生成的词元。PagedAttention将逻辑序列映射到非连续物理块,降低动态长度请求的碎片;vAttention利用虚拟内存保持地址连续^[26-27]。二者主要改善驻留布局,不减少KV信息量或注意力读取量,但通过碎片率、地址映射与写时复制影响可接纳批量。

跨请求复用可避免重复Prefill。SGLang用基数树匹配公共前缀;PromptCache以预定义模块组织可复用状态^[28-29];CacheBlend面向非前缀检索增强生成(RAG)片段,通过选择性重算修复片段间交叉注意力^[30]。从精确词元前缀、预定义模块到组合片段,复用自由度增大,位置约束、键匹配与补算成本亦相应增加。LMCache将KV扩展为可在GPU、CPU、磁盘与远端访问的数据层^[31],使一次Prefill结果可跨

请求、跨实例或跨时段使用。

缓存命中并不自动等价于端到端收益。精确前缀恢复开销低、正确性边界清晰，适合作为共享缓存的基础机制；模块化片段便于编排，但要求模板与位置一致；RAG 片段组合更灵活，但需修复跨片段注意力。缓存位置同样影响收益：本地命中主要支付查找与恢复开销，远程命中还需支付网络与排队开销；对低频短前缀，直接重算可能更快。

3.2 缓存收益与生命周期

缓存的预期收益取决于复用概率、可避免的 Prefill 计算量与再次访问时间。相关开销包括插入、索引、查找、取回、片段修复、一致性维护，以及占用显存带来的容量机会成本。命中率应按复用词元数或避免的计算时间加权，而不宜仅按请求数统计。在 HBM 中保留大量冷 KV 可降低未来恢复开销，但这可能会挤占活跃请求的批处理空间。将冷 KV 下沉虽然可以释放一定容量，但仍需在恢复截止时间前完成传输。因此，缓存策略应综合考虑下次使用时间、重算代价、共享概率与当前队列，而不宜仅依赖最近最少使用算法 (LRU)。

缓存复用需满足多维度兼容性约束，包括模型、适配器、分词器、提示模板、位置编码、KV 存储布局与数值精度的完全匹配。多租户场景下，还需将访问权限、资源所有权、版本信息与安全属性纳入缓存键的构建维度；跨版本迭代更新时，历史 KV 缓存需隔离至独立命名空间或采用渐进失效策略。本文将缓存净收益定义为：预估规避的重算耗时，扣除缓存恢复耗时与存储容量机会成本；同时在队列长度动态增长、缓存容量水位抬升时，主动降低低收益缓存任务的调度优先级。

此外，前缀共享还存在收益递减问题。系统级 Prompt 被多个请求复用时，首次请求承担构建与写入开销，后续请求才获得节省。当共享度下降或模板频繁变更时，目录、哈希与失效维护仍持续发生。RAG 片段组合可能出现部分命中、重复片段与位置不连续，此时可复用比例不宜用字节命中率代替。运行时应记录共享前缀长度、命中后补算词元数与恢复时延分位数，以判断缓存是否真正缩短 Prefill 关键路径。对于暂停的 Agent 会话，还应保留原始 Prompt、工具状态与版本元数据，以便 KV 失效时能够重算或部分恢复。

3.3 推测解码与缓存

推测解码由小模型或附加头生成候选，再由主模型并行验证，从而减少串行 Decode 步数^[32]。该方法增加候选计算与临时状态，以换取主模型串行步数减少，其收益取决于候

选接受率、验证批量与 KV 提交方式。接受长度波动会改变批量形状与 KV 增长速度。若分配器仍按固定步长预留，峰值容量压力可能上升。

共享前缀可降低候选阶段的输入处理开销，但候选分支产生的临时 KV 通常不宜跨请求复用，验证失败后还需回收未提交状态。分块 Prefill 通过混合调度降低长输入对 Decode 的阻塞^[33]，同时使活跃请求、预填充块与缓存恢复共享调度队列。因此，缓存管理需感知词元提交节奏，不宜将生成速度视为常数。推测解码与前缀缓存同时启用时，还应区分共享状态与请求私有状态：共享前缀可降低草稿阶段开销，但未提交候选 KV 不应写入跨请求缓存目录。

4 KV 存储路径

当 HBM 无法同时容纳权重、热 KV、激活与工作区时，KV 需在 CPU 内存、NVMe SSD 与远程存储池之间进行存放与恢复。对于多轮 Agent 与长会话负载，Prefill 命中后的性能往往由 KV 回到计算单元的数据通路决定，而非仅由容量是否足够决定。本节从节点内放置、I/O 通路选择、双路径加载、选择性恢复与分布式数据层几个方面展开讨论，并给出跨层组合与服务评价建议。

4.1 节点内放置

FlexGen 联合规划权重、KV 与激活的放置，并通过大批量与流水重叠来降低 CPU 与 SSD 的传输开销。FlexInfer 面向端侧异构设备动态选择卸载对象^[34-35]。此类方法首先解决了部署可行性问题，并不自动降低请求时延。较大的批量处理可扩大计算覆盖窗口，但会增加 KV 占用、排队及 TTFT。离线吞吐配置不宜直接外推到在线低时延服务。

对 KV 路径而言，关键约束是恢复截止时间：查找、解码与数据移动必须在首次使用前完成。阻塞下一 Decode 步骤的热 KV 优先级高于可延迟的权重预取；可由 Prompt 重建的临时激活优先级低于长会话状态。若权重卸载占满 PCIe，KV 恢复将被拖慢。PowerInfer 表明迁移粒度可由层细化到神经元或结构单元^[12]：粒度越细，空间利用率越高，但预测、小事务与同步开销会增加。系统应依据 I/O 队列深度、事务启动开销与预测置信度选择块大小，而非固定采用最小块。

4.2 KV I/O 通路

外层 KV 进入 GPU HBM 的常见通路包括：1) 存储经主机 DRAM 再由 cudaMemcpyAsync 写入 HBM；2) GPUDirect 存储器 (GDS) 由 NVMe 经 PCIe 直达 HBM，减少主机 CPU

拷贝^[36]；3）经计算网卡（CNIC）发起GPU远程直接内存访问（GPUDirect RDMA），将远端数据直接写入HBM，使传输进入网卡侧服务质量（QoS）调度。三者峰值带宽、CPU开销以及集合通信的隔离能力等方面有所不同。

GDS比较适用于以吞吐为主且PCIe上模型执行通信压力较低的场景：绕过主机中转可降低CPU占用并缩短存储到HBM的路径。但在Prefill/Decode（P/D）解耦与张量并行的常见部署中，同一PCIe链路上同时存在集合通信与KV加载，GDS与CUDA拷贝引擎通常无法纳入计算网卡侧的虚拟通道仲裁，KV突发可能会抬高通信尾延迟。此时，先由存储网卡读入主机DRAM，再经网卡以RDMA Write完成主机到设备的数据传输，可把KV流量置于可调度路径上，以稍长的主机中转路径换取与延迟敏感通信的隔离。而对于大量小块KV，用户态提交RDMA工作请求并批量触发doorbell，往往比逐次调用cudaMemcpyAsync更易摊薄提交开销。

路径选择应同时报告有效带宽、CPU占用、PCIe争用与集合通信P99。若评测仅给出GDS峰值吞吐而忽略通信干扰，则会高估其在在线服务中的收益。相应地，新生成KV落盘亦可经CNIC先到主机、再经存储网络写入后端，使读写两侧共用同一套QoS规则。

4.3 双路径加载

在P/D解耦架构中，传统KV加载仅采用“存储→Prefill实例”的单路径传输模式。多轮Agent负载场景下，Prefill侧存储网卡极易出现带宽饱和，而Decode侧存储网卡往往处于空闲状态，形成显著的带宽资源不对称问题。DualPath方案在保留存储至Prefill原生路径的基础上，新增存储至Decode的传输路径：将KV数据优先加载至Decode实例，再通过计算网络RDMA转发至Prefill实例，依托全局调度器在双路径间动态分配流量，实现双侧存储NIC带宽的聚合利用^[37]。该方案的核心并非扩充存储介质资源，而是重构KV数据进入计算平面的拓扑入口。

双路径机制的有效运行需满足多项约束条件：计算网络可用带宽需高于存储网络尾部带宽，确保转发流程不会显著干扰Decode通信；调度器可依据Prefill与Decode双侧的存储队列、计算队列状态及业务SLO自适应选择传输路径；加载路径与执行通信可在PCIe/NIC层级实现硬件隔离。若无法满足上述条件，经由Decode的绕行传输将同步增加Decode尾延迟与Prefill等待时延。性能评价需全面统计双侧存储网卡（NIC）利用率、路径切换占比、KV数据在存储网络与计算网络的流量分布，以及TTFT、吞吐相较单路径架构的

优化幅度，而非仅仅关注聚合带宽峰值指标。

Prefill/Decode架构解耦本身也会重构KV数据的交接链路。DistServe、Splitwise与WindServe分别从有效吞吐配置、异构资源匹配、流式动态调度维度实现了推理阶段解耦^[38-40]。Prefill阶段结束后需完成KV数据跨实例交接，网络拥塞将直接纳入TTFT关键路径。在短输入、低负载或设备互联带宽不足的场景下，阶段解耦带来的性能收益可能小于KV传输与远端排队产生的开销，此时推理共置执行方案更具优势。当P/D解耦与双路径机制叠加部署时，需严格区分“阶段间KV交接流量”与“外层存储KV加载流量”，对两类流量分别统计计量并设置差异化调度优先级。

4.4 选择性恢复与预取

即便确定数据加载通路，推理过程仍无需每次搬移完整KV数据。InfiniGen依托部分查询信息与已有Key缓存预判关键KV内容，实现选择性按需拉取^[41]，将完整KV数据留存于低速存储层，仅在预测偏差时补充加载。面向多轮大模型服务的以内存为中心的架构（MCAM）用于管理多级KV缓存的分布位置，异步KV预取机制则可利用计算间隙提前完成缓存状态恢复^[42-43]。预取策略存在明显的权衡关系：预取时机过早会占用宝贵的HBM资源，过晚则会造成GPU计算等待；预取粒度过细会受限于单次事务时延，粒度过粗则会冗余搬运无效数据。选择性数据恢复与GDS、双路径机制相互正交：前者精简数据传输字节量，后者决定剩余数据的传输拓扑路径。

CacheGen通过定制化编码方案削减KV数据跨存储、跨网络的传输字节开销^[44]。当低速链路的字节节省收益高于编码、传输、解码及临时写回的综合开销时，该机制可有效压缩推理关键路径时延。压缩传输主要优化存储网络与PCIe链路流量，可计算压缩还能进一步降低HBM与计算单元间的数据交互流量。需要说明的是，两类优化机制需加以区分，不可混为一谈。缓存迁移无须完全被计算时延掩盖即可产生优化价值，仅当容量释放带来的请求准入提升、排队时延降低等收益，超过迁移暴露开销与链路争用损耗时，能够切实提升服务性能；反之，该类优化主要体现为部署可行性层面的增益。

在故障恢复与调度公平性层面，可依据任务重算耗时与业务优先级划分缓存恢复等级。I/O调度可将缓存迁移任务拆解为可抢占式事务，在保障Decode任务截止时间的前提下，为不同请求分配固定带宽份额，同时将超时请求退回本地重算或切换至备用加载路径。

4.5 分布式 KV

分布式 KV 缓存将缓存覆盖范围从单节点拓展至集群维度。Mooncake 构建以 KV 数据为核心的分布式数据层；MemServe 依托弹性内存池与统一访问接口，支撑跨实例 KV 缓存复用^[45-46]。缓存容量与共享域的扩张，使得目录维护、权限归属、副本管理、版本迭代与故障恢复成为系统必需的基础性开销。针对推理请求，调度器需综合比对本地图中、本节点 SSD/GDS 恢复、远程内存池读取、双路径加载与重新 Prefill 等多种方式的预估完成时延，不可预设单一路径的绝对最优性。

缓存感知路由需要在数据局部性与集群负载均衡之间实现权衡。Preble 依托提示词共享特征提升前缀缓存局部性；Llumnix 通过请求动态迁移消解节点热点；LoongServe 借助弹性序列并行机制适配长上下文推理场景^[47-49]。单纯追求高缓存命中率，易使共享前缀与存储 I/O 集中于少数节点，引发负载倾斜。片面依赖队列负载均衡，则会破坏数据局部性、放大跨节点 KV 传输字节量。面向长期缓存与跨节点缓存部署，需明确缓存兼容规则与安全边界，并依据 KV 重算成本动态配置副本数量。

随着集群规模扩张，控制面调度开销将成为推理路径耗时的重要组成部分。目录更新、路径决策与副本回收流程均会消耗 CPU 资源与网络消息开销。控制周期过短会引发频繁路径切换，抵消缓存复用收益；控制周期过长则无法及时感知并消解存储热点。实际部署中需区分 GPU 资源分配、节点预取调度与集群路径调整的时间尺度差异，通过流量限速、租约管理与版本校验机制，规避多控制器对同一 KV 对象的调度争用问题。

4.6 跨层优化与评价

短上下文、高并发 Decode 负载下，系统优化优先级通常聚焦于分页机制、批处理策略与权重带宽瓶颈；而超长提示词与多轮 Agent 负载场景，需优先统计 KV 外层命中率与

恢复路径时延，并据此择优部署 GDS 加速、主机中转、双路径传输或选择性预取等优化方案。针对显存资源受限场景，可在保障数据格式兼容的前提下，先对 KV 数据进行压缩处理再落地存储，同步缩减缓存容量占用与链路传输字节量。多机制组合部署需按照不可逆优化程度逐级递增推进，同时分别统计存储 I/O、PCIe、计算网络与 GPU 计算的耗时占比，实现精细化性能拆解。

在 KV 存储路径的性能评价体系中，除 TTFT、TPOT 与有效吞吐等常规指标外，还需纳入多维度量化参数，包括各传输路径的字节占比、存储与计算网卡利用率、GDS 及主机中转引入的 CPU 与 PCIe 开销、集合通信 P99 时延，以及路径切换、路径回退的触发次数。消融实验需遵循科学递进顺序：首先固定计算策略与缓存机制，再替换底层 I/O 传输通路，最后叠加双路径调度策略，以此精准区分“精简数据搬运量”与“更换传输通路优化资源争用”两类核心收益。系统成本核算不能仅依托 GPU 数量统计，需要全面涵盖 SSD 带宽、存储网卡、计算网卡与主机内存中转产生的综合开销。典型负载场景的优化优先级与核心评价指标汇总如表 2 所示。

5 未来发展方向

从生命周期控制、压缩表征、跨引擎接口与真实负载评测 4 个维度分析，大模型推理存算协同领域仍存在若干亟待解决的开放性问题，具体如下：

第一，运行时管控机制亟需从单一访问历史统计，拓展至全生命周期管理、使用时限约束与传输路径自适应选择。共享前缀、暂停智能体任务、临时激活值与模型权重具备差异化的重算开销与下次访问时序特征；GPU 内存分配、节点预取调度、I/O 通路切换与集群副本管理需适配不同时间尺度完成精细化管控，规避多控制环路对同一缓存对象的重复迁出、路径调整与重复取回问题，减少无效系统开销。

表2 典型负载场景的优化优先级与核心评价指标

负载类型	首要约束	优先动作	核心指标	主要回退策略
交互式短问答	TTFT/TPOT、权重或 KV 带宽	I/O 重组、可计算压缩、分页与小批调度	TTFT/TPOT 分位数、HBM 带宽、批量大小	降低压缩等级或关闭选择性恢复
长上下文与 RAG	Prefill 算力、KV 写入与恢复	前缀/片段缓存、选择性恢复、预取	命中词元比例、恢复截止时间、TTFT、质量	重新 Prefill 或完整 KV 恢复
多轮 Agent	冷 KV 容量、跨轮恢复与路径拥塞	分层 KV、缓存感知路由、双路径	重用概率、路径字节、P99、回退次数	下沉冷 KV 并限制并发
批量离线生成	吞吐、单位成本与存储带宽	大批量放置、压缩传输、流水重叠	Token/s、GPU 利用率、GiB/Token、能耗	降低并行度或改用本地重算
GPU: 图形处理器 I/O: 输入/输出 RAG: 检索增强生成 TTFT: 首 Token 延迟/首词元延迟 HBM: 高带宽内存 KV: 键值 TPOT: 每输出 Token 时间/每输出词元时间				

第二,亟需构建可直接计算、可跨层传输的统一数据表征体系,并实现与多级存储通路的深度协同。优化压缩格式需支持矩阵乘、注意力等核心计算内核直接读取解析,无需额外格式转换;同时向缓存调度、数据迁移层级暴露位宽特性、分块边界与运行工作区资源需求,有效抑制格式转换、临时副本带来的性能损耗,避免抵消存算协同的路径优化收益。

第三,跨引擎KV缓存复用亟需标准化接口与可靠的异常回退机制。标准接口需统一规范模型与适配器参数、分词器配置、位置编码方式、分块边界、存储布局、数值精度、资源所有权与版本信息;当出现格式不兼容、目录失效、网络超时等异常场景时,系统需具备可靠的主动重算能力。针对数据持久化与多租户共享场景,还需明确访问控制、数据加密、操作审计与资源销毁的语义规范。若无上述统一约束,跨实例缓存复用将局限于单引擎局部优化,无法构建可规模化、可运营的分层缓存数据底座。

第四,性能评测体系需从传统峰值吞吐指标,转向真实负载下的精细化瓶颈解析。测试场景需全面覆盖长上下文推理、RAG检索增强、智能体迭代交互、请求突发到达与多租户混部等真实业务轨迹,评测指标需纳入网络尾时延、预测误差代价、迁移回滚次数、故障恢复耗时与安全管控开销。硬件架构研究表明,存储容量、数据迁移开销与集群可扩展性仍是大模型推理的长期约束条件^[50],单一峰值性能无法等价于真实端到端服务能力。此外,跨模态编码器状态、工具调用中间结果、外部知识库缓存的全生命周期管控规则尚未形成统一标准,其与文本KV缓存的资源置换、收益权衡机制仍有待深入探究。

6 结束语

大模型推理存算协同的核心逻辑,是在数据对象生命周期、输出质量、硬件资源与业务SLO的多重约束下,完成精细化资源置换策略的自适应选择。以算换存依托计算重组、数据压缩与选择性计算机制,降低推理过程的数据物化与内存访问开销;以存换算依靠KV缓存留存与复用策略,规避重复Prefill计算损耗;存算路径优化则通过跨层数据放置、多路径恢复通路与分布式KV数据层部署,扩充有效缓存容量并压缩数据恢复时延。3类优化机制支持叠加组合,其综合收益受HBM容量、CPU算力、PCIe通道、存储网络、计算网络及集合通信的资源争用状态共同制约。针对差异化推理负载,需先精准定位Prefill与Decode阶段的核心约束瓶颈,结合KV外层缓存命中特征,择优采用计算侧压缩、缓存复用或传输通路调优方案。

各类存算协同优化均存在适用边界:容量扩容仅当数据可按时序约束送达计算单元时,方可转化为真实推理性能增益;缓存复用仅在数据恢复开销显著低于重算开销时具备优化价值;跨层数据恢复与推理阶段解耦,仅在路径隔离机制与带宽收益能够覆盖绕行传输、数据交接的额外开销时,可实现正向收益。因此,需围绕数据对象特征、传输路径特性与资源置换规则构建分析框架,在每一轮优化迭代后重新拆解端到端关键路径,将局部技术改进转化为可量化、可解释的推理服务增益。

致谢

感谢华东师范大学、上海创智学院、华中科技大学武汉光电国家研究中心对本研究的支持!感谢华东师范大学闫晗同学、许珂同学、王婧老师在研究过程中给予的协助!

参考文献

- [1] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need [PP/OL]. arXiv (2017-06-12)[2026-05-26]. <https://arxiv.org/abs/1706.03762>
- [2] Miao X P, Oliaro G, Zhang Z H, et al. Towards efficient generative large language model serving: a survey from algorithms to systems [J]. ACM computing surveys, 2026, 58(1): 1-37. DOI: 10.1145/3754448
- [3] Li H, Li Y, Tian A, et al. A survey on large language model acceleration based on KV cache management [PP/OL]. arXiv (2024-12-27)[2026-05-26]. <https://arxiv.org/abs/2412.19442>
- [4] Dao T, Fu D Y, Ermon S, et al. FlashAttention: fast and memory-efficient exact attention with IO-awareness [PP/OL]. arXiv (2022-05-27)[2026-05-26]. <https://arxiv.org/abs/2205.14135>
- [5] Dao T. FlashAttention-2: faster attention with better parallelism and work partitioning [PP/OL]. arXiv (2023-07-17)[2026-05-26]. <https://arxiv.org/abs/2307.08691>
- [6] Shah J, Bikshandi G, Zhang Y, et al. FlashAttention-3: fast and accurate attention with asynchrony and low-precision [PP/OL]. arXiv (2024-07-11)[2026-05-26]. <https://arxiv.org/abs/2407.08608>
- [7] Ye Z H, Chen L Q, Lai R H, et al. FlashInfer: efficient and customizable attention engine for LLM inference serving [PP/OL]. arXiv (2025-01-02)[2026-05-26]. <https://arxiv.org/abs/2501.01005>
- [8] Jiang H Q, Li Y C, Zhang C, et al. MInference 1.0: accelerating pre-filling for long-context LLMs via dynamic sparse attention [PP/OL]. arXiv (2024-07-02)[2026-05-26]. <https://arxiv.org/abs/2407.02490>
- [9] Liu H, Zaharia M, Abbeel P. Ring attention with blockwise transformers for near-infinite context [PP/OL]. arXiv (2023-10-03)[2026-05-26]. <https://arxiv.org/abs/2310.01889>
- [10] Xiao G X, Lin J, Seznec M, et al. SmoothQuant: accurate and efficient post-training quantization for large language models [PP/OL]. arXiv (2022-11-18)[2026-05-26]. <https://arxiv.org/abs/2211.10438>
- [11] Lin J, Tang J M, Tang H T, et al. AWQ: activation-aware weight quantization for LLM compression and acceleration [PP/OL]. arXiv (2023-06-01)[2026-05-26]. <https://arxiv.org/abs/2306.00978>
- [12] Song Y X, Mi Z Y, Xie H T, et al. PowerInfer: fast large language model serving with a consumer-grade GPU [C]//Proceedings of

- the ACM SIGOPS 30th Symposium on Operating Systems Principles. ACM, 2024: 590–606. DOI: 10.1145/3694715.3695964
- [13] Liu Z, Yuan J, Jin H, et al. KIVI: a tuning-free asymmetric 2 bit quantization for KV cache [PP/OL]. arXiv (2024-02-05) [2026-05-26]. <https://arxiv.org/abs/2402.02750>
- [14] Hooper C, Kim S, Mohammadzadeh H, et al. KVQuant: towards 10 million context length LLM inference with KV cache quantization [PP/OL]. arXiv (2024-01-31) [2026-05-26]. <https://arxiv.org/abs/2401.18079>
- [15] Liu A, Liu J, Pan Z Z, et al. MiniCache: KV cache compression in depth dimension for large language models [PP/OL]. arXiv (2024-05-23) [2026-05-26]. <https://arxiv.org/abs/2405.14366>
- [16] Chang C C, Lin W C, Lin C Y, et al. Palu: compressing KV-cache with low-rank projection [PP/OL]. arXiv (2024-07-30) [2026-05-26]. <https://arxiv.org/abs/2407.21118>
- [17] Zhang Z Y, Sheng Y, Zhou T Y, et al. H₂O: heavy-hitter oracle for efficient generative inference of large language models [PP/OL]. arXiv (2023-06-24) [2026-05-26]. <https://arxiv.org/abs/2306.14048>
- [18] Liu Z C, Desai A, Liao F S, et al. Scissorhands: exploiting the persistence of importance hypothesis for LLM KV cache compression at test time [PP/OL]. arXiv (2023-05-26) [2026-05-26]. <https://arxiv.org/abs/2305.17118>
- [19] Xiao G X, Tian Y D, Chen B D, et al. Efficient streaming language models with attention sinks [PP/OL]. arXiv (2023-09-29) [2026-05-26]. <https://arxiv.org/abs/2309.17453>
- [20] Li Y H, Huang Y B, Yang B W, et al. SnapKV: LLM knows what you are looking for before generation [PP/OL]. arXiv (2024-04-22) [2026-05-26]. <https://arxiv.org/abs/2404.14469>
- [21] Cai Z F, Zhang Y C, Gao B F, et al. PyramidKV: dynamic KV cache compression based on pyramidal information funneling [PP/OL]. arXiv (2024-06-04) [2026-05-26]. <https://arxiv.org/abs/2406.02069>
- [22] Liu X, Tang Z H, Dong P J, et al. ChunkKV: semantic-preserving KV cache compression for efficient long-context LLM inference [PP/OL]. arXiv (2025-10-14) [2026-05-26]. <https://arxiv.org/abs/2502.00299>
- [23] Tang J M, Zhao Y L, Zhu K, et al. Quest: query-aware sparsity for efficient long-context LLM inference [PP/OL]. arXiv (2024-06-16) [2026-05-26]. <https://arxiv.org/abs/2406.10774>
- [24] Xiao G X, Tang J M, Zuo J W, et al. DuoAttention: efficient long-context LLM inference with retrieval and streaming heads [PP/OL]. arXiv (2024-10-14) [2026-05-26]. <https://arxiv.org/abs/2410.10819>
- [25] Sun H S, Chang L W, Bao W L, et al. ShadowKV: KV cache in shadows for high-throughput long-context LLM inference [PP/OL]. arXiv (2024-10-28) [2026-05-26]. <https://arxiv.org/abs/2410.21465>
- [26] Kwon W, Li Z H, Zhuang S Y, et al. Efficient memory management for large language model serving with PagedAttention [PP/OL]. arXiv (2023-09-12) [2026-05-26]. <https://arxiv.org/abs/2309.06180>
- [27] Prabhu R, Nayak A, Mohan J, et al. vAttention: dynamic memory management for serving LLMs without PagedAttention [PP/OL]. arXiv (2024-05-07) [2026-05-26]. <https://arxiv.org/abs/2405.04437>
- [28] Zheng L M, Yin L S, Xie Z Q, et al. SGLang: efficient execution of structured language model programs [PP/OL]. arXiv (2023-12-12) [2026-05-26]. <https://arxiv.org/abs/2312.07104>
- [29] Gim I, Chen G, Lee S S, et al. Prompt Cache: modular attention reuse for low-latency inference [PP/OL]. arXiv (2023-11-07) [2026-05-26]. <https://arxiv.org/abs/2311.04934>
- [30] Yao J Y, Li H C, Liu Y H, et al. CacheBlend: fast large language model serving for RAG with cached knowledge fusion [C]// Proceedings of the Twentieth European Conference on Computer Systems. ACM, 2025: 94–109. DOI: 10.1145/3689031.3696098
- [31] Liu Y H, Cheng Y H, Yao J Y, et al. LMCACHE: an efficient KV cache layer for enterprise-scale LLM inference [PP/OL]. arXiv (2025-10-08) [2026-05-26]. <https://arxiv.org/abs/2510.09665>
- [32] Leviathan Y, Kalman M, Matias Y. Fast inference from transformers via speculative decoding [PP/OL]. arXiv (2022-11-30) [2026-05-26]. <https://arxiv.org/abs/2211.17192>
- [33] Agrawal A, Kedia N, Panwar A, et al. Taming throughput-latency tradeoff in LLM inference with Sarathi-Serve [PP/OL]. arXiv (2024-03-04) [2026-05-26]. <https://arxiv.org/abs/2403.02310>
- [34] Sheng Y, Zheng L, Yuan B, et al. FlexGen: high-throughput generative inference of large language models with a single GPU [PP/OL]. arXiv (2023-03-13) [2026-05-26]. <https://arxiv.org/abs/2303.06865>
- [35] Du H C, Wu S Y, Kharlamova A, et al. FlexInfer: breaking memory constraint via flexible and efficient offloading for on-device LLM inference [C]// Proceedings of the 5th Workshop on Machine Learning and Systems. ACM, 2025: 56–65. DOI: 10.1145/3721146.3721961
- [36] NVIDIA. GPUDirect storage overview guide. [EB/OL]. [2026-05-26]. <https://docs.nvidia.com/gpudirect-storage/overview-guide/>
- [37] Wu Y T, Chen S Y, Zhong Y M, et al. DualPath: breaking the storage bandwidth bottleneck in agentic LLM inference [PP/OL]. arXiv (2026-02-25) [2026-05-26]. <https://arxiv.org/abs/2602.21548>
- [38] Zhong Y M, Liu S Y, Chen J D, et al. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving [PP/OL]. arXiv (2024-01-18) [2026-05-26]. <https://arxiv.org/abs/2401.09670>
- [39] Patel P, Choukse E, Zhang C, et al. Splitwise: efficient generative LLM inference using phase splitting [PP/OL]. arXiv (2023-11-30) [2026-05-26]. <https://arxiv.org/abs/2311.18677>
- [40] Feng J Q, Huang Y K, Zhang R, et al. WindServe: efficient phase-disaggregated LLM serving with stream-based dynamic scheduling [C]// Proceedings of the 52nd Annual International Symposium on Computer Architecture. ACM, 2025: 1283–1295. DOI: 10.1145/3695053.3730999
- [41] Lee W, Lee J, Seo J, et al. InfiniGen: efficient generative inference of large language models with dynamic KV cache management [PP/OL]. arXiv (2024-06-28) [2026-05-26]. <https://arxiv.org/abs/2406.19707>
- [42] Chu, K, Shen, Z, Cheng, S, et al. MCaM: efficient LLM inference with multi-tier KV cache management [C]// Proceedings of 2025 IEEE 45th International Conference on Distributed Computing Systems. IEEE, 2025: 571–581. DOI: 10.1109/ICDCS63083.2025.00062
- [43] Dong Y H, Miao Y B, Li W N, et al. Accelerating LLM inference throughput via asynchronous KV cache prefetching [J]. Proceedings of the AAAI conference on artificial intelligence, 2026, 40(25): 20844–20851. DOI: 10.1609/aaai.v40i25.39224
- [44] Liu Y, Li H, Cheng Y, et al. CacheGen: KV cache compression and streaming for fast large language model serving [C]// Proceedings of the ACM SIGCOMM 2024 Conference. ACM, 2024: 38–56
- [45] Qin R Y, Li Z M, He W R, et al. Mooncake: a KVCache-centric disaggregated architecture for LLM serving [J]. ACM transactions on storage, 2026, 22(4): 1–38. DOI: 10.1145/3773772
- [46] Hu C C, Huang H Y, Hu J H, et al. MemServe: context caching for disaggregated LLM serving with elastic memory pool [PP/OL]. arXiv (2024-06-25) [2026-05-26]. <https://arxiv.org/abs/2406.17565>
- [47] Srivatsa V, He Z, Abhyankar R, et al. Preble: efficient distributed

- prompt scheduling for LLM serving [PP/OL]. arXiv (2024-05-08) [2026-05-26]. <https://arxiv.org/abs/2407.00023>
- [48] Sun B, Huang Z, Zhao H, et al. Llumnix: dynamic scheduling for large language model serving [PP/OL]. arXiv (2024-06-05) [2026-05-26]. <https://arxiv.org/abs/2406.03243>
- [49] Wu B Y, Liu S Y, Zhong Y M, et al. LoongServe: efficiently serving long-context large language models with elastic sequence parallelism [PP/OL]. arXiv (2024-04-15) [2026-05-26]. <https://arxiv.org/abs/2404.09526>
- [50] Ma X Y, Patterson D. Challenges and research directions for large language model inference hardware [J]. Computer, 2026, 59(5): 55-64. DOI: 10.1109/mc.2026.3652916

作者简介



汪俊峰，华东师范大学在读硕士研究生；主要研究方向为大模型存储系统和推理性能优化。



刘璇，华东师范大学在读硕士研究生；主要研究方向为大模型存储系统和多模态数据湖仓。



李亚奇，华东师范大学、上海创智学院在读博士研究生；主要研究方向为智能存储系统、高带宽闪存。



石亮，华东师范大学教授、博士生导师，上海创智学院全时导师；主要研究方向为存储系统和操作系统，涵盖大模型与人工智能、大数据与云计算、嵌入式系统与物联网等领域。

大规模向量存储系统研究综述



Survey on Large-Scale Vector Storage Systems

郭昊/Guo Hao¹, 谢旻晖/Xie Minhui², 陆游游/Lu Youyou¹

(1. 清华大学, 中国 北京 100084;

2. 南京大学, 中国 南京 210023)

(1. Tsinghua University, Beijing 100084, China;

2. Nanjing University, Nanjing 210023, China)

DOI: 10.12142/ZTETJ.202604007

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260828.1138.002>

网络出版日期: 2026-08-28

收稿日期: 2026-06-05

摘要: 大规模向量存储系统作为面向海量非结构化数据的高维向量表示载体, 是提供高效存储与检索能力的核心基础设施。针对数据规模激增所带来的内存容量瓶颈与成本压力, 聚焦于大规模向量索引与检索系统的演进路径。首先, 概述内存图索引与聚类索引的基本机制; 随后, 重点分析十亿级乃至更大规模场景下的向量量化压缩、基于固态硬盘(SSD)的索引架构以及异构存储硬件加速技术, 并从精度、延迟、吞吐量、构建更新效率及资源开销等维度对比各方案的差异; 进一步, 探讨向量检索与检索增强生成(RAG)的端到端协同优化; 最后, 总结不同技术路线的适用场景、主要瓶颈与未来研究方向。

关键词: 大规模向量存储; 近似最近邻检索; SSD; RAG

Abstract: Large-scale vector storage systems serve as the core infrastructure for efficiently storing and retrieving high-dimensional representations of massive unstructured datasets. To address the memory capacity bottlenecks and cost pressures arising from the explosive growth of data volumes, the evolution of large-scale vector indexing and retrieval systems is reviewed. In-memory graph-based and clustering-based indexing methods are first outlined. Then, vector quantization, solid-state drive (SSD)-based index architectures, and heterogeneous storage-computing hardware acceleration for billion-scale and even larger datasets are examined. These approaches are compared in terms of accuracy, latency, throughput, construction/update costs, and resource overhead. End-to-end collaborative optimization between vector retrieval and Retrieval-Augmented Generation (RAG) is further discussed. Finally, applicable workloads, major bottlenecks, and future research directions are summarized.

Keywords: large-scale vector storage; approximate nearest neighbor search; SSD; RAG

引用格式: 郭昊, 谢旻晖, 陆游游. 大规模向量存储系统研究综述 [J]. 中兴通讯技术, 2026, 32(4): 41-46. DOI: 10.12142/ZTETJ.202604007

Citation: Guo H, Xie M H, Lu Y Y. Survey on large-scale vector storage systems [J]. ZTE technology journal, 2026, 32(4): 41-46. DOI: 10.12142/ZTETJ.202604007

随着人工智能大模型的持续发展, 向量正逐步成为一种新兴的数据表示范式。文本、图像、视频等非结构化数据经嵌入模型处理后, 可转化为数百至数千维的高维向量。语义相近的数据在向量空间中呈现相近的几何距离, 这为基于语义的向量检索带来了新的可能性。在此过程中, 向量存储系统需根据查询所生成的嵌入向量, 在海量数据集中检索并返回若干个最近邻向量, 同时还需支持高吞吐的向量更新等操作。

当数据规模扩展至百万级或更高时, 精确向量检索的计

算开销呈指数增长, 查询延迟往往高达秒级以上。传统的 K-D 树等精确检索方法在低维场景下表现良好, 但在高维场景中易受“维度灾难”影响, 导致空间划分失效, 搜索效率退化至与暴力扫描相当。为此, 学术界与工业界普遍采用近似最近邻检索 (ANNS)。该方法以极小的召回精度损失为代价, 实现检索效率的数量级提升, 将单次查询延迟降低至毫秒级甚至更短。

1 向量存储的主流索引算法

随着数据规模的增长, 百万级别的向量通常占据百 MB 至 GB 级存储空间, 仍可完全驻留于内存。因此, 传统向量索引大多基于内存构建, 主要分为基于图的索引与基于聚类的索引两类。

基金项目: 国家重点研发计划项目 (2024YFE0203300); 国家自然科学基金项目 (62332011, 62502521); 北京市自然科学基金项目 (L242016, 4254083); 北京市科学技术协会青年人才托举工程项目 (20250982); 霍英东教育基金会高等院校青年教师基金项目; CCF-腾讯犀牛鸟科研基金项目

1.1 基于图的向量索引

图索引结构将所有向量组织为一张有向图，每个向量作为图节点，节点间通过边相连。其优势在于支持细粒度的图上游走，能够在保证高召回率的同时实现低检索延迟。为保证连通性与检索效率，图索引既需包含距离相近节点间的短边，以便在逼近目标时实现精确收敛，也需包含跨度较大的长边，以便在搜索初期快速拉近与查询向量的距离。

分层可导航小世界图（HNSW）^[1]、导航扩展图（NSG）^[2]及 DiskANN^[3]等主流图索引采用了不同的选边策略。基础策略通常依赖于三角不等式，即对最长边进行裁剪。例如，对于即将插入的向量 A ，若 A 到 B 的边长大于 A 到 C 与 B 到 C 的边长之和，则该边会被移出 A 的邻居集合。HNSW 整体采用该策略，并额外引入多层级结构以增加长边连接：最底层包含全量向量，每向上一层，节点规模按固定比例（如 1%）指数级递减。检索时自顶向下逐层贪心路由，直至底层局部最优。相比之下，NSG 和 DiskANN 采用单层图结构，其中 DiskANN 进一步放宽了三角不等式的限制，允许长度小于其他边 1.2 倍的长边保留，以增强图的连通性。

尽管图拓扑结构各异，其查询过程大体遵循启发式贪心遍历策略：从入口节点出发，将邻居按与查询的距离加入候选池，反复扩展当前最优的未访问节点，直至候选集收敛。传统方法通常需要对被访问节点的绝大多数邻居执行精确距离计算；分区极值次序统计量（PEOs）^[4]则利用空间分区与随机投影构造带概率保证的路由判定，仅对更有可能改善搜索结果的邻居计算精确距离。原文实验表明，将 PEOs 接入 HNSW 和 NSG 后，在不同召回率条件下可将查询吞吐量提升 1.6 ~ 2.5 倍，体现了其相较启发式邻居剪枝方法更具可分析性与稳定性。

图索引虽检索性能优异，但其构建与更新受节点间依赖关系影响，开销较大。HNSW 与新版本 DiskANN 采用逐点插入方式：每加入一个新节点，需在现有图中搜索候选邻居，并按度数上限 R 进行裁剪；NSG 与初代 DiskANN 则以 k 近邻（kNN）图或随机连通图为起点，通过全局重构优化邻居连接。上述两类方法均需反复执行邻居搜索与选边操作，构建成本随数据规模增长而快速上升。

1.2 基于聚类的向量索引

聚类索引通过 K-Means 等方法将向量空间划分为若干簇，并为每个簇提取中心向量。查询时，先定位距离最近的若干中心，再扫描相应簇内的候选向量；典型工作包括倒排文件索引（IVF）^[5]与 IVF-分层 HNSW^[6]，后者采用图索引管理聚类中心，以降低路由开销并支持更细粒度的空间划分。

传统 IVF 通常将每个向量分配至单一簇，边界向量易被漏检；基于正交增强残差的向量副本方法（SOAR）^[7]则利用正交放大残差损失为向量选择冗余分区，使次级分区专门补偿主分区失效的查询方向。与简单复制到第二近簇或独立训练多个索引相比，SOAR 侧重于降低不同分区失败事件之间的相关性，在较低额外存储开销下改善了召回率与扫描效率。

聚类索引的构建主要依赖 K-Means 迭代过程：聚类数较少时可通过采样快速训练，超大规模聚类则需借助层次化或平衡聚类来降低多轮全量扫描的开销。与图索引相比，聚类之间的数据依赖较弱，更便于并行构建与局部更新，但其检索质量更易受到聚类边界效应的影响。

2 大规模向量存储系统

在真实生产环境中（如在线大模型服务的全网数据检索），向量规模通常高达十亿乃至千亿级别。将如此海量的高维数据全量驻留于单机内存，不仅成本高昂，亦受限于物理容量。当前主流的扩展思路有两种：一是采用精度有损的向量量化压缩技术；二是将数据下沉至闪存固态硬盘（SSD）等大容量非易失性存储介质中。

2.1 向量量化

向量量化是一种有损压缩技术，不仅能成倍缩减存储占用，还能显著降低单次距离计算的算力开销。量化技术与上述图/聚类检索方法相互正交，既可独立应用于每个向量，也可嵌入索引遍历过程中以加速距离测算。常用量化技术主要分为标量量化（SQ）与乘积量化（PQ）两类。

SQ 将向量每一维度的浮点值线性映射至固定区间。例如，8-bit 量化将极值区间划分为 256 段，用整型值替代原始浮点数。RaBitQ^[8]提出一种改进型 SQ 量化方法，通过随机正交变换与残差量化降低精度损失，并利用单指令多数据流（SIMD）指令集加速扫描过程。总体而言，标量量化计算速度极快，但固定步长的离散化方式会引入一定程度的精度误差。

PQ 则将高维向量沿维度切分为若干子段，在每个子段内独立进行 K-Means 聚类（如设定 256 个中心），随后将向量编码为对应子段聚类中心的索引号。检索时，通过预先计算查询向量各子段到所有聚类中心的距离，并构建非对称距离表，即可通过查表求和快速估算近似距离。相较于标量量化，PQ 在同等压缩率下具有更高的保真度，但查表操作破坏了内存连续访问模式，使得单步计算开销略高于基于 SIMD 的并行计算方法。ScaNN^[9]进一步将空间分区与各向异

性向量化相结合：其不再对所有重构误差进行等权处理，而是对与原始向量方向平行的残差施加更大惩罚，从而减少最大内积搜索中的排序误差。与传统 PQ 主要致力于最小化全局重构误差不同，ScaNN 直接围绕检索目标优化量化损失，通过分区缩小候选集，并借助量化加速候选向量的打分过程。

2.2 盘上向量存储——基于图索引

除了对向量进行压缩之外，部分研究还通过引入 SSD 等非易失性存储介质来扩展容量。相较于内存，SSD 具有更大的存储容量和更低的单位成本，能够支持十亿乃至百亿级别的向量数据。然而，SSD 的读写吞吐量显著低于内存，且读写延迟高出数个数量级。现有基于 SSD 的向量存储系统主要围绕这两方面进行优化。

在降低输入/输出（I/O）读写量与提升吞吐方面，DiskANN^[9]采用了内存与 SSD 协同的设计：在内存中仅缓存所有向量的压缩量化版本（如 PQ 码），而将完整的图索引拓扑结构与全量高维向量存储于 SSD 上。检索时，针对邻居向量的距离比较与图遍历可直接在内存中的量化向量上完成，系统仅在读取节点的邻居编号及进行向量重排时访问 SSD，无需读取大量邻居向量的原始数据，从而显著降低了整体 I/O 读取量。Starling^[10]在此基础上进一步优化了磁盘数据布局，通过图节点重排技术，将图结构中彼此相邻、极易被共同访问的节点聚集存放于 SSD 的同一数据页面或相邻块中。这种设计大幅提升了单次 I/O 的数据命中率，减少了磁盘带宽的浪费。

在缓解高 I/O 延迟方面，DiskANN 摒弃了传统的逐节点贪心搜索机制，转而采用集束搜索。该方法在每一轮迭代中并发探索多个候选节点，向 SSD 一次性发出多个并行 I/O 请求，充分利用 SSD 内部并发度来均摊单次 I/O 延迟。PipeANN^[11]则进一步放宽了贪心搜索的约束，仅探索当前局部最优节点，允许每次探索时存在尚未完成的节点读取操作，而不必等待这些读取完成。通过这种方式，PipeANN 消除了步间依赖关系，使得 SSD I/O 与计算能够重叠并流水线执行，从而有效掩盖 I/O 延迟。

除了向量检索外，向量存储还需支持数据动态更新，以适应不断变化的向量数据集。针对图向量索引的动态更新，现有方法主要分为两类：增量索引合并与直接插入。

增量索引合并的代表系统为 FreshDiskANN^[12]。其在内存中开辟独立空间，维护一个动态增量图以临时容纳新插入的向量。当内存缓冲区达到容量上限后，系统通过后台操作将增量图批量合并至 SSD 的主图索引中。该方式能有效减少写

放大，但合并过程中产生的突发 I/O 可能干扰前台检索请求。直接插入的代表系统为 OdinANN^[13]。其直接将新向量插入磁盘上的全量图索引中，将插入过程的 I/O 开销均匀摊薄至持续操作流中，使得前台查询延迟在持续插入场景下更为稳定，但代价是写放大较高。

2.3 盘上向量存储——基于聚类索引

将传统聚类索引迁移至 SSD 是一种直观的扩展思路。该系统在内存中维护聚类中心表，并将各聚类内的向量存储于 SSD。检索时，通过内存中的聚类中心定位到距离最近的若干个聚类，再从 SSD 中并行读取相应聚类内的向量进行比较。该方式虽然具备可扩展性，但仍面临两方面的性能瓶颈：一方面，传统聚类中心数量较少，即使单次检索只查询少数聚类，仍需读取大量聚类内向量，导致 I/O 开销较大；另一方面，聚类算法本身效率较低，K-Means 聚类需流式读取全量数据。当数据规模达到十亿级时，聚类复杂度显著上升，若内存容量有限，聚类迭代过程需进行数遍乃至数十遍全量数据扫盘，计算与 I/O 开销均十分庞大。

针对上述问题，SPANN（微软的系统名）^[14]提出了一种海量聚类方法，每个聚类仅包含数十个向量（远小于传统倒排文件索引 IVF 中数千至数万个向量的规模），并在内存中维护一个规模约为总向量数 1/10 的聚类中心导航图，以快速定位至目标聚类中心。通过该架构，SPANN 对单个聚类进行遍历的 I/O 开销显著降低，使其能够在单次查询中访问多个精细聚类以提升搜索精度，同时保留聚类索引并行读取的特性，从而保持较低的查询延迟。为缓解海量聚类收敛慢的问题，SPANN 采用层次化平衡聚类策略，通过层次聚类加速收敛过程，并确保各聚类长度相对均衡。针对聚类索引的边界效应，SPANN 引入副本机制，将边界向量复制并存储于多个相邻倒排列表中，从而提升系统召回率。在此基础上，SPFresh^[15]进一步设计了聚类分裂与合并机制，以支持索引的动态更新。

2.4 基于异构存算硬件的向量存储

除基于中央处理器（CPU）和 SSD 的架构外，另有研究探索在混合内存、图形处理器（GPU）等场景下进行大规模向量检索，以突破 CPU 的算力瓶颈和 SSD 的带宽限制。

图向量索引方面，基于异构内存的近似最近邻搜索算法（HM-ANN）^[16]针对异构内存的图索引进行了探索。HM-ANN 利用了持久性内存大容量与动态随机存取存储器（DRAM）高速度的物理特性，设计了一种层次化的图索引放置策略。HM-ANN 将图索引的上层导航节点存放在

DRAM中以保证高速的初期路由,而将底层全量图拓扑及未压缩的原始高维向量持久化于非易失内存中。这种设计不仅规避了量化带来的精度损失,还能在单机架构下无损支撑十亿级向量检索。GustANN^[17]则针对GPU-SSD结构的向量检索,提出了一种以GPU为中心、CPU辅助的架构。与传统的基于GPU直访的I/O不同,GustANN利用CPU进行I/O操作,通过仅将需要的部分传输到GPU中以缓解外围组件互连高速总线(PCIe)带宽瓶颈。与此同时,GustANN设计了一种GPU高效的检索内核,以减少搜索过程的分支。

聚类向量索引方面,内存处理近似最近邻搜索系统(PIMANN)^[18-19]首次将存内计算技术应用于向量检索以应对带宽瓶颈。传统聚类检索在扫描海量倒排列表时极易耗尽内存带宽,而PIMANN通过将距离比较逻辑直接卸载至靠近存储单元的存内处理(PIM)协处理器,利用其提供的极高内部聚合内存带宽,大幅减少了向量数据在内存总线与CPU之间的搬移开销,从而加速了聚类内部的距离扫描过程。FusionANNS^[20]则提出了一种面向CPU、GPU与SSD的三级异构协同架构。FusionANNS在SSD上持久化存储全量原始向量,在GPU的高带宽内存中缓存PQ量化向量,同时在CPU主机内存中仅维护轻量级的聚类导航图与倒排链表ID。检索时,CPU首先在内存中快速定位至最近的若干聚类,随后仅将向量ID传输至GPU;GPU利用其庞大的并发线程与高带宽存储器(HBM)带宽完成大规模近似距离计算与过滤;最后,系统根据GPU的筛选结果,从SSD中读取极少量原始向量进行精确重排。该协同机制充分利用了GPU的算力以及CPU在分支处理上的优势,以较低的硬件成本兼顾了高吞吐、低延迟与高精度。

表1从精度、查询性能、构建更新和资源开销4个方面对代表性技术路线进行了比较,其中“高”“中”“低”表示在相近召回率与硬件预算条件下的相对水平。

因此,内存方案优先服务于容量可控且追求极低延迟与高吞吐的负载;盘上图索引更适用于查询密集、更新较少的数据集;盘上聚类索引则更适合持续写入与批量更新的场景;异构方案以更复杂的数据放置与调度策略为代价,换取更高的查询吞吐量。

3 向量存储与RAG的协同优化

从端到端的视角来看,检索增强生成(RAG)系统包含离线或增量的数据处理链路以及在线查询链路。以用户上传PDF为例,系统需先完成解析或光学字符识别(OCR)、清洗与分块,再由嵌入模型生成向量并构建或增量更新索引;在线阶段则依次执行查询编码、近似最近邻检索、候选重排序与上下文组装,最后将检索结果送入生成模型。向量检索位于两条链路的交汇点,其召回率、更新时效和尾延迟会直接影响后续重排质量、上下文长度及端到端响应时间。

上述环节之间存在明显的耦合关系:图索引、聚类索引、SSD布局与量化精度共同决定候选质量与检索时延;热点文档块、聚类中心及查询结果可跨请求缓存;CPU侧的解析检索与GPU侧的嵌入生成需统一调度,以减少数据搬移并实现各阶段执行的重叠。因此,RAG系统的协同优化不应仅追求向量索引的独立吞吐量,而应在质量约束下,联合优化数据摄取、索引更新、在线检索、重排序、缓存与模型推理的端到端尾延迟及资源成本^[21-22]。

检索增强生成优化器(RAGO)^[21]从全局资源调度与配置的角度进行协同优化。实际部署的RAG系统包含多样的配置(如不同的数据库规模、是否包含文档编码器、查询重写器和结果重排器等)。为此,RAGO提出了一种结构化的负载抽象框架RAGSchema,用来统一描述这些复杂的应用流水线与特征。基于这种抽象,RAGO能够在系统资源受限的前提下,自动探索并搜索出最优的调度策略,这包括:决

表1 大规模向量索引与检索技术路线比较

技术路线	精度	查询延迟与吞吐	构建与更新	内存/存储开销	适用负载与主要瓶颈
内存索引及量化(HNSW、NSG、ScANN + PQ、RaBitQ)	中-高;受量化率影响	延迟低、吞吐高	量化快;图构建偏慢,更新依基础索引	DRAM高;量化可压缩	适合极低延迟、极高吞吐的负载;瓶颈为内存容量
盘上图索引(DiskANN、PipeANN)	高	延迟较低、吞吐受限于盘随机访问IOPS	构建和增量更新开销高	DRAM低;SSD容量开销高	适合查询密集的大规模负载;瓶颈是随机IOPS
盘上聚类索引(SPANN、SP-Fresh)	中	延迟较低,I/O并发好;吞吐低	构建和增量更新相对友好	DRAM低;SSD存储开销高	适合更新密集的大规模负载;瓶颈是扫描带宽与边界效应
异构硬件加速(GustANN、PIMANN、FusionANNS)	取决于底层索引	吞吐高、延迟较高(受PCIe传输影响)	数据布局和预处理复杂,动态更新较难	需HBM、持久性内存或PIM等专用资源	适合稳定高吞吐负载;瓶颈是PCIe/内存带宽与数据搬移
ANN: 近似最近邻 DRAM: 动态随机存取存储器 HBM: 高带宽存储器 HNSW: 分层可导航小世界图		IOPS: 每秒输入/输出操作次数 NSG: 导航扩展图 PCIe: 外围组件互连高速总线 PIM: 存内计算		PIMANN: 存内计算近似最近邻搜索 PQ: 乘积量化 ScANN: 可扩展最近邻搜索 SPANN: 微软的系统名 SSD: 固态硬盘	

定各个模型组件是分离部署还是混合部署任务放置、各阶段分配多少计算资源,以及采用多大的批处理等。

HedraRAG^[22]针对细粒度流水线进行协同优化。面对多跳推理、迭代检索等高度异构且复杂的多阶段RAG workflow, HedraRAG将整个处理过程抽象为图结构。为避免检索与生成相互阻塞, HedraRAG通过动态图变换(如节点拆分)将大任务切分为耗时相近的微阶段,使CPU端的检索与GPU端的生成形成高效的并行流水线。同时,利用复杂请求在多个阶段间的语义相似性进行推测执行:在完整检索结果返回之前,大模型可借助早期获得的部分检索结果提前启动生成过程;若最终检索结果一致,则直接采纳,从而有效掩盖检索延迟。此外, HedraRAG还通过缓存热门聚类来提高跨请求的检索效率。

4 未来趋势展望

预计在未来较长一段时期内,图索引与聚类索引仍将保持共存。图索引的优势在于能够通过精细的图结构实现高精度与低延迟,但其构建过程算力开销大、节点依赖性强,且拓扑结构复杂,导致扩展困难、更新成本高。相比之下,聚类索引结构简单,聚类间无数据依赖,构建与更新易于实现,但检索精度相对较低。预期未来研究将围绕这两类索引分别展开:一方面,打破图索引的强依赖关系,进一步提升其构建性能与水平扩展能力;另一方面,优化聚类划分策略,在保持高扩展性的同时提高聚类索引的检索精度。

随着大模型应用进一步发展,向量存储系统也面临新的挑战,未来主要呈现以下3个技术趋势:

1) 超大规模检索。随着嵌入模型的演进,对单个数据的表征正从传统的单向量表示转向多向量表征。加之数据规模的持续增长,向量检索的规模预计将从十亿级扩展至千亿级乃至万亿级。以全网搜索等场景为例,其总数据规模已达千亿级别。面向十亿级规模的单机SSD检索方案难以满足千亿级规模下PB级索引的存储需求。如何利用新型网络与存储硬件实现分布式构建与检索,在大规模条件下兼顾精度与性能,是亟待解决的新挑战。

2) 混合检索。RAG对底层存储的需求已不再局限于纯粹的语义向量相似度计算,实际业务场景往往需要复合过滤条件,例如“仅从权威来源中筛选”或“优先检索最新数据”。这要求未来的向量存储系统具备高效的混合检索能力。如何在保证向量检索召回率的前提下,将倒排索引、列式标量存储等技术与向量检索有效结合,以支持多维度的标量筛选,是提升RAG系统准确性与合规性的关键所在。

3) 向量存储-应用协同优化。以“边想边搜”及 Deep

Research 长程推理等为代表的新型AI负载,呈现出高度动态、多轮迭代和多跳检索的语义特征,使得检索与大模型生成之间不再是简单的串行执行关系。向量存储系统需针对此类负载进行定制化设计,例如:感知多轮对话中的语义变化以支持索引预取,针对多跳推理的中间状态进行缓存优化等。通过存储层与应用层的协同设计,可有效掩盖I/O与检索延迟,从而提升端到端的响应速度。

5 结束语

向量存储作为人工智能的关键支撑技术,正经历从单一算法优化向全栈系统协同乃至应用协同设计的演进。从内存态的图与聚类算法,到基于大容量固态盘的盘上架构,再到异构软硬件加速,向量存储系统已能够以较低成本实现十亿级别的高效检索。展望未来,向量存储仍面临超大规模检索、混合检索以及向量存储-应用协同优化等关键挑战。通过攻克这些挑战,向量存储将为下一代通用人工智能提供更为坚实、智能的数据底座。

参考文献

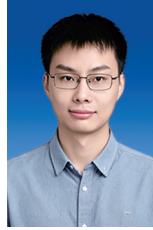
- [1] Malkov Y A, Yashunin D A. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs [J]. IEEE transactions on pattern analysis and machine intelligence, 2020, 42(4): 824-836. DOI: 10.1109/TPAMI.2018.2889473
- [2] Fu C, Xiang C, Wang C X, et al. Fast approximate nearest neighbor search with the navigating spreading-out graph [J]. Proceedings of the VLDB endowment, 2019, 12(5): 461-474. DOI: 10.14778/3303753.3303754
- [3] Jayaram Subramanya S, Devvrit F, Simhadri H V, et al. DiskANN: fast accurate billion-point nearest neighbor search on a single node [EB/OL]. (2019-12-08)[2026-06-12]. <https://dl.acm.org/doi/abs/10.5555/3454287.3455520>
- [4] Lu K, Xiao C, Ishikawa Y. Probabilistic routing for graph-based approximate nearest neighbor search [PP/OL]. arXiv(2024-02-17)[2026-06-15]. <https://arxiv.org/abs/2402.11354>
- [5] Babenko A, Lempitsky V. The inverted multi-index [J]. IEEE transactions on pattern analysis and machine intelligence, 2015, 37(6): 1247-1260
- [6] Baranchuk D, Babenko A, Malkov Y. Revisiting the inverted indices for billion-scale approximate nearest neighbors [PP/OL]. arXiv(2018-07-23) [2026-06-15]. <https://arxiv.org/abs/1802.02422>
- [7] Sun P, Simcha D, Dopson D, et al. SOAR: improved indexing for approximate nearest neighbor search [PP/OL]. arXiv(2024-05-31)[2026-06-14]. <https://arxiv.org/abs/2404.00774>
- [8] Gao J Y, Long C. RaBitQ: quantizing high-dimensional vectors with a theoretical error bound for approximate nearest neighbor search [J]. Proceedings of the ACM on management of data, 2024, 2(3): 1-27. DOI: 10.1145/3654970
- [9] Guo R, Sun P, Lindgren E, et al. Accelerating large-scale inference with anisotropic vector quantization [PP/OL]. arXiv (2019-08-27)[2020-12-04]. <https://arxiv.org/abs/1908.10396>
- [10] Wang M, Xu W, Yi X, et al. Starling: an I/O-efficient disk-

- resident graph index framework for high-dimensional vector similarity search on data segment [J]. Proceedings of the ACM on management of data, 2024, 2(1): 1–27
- [11] Guo H, Lu Y. Achieving low-latency graph-based vector search via aligning best-first search algorithm with SSD [EB/OL]. (2025-07-07) [2026-06-14]. <https://www.usenix.org/system/files/osdi25-guo.pdf>
- [12] Singh A, Subramanya S J, Krishnaswamy R, et al. FreshDiskANN: a fast and accurate graph-based ANN index for streaming similarity search [PP/OL]. arXiv(2021-05-20) [2026-06-18]. <https://arxiv.org/abs/2105.09613>
- [13] Guo H, Lu Y. OdinANN: direct insert for consistently stable performance in billion-scale graph-based vector search [EB/OL]. [2026-06-14]. <https://storage.cs.tsinghua.edu.cn/papers/fast26-odinann.pdf>
- [14] Chen Q, Zhao B, Wang H, et al. SPANN: highly-efficient billion-scale approximate nearest neighborhood search [PP/OL]. arXiv (2021-11-05)[2026-06-11]. <https://arxiv.org/abs/2111.08566>
- [15] Xu Y M, Liang H Y, Li J, et al. SPFresh: incremental in-place update for billion-scale vector search [C]//Proceedings of the 29th Symposium on Operating Systems Principles. ACM, 2023: 545–561. DOI: 10.1145/3600006.3613166
- [16] Somboonmark W, Chumnaul J. Enhancing AQI forecasting accuracy: integrating ARIMA, ANN, and regression techniques with the development of HM4AQI web application [J]. Journal of big data, 2026, 13(1): 10672–10684. DOI: 10.1186/s40537-026-01410-y
- [17] Jiang H D, Guo H, Xie M H, et al. High-throughput, cost-effective billion-scale vector search with a single GPU [J]. Proceedings of the ACM on management of data, 2025, 3(6): 1–27. DOI: 10.1145/3769799
- [18] Wu P Q, Xie M H, Zhao E R, et al. Turbocharge ANNS on real processing-in-memory by enabling fine-grained per-PIM-core scheduling [C]//Proceedings of the 2025 USENIX Conference on Usenix Annual Technical Conference. ACM, 2025: 1223–1241. DOI: 10.5555/3768039.3768111
- [19] Wu P Q, Xie M H, Zhao E R, et al. Unlocking PIM-core capabilities for efficient ANNS on commodity processing-in-memory hardware [J]. ACM transactions on storage, 2026, 22 (3): 1–39. DOI: 10.1145/3806055
- [20] Tian B, Liu H K, Tang Y H, et al. Towards high-throughput and low-latency billion-scale vector search via CPU/GPU collaborative filtering and re-ranking [C]//Proceedings of the 23rd USENIX Conference on File and Storage Technologies. ACM, 2025: 171–185. DOI: 10.5555/3724648.3724659
- [21] Jiang W Q, Subramanian S, Graves C, et al. RAGO: systematic performance optimization for retrieval-augmented generation serving [C]//Proceedings of the 52nd Annual International Symposium on Computer Architecture. ACM, 2025: 974–989. DOI: 10.1145/3695053.3731093
- [22] Hu Z D, Murthy V, Pan Z F, et al. HedraRAG: co-optimizing generation and retrieval for heterogeneous RAG workflows [C]// Proceedings of the ACM SIGOPS 31st Symposium on Operating Systems Principles. ACM, 2025: 623–638. DOI: 10.1145/3731569.3764806

作者简介



郭昊，清华大学计算机系在读博士研究生；主要研究领域为文件系统和向量存储系统。



谢旻晖，南京大学讲师；主要研究领域为存储系统和人工智能系统；主持国家自然科学基金青年基金项目、北京市自然科学基金项目等；入选北京市青年托举人才工程，获得ACM中国优秀博士学位论文奖等；发表论文20余篇。



陆游游，清华大学计算机系长聘副教授、博士生导师，清华大学（计算机系）-软通计算机人工智能系统联合研究中心主任，青年北京学者，FAST、SOSP、OSDI、USENIX ATC、EuroSys等国际会议组织委员会或程序委员会委员；主要研究方向为存储与人工智能系统；曾获国家优秀青年科学基金项目资助，作为项目负责人主持了新一代人工智能国家科技重大专项项目、国家重点研发计划项目、国家自然科学基金重点项目等；6次获得最佳国际论文奖项，参与研制的高性能文件系统SuperFS部署于鹏城云脑II，2023年—2025年该系统多次蝉联世界超算存储IO500榜单第一名；发表论文80余篇。

端侧具身智能系统:挑战与展望



On-Device Embodied Intelligence System: Challenges and Opportunities

徐梦炜/Xu Mengwei, 徐大亮/Xu Daliang

(北京邮电大学计算机学院(国家示范性软件学院), 中国 北京 100876)
(School of Computer Science (National Pilot Software Engineering School), Beijing University of Posts and Telecommunications, Beijing 100876, China)

DOI: 10.12142/ZTETJ.202604008

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260828.1141.004>

网络出版日期: 2026-08-28

收稿日期: 2026-06-10

摘要: 端侧具身智能将具身大模型从云端远程推理下沉至设备本体并实现闭环执行, 是机器人进入真实物理世界的关键系统基础。揭示视觉-语言-动作模型(VLA)与世界-动作模型(WAM)两类模型在多模态感知、扩散动作生成及世界模型滚动中产生的高计算与高访存开销, 与端侧硬件在10~100 Hz控制频率、有限功耗及网络不稳定条件下可靠运行之间的系统性矛盾。针对该矛盾, 提出以动作成功率、闭环时延与能效为共同优化目标, 围绕少步化生成、动作感知压缩、缓存复用、异步分块执行与专用运行时等策略, 开展算法与系统层面的协同设计。

关键词: 端侧智能; 具身智能; 端侧推理优化

Abstract: Endowing embodied intelligence on the device side shifts large embodied models from cloud-based remote inference to on-device closed-loop execution, serving as a crucial system foundation for robots to operate in real physical environments. Two representative model families—vision-language-action models (VLA) and world-action models (WAM)—introduce high computational and memory-access overheads through multimodal perception, diffusion-based action generation, and world model rolling. These overheads create a systematic contradiction with the requirements of end-side hardware, which must support 10–100 Hz control frequencies, operate under limited power budgets, and maintain reliability despite unstable network conditions. To address this contradiction, action success rate, closed-loop latency, and energy efficiency are adopted as the joint optimization objectives. Algorithm-system co-design strategies are proposed, centering on few-step generation, action-aware compression, cache reuse, asynchronous chunk execution, and specialized runtimes.

Keywords: on-device intelligence; embodied intelligence; on-device inference optimizations

引用格式: 徐梦炜, 徐大亮. 端侧具身智能系统: 挑战与展望 [J]. 中兴通讯技术, 2026, 32(4): 47–55. DOI: 10.12142/ZTETJ.202604008

Citation: Xu M W, Xu D L. On-device embodied intelligence system: challenges and opportunities [J]. ZTE technology journal, 2026, 32(4): 47–55. DOI: 10.12142/ZTETJ.202604008

莫拉维克悖论指出, 人类感到困难的高级抽象推理, 对计算机而言往往轻而易举; 而儿童都能完成的感知、运动及物理交互, 对机器来说却极具挑战。过去十余年, 深度学习尤其是大模型技术在语言、视觉、代码等数字空间任务中取得了显著突破, 但仍难以使机器人在开放物理环境中可靠地完成抓取、导航、协作、维护等基础实体操作任务。随着人工智能应用从屏幕、文本及云端服务扩展至家庭、工厂、交通、医疗和空间设施等真实场景, 面向物理世界交互的具身智能正成为人工智能下一个十年的核心挑战。具身智能系统需在连续观测、有限算力和实时约束条件下, 将模型能力转化为可执行动作, 其成效直接决定大模型能否从数字

助手发展为物理代理。

近年来, 具身智能模型架构快速演进, 形成了两种具有代表性的技术路线: 第一类为视觉-语言-动作模型(VLA), 以视觉-语言模型为主干, 将相机观测、语言指令及历史状态统一编码, 直接映射为机器人动作; 第二类为世界-动作模型(WAM), 以预测未来状态(如图像、隐空间特征或其他表征形式)为核心训练目标。该模型继承世界模型的思想, 并针对真实机器人的动作生成需求进行了专项优化。这两类模型共同推动具身智能从任务专用策略向通用基础模型演进, 使机器人有望在更复杂、更开放的场景中理解指令, 预判环境变化并生成连续动作。然而, 具身智能从实验室走向真实部署, 仍面临系统效率方面的关键挑战。机器人在与物理世界实时交互过程中需持续决策, 网络抖动将直接导致控制性能退化、断连, 甚至可能引发任务失败。为此, 具身智能正从依赖云端大模型的远程推理, 转向模型在设备本体

基金项目: 国家自然科学基金项目(62522202)

的就地部署与闭环控制。然而，机器人实现稳定控制通常需要 10~100 Hz 的控制频率，而参数量达数十亿的 VLA 模型及基于视频扩散的 WAM 模型，在端侧计算平台上往往难以满足这一推理效率要求。基于此，本文从算法与系统两个层面面对高效端侧具身智能系统展开讨论与展望，二者的关键技术及其相互关系如图 1 所示。

1 模型架构与瓶颈

端侧具身智能主要由两类大模型驱动：一类是以视觉-语言模型为主干，将观测与指令直接映射为动作的 VLA；另一类则通过预测未来观测来支撑规划与控制的 WAM。二者的推理架构与计算特性不同，所面临的端侧瓶颈亦各有侧重。本节首先概述两类模型的执行流程，继而剖析其各自存在的瓶颈，为后续各层优化提供依据。

VLA 的推理流程如下：一个 VLA 模型通常由视觉编码器、视觉-语言主干网络及动作头（action head）级联组成。推理时，机器人当前的视觉观测首先由编码器转换为视觉词元，这些词元与经分词处理的语言指令共同输入主干网络，通过一次前向计算（即预填充阶段）获得融合场景理解与指令意图的上下文表征；随后，动作头以该表征为条件，生成低层控制指令，通常以动作分块的形式一次性输出未来若干步动作并按序执行。就动作头的实现方式而言，早期方案将连续动作离散化后以逐词元自回归方式生成；近年来的高性能模型则多采用基于扩散（diffusion）或流匹配（flow

matching）的动作专家（action expert）。这类方法从随机噪声出发，通过多步去噪迭代生成连续动作向量，因其在连续控制与多模态动作分布建模方面的优势，正逐渐成为主流。VLA 和 WAM 的模型架构及推理过程如图 2 所示。

VLA 的端侧瓶颈主要体现在以下几个方面，且彼此关联，共同制约系统效率。首先，视觉编码器与主干网络承担了推理过程中的主要算力开销：单帧图像通常被展开为数百至上千个视觉词元，其数量远超语言与动作词元，导致预填充阶段的注意力计算与前馈网络运算量显著偏大。由于该阶段每次访存均伴随大量浮点运算，性能主要受限于算力而非访存带宽，因此基于 roofline 模型可判定，视觉编码与主干部分属于计算受限型。

相比之下，动作专家的多步去噪过程在每一步均需反复读取模型权重与键值缓存，而单步运算量相对有限，这导致其性能瓶颈转向内存带宽。本文在两类端侧图形处理器（GPU）上的实测结果（如表 1 所示）进一步表明：在 Jetson Thor 上， $\pi 0.5$ 模型的动作专家时延占比约为 38%；而在消费级 RTX 5090 上，GR00T N1.7 模型的动作头时延占比升至 59.2%，但其模型算力利用率仅为 13.2%，而主干部分则达到 33.0%。主干网络加速越充分，动作生成环节的相对开销反而越突出，单纯提升峰值算力无法有效消除这一瓶颈。

这一计算受限的主干网络与访存受限的动作专家并存的两阶段结构，直接导致了控制频率与模型推理时延之间的失配：未经优化的 3B~7B 参数级 VLA 在边缘平台上通常仅能

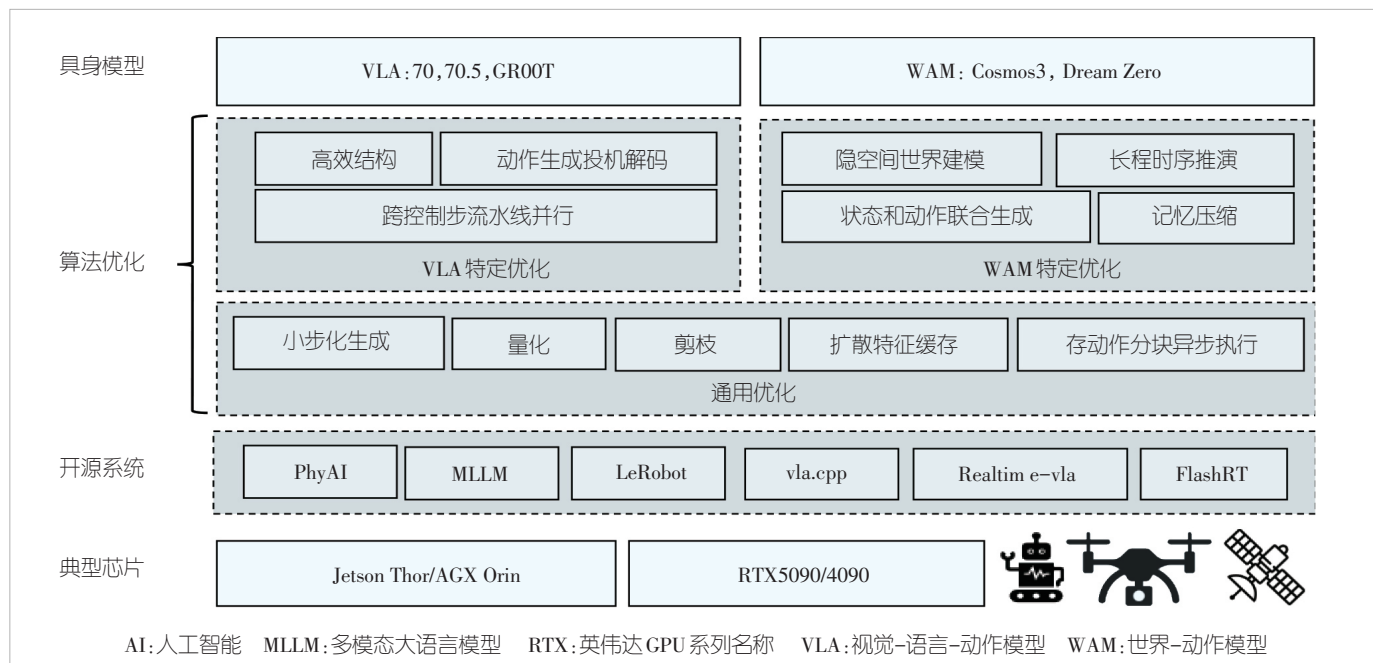


图1 端侧具身智能算法、系统与芯片的全栈协同总览

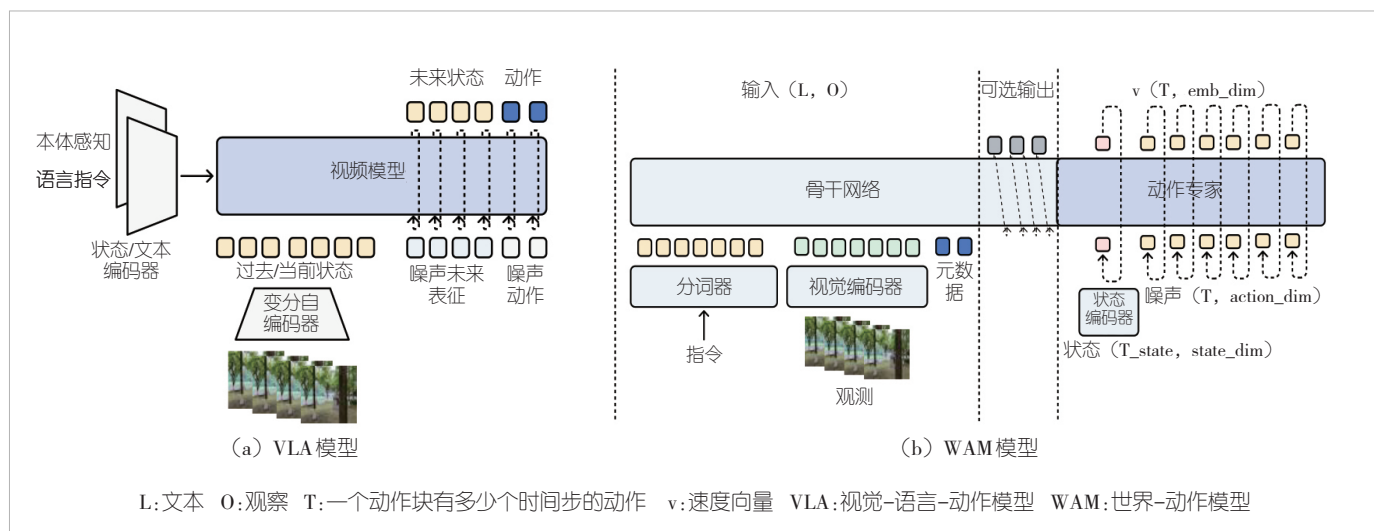


图2 VLA和WAM模型架构及推理过程

表1 $\pi 0.5$ 、GR00T N1.7 与 Cosmos3-Nano-Policy 的阶段剖析与瓶颈定位(批大小为1)

指标	$\pi 0.5$ (Jetson Thor)	GR00T N1.7 (RTX 5090)	Cosmos3-Nano (Jetson Thor)
端到端时延/ms	124.6	20.5	8 121
等效频率/Hz	8.00	48.80	0.12
主干/条件阶段/ms	76.8 (61.6%)	8.3 (40.8%)	1 824 (22.5%)
动作头/生成阶段/ms	47.8 (38.4%)	12.1 (59.2%)	6 298 (77.5%)
该阶段算术强度/ (FLOP·byte ⁻¹)	50.8	135.8	926.2
设备性能瓶颈拐点/ (FLOP·byte ⁻¹)	450.4	139.4	448.9
该阶段模型算力 利用率/%	7.6	13.2	65.8
瓶颈判定	访存受限	访存受限	计算受限

FLOP:浮点运算量 RTX:英伟达GPU系列名称

达到约 6~19 Hz 的处理频率。本文在典型端侧设备 Jetson Thor 上对当前最先进的开源模型 $\pi 0.5$ 进行了性能度量分析,实验数据如表 1 所示。结果表明,官方参考实现的等效推理频率仅为 5.6 Hz,即使经过系统优化,也仅提升至 9.3 Hz,均远低于灵巧操控任务所期望的 30 Hz 目标频率。

此外,预填充阶段与动作生成阶段所关注的词元并不一致:前者侧重于全局场景理解,后者则聚焦于与当前操作密切相关的局部区域。因此,若将面向通用视觉-语言模型设计的视觉词元剪枝方法直接迁移至 VLA 场景,往往会导致动作所依赖的关键词元被误删,进而损害任务成功率。这一

现象要求端侧 VLA 的压缩策略必须具备动作感知能力。

WAM 的推理流程与瓶颈如下。与 VLA 不同,世界模型在具身场景中两种典型用法:一是作为动作生成器,在预测未来的同时直接输出动作;二是作为可闭环滚动的环境模拟器,对候选动作序列进行前瞻推演并择优。无论采用何种方式,其计算代价均集中于未来预测的生成环节。未来帧或隐状态需经多步扩散采样方可获得,单次预测的开销已较为高昂;面向规划的长程滚动生成则进一步放大这一代价,一方面使内存占用随滚动步数线性增长,另一方面逐步累积预测误差,二者共同限制了可用的规划深度。此外,在像素空间逐帧生成的代价远高于在隐空间中进行滚动生成。

本文对 Cosmos3-Nano 的实测结果亦显示,在端到端时延中,单次动作分块生成所包含的 4 步去噪循环占比约 78% (Jetson Thor, 见表 1)。与 VLA 的动作专家不同, WAM 的生成模块在批大小为 1 时即已呈现计算受限特征,其算术强度约为 926 FLOP/byte, 远高于 Jetson Thor 的性能瓶颈拐点(约 450 FLOP/byte),模型算力利用率达 65.8% (图 3 (b))。因此,其时延主要由多步去噪的原始算力需求决定,单纯提升访存带宽的收益有限。更为严峻的是显存约束:该模型单次推理的峰值显存已达 30 GiB 量级,在 32 GB 的消费级 RTX 5090 上因显存溢出而无法运行。此外,当前能够实现实时滚动的世界模型仍多依赖服务器级 GPU,如何将其压缩至端侧设备并在控制节拍内完成滚动生成,仍是一个亟待解决的关键问题。

2 高效算法

高效算法在不依赖专用硬件的前提下降低推理成本,既

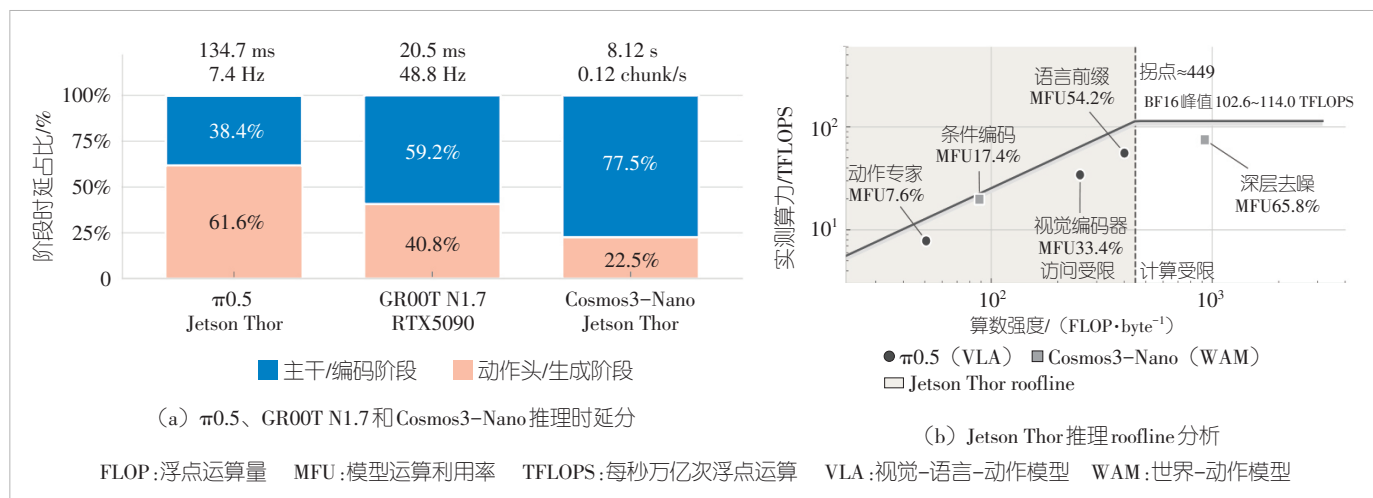


图3 $\pi 0.5$ (VLA)、GR00T N1.7(VLA)与 Cosmos3-Nano(WAM)在 Jetson Thor 与 RTX 5090 上的阶段时延构成及 Jetson Thor 上推理的 roofline 分析

包括直接对模型自身进行改动的方法,也包括对推理执行过程进行改造的方法,二者均属于算法层面的手段。鉴于 VLA 的扩散动作专家与 WAM 的视频扩散 Transformer (DiT) 均以迭代式扩散为计算核心,本节首先论述二者共享的扩散生成加速策略,随后分别讨论 VLA 与 WAM 各自利用具身结构特征的专有优化方法。

2.1 通用算法优化

1) 少去噪步生成。扩散与流匹配生成过程需经多步迭代去噪,推理时延随步数增加而近似线性增长,因此压缩采样步数是最直接的加速途径。一致性流匹配蒸馏方法将多步教师模型压缩为单步或少步学生模型,在尽量保持动作生成质量的同时显著提升推理速度。例如,OneDP^[1]以最小化扩散链 KL 散度得分差为训练目标,将多步扩散策略蒸馏为单步生成器,在 NVIDIA V100 GPU 上使动作预测频率由原来的 1.5 Hz 提升至 62 Hz;FreqPolicy^[2]进一步考虑动作本质为时序信号,在频域施加一致性约束以保障单步生成的动作质量;Streaming Diffusion Policy^[3]则维护一个含变噪声的动作缓冲区,仅对当前待执行动作完全去噪,后续动作保留噪声状态,从而以较少计算量产出可执行动作。

2) 降低权重或激活精度。量化是另一条与减步正交的加速途径。面向视频 DiT 的后训练量化方法(如 ViDiT-Q、PTQ4DiT)可将权重与激活值压缩至 4~8 bit,从而有效削减访存开销;而 VLA 侧的量化需与控制任务紧密耦合,须以操控成功率为核心约束条件。

3) 扩散特征缓存。在迭代去噪过程中,相邻步骤(或相邻生成块)的中间特征通常高度相似,复用这些特征可在不重新训练模型的前提下跳过冗余计算。需要指出的是,通

用扩散缓存方法因架构差异难以直接迁移至策略与世界模型,因此已有研究提出了针对性设计。例如,块级自适应缓存(BAC)^[4]为扩散策略设计了块级自适应缓存机制,并指出不同时间步与网络块之间的特征相似性呈非均匀变化。

4) 动作分块异步执行。动作分块虽能提升吞吐量,但会在分块边界引入响应滞后:生成下一分块所需的推理时延,可能导致机器人在动态环境中无法及时响应。异步执行策略的基本思路是,在执行当前分块的同时生成下一分块,从而将推理时延隐藏于执行过程之中。由于该方法仅涉及分块的生成与执行调度,可适用于任意输出动作分块的策略,因此具有良好的通用性。其代表性工作实时分块执行(RTC)^[5]将该问题形式化为图像修复问题:将已承诺执行的动作冻结为已知量,其余部分作为待修复的未知量,通过引导项确保衔接平滑。该方法无须训练,可适配任意扩散或流匹配策略,能够容忍超过 300 ms 的推理时延。

2.2 VLA 模型特定优化

VLA 模型特定优化的共性,在于显式利用机器人时序连续性、动作相关性等结构化特征,可归纳为高效架构、动作感知压缩、动作生成加速与推理执行级加速 4 类。

1) 高效架构。此类工作针对主干网络与编码器的高开销问题,从多条路径对模型结构进行重塑。为缓解自注意力的二次复杂度,RoboMamba^[6]以线性复杂度的状态空间模型替代标准注意力主干;为降低整体模型规模,TinyVLA、SmolVLA 等采用亚十亿参数级主干网络,并配合扩散动作专家实现轻量化;针对第 1 节中指出的视觉词元主导预填充阶段计算量的问题,FastVLM^[7]从编码器设计层面入手,直接削减视觉词元数量,使首词元时延显著下降。

2) 动作感知压缩。与通用模型压缩不同,这类方法以操控成功率而非重建精度作为评价准则,将有限的比特或词元预算优先分配给对动作生成最为关键的部分。在量化方面,基于显著性的量化模仿学习(SQIL)^[8]以动作显著性与量化鲁棒的行为克隆损失为依据,将精度预算集中于对控制性能影响较大的关键状态,使得四比特模型在Jetson GPU上提速约2.5倍,且性能近乎无损;BitVLA^[9]进一步采用原生三值权重,模型内存开销仅为1.4 GB,相当于OpenVLA原始模型的1/11。在剪枝方面,VLA-Pruner^[10]针对预填充阶段与动作解码阶段注意力分布存在差异的语义-动作鸿沟,对每个词元同时按语义重要性与动作相关性进行双级评分,从而在删减冗余的同时保留动作真正依赖的关键词元。

3) 动作生成加速。该方法直接缩短动作专家(或自回归动作头)的解码过程。一类工作聚焦于减少迭代步数或实现并行化解码,例如CEED-VLA^[11]通过一致性蒸馏与早退策略改造Jacobi解码,在NVIDIA H100 GPU上将OpenVLA模型的推理加速约4.1倍。另一类研究则改进动作的表示与解码形式,例如OpenVLA-OFT^[12]以并行解码、动作分块以及连续动作表示替代原有的自回归离散生成方式,在NVIDIA A100 GPU上实现约26倍加速,并在ALOHA真机上达到25 Hz的动作控制频率;FAST^[13]则通过离散余弦变换压缩动作词元序列,从表示层面降低生成代价。

4) 推理执行级加速。除对模型结构与权重进行改动外,现有方法还致力于在不改变模型本身的前提下,重塑解码与服务执行流程。投机动作解码通过草稿-验证机制缓解逐词元生成带来的串行瓶颈,Spec-VLA^[14]与FLASH^[15]分别覆盖自回归与扩散两类动作头,后者借助动作专家并行验证流一致性,在单张RTX 4090D GPU上取得约3.04倍加速。键值(KV)缓存复用利用帧间冗余降低计算开销,如VLA-Cache^[16]复用相邻帧中变化极小的静态视觉词元的键值表示,并选择性重算任务相关词元,从而削减每步视觉处理开销,在单张RTX 4090 GPU上对OpenVLA模型进行推理时最高取得约1.7倍加速。跨请求流水线方法ActionFlow^[17]进一步将访存受限的解码阶段与计算受限的预填充阶段跨控制步进行打包调度,使算力与带宽资源同时趋于饱和。

2.3 WAM模型特定优化

WAM的端侧效率,归根结底取决于如何以较低计算代价从世界模型中生成动作。本文从两个正交维度刻画其动作生成范式:其一为预测与动作在时序上的耦合方式,可分为先完成未来预测再据此导出动作的顺序模式,以及在同一生成过程中同时产出预测与动作的联合模式;其二为动作所依

赖的隐状态成熟度,即依赖于完全去噪后的预测结果,还是依赖于尚未完成去噪的中间表征。在上述二维划分基础上,本节进一步补充两条与之正交的降本路径:采用隐空间表征替代显式像素生成,以及面向长程推演的执行级优化。

1) 顺序范式:从完整状态预测到动作生成。顺序范式遵循先预测、后动作的执行流程。其早期实现依赖完整的未来状态预测,保真度较高但计算开销大,例如Seer^[18]通过预测未来帧,再由逆动力学模型从中导出动作;DreamVLA^[19]则转而预测动态区域、深度与语义等紧凑型世界知识表征,而非重建整幅未来图像,以降低预测成本。

2) 联合范式:状态与动作联合生成。联合范式通过共享的去噪或前向过程同时产出未来状态与动作,使二者能够利用共同的物理动态先验信息。例如,PAD^[20]在单一扩散过程中联合去噪图像与动作,WorldVLA^[21]则以自回归方式统一建模动作与世界模型。该范式的局限在于,动作需等待完整去噪过程结束后方可获得。因此,如何在联合框架内从中间表征提前产出动作,以兼顾信息共享与低时延,迄今尚无成熟的代表性工作,是一个值得探索的开放问题;不过,统一世界模型(UWM)^[22]已为各模态设置独立的噪声时间步,为从中间表征提前生成动作提供了基础支持。

3) 隐空间世界建模:一条更为根本的降本思路是放弃像素级生成,转而在隐空间中完成预测。LAPA^[23]通过无标注视频学习离散隐动作并据此进行预训练,FLARE^[24]则使策略特征对齐未来观测的隐嵌入,而不生成完整的未来像素表征。二者均避免了显式像素级预测,从而有效降低世界模型在表示与生成层面的计算负担。

4) 长程时序推演与记忆压缩:在执行层面,流式滚动与长程记忆通过联合去噪、注意力汇聚及历史记忆压缩,抑制长程误差累积,并管理日益膨胀的键值缓存。代表性工作Rolling Forcing^[25]同时采用了3项设计:以联合去噪方式对窗口内多帧施加递增噪声,一次性更新整个窗口而非逐帧迭代采样,从而放松相邻帧间的严格因果约束;以注意力汇聚长期保留首帧的KV状态作为全局锚点,并调整其旋转位置编码偏移,使相对位置保持有界;在显著加长的去噪窗口上施行少步蒸馏,以自生成历史为条件缓解暴露偏差。

3 高效端侧具身推理系统

面向端侧具身负载的推理引擎与运行时近年来集中涌现,开源生态已初具规模:既有面向物理智能模型进行全栈优化的专用框架,也有从端侧大语言模型演进而来、覆盖多种异构后端的通用底座,还有由机器学习社区维护的端到端参考实现。本文在Jetson Thor与RTX 5090两类端侧GPU

上,对上述运行时承载 $\pi 0$ 、 $\pi 0.5$ 、GR00T N1.7与Cosmos3–Nano4个典型具身模型的单请求时延进行了系统性度量,结果见表2;各运行时所支持的模型族、可用精度、硬件后端及端云协同形态等方面的差异见表3。

1) PhyAI^[26] (github.com/mingti-org/phyai): 面向物理智能模型(VLA和WAM)的高性能推理框架。其核心特点是以同一套代码路径贯通端侧本体、边缘服务器与云端:模型评测、云端强化学习回放、边缘GPU服务与机器人本体部署共用同一份检查点与同一条推理路径,无须为每种场景单独编写推理程序。该框架支持NVFP4(NVIDIA 4位浮点格式)/INT4(4位整数)等低比特推理优化,并实现了DiT缓存、算子融合、视觉–语言主干与扩散动作专家分离部署等关键技术;模型适配层与运行时相互解耦,新架构只需新增一个适配器即可接入,调度、内存管理、算子选择与并行服务则由运行时统一提供。在Jetson Thor上,该框架将 $\pi 0.5$ 的

单请求时延由官方参考实现的179.5 ms压缩至107.7 ms(约9.3 Hz,加速1.67倍),已逼近操控所需的10 Hz下限;在消费级RTX 5090上进一步降至28.6 ms(约35 Hz),达到灵巧操控所需的实时控制频率区间。

2) mllm^[27] (github.com/UbiquitousLearning/mllm): 面向移动端的多模态大模型推理引擎,以C/C++实现,不依赖重型运行时,支持W4A8、W8A8与W4A16等多档权重与激活量化方式。其突出优势在于后端覆盖广泛:除统一计算设备架构(CUDA)与OpenCL外,还支持高通Hexagon神经网络处理单元(NPU)、昇腾NPU与Arm CPU,这些正是其余框架尚未覆盖的端侧异构算力资源。在该引擎基础上进一步提出的llm.npu系统^[32],通过提示分块、离群值提取与块级乱序调度,化解了NPU对动态形状与激活离群值的不友好特性,将十亿级参数模型的预填充阶段提升至每秒超过1 000词元。目前,mllm尚未包含VLA/WAM模型,因此在表2中

表2 端侧具身智能推理系统在GPU上的单请求时延性能对比¹

模型	设备	LeRobot/ms	PhyAI/ms	FlashRT/ms	vla.cpp/ms	realtime–vla/ms	mllm/ms
$\pi 0$	Jetson Thor	321.0	121.5	61.0	267.8	116.5	NA
	RTX 5090	127.5	31.5	22.2	61.6	32.7	NA
$\pi 0.5$	Jetson Thor	179.5	107.7	86.9	309.1	123.8	NA
	RTX 5090	52.1	28.6	28.5	54.2	27.5	NA
GR00T N1.7	Jetson Thor	133.5	95.4	65.8	164.6	NA	NA
	RTX 5090	47.0	20.6	19.3	32.7	NA	NA
Cosmos3–Nano	Jetson Thor	NA	8 121.4	8 027.5	NA	NA	NA

GPU:图形处理器

表3 端侧具身智能推理系统对比²

系统	支持模型	支持精度		硬件后端	是否支持端云协同推理
PhyAI ^[26]	VLA + WAM	BF16、FP8、INT8、NVFP4 / MXFP4 / INT4		Jetson Thor、RTX 5090/4090、A40/A100、H20/H100	是(DP / TP / CFG并行)
mllm ^[27]	通用多模态大语言模型	W4A8、W8A8、W4A16		Arm CPU、高通Hexagon NPU、昇腾NPU、OpenCL、CUDA	否
LeRobot ^[28]	VLA + WAM	FP32 / BF16		Jetson Thor、RTX5090/4090、A40/A100、H20/H100	是
vla.cpp ^[29]	VLA	W2A8		Jetson Orin Nano/Thor	是
realtime–vla ^[30]	VLA	BF16		Jetson Thor、RTX 4090/5090	否
FlashRT ^[31]	VLA + WAM	BF16、FP8、NVFP4、INT8 / INT4		Jetson Thor/Orin、RTX 5090/4090	否

BF: BrainFloat16位浮点格式
CFG: 无分类器引导
CUDA: 统一计算设备架构
DP: 数据并行
FP: 浮点数
INT: 整数
MLLM: 多模态大语言模型
MXFP: 微缩放浮点格式
NPU: 神经网络处理单元
NVFP: 英伟达浮点格式
RTX: 光线追踪极致纹理
TP: 张量并行
VLA: 视觉–语言–动作模型
WAM: 世界–动作模型

1 性能评测依据各项目官方仓库(截至2026年8月2日版本)本地测试所得;NA表示该运行时不支持该模型或无可用数据。

2 软件能力依据各项目论文及官方仓库截至2026年8月2日的公开状态核对。

其各项时延均记为不适用 (NA); 但其可作为承载 VLA 视觉-语言主干、并向 NPU 侧迁移的通用底座。

3) LeRobot^[28] (github.com/huggingface/lerobot): HuggingFace 维护的端到端机器人学习开源框架, 面向真实机器人场景, 兼容低成本与边缘硬件, 提供检查点加载、数据处理、训练代码、参考实现行为以及通用的异步策略推理栈, 已集成 SmolVLA、基于 Transformer 的动作分块 (ACT)、DiT、 $\pi 0$ 等可直接在端侧运行的模型。该框架的定位覆盖模型全生命周期并提供参考实现, 而非追求算子级的极致优化, 因此以 FP32/BF16 参考实现为主。本文实测其 $\pi 0.5$ 推理路径在 Jetson Thor 上的时延为 179.5 ms (约 5.6 Hz), 在 RTX 5090 上为 52.1 ms (约 19.2 Hz)。

4) vla.cpp^[29] (github.com/VinRobotics/vla.cpp): 首个 ggml/llama.cpp 级、原生支持流匹配与扩散动作专家的统一 C++ 运行时, 已覆盖多种 VLA 架构。该运行时支持视觉-语言前缀的键值跨求解复用, 避免在每个去噪步上重复计算前缀; 并采用三值 1.58 bit (W2A8) 权重量化, 以压缩访存与显存占用, 使其能够在 8 GB 显存的 Orin Nano 等紧显存设备上运行; 通过零消息队列 (ZMQ) 客户端/服务端形态支持端云分离部署。本文实测其 $\pi 0.5$ 模型在 Jetson Thor 上的时延为 309.1 ms (约 3.2 Hz), 在 RTX 5090 上为 54.2 ms (约 18.4 Hz), 表明其广泛的可移植性以一定的时延为代价。

5) realtime-vla^[30] (github.com/Dexmal/realtime-vla): 将 $\pi 0.5$ 模型拆解为 24 个通用矩阵乘法 (GEMM) 类算子, 逐级施以 Triton 内核优化, 实现全流式推理, 目前已覆盖 $\pi 0$ 、 $\pi 0.5$ 与 DM0 等模型。该框架不引入量化, 仅依靠 BF16 内核调优与流式调度获得加速效果, 因此精度与参考实现完全对齐。本文实测其 $\pi 0.5$ 模型在 Jetson Thor 上的时延为 123.8 ms (约 8.1 Hz), 在 RTX 5090 上为 27.5 ms (约 36.4 Hz)。

6) FlashRT^[31] (github.com/flashrt-project/FlashRT): 面向 VLA、WAM 等小批量、低时延负载的实时推理引擎, 支持 FP8、NVFP4 及 INT8/INT4 等多档低精度选项, 硬件后端覆盖 Jetson Thor、RTX 系列与 Jetson AGX Orin; 除 VLA 外, 亦已支持 Cosmos3 等 WAM 模型。本文实测其 $\pi 0.5$ 模型在 Jetson Thor 上的时延为 86.9 ms (约 11.5 Hz), 在 RTX 5090 上为 28.5 ms (约 35.1 Hz)。

综合表 2 与表 3 可见, 6 个系统各有侧重。就推理速度而言, 专用运行时在特定模型与硬件组合上确实更快: FlashRT 在 Jetson Thor 上推理 $\pi 0.5$ 与 GR00T N1.7 的时延分别为 86.9 ms 与 65.8 ms, realtime-vla 在 RTX 5090 上推理 $\pi 0.5$ 的时延为 27.5 ms, 均略优于 PhyAI 的对应结果 (107.7 ms、95.4 ms 与 28.6 ms)。但专用运行时的适用面明显更窄:

realtime-vla 仅覆盖 π 系列与 DM0, 其与 vla.cpp 均不支持世界-动作模型 (WAM), 且二者连同 FlashRT 均未提供端云协同推理能力; mllm 虽后端覆盖范围最广, 却尚未包含任何具身模型。相比之下, 在本文测试的 6 个系统中, PhyAI 是唯一在全部 4 个模型上均完成了实测、同时覆盖 VLA 与 WAM 两类架构, 并支持以数据并行 (DP)/张量并行 (TP)/无分类器引导 (CFG) 并行方式进行云侧扩展的框架。其时延与最快的专用运行时处于同一梯队: 在 RTX 5090 上推理 $\pi 0.5$ 时, PhyAI 仅比 FlashRT 慢 0.1 ms, 比 realtime-vla 慢 1.1 ms。因此, 在本文的测试范围内, PhyAI 在覆盖面与时延之间取得了较好的折中——它以一套代码路径与一份检查点统一覆盖端、边、云 3 类部署场景, 避免为不同环境重复实现推理程序, 同时将时延维持在专用运行时的量级水平。

4 未来展望

尽管前文已在算法与系统两个层面梳理出相当丰富的优化手段, 端侧具身智能距离实际部署仍有明显差距: 即便经过优化, 现有系统往往仍难以稳定达到操控所需的 10 Hz 控制频率, 更难以企及操作任务所期望的 30 Hz 甚至更高频率。本文实测结果亦印证了这一差距: $\pi 0.5$ 在 Jetson Thor 上的官方参考实现仅为 5.6 Hz, 经优化后也仅提升至 9.3 Hz; Cosmos3-Nano 则低至每秒 0.12 个动作分块, 且在消费级 RTX 5090 上因显存溢出而无法运行。两类模型的性能瓶颈各有不同: VLA 在机器人必须采用的单样本批处理下, 访存与启动开销成为主要瓶颈, 模型算力利用率在 Jetson Thor 上仅为 7.6%, 在 RTX 5090 上 GR00T N1.7 的动作头也仅为 13.2%; WAM 的生成模块在批大小为 1 时即呈现计算受限特征, 利用率高达 65.8%, 其时延主要由多步去噪的原始算力需求决定。前者需降低访存与启动开销, 后者需减少去噪所需的算力需求。图 4 展示了算法-系统-芯片协同优化的潜在研究方向。

1) 评测层: 构建面向具身的端到端性能评测指标体系。一切优化均以正确的衡量为前提, 而当前对具身大模型推理的评测仍主要沿用传统大模型的时延与能耗指标, 缺乏具身特定的视角。这类指标既未刻画控制频率与动作成功率之间的耦合关系, 也未反映感知-动作闭环条件下的实时性与能效约束, 因而难以有效指导面向操控任务的优化。本文认为, 亟需从具身角度定制测量方法与指标体系, 将控制频率、动作成功率、端到端时延与闭环能效进行联合考量; 唯有评测先行, 后续算法与系统层的优化方能具备明确的优化目标与可比的衡量基准。

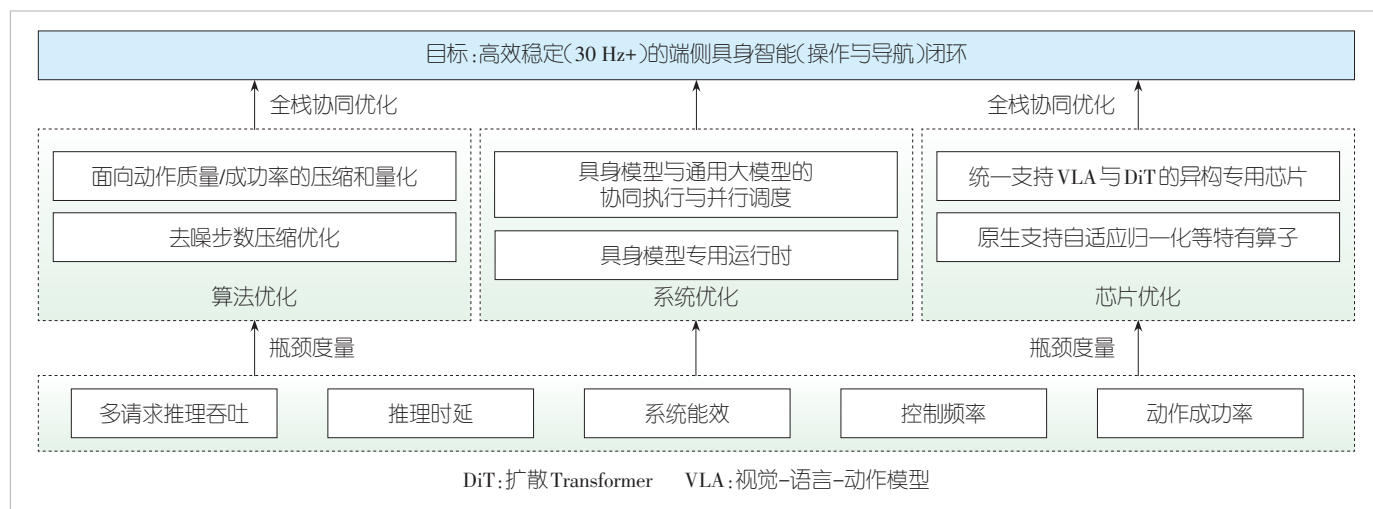


图4 算法-系统-芯片协同优化

2) 算法层: 以动作质量为目的的少步化与压缩。针对上述两类问题, 算法层的优化策略也应有所区分。对访存受限的 VLA 动作专家而言, 收益主要来自两方面: 一是将去噪步数压缩至单步, 二是通过低比特量化降低权重与键值缓存的访存量。对计算受限的 WAM 生成模块, 则应减少或简化去噪步数、压缩隐空间维度, 因为其时延随原始算力需求增长而增加。值得关注的是优化目标的转变: 扩散缓存、量化等技术虽对降低内存占用与提升速度具有重要意义, 但目前其在 VLA/WAM 上的应用仍较为有限, 且大多沿用最小量化误差或重建误差的通用目标。然而, 在具身场景下, 误差最小并不等同于操控最优; 更合理的做法是直接以动作成功率作为优化目标与评判准则, 使压缩与加速过程对动作负责, 而非仅对像素或张量负责。

3) 系统层: 专用运行时与多模型协同。当前, 专用具身推理运行时 (如 vla.cpp、realtime-vla) 尚处于起步阶段, 多数实机部署仍沿用并非为具身场景设计的通用大模型引擎, 缺乏对控制频率约束及感知-动作闭环特性的针对性优化。算法报告的加速效果与其在真实 CPU、GPU、NPU 上的实际执行效率之间仍存在明显差距, 需要通过算法-系统跨层协同设计来缩小这一差距。另一方面, 端侧具身设备上往往不止运行控制模型, 还并存全模态 (OMNI) 式的通用交互大模型, 而现有工作大多仅考虑单模型执行场景。控制模型与交互模型如何在有限的端侧资源上实现协同调度与并行执行, 是一个尚未被充分研究的关键系统问题。

4) 芯片层: 统一支持 VLA 与 DiT 的具身专用加速器。面向具身负载的专用芯片研究尚少, 其核心难点在于异构执行需求: VLA 中视觉-语言主干为计算受限型, 而扩散动作专家为访存受限型且需多步迭代, 二者执行模式差异显著。

如何使同一加速器同时高效支持这两种计算模式, 并原生支持自适应归一化等具身与扩散特有的算子, 是关键的结构问题。总体而言, 该层的研究仍比较有限, 但值得深入探索。

参考文献

- [1] Wang Z D, Li Z S, Mandlekar A, et al. One-step diffusion policy: fast visuomotor policies via diffusion distillation [PP/OL]. arXiv (2024-10-28)[2026-04-09]. <http://arxiv.org/html/2410.21257v1>
- [2] Su Y F, Liu N, Chen D, et al. Freqpolicy: efficient flow-based visuomotor policy via frequency consistency [PP/OL]. arXiv(2026-02-18)[2026-04-08]. <http://arxiv.org/abs/2506.08822>
- [3] Høeg S H, Du Y, Egeland O. Streaming diffusion policy: fast policy synthesis with variable noise diffusion models [PP/OL]. arXiv (2024-06-07)[2026-04-09]. <http://arxiv.org/html/2406.04806v1>
- [4] Ji K Y, Meng Y, Cui H Y, et al. Block-wise adaptive caching for accelerating diffusion policy [PP/OL]. arXiv (2025-06-16) [2026-04-08]. <http://arxiv.org/abs/2506.13456>
- [5] Black K, Galliker M Y, Levine S, et al. Real-time execution of action chunking flow policies [PP/OL]. arXiv(2025-06-09) [2026-04-08]. <http://arxiv.org/abs/2506.07339>
- [6] Liu J M, Liu M Z, Wang Z Y, et al. Robomamba: efficient vision-language-action model for robotic reasoning and manipulation [PP/OL]. arXiv(2024-06-06) [2026-04-08]. <http://arxiv.org/abs/2406.04339>
- [7] Vasu P K A, Faghri F, Li C L, et al. FastVLM: efficient vision encoding for vision language models [PP/OL]. arXiv (2024-12-17) [2026-04-08]. <http://arxiv.org/abs/2412.13303>
- [8] Park S M, Kim H M, Kim S W, et al. Saliency-aware quantized imitation learning for efficient robotic control [PP/OL]. arXiv(2025-05-24)[2026-04-08]. <http://arxiv.org/abs/2505.15304>
- [9] Wang H Y, Xiong C Y, Wang R P, et al. BitVLA: 1-bit vision-language-action models for robotics manipulation [PP/OL]. arXiv (2025-06-10)[2026-04-08]. <http://arxiv.org/abs/2506.07530>
- [10] Liu Z Y, Chen Y Q, Cai H Y, et al. Bridging the semantic-action gap in visual token pruning for efficient VLA inference [PP/OL]. arXiv(2025-11-20) [2026-04-08]. <http://arxiv.org/abs/2511.16449>

- [11] Song W X, Chen J Y, Ding P X, et al. CEED-VLA: consistency vision-language-action model with early-exit decoding [PP/OL]. arXiv(2025-06-16) [2026-06-08]. <http://arxiv.org/abs/2506.13725>
- [12] Kim M J, Finn C, Liang P, et al. Fine-tuning vision-language-action models: optimizing speed and success [PP/OL]. arXiv(2025-02-27)[2026-04-08]. <http://arxiv.org/abs/2502.19645>
- [13] Pertsch K, Stachowicz K, Ichter B, et al. FAST: efficient action tokenization for vision-language-action models [PP/OL]. arXiv(2025-01-17)[2026-04-08]. <http://arxiv.org/abs/2501.09747>
- [14] Wang S S, Yu R C, Yuan Z H, et al. Spec-VLA: speculative decoding for vision-language-action models with relaxed acceptance [PP/OL]. arXiv(2025-07-29) [2026-04-08]. <http://arxiv.org/abs/2507.22424>
- [15] Niu J H, Gu K F, Zhao Y C, et al. Realtime-VLA FLASH: speculative inference framework for diffusion-based VLAs [PP/OL]. arXiv(2026-05-13) [2026-04-08]. <http://arxiv.org/abs/2605.13778>
- [16] Xu S Y, Wang Y K, Xia C H, et al. VLA-cache: efficient vision-language-action manipulation via adaptive token caching [PP/OL]. arXiv(2025-10-21) [2026-04-12]. <https://arxiv.org/abs/2502.02175>
- [17] Dai Y T, Gu H, Wang T, et al. ActionFlow: a pipelined action acceleration for vision language models on edge [PP/OL]. arXiv(2025-12-24)[2026-04-08]. <http://arxiv.org/abs/2512.20276>
- [18] Tian Y, Yang S Z, Zeng J, et al. Predictive inverse dynamics models are scalable learners for robotic manipulation [PP/OL]. arXiv(2024-12-19) [2026-04-08]. <http://arxiv.org/abs/2412.15109>
- [19] Zhang W Y, Liu H S, Qi Z K, et al. DreamVLA: a vision-language-action model dreamed with comprehensive world knowledge [PP/OL]. arXiv(2025-07-07) [2026-04-08]. <http://arxiv.org/abs/2507.04447>
- [20] Guo Y J, Hu Y C, Zhang J K, et al. Prediction with action: visual policy learning via joint denoising process [PP/OL]. arXiv(2024-11-27)[2026-04-08]. <http://arxiv.org/abs/2411.18179>
- [21] Cen J, Yu C H, Yuan H J, et al. WorldVLA: towards autoregressive action world model [PP/OL]. arXiv(2025-06-26) [2026-04-08]. <http://arxiv.org/abs/2506.21539>
- [22] Zhu C N, Yu R, Feng S Y, et al. Unified world models: coupling video and action diffusion for pretraining on large robotic datasets [PP/OL]. arXiv(2025-04-03) [2026-04-08]. <http://arxiv.org/abs/2504.02792>
- [23] Ye S H, Jang J, Jeon B G, et al. Latent action pretraining from videos [PP/OL]. arXiv(2024-10-15) [2026-04-08]. <http://arxiv.org/abs/2410.11704>
- [24] Zheng R J, Wang J, Reed S, et al. Flare: robot learning with implicit world modeling [PP/OL]. arXiv(2025-05-23) [2026-04-08]. <http://arxiv.org/abs/2505.15659>
- [25] Liu K H, Hu W B, Xu J L, et al. Rolling forcing: autoregressive long video diffusion in real time [PP/OL]. arXiv(2025-03-24) [2026-04-08]. <http://arxiv.org/abs/2503.18452>
- [26] Wang C H, Xu D L, Cai D Q, et al. PhyAI: real-time physical AI at the edge, scalable rollouts in the cloud [PP/OL]. arXiv(2026-04-08). <http://arxiv.org/abs/2608.03682>
- [27] Yi R J, Li X, Lu Z Y, et al. mllm: fast and lightweight multimodal LLM inference engine for mobile and edge devices [PP/OL]. arXiv(2023-08-30) [2026-04-08]. <https://github.com/UbiquitousLearning/mllm>
- [28] Cadene R, Aliberts S, Capuano F, et al. Lerobot: an open-source library for end-to-end robot learning [PP/OL]. arXiv(2025-02-08)[2026-04-08]. <http://arxiv.org/abs/2502.05637>
- [29] Nguyen K D, Ho H T, Nguyen C T, et al. vla.cpp: a unified inference runtime for vision-language-action models [PP/OL]. arXiv(2026-06-16) [2026-04-08]. <http://arxiv.org/abs/2606.08094>
- [30] Ma Y C, Zhou Y Z, Yang Y H, et al. Running VLAs at real-time speed [PP/OL]. arXiv(2025-10-31)[2026-04-08]. <http://arxiv.org/abs/2510.26742>
- [31] Su L. Execution-state capsules: graph-bound execution-state checkpoint and restore for low-latency, small-batch, on-device physical-AI serving [PP/OL]. arXiv(2026-06-30) [2026-04-08]. <http://arxiv.org/abs/2606.20537>
- [32] Xu D L, Yang L Y, Liu R Q, et al. Fast on-device LLM inference with NPUs [EB/OL]. (2025-08-06) [2026-04-10]. <http://dl.acm.org/doi/proceeding/10.1145/3676642>

作者简介



徐梦炜, 北京邮电大学副教授; 主要研究方向为端侧具身智能算法和系统; 先后主持和参与基金项目10余项, 获得2项科研成果奖; 发表论文50余篇。



徐大亮, 北京邮电大学副研究员; 研究方向为端侧大模型与具身智能系统; 先后主持和参与基金项目2项; 发表论文10余篇。

城域光网络演进与fgOTN技术的应用



Evolution of Metro Optical Networks and Application of fgOTN Technology

王光全/Wang Guangquan, 董姗/Dong Shan

(中国联通研究院, 中国 北京 100048)
(China Unicom Research Institute, Beijing 100048, China)

DOI: 10.12142/ZTETJ.202604009

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260901.1427.006>

网络出版日期: 2026-08-28

收稿日期: 2026-05-16

摘要: 随着人工智能与产业数字化的深入发展, 智算中心互联及高品质企业上云的业务需求日益增长, 传统城域光网络在带宽、时延、架构及服务感知能力等方面面临严峻挑战。分析了城域光网络关键技术的发展现状, 提出了“极简、扁平、融合”的城域光网络演进架构, 阐述了细粒度光传送网(fgOTN)作为小颗粒高品质专线承载及智算业务接入技术在下一代OTN演进中的关键地位, 探讨了fgOTN的典型应用场景, 并对下一代城域光网络的发展方向进行思考和展望。

关键词: 城域光网络; 网络架构; fgOTN; 应用

Abstract: With the development of artificial intelligence and industrial digitalization, the demand for interconnection among intelligent computing centers and high-quality enterprise cloud access continues to grow. Traditional metro optical networks face severe challenges in bandwidth, latency, architecture, and service awareness capabilities. This paper analyzes the key technology trends in metro optical networks and proposes a "simplified, flattened, and converged" evolution architecture. It elaborates on the important role of fine-grained optical transport network (fgOTN) both as a bearer technology for small-granularity high-quality private lines in the evolution of next-generation OTN and as an access technology for AI computing workloads. The paper also explores the application scenarios of fgOTN and concludes with perspectives on the future development of metro optical networks.

Keywords: metro optical network; network architecture; fgOTN; application

引用格式: 王光全, 董姗. 城域光网络演进与fgOTN技术的应用 [J]. 中兴通讯技术, 2026, 32(4): 56-62. DOI: 10.12142/ZTETJ.202604009

Citation: Wang G Q, Dong S. Evolution of metro optical networks and application of fgOTN technology [J]. ZTE technology journal, 2026, 32(4): 56-62. DOI: 10.12142/ZTETJ.202604009

相较于传统电信业务, 人工智能(AI)业务具备超大规模数据流转、跨数据中心的分布式训练以及边缘推理的实时接入三大显著特征, 城域光网络正面临以下颠覆性挑战:

1) 传输速率瓶颈: 智算互联要求单波 800 Gbit/s 甚至 1.2 Tbit/s 的超高速传输, 但数据中心互联(DCI)的传输距离通常为 80~120 km, 传统的 100 Gbit/s、200 Gbit/s 密集波分复用(DWDM)系统难以经济高效地满足这一需求。

2) 多级架构带来的时延问题: 传统“核心+汇聚+接入”三级架构引入了过多的转接跳数, 且多次光电转换导致AI训练集群间的通信时延与抖动不可控, 严重制约了网络传输效率。因此, 迫切需要改变以光传送网(OTN)为基础的电层组网模式, 转向基于可重构光分插复用器(ROADM)的全光城域组网。该模式不仅能提供基于波长的低时延高速

业务传输, 还可通过合理的光电联合组网, 实现仅在电路源端和宿端进行业务配置, 从而大幅缩短业务开通时间, 有效提升城域光网络的运行效率。

3) 小颗粒业务的管理和运维问题: 传统OTN最初定位于城域网和骨干网^[1], 重点解决了大于 1 Gbit/s 客户信号的高效承载, 但难以高效承载大量 10 Mbit/s~1 Gbit/s 的高价值政企专线、金融交易和工业控制业务。低速业务通常通过同步数字体系(SDH)、无源光网络(PON)等方式接入, 并经过复杂的基于SDH的以太网(EoS)映射及汇聚处理后, 再进入OTN传输通道。传输设备内部集成了光数据单元(ODU)、以太网、虚拟容器(VC)等多种颗粒的交叉能力, 设备卡种类繁多; 多协议层叠加也显著增加了传输设备和网络在应用与维护管理方面的复杂度。因此, 迫切需要一种更为直接、简单的小颗粒业务复用方式, 在提升通道利用率的

同时,通过网络自身的管理能力简化网络架构,提升管控水平。城域光网络亟需进行系统性演进,而细粒度光传送网(fgOTN)技术正是解决小颗粒业务高品质承载问题的关键路径,能够在提高业务承载效率的同时,进一步简化网络并降低运维复杂度。

1 城域光网络演进方向

传统城域光网采用分层架构,业务流量以南北向逐层汇聚为主,适用于业务相对固定的静态网络。而AI时代的城域光网则面临东西向业务流量的急剧增长,网络资源需根据算力业务需求实现面向多种带宽的快速灵活调度,网络架构逐步向扁平化演进。基于ROADM的光层连接需延伸至综合接入点(CO),以满足算力业务对网络品质的更高诉求。

1.1 基于全光底座的网络架构

传统的城域光网络通常采用“核心+汇聚+接入”三级架构,一个数据流从源数据中心到目标数据中心,可能需要经过多级基于不同OTN颗粒的汇聚复用,最终形成基于DWDM或光纤直驱的电路连接。随着大规模数据中心(DC)的部署,基于波长级的业务颗粒逐步成为城域网业务承载的主流方向,也对光层的可用性提出了更高要求。因此,引入ROADM以适配城域核心层网状网及汇聚接入层环形的光纤组网架构,特别是在接入层引入池化波分技术,在实现接入层波长资源共享的基础上,支持端到端波长级业务的调度、穿通或上下路,从而实现从接入到核心汇聚层的全光一级组网,达成任意核心DC间的一跳直达,满足城域内边缘DC与核心DC的分钟级光路直连需求,提升网络的灵活性和可用性。网络架构如图1所示。

池化波分采用简化波长选择开关(WSS)、极简相干线路模块和高灵敏度光层数字化等技术,实现城域接入层全光网络任意站点间的波长直达,提供速率和波长间隔的按需选择能力,在降低成本的同时降低光层复杂度,实现高集成、

低功耗、灵活易用的大带宽接入,适合在城域汇聚接入、国内县乡区域波分等场景中部署。

城域光网络的核心层向Mesh化方向演进,按需开通波长交换光网络(WSON),汇聚层与接入层则趋于扁平化融合。这一架构在实现资源集约化利用的同时,精简了网络层级,使业务能够跨网层在光层直接打通,支持城域内任意站点间的波长直达,从而减少电层中继与转接,有效降低网络时延和建网成本^[2]。网络架构的系统性精简推动从“节点冗余”向“扁平高效”的转变,而资源的盘活与集约化利用则实现了从“存量闲置”到“循环高效”的转变^[3]。面向东西向流量,网络围绕DC为中心进行建设,光交叉连接(OXC)/ROADM覆盖DC站点,并与核心层融合形成Mesh化组网。算力优先集中部署,再通过高品质网络保障低时延传输,以实现算力更广泛的高品质覆盖^[2]。同时,引入数字孪生和AI算力路由,动态匹配最优光缆资源与光层传输能力,实现算力感知,并协同资源供给、业务调度和网络运维。

综上,通过网络架构升级,构建以“极简、扁平、融合”为核心的城域光网络,可实现东西向算力的高效流转与南北向原生数据的灵活入算。

1.2 G.654.E光纤下沉城域网

随着DCI直连带宽达到800 Gbit/s及以上,当前城域网主要采用的G.652.D光纤性能已难以较好匹配800 Gbit/s高速传输需求。实验室及现网测试数据均表明,G.652.D光纤在承载800 Gbit/s光传输设备时,受传输损耗、非线性效应等因素影响,在接收端预留5 dB维护余量的情况下,满波无电中继的实际有效传输距离不足200 km。而对城域数据中心的调研表明,省会及以上城市大部分算力核心节点间的跨段传输距离超过300 km,G.652.D光纤无法满足800 Gbit/s光层ROADM组网的传输需求,需增加电中继器,这大大增加了网络建设和运维成本,制约了高速城域光网络的发展。

G.654.E光纤具备更低的衰减系数(0.17 dB/km)和更大的有效模场面积(110~130 μm^2),可降低非线性效应,提升入纤光功率。该光纤目前主要用于骨干网,以延长100 Gbit/s、400 Gbit/s及以上高速传输系统的无电中继距离,同时能够有效应对800 Gbit/s在城域网多节点多路由ROADM光层组网及光链路损耗较大的传输场景,提升800G系统的无中继传输距离,从根本上解决G.652.D光纤

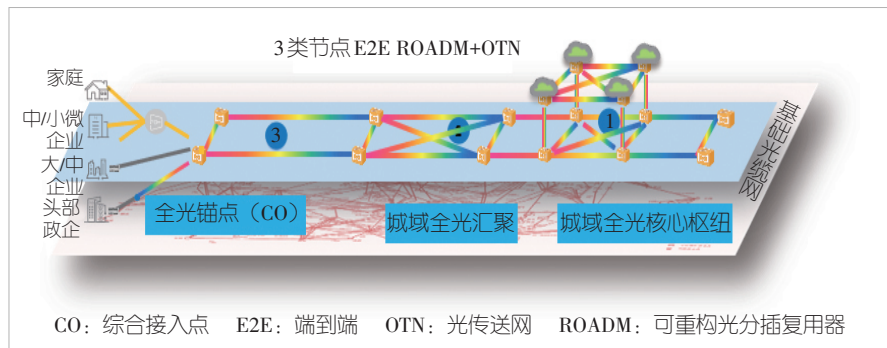


图1 基于全光互联的城域网架构

非线性效应较高的问题。采用G.654.E光纤可实现城域内算力枢纽的高效互联，减少中继站数量，降低传输时延，为算力资源的跨区域调度和AI业务的跨集群协同提供底层光传输支撑。此外，G.654.E光纤支持更宽波段的传输能力，可满足未来1.6T等更高速率系统的波段拓展需求，其性能优势能够支撑城域网未来15~20年的业务发展，避免因光纤性能不足导致的重复建设，从而降低网络全生命周期成本。

1.3 光电融合保护恢复与调度

AI智算互联对城域光网络的可靠性提出了高达99.999%的要求，可采用光层与电层协同、保护与恢复相结合的WSN/自动交换光网络（ASON）技术。未来可引入改进的纯光层WSN保护恢复技术，实现低于50 ms的保护倒换。初期可针对DC间流量采用光电协同WSN/ASON技术，通过光+电协同保护与恢复的方式，提供低成本、高可靠的业务动态恢复能力。后续随着WSS快速切换、光放快速响应、线路模块快速收敛及协议并行硬化等关键技术的突破，可在城域网络中将WSN的波长重路由恢复时间从秒级压缩至50 ms以内。结合Mesh化ROADM/OXC组网架构，无需额外电层资源，即可提升波长级业务可靠性，提供高性价比、高可用率、极简网络的城域OTN算力网络。

ROADM网络更适用于波长直达的业务颗粒，可实现快速的波长级业务调度，但对于快速以太网（FE）、千兆以太网（GE）、10GE等更细颗粒的业务，仍需进行分级的汇聚与复用。AI时代的城域光网络不能单纯依赖全光ROADM或OTN电交叉，而应采用光电融合调度策略，按业务需求动态选择光交换或电交换：100 Gbit/s以上的大颗粒流量通过ROADM光层调度，中小颗粒业务则通过合理配置的OTN或fgOTN设备进行汇聚交换。该模式旨在尽量减少交换层级，在ROADM光层“波长过粗”与OTN电层“ODUflex交换层次过多”之间取得平衡，从而提升整体网络的运行效率。

1.4 一跳入算极简城域网

下一代城域光网络的核心汇聚层采用40/80×800 Gbit/s DWDM超高速传输技术，以满足智算互联对DC间超宽管道的需求，并降低每比特成本与能耗。接入层采用池化波分技术，实现资源共享和业务弹性接入，通过将边缘接入节点的波长资源池化，提升资源利用率。电层接入节点则采用fgOTN、fgOTN点到多点（P2MP）、G.metro纯光接入及FTTx等多种接入技术，支持单波长在城域边缘为多个站点提供独立子通道，无需光层汇聚，从而简化接入层拓扑，构建一跳入算的极简城域光网络。同时，光电融合调度支持按业务需

求动态调整交换模式。一跳入算极简城域网架构如图2所示。

算力下沉至城域推动了波分技术的进一步下沉。通过池化波分技术将边缘接入节点的波长资源池化，可打造全光接入锚点，将光锚点部署于距用户500 m~1 km范围内（如图2所示），从而实现不同场景下用户的快速、灵活、高品质接入。在具体接入方式上，OTN结合P2MP技术可实现中小企业楼宇的覆盖；光纤到办公室（FTTO）技术将光纤从弱电机房延伸至每个房间，实现园区全光覆盖；家庭及其他入算业务可通过OTN对接光线路终端（OLT），并利用光纤到房间（FTTR）技术实现家庭用户的差异化品质承载；高价值企业专线则可采用fgOTN进行端到端承载。

2 fgOTN 技术优势

fgOTN技术是对现有OTN技术在小粒度业务承载方面的扩展，定义了面向业务的细粒度灵活光数据单元（fgODUflex）容器，可提供10 Mbit/s~1 Gbit/s级别（最高支持1 190 Mbit/s）的业务承载能力，并具备完整的层网络功能。相较于传统的多业务传送平台（MSTP）、SDH等技术，fgOTN与OTN之间具有更简洁的映射复用关系，具有明显的代际优势和演进特征。

相较于传统OTN技术，fgOTN在小颗粒业务承载、低时延传输和快速无损带宽调整方面具备显著优势。传统OTN技术的最小管道为ODU0（1.25 Gbit/s），小颗粒业务需借助

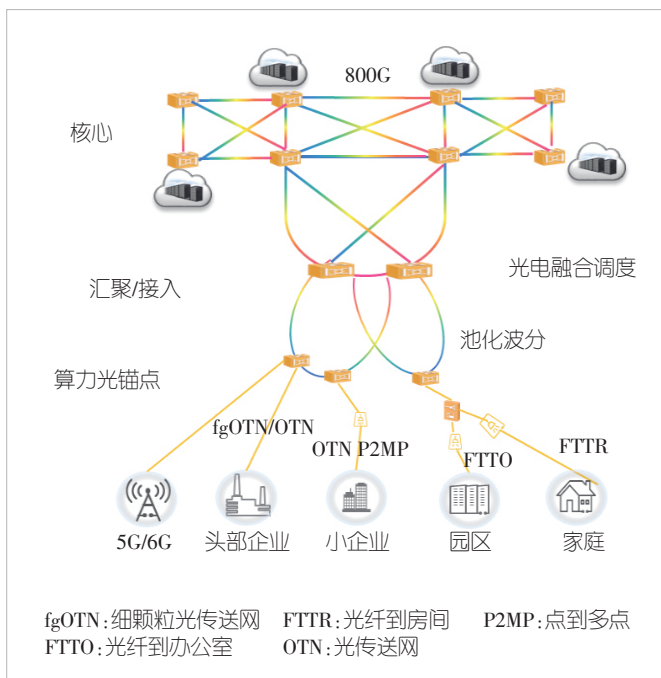


图2 一跳入算极简城域网^[2]

VC技术实现承载，但VC带宽固定，易产生碎片化时隙，且经过多层级适配与复用后，净荷占比低，承载效率不高。相比之下，fgOTN可提供 $N \times 10$ Mbit/s的fgODUflex灵活容器，实现小颗粒业务的高效承载。同时，fgOTN对传统OTN中的无损带宽调整机制（G.HAO协议）进行了全新改良，简化并继承了部分交换协议指示，引入一步调整机制，具备100 ms级别的快速无损带宽调整能力。

当前城域光网络承载的主要业务为FE、GE、10GE等中低速以太网业务。传统OTN网络承载这些业务需经过复杂的EoS（低速IP映射进入SDH，SDH映射复用进入OTN，OTN解复用为中速IP）过程，涉及城域网接入、汇聚、核心等多个层级的网元，主要存在以下问题：

一是为提升网络利用率，存在大量以太网上下电路，无法形成端到端的网络通道级监测与管理开销通道，因而难以实现网络自身的端到端运行、维护和管理（OAM）能力。业务多次上下电路也导致无法形成端到端保护，即便对需要保护的業務，配置也相对复杂。

二是业务需经过多个节点不同层级的交叉调度，增加了业务复用与解复用的次数和时延，也显著加大了网络运维管理的难度。

三是在主要交换节点内部，大量采用SDH VC级联功能以满足以太网业务映射，导致网管系统对设备和业务通道的管理能力降低，设备利用率也明显下降。

fgOTN可有效解决上述问题。fgOTN继承了SDH基于TDM固定时隙的信号结构，可提供端到端的刚性管道，保障业务安全传输，无须经过多层级交换，从而降低业务时延。在管控层面，fgOTN具备业务感知、自动开通和在线无损带宽调整能力，结合单管理业务通道、纳秒级时延测量及串联监测等智能运维手段，可显著降低海量小颗粒业务的管控复杂

度，大幅提升运维效率。在网络运维方面，fgOTN真正实现了端到端的自动化OAM，简化了运维复杂度，推动网络架构从极简向运维极简方向发展。表1展示了分组增强型光传送网（PeOTN）向fgOTN演进各阶段设备与网络特征对比结果。

3 fgOTN应用场景和方案

3.1 fgOTN网络应用位置

fgOTN技术主要面向小颗粒业务，在实际网络应用中，主要用于末端客户节点、本地综合业务接入节点、核心汇聚节点以及OTN专线的云化网络节点，如图3所示。

为同时支持fgODUflex业务（如10 Mbit/s ~ GE分组业务、STM-1/4等，其中STM- N 为同步传送模块， $N=1、4$ ）和基于ODUk的业务（如GE以上分组业务、STM-4以上的TDM业务等），本地OTN网络应同时具备fgODUflex与ODUk的交换能力。在本地OTN网络的核心节点或二干出省节点上，可将发往其他本地网络或其他省份的fgODUflex业务调度并复用至ODUk中，骨干OTN网络可沿用现有OTN架构，仅需支持ODUk交换即可。

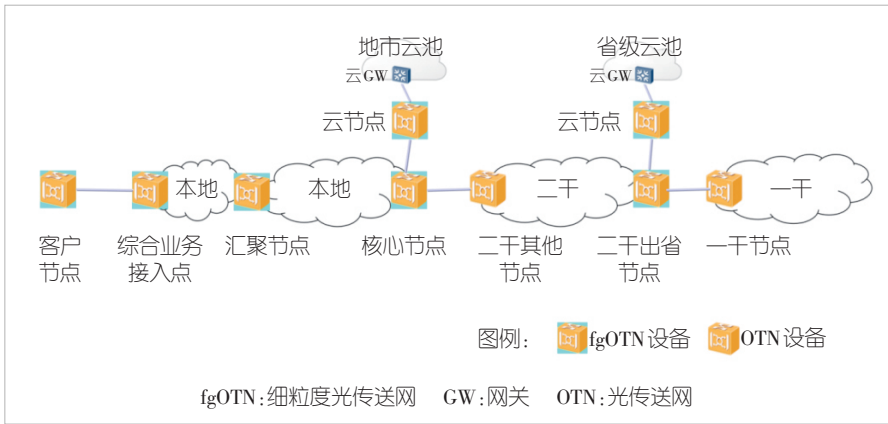


图3 fgOTN设备网络定位

表1 PeOTN向fgOTN演进各阶段设备与网络特征对比

维度	现阶段:PeOTN(VC/Package/ODU)	过渡阶段:PeOTN与fgOTN (VC/Package/ODU/fg)	未来阶段:ODU/fgODUflex
设备架构	OTN、分组、VC共3个交换平面	OTN、分组、VC、fg共4个交换平面	OTN/fgOTN共2个交换平面， 简化设备架构、协议与维护
板卡种类	板卡类别多,复杂度高	板卡类别更多,复杂度更高	板卡归一化,维护简化
业务承载	时延高,2 Mbit/s ~ 155 Mbit/s、 155 Mbit/s ~ 1 Gbit/s两档无损带宽调整	时延低,10 Mbit/s ~ 1 Gbit/s无损带宽调整	时延低,10 Mbit/s ~ 1 Gbit/s无损带宽调整
智能化	模型复杂,ODUk隧道预规划	模型复杂,ODUk隧道预规划	北向业务模型简化
fgODUflex:细粒度灵活光数据单元 fgOTN:细粒度光传送网 ODU:光数据单元 OTN:光传送网 PeOTN:分组增强型光传送网 VC:虚拟容器			

3.2 fgOTN设备架构和部署方案

fgOTN设备可分为统一交叉与二级交叉两种架构。图4展示了核心汇聚层设备在两种架构下的实现方式。统一交叉架构通过同一块交叉板卡实现VC/Ethernet/ODUk/fgODUflex的统一交换，该架构下需新增支持fgOTN的线路侧板卡和客户侧板卡，以实现fgOTN业务的调度与接入；二级交叉架构则通过独立的fgOTN交叉板卡完成fgOTN的调度，该架构可利旧原有OTN线路侧板卡。

统一交叉与二级交叉两种架构均需在综合业务接入点对交叉板卡进行软件升级，并新增线路板卡、支路板卡及客户

端设备，两者的部署差异主要体现在核心汇聚层。统一交叉架构需新增核心汇聚层线路板，部署成本较高，但业务时延较低，线路容量有所提升；二级交叉架构部署成本较低，但设备需经过两次交叉转接，时延略高，且二级交叉板卡接入业务量较小，线路容量保持不变。因此，实际部署中通常需综合考虑建设成本与本地网业务规模：在业务热点城市及建网预算充足的地区，优先选用统一交叉架构；在业务量较小且对成本敏感的地区，优先选用二级交叉架构。

同时，如图5所示，在新建电层场景中，原则上采用统一交叉架构部署fgOTN能力；在利旧电层场景中，若现有OTN链路时隙较足或存在空闲线路侧端口，则优先采用二级交叉架构；若需新增线路板卡，则原则上采用统一交叉架构。两种交叉架构的设备可混合组网。

3.3 政企精品网fgOTN业务开通方案

传统政企网络架构存在组网僵化、承载能力不足、运维效率低等问题，难以支撑高带宽、低时延业务的快速增长^[4]，如图6所示。在当前OTN政企精品网中，省际干线层具备省际大颗粒调度能力，作为大颗粒刚性管道网络，为各省间提供ODUk级别的通达能力，但未配置VC能力；省内干线具备跨地市大颗粒调度能力，并按需配置VC高阶调度能力，用于地市间VC4调度，但未配置VC低阶交叉能力；本地网具备地市内大颗粒调度能力，同时配置VC高低阶调度能力，但在业务量较大的本地网中，核心汇聚层网络资源（如低阶VC交叉利用

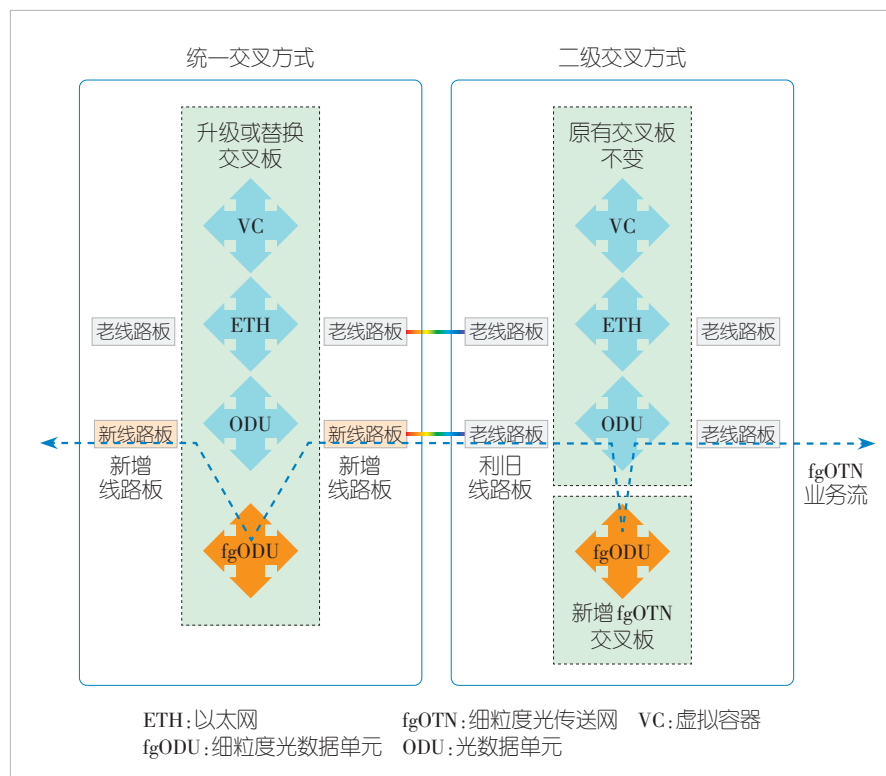


图4 核心汇聚层设备统一交叉和二级交叉架构对比

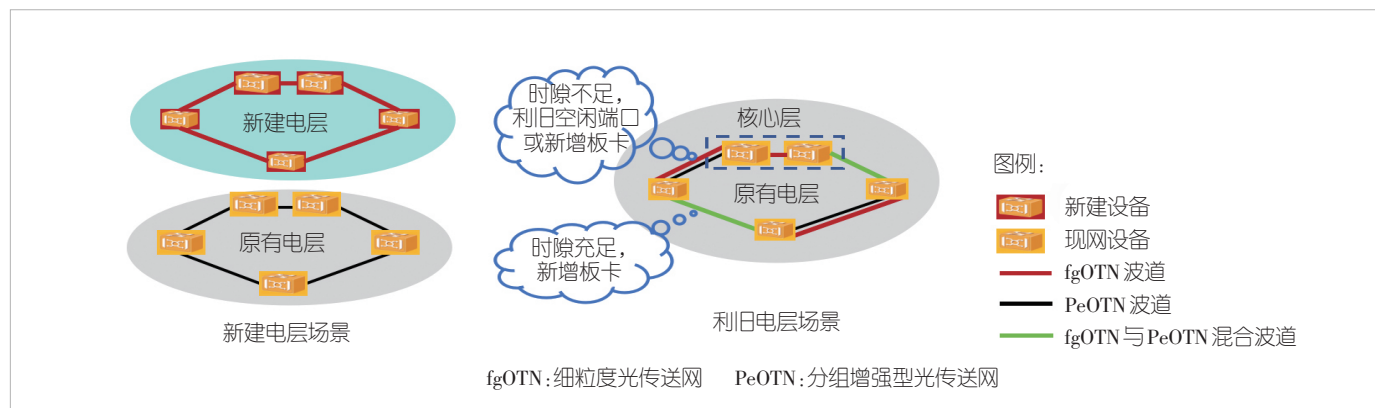


图5 fgOTN新建和利旧电层场景示意图

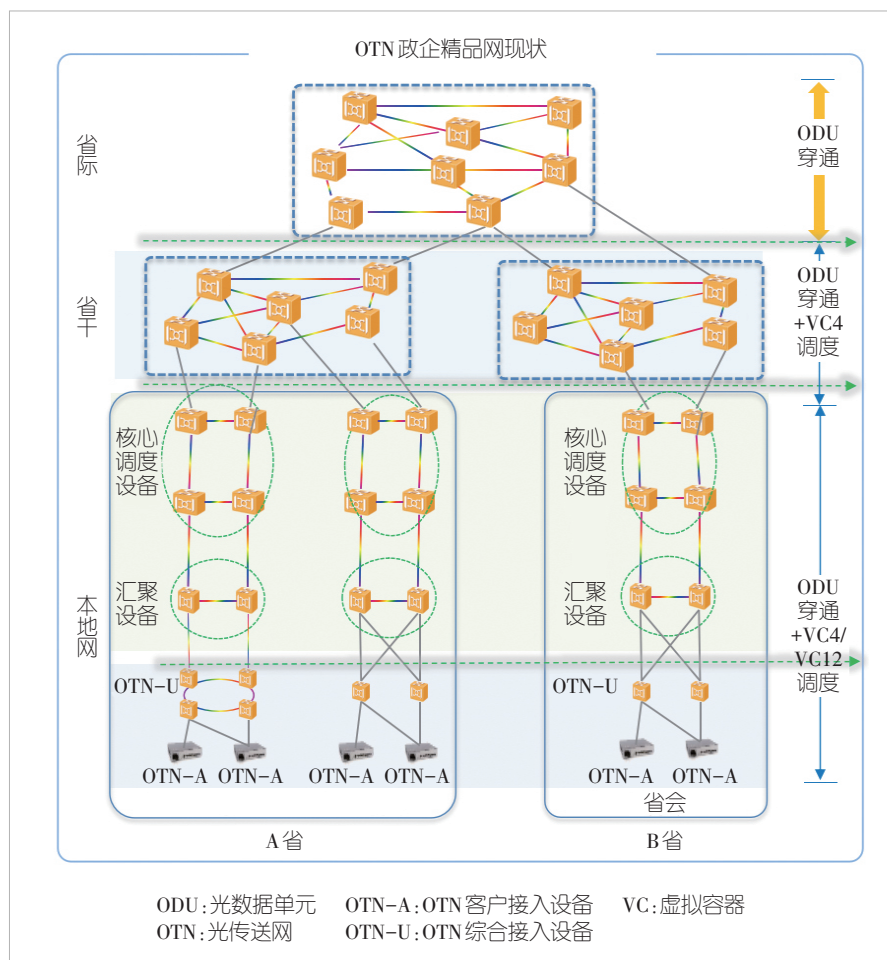


图6 OTN政企精品网现状

率、虚端口配置率等)存在一定瓶颈,难以充分发挥政企OTN网络的整体效能。

图7展示了在政企精品网建设与优化中,本地OTN网络在接入层和核心汇聚层同时引入fgOTN技术实现端到端业务调度的开通方案。在资源利用率方面,10 Mbit/s级别的细粒度时隙通过灵活的时隙复用与交叉调度,可将多条小带宽业务高效汇聚至同一波长通道和高阶ODUk通道中,从而大幅提升带宽利用率,显著节省纤芯和波长资源,提高投资回报率。在业务品质方面,基于fgOTN的管道具备硬隔离和低时延特性,支持端到端时延、丢包等性能感知,有助于提升差异化竞争力。在智能化方面,该方案有利于形成统一的资源池与管控视图,易于维护的同时,为后续引入AI预测性维护、自动弹性调

速等智能化能力奠定基础,且业务智能化开通难度较低。综合考虑以上因素,建议政企精品网在核心汇聚层根据节点功能实现新增业务的fgOTN汇聚与保护,并统筹接入层设备的fgOTN接入及交叉功能引入。

3.4 fgOTN在极简网络中的应用

在“极简网络”的建设过程中,应重点推动“OTN CPE+极简核心”模式的应用。fgOTN在此框架下发挥着关键作用。

在AI边缘节点接入智算中心场景中,网络对时延有极高要求。2025年工业和信息化部办公厅印发《关于开展城域“毫秒用算”专项行动的通知》,明确提出“面向基础设施实现算力中心毫秒互连(<1 ms),面向重点场所实现算力资源毫秒接入(<1 ms),面向应用终端实现算力应用毫秒可达(网络时延<10 ms)”的硬性指标^[5]。AI边缘推理服务器产生的多路1 Gbit/s~10 Gbit/s信号流,可通过fgOTN直接汇聚并映射至骨干OTN网络,无需在边缘侧进行L2/L3交换,从而实现小于50 μs的确定性时延。

在金融高频交易场景中,证券、期货公司需从城区机房直连交易所数据中心,fgOTN可提供10 Mbit/s~1 Gbit/s的物理隔离通道,以刚性管道保障业务安全性。

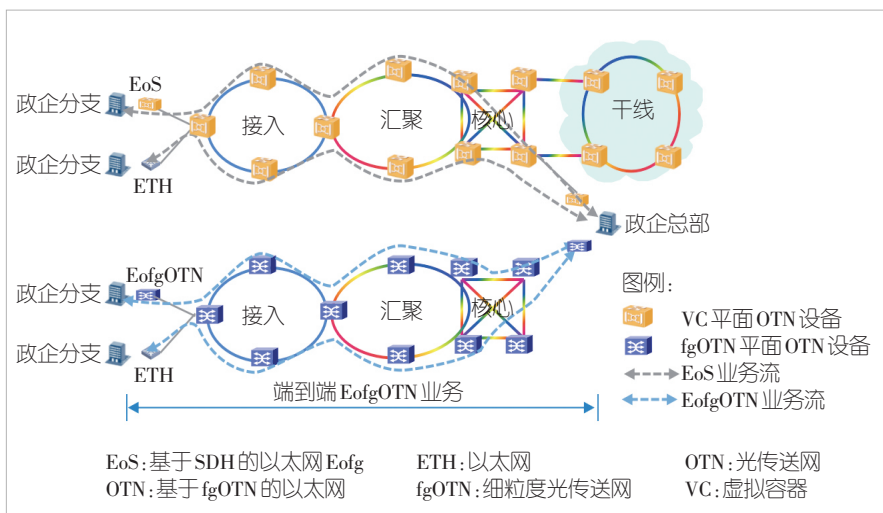


图7 政企精品网fgOTN业务开通方案

在党政军及大企业高品质专线场景中，fgOTN可用于替换老旧SDH、MSTP网络，直接承载E1、STM-1等传统信号，实现SDH类业务向IP/光融合网络的平滑迁移。

4 城域光网络发展思考

基于上述分析，我们对城域光网络未来3~5年的发展提出以下建议：

1) 确立“全光底座+fgOTN顶层”的双层架构：以“800G+ROADM+OXC”构建大带宽、低时延的物理光层底座，以fgOTN为核心构建小颗粒高品质业务承载顶层，逐步替代老旧SDH网络。

2) 加速推进G.654.E光纤在城域核心的规模化部署：新建城域核心光缆直接采用G.654.E光纤，存量G.652.D光纤逐步替换或采用混模传输方式。

3) 推动智能管控平面向AI原生演进：引入数字孪生和AI算力路由，根据智算任务实时负载，动态计算最优光路与fgOTN时隙分配，实现“光算协同”。

4) 坚持标准先行，确保平滑兼容：保障fgOTN与现有OTN网络的混合组网能力，支持业务在传统ODUk与fgOTN时隙间的无损切换，保护运营商既有投资。

AI驱动下的城域光网络正加速向算力感知、品质确定的方向演进。构建“极简、扁平、融合”的城域光网络，将是应对未来智算互联与泛在高品质接入挑战的最优路径，特别是在数据中心内部（DCN）与数据中心间（DCI）实现光传输与光交换的统一。fgOTN作为OTN技术演进的必然方向，精准填补了小颗粒高品质服务的空白，不仅是技术演进，更是产业数字化新型基础设施实现商业成功的关键，对下一代城域光网络的演进，以及数据中心接入（DCA）等多

种接入与承载场景，均具有重要的应用前景。

参考文献

- [1] 瞿少凯, 贺政, 豆晓雷, 等. fgOTN技术及部署策略研究[J]. 长江信息通信, 2025, 38(8): 187-191. DOI: 10.20153/j. issn. 2096-9759.2025.08.056
- [2] 中国联通. AI时代的全光底座白皮书[R]. 2025
- [3] 李俊, 李有胜, 彭纪源. 极简网络在通信运营商中的实施路径与价值重构研究[J]. 江西通信科技, 2025(4): 12-14+44. DOI: 10.3969/j. issn.1009-0940.2025.04.005
- [4] 闫冰, 王玉潇, 武斌, 等. 融合统一的政企专网架构设计与创新实践[J]. 山东通信技术, 2025, 45(2): 16-19
- [5] 工业和信息化部办公厅. 工业和信息化部办公厅关于开展城域“毫秒计算”专项行动的通知[EB/OL]. (2025-10-14)[2026-06-16]. https://www.gov.cn/zhengce/zhengceku/202510/content_7044694.htm

作者简介



王光全，中国联通研究院总工程师、中国联通领军人才、网络通信科学家，教授级高工，享受国务院政府特殊津贴专家；主要研究方向为下一代光网络、下一代互联网和算力网络；主持国家大中型工程设计多项，主持和参与国家“863计划”及重大科研专项多项，曾获2019年WSIS最高奖（C5类）、国家科技进步奖二等奖、国家技术发明奖二等奖、国家优秀设计奖、国家优质工程奖等奖项。



董姗，中国联通研究院光传输研究员；主要研究方向为传输网络、下一代光网络；曾获中国电子学会科技进步奖一等奖、世界互联网大会领先科技奖等奖项。

存算一体——重塑芯片架构新势力



CIM/PIM/PNM: An Emerging Force Reshaping Chip Architecture

刘新阳/Liu Xinyang, 李婷宇/Li Tingyu, 巨鸣/Ju Ming

(深圳市中兴微电子技术有限公司, 中国 深圳 518055)
(Sanechips Technology Co., Ltd, Shenzhen 518055, China)

DOI: 10.12142/ZTETJ.202604010

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260904.1059.002>

网络出版日期: 2026-08-24

收稿日期: 2026-06-02

摘要: 存算一体技术通过将计算单元与存储单元深度融合或物理级邻近部署, 从架构层面缓解了数据搬运瓶颈带来的压力。大语言模型 (LLMs) 及智能体 (Agentic AI) 的持续演进, 对计算架构的高带宽、低延迟与高效能提出了严苛要求。系统梳理了基于静态随机存取存储器 (SRAM)、动态随机存取存储器 (DRAM)、闪存 (Flash) 及新兴非易失性存储器的存算一体技术路径, 涵盖存内计算 (CIM)、存内处理 (PIM) 与近存处理 (PNM) 三大主流范式, 综述其研究进展、典型应用案例及面临的技术挑战。分析表明, 不同技术路径展现出差异化的性能与能效优势, 可通过场景驱动的技术融合、软硬件协同设计及产业链生态协同实现优势互补, 为下一代人工智能芯片的架构创新提供有益参考。

关键词: 大语言模型; 智能体; AI芯片架构; 存算一体

Abstract: Compute-in-Memory (CIM), Processing-in-Memory (PIM), and Processing-Near-Memory (PNM) hold the promise of fundamentally resolving the data movement bottleneck by integrating computing logic with memory units or placing them in close physical proximity. A systematic review is presented of CIM, PIM, and PNM technical paths implemented on static random-access memory (SRAM), dynamic random-access memory (DRAM), Flash, and emerging non-volatile memories, covering their research progress, representative use cases, and associated challenges. The analysis reveals that each technical path exhibits distinct performance and energy-efficiency advantages. It is argued that scenario-driven technology fusion, hardware-software co-design, and industry-wide ecosystem collaboration are essential for harnessing complementary strengths across these paths. This work is intended to serve as a valuable reference for architectural innovation in next-generation AI chips.

Keywords: large language models; Agentic AI; AI chip architecture; computing-in-memory

引用格式: 刘新阳, 李婷宇, 巨鸣. 存算一体——重塑芯片架构新势力 [J]. 中兴通讯技术, 2026, 32(4): 63–71. DOI: 10.12142/ZTETJ.202604010

Citation: Liu X Y, Li T Y, Ju M. CIM/PIM/PNM: an emerging force reshaping chip architecture [J]. ZTE technology journal, 2026, 32(4): 63–71. DOI: 10.12142/ZTETJ.202604010

1 内存墙成为 AI 系统核心瓶颈

1.1 从大语言模型到智能体

近年来, 随着生成式人工智能 (Generative AI) 的快速发展, 以 Transformer 架构为核心的大语言模型 (LLM) 凭借其强大的语言理解、内容生成与推理能力, 成为推动通用人工智能发展的核心引擎。进入 2026 年, AI 应用正从传统聊天机器人 (Chatbot) 的单次问答推理, 向多轮自主智能体 (Agent) 交互范式深度演进。然而, 算法与应用层面的繁荣, 正对底层计算硬件提出前所未有的挑战。在冯·诺依曼存算分离架构下, 经典的“内存墙 (Memory Wall)”问题^[1]被 LLM 推理与 Agent 应用持续放大, 数据搬移而非计算本身成为制约系统性能与能效的核心瓶颈。

大语言模型推理面临以下几个方面的内存墙方面挑战:

1) 带宽瓶颈。在 Transformer 的自注意力计算中, 每个 Token 需与所有历史 Token 进行交互, 这意味着即使参数量相对适中, 其推理过程仍须以极高的带宽频繁访问大规模的权重数据和键值 (KV) 缓存^[2]。

2) 容量瓶颈。大语言模型的全量参数往往远超单一芯片的存储容量, 必须依赖分布式存储与并行计算技术。这不仅增加了系统设计的复杂性, 也带来了严重的芯片间通信开销。

3) 能耗瓶颈。对于大语言模型这类访存密集型的负载, 数据搬运往往主导着系统总能耗^[3]。从 Mark Horowitz 的经典数据^[4]来看, 在 45 nm 工艺下, 一次动态随机存取存储器 (DRAM) 访问的能耗约为一次 64 位浮点运算能耗的 100~1 000 倍。由于工艺微缩对 DRAM 访存能耗的改善远滞后于对计算逻辑能耗的改善, 这一比例在更先进的工艺节点下可能进一步扩大。

如表1所示,智能体不仅是一个被动的对话模型,还具备环境感知、多轮迭代、长期记忆、工具调用及多智能体协作等能力^[5]。这些特性进一步加剧了内存墙问题,主要体现在以下几个方面:

1) 多模态融合带来的数据规模膨胀。现代智能体系统通常需要感知并处理文本、图像、音频、视频等多种模态的信息。这种多模态融合的处理模式意味着系统所需存储与访问的数据规模相比纯文本模型将提升1~2个数量级。

2) 调用次数的倍增。智能体多轮迭代的执行机制使访存密集的解码阶段被反复触发。一个包含 k 步推理的智能体任务,其总访存开销约为单次推理访存量的 k 倍,呈线性甚至超线性增长。

3) 上下文长度的累积与生命周期的延长。智能体在执行过程中需不断将检索增强生成结果、历史观察、中间结果及工具返回信息追加至上下文,导致序列长度持续增长。与此同时,KV Cache管理需支持跨轮生命周期及长期记忆。这将带来KV Cache存储与访存量的同步膨胀,使长上下文下的存储容量和带宽压力进一步加剧。

4) 多模型或多智能体协作。智能体系统通常需并发处理多个子任务,多个智能体可能并行运行并通过消息传递进行协作。每个智能体实例均需独立的模型副本、上下文缓存及中间状态存储,因此当智能体数量增加时,系统对内存容量和带宽的需求呈线性增长。

综上,智能体应用将原本已属访存受限的LLM推理进一步推至“访存极度受限”的境地。可以预见,随着智能体自主性与复杂度的提升,内存墙将成为制约其规模化部署的核心障碍。

1.2 芯片架构的局限性和发展方向

过去20年,处理器算力的提升速度远高于内存DRAM带宽和互连带宽的增长速度。硬件峰值算力增长了约60 000倍,而DRAM带宽和互连带宽仅分别增长100倍和30倍^[1]。这意味着,单纯增加计算单元无法带来等比例的性能提升,反而会导致硅片利用率下降和能耗剧增。

如图1所示,随着访问距离增大,传统存储层级延迟、带宽、功耗瓶颈加剧。当前,叠加模型与应用端范式的转变,AI芯片的竞争逻辑正在重构:竞争焦点不再局限于算力峰值,而拓展至内存容量、带宽、能耗及成本控制的综合较量。如何突破内存墙、优化数据移动路径、重构内存层级,已成为决定AI芯片成败的核心技术挑战。

为应对这一挑战,芯片架构设计正从“以计算为中心”转向“以存储为中心”,其核心原则是以存储容量最大化、

表1 从LLM Chatbot到Agent,内存墙瓶颈加剧

	LLM Chatbot	Agent
模态	文本	文本、图像、音频、视频等
调用模式	单轮问答	多轮迭代
上下文长度	数千Tokens	百万Tokens(累积工具调用、记忆、文档等)
KV Cache 生命周期	会话级,可丢弃	跨轮持久化,需长期驻留或反复加载
模型组合	单一大模型	多模型/多智能体协作等异构组合

KV: 键值 LLM: 大语言模型

数据访问带宽最大化、数据移动能耗最小化及系统成本最小化为目标,优化整个计算-存储数据流。主要实现路径包括:

1) 基于软硬协同的存储层级重构:根据LLM与Agent的网络结构与数据流向,重新设计由不同存储介质构成的存储层级,缩短或消除冗余访问路径。

2) 基于存算一体的数据就近计算:从器件、电路、芯片、系统等不同层面,拉近计算单元与数据存储的物理距离,从根本上消除数据在存储器与处理器之间的频繁搬运^[6]。

2 存算一体技术路径分析

存算一体技术通过使计算单元与数据存储单元在物理上相互靠近乃至融合,使数据在存储位置即可完成计算,从而大幅减少数据搬运的距离与次数,提升访存带宽并降低访存功耗。依据以下3个维度,可形成多种存算一体技术路径。

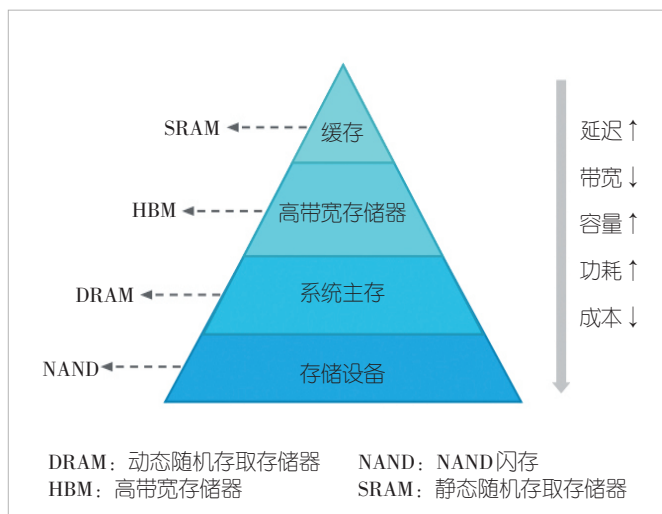


图1 典型内存层级架构

1) 按存储类型分类。依据所用存储介质的不同,可将存算一体技术路径分为两类:基于成熟存储介质的技术路径,包括静态随机存取存储器(SRAM)、DRAM、闪存(Flash)等;基于新型非易失性存储介质的技术路径,包括阻变随机存取存储器(ReRAM)、磁阻随机存取存储器(MRAM)、相变随机存取存储器(PCRAM)、铁电随机存取存储器(FeRAM)等。

2) 按存-算距离分类。依据计算单元与存储单元在物理上的距离与集成方式,可分为以下3种:

(1) 存内计算(CIM):存储单元与计算单元集成于同一芯片内,且两者紧密耦合,需对存储阵列进行改动,使存储单元可直接参与计算操作。

(2) 存内处理(PIM):存储单元与计算单元同样集成于同一芯片内,但两者相对独立,一般无须改动存储阵列,存储单元仅提供数据访存功能。

(3) 近存处理(PNM):存储单元与计算单元位于不同芯片,无须改动存储阵列,通过2.5D或3D先进封装技术缩短芯片间的物理距离。

3) 按封装方式分类。依据存储芯片与计算芯片的封装集成形式,可分为以下3种:

(1) 2D封装:该方式一方面代表传统存算分离的封装形式,如独立的低功耗双倍数据速率内存(LPDDR)、图形双倍数据速率内存(GDDR)、双倍数据速率内存(DDR)等;另一方面涵盖在同一芯片内实现的CIM。

(2) 2.5D封装:计算芯片与存储芯片通过中介层(Interposer)在水平方向进行互连,典型应用于高带宽存储器(HBM)或高带宽闪存(HBF)与计算芯片的封装集成。

(3) 3D封装:计算芯片与存储芯片通过混合键合和硅通孔(TSV)等技术,在垂直方向进行三维堆叠集成。

下文将依据上述分类维度,依次剖析各条技术路线的核心原理与特点。其中,基于SRAM、DRAM和Flash的存算一体方案,技术架构如图2^[7-10]、图3^[11-13]和图4^[14]所示,后续分析将围绕这3类主流存储介质展开。

2.1 基于SRAM的存算一体技术

2.1.1 SRAM CIM

SRAM CIM架构在SRAM阵列内部将SRAM Bitcell与算术逻辑电路(如乘法器、加法树、移位累加单元等)紧密耦合,从而支持乘累加(MAC)计算。在相同先进工艺节点下,SRAM CIM Macro在实现相同功能时,相较于传统数字

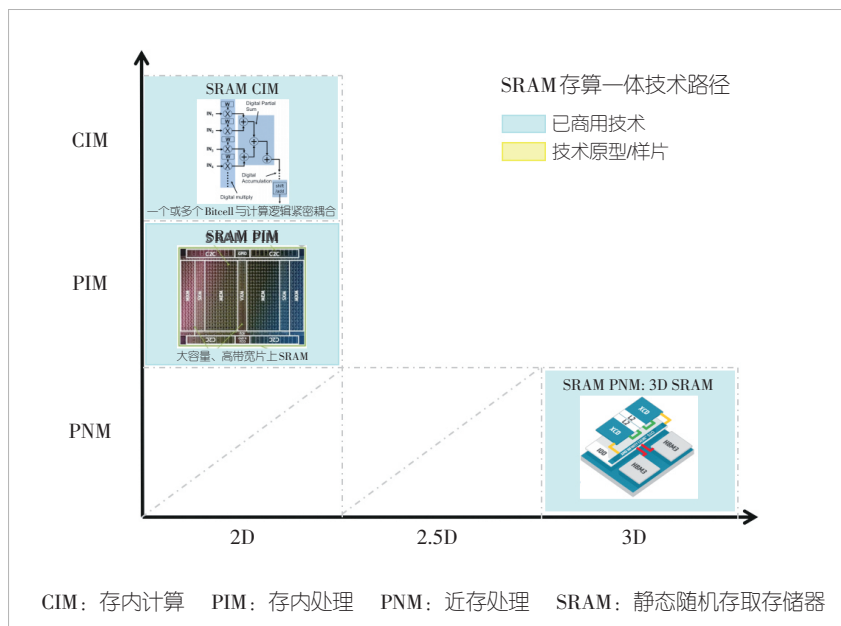


图2 SRAM存算一体技术路径^[7-10]

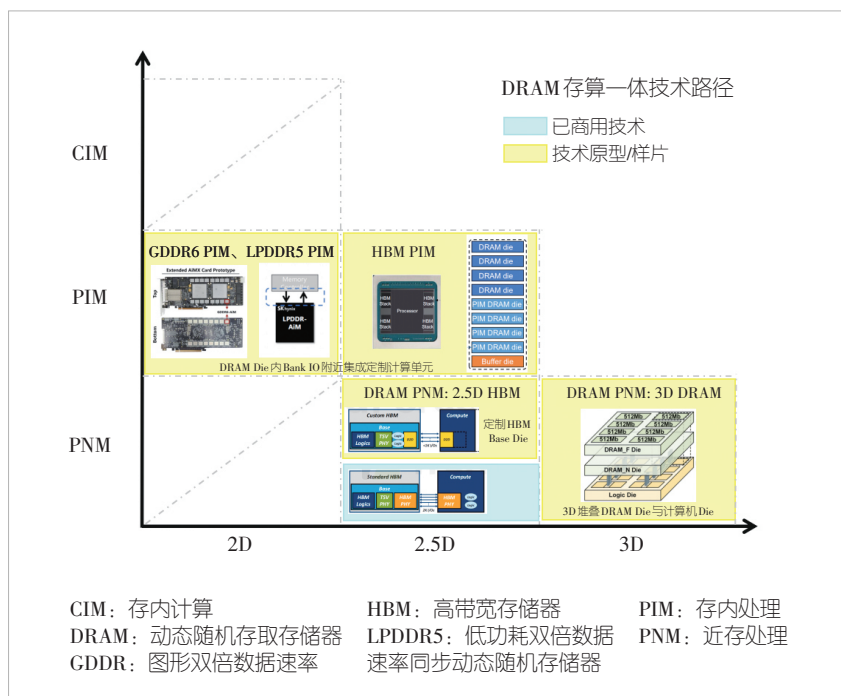
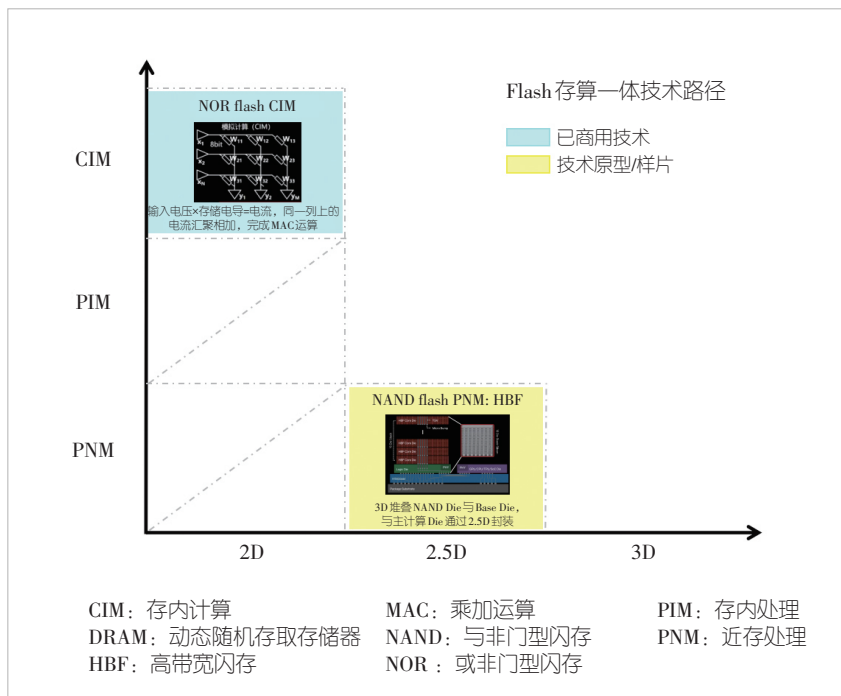


图3 DRAM存算一体技术路径^[11-13]

图4 Flash存算一体技术路径^[14]

计算单元可获得更高的能效与面积效率，如表2所示^[15-16]。中兴微电子与北京大学合作了基于5 nm工艺的SRAM CIM Macro模型，在0.6 V供电电压、50%稀疏度条件下，能效达到111 8b-TOPS/W；同时，双方还开发了集成该模型的仿真套件，用于SRAM CIM架构的性能、功耗、面积（PPA）仿真预测。

SRAM CIM Macro的容量通常限于数十kb，其原因在于：当阵列规模增大时，输入并行度随之增加且走线距离变长，从而导致布线拥塞、延迟增加及可靠性下降。鉴于单个SRAM CIM Macro的容量受限，实际往往通过拼接多个Macro来扩展容量与算力，但拼接本身也引入了Macro间互连带宽、延迟及面积开销等问题。

d-Matrix基于6 nm工艺的Corsair计算卡是采用SRAM CIM架构的代表性产品。该产品结合Chiplet与精简指令集计算机体系结构（RISC-V）内核等技术，实现了8个Chiplet、总算力2.4千万亿次浮点运算每秒（PFLOPS）（MXINT8）、SRAM总带宽150 TB/s及SRAM总容量2 GB的规格。然而，

2 GB的片上容量对于大语言模型而言依然十分有限，仍需借助低功耗双倍数据速率同步动态随机存储器（LPDDR5）等片外存储进行扩展。受SRAM存储容量制约，该产品的应用场景主要面向10B以下的小语言模型（SLM）以及Whisper等语音模型，旨在使模型尽可能驻留于片上，减少数据搬移，从而实现极低时延的AI推理。在成本方面，采用SRAM加LPDDR5、不使用HBM的架构提供了一种低成本的推理方案。在能效方面，尽管SRAM CIM单元本身的能效可达约20~40 TOPS/W，但其功耗仅占计算卡整体功耗的20%左右，计算卡中的矢量计算单元、交叉开关、输入输出（IO）模块及片外DDR等部件会显著稀释SRAM CIM所带来的能效收益。

2.1.2 SRAM PIM

SRAM PIM架构通过减少控制与计算单元的面积开销，换取更大的片上SRAM存储。

业界典型方案包括Groq 3语言处理单元（LPU）、SambaNova SN50及Cerebras WSE-3等。以Groq 3 LPU为例，该芯片基于三星4 nm工艺，片上SRAM面积占比高达40%，容量可达500 MB，聚合带宽达150 TB/s，有效降低了片外访存的带宽压力与功耗。同时，其结合编译器针对确定性负载预先设定数据流动路径与指令排布，从而实现极低延迟推理。

然而，即使在此类芯片架构中配置如此高比例的SRAM，受限于SRAM的存储密度，单芯片仍难以满足大模型部署需求。针对这一问题，目前存在两种主要解决方案。一种是通过片间与机间互联实现多芯片、多机柜扩展，进而支持系统级并行切分与部署，该方案对芯片间互联带宽要求较高。Groq 3 LPU即采用此方案，而晶圆级芯片Cerebras WSE-3本质上是片间互联转换为Die间的互联。另一种方案是结合HBM、DDR等多级片外存储进行容量扩展，虽然在一定程度上牺牲了纯SRAM架构的高性能与确定性，但显著降低了最小部署规模所需的芯片数量与成本，SambaNova

表2 SRAM CIM Macro与传统数字计算单元收益对比^[15-16]

	工艺	能效	面效
SRAM CIM Macro(台积电)	5 nm	~150 4b-TOPS/W @0.5V (50% sparsity)	221 4b-TOPS/mm ² @0.9V
传统数字计算单元		~90 4b-TOPS/W @0.46V (50% sparsity)	~100 4b-TOPS/mm ² @1.1V

CIM: 存内计算

是该方案的代表。

2.1.3 SRAM PNM

3D SRAM通过3D封装技术将计算Die与SRAM Die进行垂直堆叠。业界的代表产品包括AMD MI300/MI400 GPGPU系列，该系列通过优化末级缓存(LLC)的容量与带宽，在未牺牲片上算力资源的前提下将片上SRAM的容量扩展至256 MB。其3D SRAM带宽达17.2 TB/s。此外，计算Die与SRAM Die可采用不同工艺制程，有助于节约制造成本。然而，该技术对3D封装工艺与系统协同设计的要求较高，封装成本亦相对较高。

2.2 基于DRAM的存算一体技术

2.2.1 DRAM PIM

如表3所示，DRAM PIM架构在DRAM Die内的存储Bank附近集成计算单元，借助Bank级并行I/O大幅提升内部计算单元的访存带宽。相较于外部接口，内部带宽可提升约一个数量级，同时显著减少DRAM与AI芯片之间的数据搬运延迟与功耗。然而，DRAM Die基于DRAM工艺制造，其内部集成的计算单元在算力性能上面临限制。究其原因，DRAM工艺以存储密度最大化与漏电流最小化为核心目标，而计算单元的性能提升高度依赖于逻辑工艺所具备的高晶体管开关速度与多层互连能力。此外，计算单元的引入需占用DRAM存储面积，导致存储容量下降，散热与供电负担增加。因此，DRAM PIM中的计算单元通常采用较低的工作频率与数据传输速率，主要用于卸载算力需求较低但带宽需求较高的通用矩阵-向量乘法(GEMV)计算。三星与SK海力士等厂商已基于该技术路线研制出原型产品，但由于其计算能力有限且容量不足，难以满足日益增长的AI工作负载需求，商业应用推进缓慢。

2.2.2 DRAM PNM

标准HBM正逐渐逼近其性能瓶颈。一方面，为支持更高带宽，DRAM Die需容纳更多硅通孔输入/输出接口(TSV I/O)，导致其容量密度显著低于普通双倍数据速率同步动态随机存储器(DDR)；另一方面，为提升容量需增加堆叠层数，但封装与散热难度随之增大，良率下降，导致HBM成本过高，性价比比较低。因此，业界开始推动DRAM PNM架构研究。与PIM不同，DRAM PNM的计算与存储位于不同Die，工艺互不受限，可各自独立优化。按封装方式划分，DRAM PNM分为2.5D和3D两条技术路线。

2.5D路线在标准HBM基础上引入定制HBM Base Die。定制HBM Base Die的主要功能包括：1) 集成部分计算单元或控制逻辑，以节约AI芯片面积；2) 集成定制芯片对芯片物理层接口知识产权(D2D PHY IP)，优化处理器与HBM之间的接口IO密度与速率；3) 集成LPDDR控制器^[7]，使AI芯片无须通过中央处理器(CPU)中转即可直接经AI芯片—HBM—LPDDR路径访问LPDDR数据，实现多层级存储融合。

以Marvell定制HBM Base Die方案为例^[8]：1) 将原本位于主计算Die的HBM控制器集成至HBM Base Die，以节省主计算Die面积；2) 将标准HBM PHY转换为高密度、低功耗的定制D2D PHY，使线带宽密度从5 Tbit·s⁻¹·mm⁻¹提升至20~30 Tbit·s⁻¹·mm⁻¹，同时减少约75%的I/O功耗。2026年台积电技术研讨会展示了2023—2029年AI系统存储带宽的提升路径：通过采用基于先进工艺的定制HBM Base Die，结合更高代际的HBM颗粒与2.5D单颗晶圆上芯片封装(CoWoS)技术，预计CoWoS封装内总HBM带宽可提升34倍^[9]。

3D路线通过3D封装垂直堆叠计算Die与DRAM Die。与2.5D封装相比，系统封装面积约减少一半。计算Die内通过定制的数据流计算架构、DRAM控制器及PHY等，实现计

表3 各DRAM-PIM内部带宽、算力对比

内部带宽		算力	容量或面积损失	收益
GDDR6-PIM ^[10]	16×外部带宽:512 GB/s	512 GFLOPS	每个计算单元需0.19 mm ² 换取32 GFLOPS算力,每Die共16个计算单元	执行MHA计算时,相比传统GPU系统,该方案计算延迟降至原来的2/5
LPDDR5x-PIM ^[10]	8×外部带宽:153.6 GB/s	307.2 GOPS	/	/
HBM2-PIM ^[11]	4×外部带宽: 1.22 9 TB/s@300 MHz	1.2 TFLOPS	共8个Die,改造4个Die,每Die损失1/4容量	GEMV计算能效最高提升8.25倍

GDDR: 图形双倍数据速率
GEMV: 通用矩阵-向量乘法
GFLOPS: 每秒十亿次浮点运算
GOPS: 每秒十亿次整数运算

CPU: 图形处理器
HBM: 高带宽存储器
LPDDR: 低功耗双倍数据速率同步动态随机存储器
MHA: 多头注意力

PIM: 存内处理
TFLOPS: 每秒万亿次浮点运算

算单元对 DRAM Bank 的垂直并行访问，在单位面积内形成海量并行数据通道，大幅缩短 DRAM Die 的访问路径。与 2.5D 封装相比，在堆叠相同数量的 DRAM Die 条件下，带宽可提升约 10 倍，功耗降低至 60%^[20]。该技术路径需解决堆叠架构带来的散热与供电压力，权衡封装工艺的良率与成本，并考虑 TSV 及堆叠层数引入的计算 Die 面积开销，在设计、制造、封装与测试全流程中均面临较高的系统复杂度。

由于上述系统工程挑战，3D DRAM PNM 预计将率先在边缘侧实现应用，并有望集成于中兴微电子边缘侧 AI 产品中。当前业界可堆叠 DRAM 层数为 1~4 层，单层面积约为 50~100 mm²，单层容量为 2~4 GB，堆叠后的总带宽可达 1~4 TB/s，动态功耗约为 0.6 pJ/bit（约为同代 LPDDR 功耗的五分之一）。由此可见，该技术路线的核心优势在于高带宽与低功耗，容量虽显著高于 SRAM，但仍受限于堆叠面积与层数，难以满足较大参数量模型的需求，因此需要结合 LPDDR 或 Flash 等存储介质进行系统级架构设计。

2.3 基于 Flash 的存算一体技术

2.3.1 Nor Flash CIM

基于 Flash 的模拟存内计算，存储密度高于 DRAM，器件成本低，但 Flash 读写速度差（几十到一百多 MB/s），耐久有限，只适合模型固定的小算力低功耗应用场景。例如知存科技 WTM2101 是基于 NOR Flash 的存算一体 AI 片上系统（SOC）芯片，算力仅 50 每秒十亿次整数运算（GOPS），面向智能语音和智能健康。

2.3.2 Nand Flash PNM

HBF 定位为介于 DRAM 与 Flash 之间的新型存储层级。根据闪存厂商（SanDisk）的产品规划，其第一代 HBF 产品

通过 3D 堆叠技术集成 16 个与非型闪存（NAND）Core Die，单颗容量可达 512 GB，两个 Stack 即可实现 TB 级以上容量，满足低成本大容量扩展需求。同时，通过底部 Base Die 对 NAND Core Die 的并行访问，HBF 可实现高达 1.6 TB/s 的带宽，性能与 HBM4 相当。然而，该技术仍面临写入寿命有限与访问延迟较高等短板，需在实际系统设计中予以权衡。

HBF 适合推理场景中只读或确定性访问模式的工作负载，但不适合训练等写密集型任务。HBF 的潜在应用方案如图 5 所示，主要包括以下 3 类：

1) 各种类型的处理器（xPU）直连 HBF：以 HBF 替代部分 HBM，在相同芯片边长下可实现 8~16 倍的读存储容量扩展。为避免因 HBF 延迟带来的性能损失，需在 HBF 内部对权重与 KV Cache 进行分区分配。

2) xPU - HBM - HBF 混合架构：该方案可减少 xPU 通过 CPU 访问内存或 SSD 所产生的延迟与功耗开销^[21]。其中，HBM Base Die 负责处理 xPU 的访问请求路由与地址解码，并集成 SRAM Buffer 对 HBF 进行数据预取，以进一步隐藏访问延迟。仿真实验表明，在部署 LLaMa 3.1 405B 模型（上下文序列长度为 10M）的条件下，该方案相比纯 HBM 方案，吞吐量提升了 6.14 倍，能效提升了 2.69 倍。

3) 3D DRAM + HBF 架构：该方案无需 HBM，通过与 3D DRAM PNM 技术结合，利用 HBF 来突破 3D DRAM 容量不足的瓶颈。

2.4 基于新型存储的存算一体技术

新型存储主要包括 MRAM、ReRAM、PCRAM、FeRAM 等非易失性存储技术，如图 6（a）^[22]所示，具备以下特点：

1) 容量密度高于 SRAM，但受限于良率与成本，难以扩展至 Gb 级以上规模；2) 写入延迟介于 SRAM 与 eFlash 之间，其中 MRAM 与 ReRAM 优于相变存储器（PCM）与铁电场效

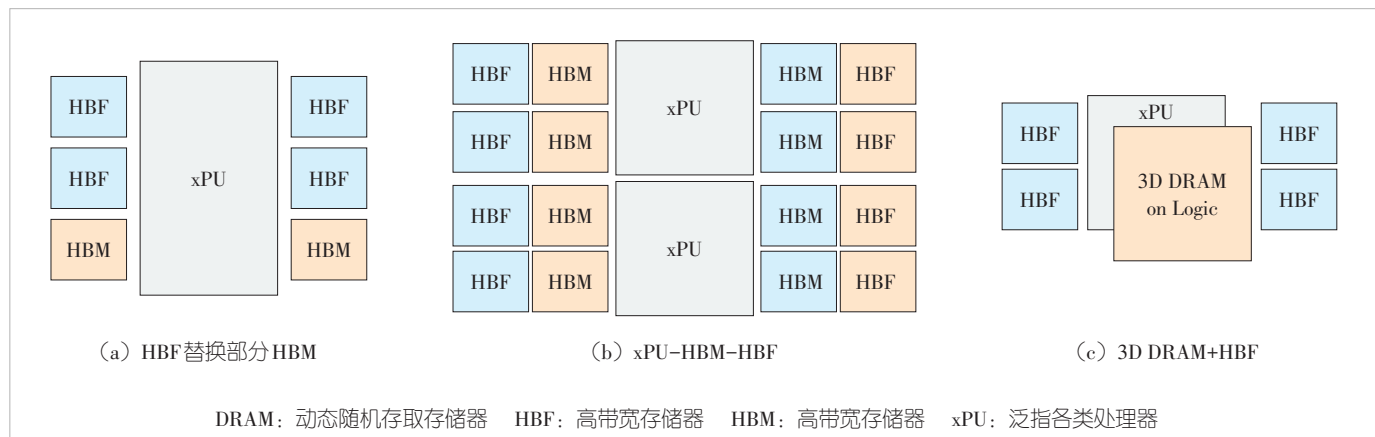
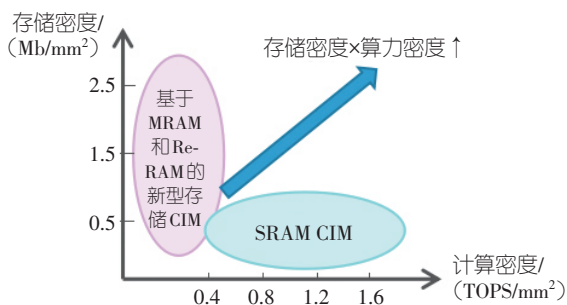


图 5 HBF 潜在应用方案

	SRAM	MRAM	ReRAM	PCM	FeFET
写延迟	< 1 ns	< 1 ns	100 ns	150 ns	175 ns
写功耗	< 0.1 nJ	< 1 nJ	2 nJ	6 nJ	0.1 nJ
静态功耗	高	低	低	低	低
容量密度	低	中	高	高	高
耐久	> 10 ¹⁶	> 10 ¹⁵	10 ⁵	10 ⁷	10 ¹⁰

(a) SRAM与新型存储介质的关键指标对比



(b) SRAM CIM与基于新型存储CIM的关键指标对比

CIM: 存内计算

MRAM: 磁阻随机存取存储器

ReRAM: 阻变随机存取存储器

TOPS: 每秒万亿次操作

FeFET: 铁电效应晶体管

PCM: 相变存储器

SRAM: 静态随机存取存储器

图6 SRAM与新型存储介质对比及其各自CIM技术^[22]

应晶体管(FeFET); 3) 非易失性存储无须定期刷新, 静态功耗较低, 其中MRAM与ReRAM优于PCM; 4) MRAM与ReRAM相对PCM与FeFET, 与互补金属氧化物半导体(CMOS)工艺兼容性更佳。

目前, 业界将MRAM与ReRAM定位为替代eFlash(嵌入式闪存, 容量通常为数十Mb)的方案, 主要布局于车端微控制器(MCU)。台积电22 nm ReRAM节点已进入量产阶段, 12 nm ReRAM完成技术验证; 三星则计划于2026年实现8 nm MRAM的量产。

在CIM改造方面, MRAM和ReRAM相较于PCM和FeFET, 在写延迟与写功耗方面更具优势, 从而能够实现更高的计算能效^[23]。其中: 1) MRAM与CMOS工艺兼容性更好, 适合用于数字CIM的改造; 2) ReRAM结构简单, 在外加电压下呈现连续的阻态变化, 能够支持多比特权重表示, 天然适用于模拟CIM。其工作原理基于欧姆定律与基尔霍夫定律: 输入电压与本地存储电导的乘积产生电流, 同一列上的电流汇聚相加, 完成乘加(MAC)运算^[24]。

当前, 产业界与学术界正积极探索基于新型存储的存算一体技术。如图6(b)所示, 新型存储可与SRAM形成互补协同, 在容量与能效之间取得更好的平衡。例如, 台积电于2025年在Nature上发表了基于异构RRAM与SRAM的CIM AI边缘处理器设计, 在ResNet-20和MobileNet-V2上分别实现了40.91 TFLOPS/W与28.63 TFLOPS/W的能效。未来, 随着存储技术的持续演进, 有望出现同时具备非易失性、CMOS工艺兼容性、高容量密度的更理想存储介质。例如, 2026年IEEE国际固态电路会议(ISSCC)上中科院提出的基于电荷陷阱晶体管(CTT)的方案^[22], 即旨在兼顾高算力密度与高存储密度, 以适配更多应用场景需求。

3 存算一体技术趋势

如表4所示, 本文现对存算一体技术路径进行技术趋势总结^[25]。其中, 带有颜色标注的为具备潜力的存算一体技术路径: 蓝色表示已商用技术, 黄色表示已有原型或样片的技术。

4 总结与展望

存算一体技术为突破云边端LLM与智能体(Agent)应用面临的“内存墙”瓶颈提供了多种技术路径。随着模型规模的持续增长与应用场景的日益复杂, 单点的硬件突破已难以满足产业需求, 亟需通过多层次深度协同, 实现系统级优化。

1) 存算一体技术协同。单一存储介质难以同时满足大容量、高速度、低功耗及低成本等多重极限需求, 因此, 基于多种存储介质协同组合的存算一体架构正成为重要发展趋势。例如, 在同一系统中可采用“SRAM CIM + 3D DRAM”或“SRAM CIM + 新型存储CIM”的异构存算架构, 利用SRAM处理高频访问的局部动态数据, 而将静态数据存储于大容量DRAM或高密度新型存储介质中, 实现优势互补, 从而突破单一介质的物理性能天花板。

2) 芯片与硬件基础设施系统协同。在芯片层面, 存算一体技术与先进封装、Chiplet等技术的结合是突破单芯片容量与带宽限制的关键路径; 在硬件基础设施层面, 存算一体芯片需与计算集群、算力网络、散热及供电等基础设施协同优化, 共同决定最终产品的能效表现。未来竞争将更多体现于系统级协同设计能力。

3) 软硬件协同。成熟的软件生态是创新架构应用推广的重要保障, 需构建涵盖算法层、系统层与芯片层的完整优

表4 存算一体技术趋势总结

存储介质	技术路径	产品化进展	适用场景及技术趋势
SRAM	SRAM CIM	d-Matrix Corsair Chiplet、后摩智能 M50	• 适合边侧和端侧的中小规模模型,尤其是对功耗敏感的场景 • 受限于 SRAM 容量,当前需要结合片外存储,导致 SRAM CIM 收益稀释;未来技术趋势为结合 DRAM PNM 技术,扩展容量的同时保持存算一体收益,已体现在此类产品下一代规划中
	SRAM PIM	Groq 3 LPU、Sambanova SN50、Cerebras WSE-3	• 适合云侧和边侧场景,尤其是对时延敏感的场景 • 受限于 SRAM 容量,往往多 Die、多芯片、多机柜规模部署,对互联带宽要求高;由于简化了硬件控制逻辑,软件编译和调度能力更加重要;技术趋势为整机软硬件系统深度协同优化
	SRAM PNM	AMD MI300/MI400	• 适合云侧和边侧场景 • 承受了 3D 封装的代价和挑战,但相比 DRAM PNM,SRAM PNM 并不能从根本上解决 AI 存储容量限制
DRAM	DRAM PIM	三星 HBM PIM、SK 海力士 GDDR PIM 和 LPDDR PIM	• 根据不同存储类型应用于云边缘场景 • 受限于算力主要用于 GEMV 计算;由存储厂商对存储颗粒产品内部进行改造
	DRAM PNM: 2.5D HBM 定制 Base Die	自 HBM4 起引入定制 Base Die	• 适合云侧和边侧场景,是实现大容量高带宽的主要路径 • 是 2026 年之后提升 HBM 带宽的重要驱动技术;需要设计厂商、晶圆代工厂商、存储厂商、封装厂商等协同设计
	DRAM PNM: 3D DRAM	d-Matrix 和后摩智能下一代产品、瑞芯微等国产 AI 协处理器	• 适合云边缘各类场景,是实现大容量、高带宽、低功耗的主要路径 • 对于 3D 封装技术和复杂系统协同设计要求较高,预计边缘侧落地早于云侧落地
Flash	Nor Flash CIM	知存 WTM2101	• 只适合模型固定的端侧小算力、低功耗场景
	Nand Flash PNM: HBF	闪迪和海力士战略合作等	• 适合云边缘各类场景的静态数据存储,构建介于 HBM 与传统 SSD 之间的新型存储层级 • 闪迪路线图显示,HBF 将沿着容量提升、带宽增长、功耗优化的方向持续迭代
新型存储	新型存储 CIM	亿铸科技 ReRAM-CIM 原型,北京大学、香港科技大学等高校研究	• 适合端侧小算力、功耗敏感场景 • 当前器件良率和成本面临挑战,容量扩展受限;随着材料和工艺的技术突破,有望发挥革命性作用
CIM: 存内计算		HBF: 高带宽闪存	PIM: 存内处理
DRAM: 动态随机存取存储器		HBM: 高带宽存储器	PNM: 近存处理
GDDR: 图形双倍数据速率存储器		LPDDR: 低功耗双倍数据速率同步动态随机存储器	SRAM: 静态随机存取存储器
GEMV: 通用矩阵-向量乘法		LPU: 语言处理单元	SSD: 固态硬盘

化空间。具体包括：面向存算一体架构的模型结构设计、量化与剪枝等算法优化技术；面向软件栈的编译优化、运行时调度及内存管理机制；建立统一的编程模型与硬件抽象层，使存算一体硬件能够无缝接入主流 AI 框架，实现硬件透明、软件友好的开发体验。

4) 产业链生态协同。存算一体技术是一项涉及新材料与器件研发、芯片设计、晶圆制造、先进封装、系统集成及应用部署的系统工程，产业链各环节的紧密协作是推动该技术从实验室走向大规模落地的关键。

中兴通讯与中兴微电子以芯片与硬件基础设施层为核心发力点，以系统总拥有成本（TCO）最优的全栈算力解决方案为目标，通过产学研协同、创新架构研究及产业合作伙伴价值共创等方式，积极探索基于 SRAM、DRAM、Flash 及新型存储的多元化存算一体技术与产品应用。未来，通过底层异构存储介质的技术协同、系统级芯片与硬件基础设施的协同、全栈式软硬件协同以及全景式产业链生态协同，存算一

体有望成为重塑计算体系架构的关键驱动力，为后摩尔时代人工智能的广泛普及与应用提供坚实基础。

致谢

感谢深圳市中兴微电子有限公司陈桂显专家对本论文做出的贡献！

参考文献

[1] Gholami A, Yao Z, Kim S, et al. AI and memory wall [EB/OL]. (2024-03-25) [2026-06-10]. <https://ieeexplore.ieee.org/abstract/document/10477550>

[2] 韩炳涛, 刘涛. 大模型关键技术与应用 [J]. 中兴通讯技术, 2024, 30 (2): 76-88. DOI: 10.12142/ZTETJ.202402012

[3] Wolters C, Yang X X, Schlichtmann U, et al. Memory is all you need: an overview of compute-in-memory architectures for accelerating large language model inference [PP/OL]. arXiv(2024-06-12)[2026-06-13]. <https://arxiv.org/abs/2406.08413>

[4] Horowitz M. 1.1 computing's energy problem (and what we can do about it) [C]//Proceedings of IEEE International Solid-State

- Circuits Conference Digest of Technical Papers (ISSCC). IEEE, 2014: 10–14. DOI: 10.1109/isscc.2014.6757323
- [5] 包义明, 林阳, 屠要峰. 人工智能技术与应用前沿 [J]. 中兴通讯技术, 2025, 31(4): 64–74. DOI: 10.12142/ZTETJ.202504010
- [6] 何斯琪, 穆琛, 陈迟晓. 基于存算一体集成芯片的大语言模型专用硬件架构 [J]. 中兴通讯技术, 2024, 30(2): 37–42. DOI: 10.12142/ZTETJ.202402006
- [7] Chih Y D, Lee P H, Fujiwara H, et al. 16.4 an 89TOPS/W and 16.3TOPS/mm² all-digital SRAM-based full-precision compute-in memory macro in 22nm for machine-learning edge applications [C]//Proceedings of IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2021: 252–254. DOI: 10.1109/isscc42613.2021.9365766
- [8] Prabhakar R, Sivaramakrishnan R, Gandhi D, et al. SambaNova SN40L: scaling the AI memory wall with dataflow and composition of experts [PP/OL]. arXiv(2024–11–05) [2026–06–11]. <https://arxiv.org/abs/2405.07518>
- [9] NVIDIA Corporation. A decade of AI infrastructure innovation: from NVIDIA DGX-1 to NVIDIA vera rubin [EB/OL]. [2026–03–06] [2026–06–11]. <https://www.nvidia.com/en-us/on-demand/session/gtc26-d8006/>
- [10] Tom's Hardware. AMD shares new second-gen 3D v-cache chiplet details: up to 25 TB/s [EB/OL]. (2023–06–01)[2026–06–11]. <https://www.tomshardware.com/news/amd-shares-new-second-gen-3d-v-cache-chiplet-details-up-to-25-tbs>
- [11] Kim G, Kim J, Kim N, et al. SK hynix AI-specific computing memory solution: from AiM device to heterogeneous AiMX-xPU system for comprehensive LLM inference [C]//Proceedings of IEEE Hot Chips 36 Symposium (HCS). IEEE, 2024: 1–26. DOI: 10.1109/hcs61935.2024.10664793
- [12] Kim K. Advancements in co-packing and co-integration technologies for high-performance memory system [C]//2024 IEEE VLSI Symposium on Technology and Circuits. IEEE, 2024
- [13] Cao Y, Jiang J H, Jiang H J, et al. A 1.2GHz 12.77GB/s/mm² 3D two-DRAM-one-logic process-near-memory chip for edge LLM applications [C]//Proceedings of IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2026: 524–526. DOI: 10.1109/ISSCC49663.2026.11409184
- [14] SanDisk. High Bandwidth Flash (HBF) tutorial [R]. 2026 IEEE International Memory Workshop (IMS), 2026
- [15] Fujiwara H, Mori H, Zhao W C, et al. A 5-nm 254-TOPS/W 221-TOPS/mm² fully-digital computing-in-memory macro supporting wide-range dynamic-voltage-frequency scaling and simultaneous MAC and write operations [C]//Proceedings of IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2022: 1–3. DOI: 10.1109/ISSCC42614.2022.9731754
- [16] Keller B, Venkatesan R, Dai S, et al. A 95.6-TOPS/W deep learning inference accelerator with per-vector scaled 4-bit quantization in 5 nm [J]. IEEE journal of solid-state circuits, 2023, 58(4): 1129–1141. DOI: 10.1109/JSSC.2023.3234893
- [17] Kaist Teralab. HBM roadmap ver 1.7 workshop: overview of next generation HBM architectures [EB/OL]. [2026–06–13]. <https://tera.kaist.ac.kr/home>
- [18] Marvell Semiconductor, Inc. Marvell Custom AI Investor Event 2025 [EB/OL]. [2026–06–02]. <https://www.marvell.com/content/dam/marvell/en/company/assets/marvell-custom-ai-investor-event-2025.pdf>
- [19] TSMC. TSMC Technology symposium press briefing [EB/OL]. (2026–04–26) [2026–06–02]. <https://www.eet-china.com/mp/a490419.html>
- [20] Lim D. Packaging-aware design and optimization of HBM power delivery in AI platforms [R]. ISSCC 2026 Forum, 2026
- [21] Ha M, Kim E, Kim H. H3: hybrid architecture using high bandwidth memory and high bandwidth flash for cost-efficient LLM inference [J]. IEEE computer architecture letters, 2026, 25 (1): 49–52. DOI: 10.1109/LCA.2026.3660969
- [22] Shen J Z, Zhou Z D, Zha W F, et al. A 12nm 4Mb 104.56-to-137.75TFLOPS/W charge-trap transistor-based computing-in-memory macro using analog-predict-digital-compute for AI edge devices [C]//Proceedings of IEEE International Solid-State Circuits Conference (ISSCC). IEEE, 2026: 514–516. DOI: 10.1109/ISSCC49663.2026.11409272
- [23] Khwa W S, Wen T H, Hsu H H, et al. A mixed-precision memristor and SRAM compute-in-memory AI processor [J]. Nature, 2025, 639(8055): 617–623. DOI: 10.1038/s41586-025-08639-2
- [24] 朱炫鹏, 姚海东, 刘隽, 等. 大语言模型算法演进综述 [J]. 中兴通讯技术, 2024, 30(2): 9–20. DOI: 10.12142/ZTETJ.202402003
- [25] 廖俊淇, 魏昕, 周亮. 人工智能驱动的跨模态语义通信系统 [J]. 中兴通讯技术, 2024, 30(S1): 33–39. DOI: 10.12142/ZTETJ.2024S1005

作者简介



刘新阳, 深圳市中兴微电子技术有限公司副总经理、中国通信学会集成电路专业委员会副主任委员、中兴通讯股份有限公司多媒体与移动网络国家重点实验室IC研发中心主任; 负责中兴微电子芯片战略与技术规划、运营工作等, 拥有20余年通信集成电路规划、研发、经营管理经验。



李婷宇, 深圳市中兴微电子技术有限公司战略规划工程师; 主要从事人工智能芯片技术研究与规划工作, 包括人工智能芯片架构技术、存储技术、互联技术和先进封装技术等。



巨鸣, 深圳市中兴微电子技术有限公司战略规划工程师; 主要从事人工智能芯片技术研究与规划工作, 包括人工智能芯片存储技术、存算一体技术、新型存储技术等。

网络运行安全完整性等级确定方法研究



Research on Method for Determination of Network Operation Safety Integrity Level

路晔绵/Lu Yemian, 郑梦青/Zheng Mengqing,
吴荣春/Wu Rongchun, 张治兵/Zhang Zhibing,
王晓颖/Wang Xiaoying

(中国信息通信研究院, 中国 北京 100191)
(China Academy of Information and Communications Technology, Beijing
100191, China)

DOI: 10.12142/ZTETJ.202604011

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260901.1347.004>

网络出版日期: 2026-08-28

收稿日期: 2025-09-25

摘要: 设备故障或系统失效经常引发大面积断网或业务暂停, 亟需引入安全完整性体系和新的安全完整性等级确定方法。通过分析总结现有多个领域中的安全完整性定级方法, 结合信息通信领域特点, 提出了网络运行安全完整性等级 (NSIL) 确定方法。通过实际案例分析, 验证了在信息通信领域引入安全完整性体系并确定关键设备或系统的 NSIL 等级的必要性, 为后续建立完善的信息通信领域网络运行安全完整性体系提供理论基础。

关键词: 网络运行事故; 安全完整性等级; 故障; 失效

Abstract: Equipment failures or system outages frequently cause large-scale network disruptions or service interruptions, underscoring an urgent need to introduce the safety integrity system and a new method for determining safety integrity levels. By analyzing and summarizing the existing determination method of Safety Integrity Level in multiple fields, combining with the characteristics of the information and communication field, a method for determining the network operation safety integrity level (NSIL) is proposed. Through practical case analysis, the necessity of introducing the safety integrity system in the information and communication field and determining the NSIL level of key equipment or systems is verified, providing a theoretical basis for establishing the well developed network operation safety integrity system in the information and communication field in the future.

Keywords: network operation accident; safety integrity level; fault; failure

引用格式: 路晔绵, 郑梦青, 吴荣春, 等. 网络运行安全完整性等级确定方法研究 [J]. 中兴通讯技术, 2026, 32(4): 72-78. DOI: 10.12142/ZTETJ.202604011

Citation: Lu Y M, Zheng M Q, Wu R C, et al. Research on method for determination of network operation safety integrity level [J]. ZTE technology journal, 2026, 32(4): 72-78. DOI: 10.12142/ZTETJ.202604011

数字经济时代, 信息通信网络已成为支撑社会正常运转的重要基石。然而近年来, 由网络设备、电源系统、空调系统等关键设备或系统的故障及失效引发的网络运行事故频发, 常常造成大范围网络中断或重要服务异常, 严重影响日常生产生活。目前, 在工业控制、机械、汽车等领域, 已广泛应用功能安全及安全完整性体系来应对故障和失效导致的安全风险, 其有效性已得到实际验证。因此, 有必要在信息通信领域引入安全完整性体系, 明确各类设备和系统应具备的安全功能及其安全完整性等级, 并达到相应等级的安全完整性要求, 以降低因设备或系统故障及失效导致的网络

运行事故概率。其中, 如何确定安全完整性定级方法是应用安全完整性体系时首要解决的问题。然而, 现有安全完整性定级方法主要基于故障失效风险对人身安全的影响划分等级, 而信息通信领域的故障和失效问题在绝大多数情况下并不会对人身安全构成威胁, 因此无法直接适用现有的方法。

本文通过分析工业控制、机械、汽车、轨道交通等领域中安全完整性等级的确定方法, 借鉴其设计理念, 并结合信息通信领域自身特点, 提出网络运行安全完整性等级 (NSIL) 的定级方法, 为后续构建网络运行安全完整性体系奠定基础。同时, 结合实际案例, 展示 NSIL 定级方法的应

用,验证了在信息通信领域引入安全完整性体系的必要性以及该方法的合理性。

1 技术现状

安全完整性属于功能安全领域的概念,在国际标准 IEC 61508-4^[1](对应中国标准 GB/T 20438-4^[2])中提出,指在规定条件和规定时间内,系统成功执行安全功能的概率。安全完整性与信息安全领域中的“完整性”极易混淆,后者在信息安全术语标准 GB/T 25069^[3]中被定义为“准确和完备的性质”,与安全完整性分属两个不同领域,是完全独立的概念。

安全完整性等级(SIL)是在系统整体安全生命周期框架下构建的安全度量体系,是衡量安全系统降低故障或失效风险能力的有效途径。SIL等级越高,表明因系统故障或失效引发事故的概率或影响越低。通常情况下,需通过对系统进行危险与风险分析,以确定其所应实现的安全功能及对应的安全完整性等级。

作为安全完整性体系的基础标准,IEC 61508-5^[4](对应中国标准 GB/T 20438-5^[5])给出了几种基于定量和定性方法的安全完整性等级确定方法。周荣义等^[6]对包含这些方法在内的多个定级方法进行了对比分析,并给出了选择建议。总体而言,定量方法主观性最小,但需要大量已有故障数据作为计算支撑,在实际应用中难以获得,且系统性失效无法进行量化计算。因此,实践中使用较多的仍是定性方法,其中风险图方法是目前应用最为广泛的方法。

风险图方法选取危险事件的后果(C)、人员暴露于危险区域的频率/时间(F)、避免危险事件的可能性(P)以及不希望事件发生的概率(W)作为定级参数,对每个参数进行分级,并通过分支图确定不同参数等级组合所对应的SIL等级。该方法通过多因素综合分析,能较好地反映故障和失效的危险程度,因此被许多领域借鉴和采用。例如,过程工业领域功能安全标准 IEC 61511-3^[7](对应中国标准 GB/T 21109.3^[8])基本沿用了IEC 61508中的风险图方法,仅对参数赋值进行了校准;机械领域标准 IEC 62061^[9](对应中国标准 GB 28526^[10])选择了与风险图方法类似的参数,但对完整性等级的推导方式作了修改;而机械领域标准 ISO 13849^[11](对应中国标准为 GB/T 16855.1^[12])则在风险图方法基础上删去W参数,从严定级,仅考虑最坏情况下发生危险事件的可能性。

汽车领域根据其产品特点,在标准 ISO 26262-3^[13](对应中国标准 GB/T 34590.3^[14])中定义了自身的安全完整性等级(ASIL)定级方法,考虑3个参数,即危害的严重性等级(S)、暴露在危害中的概率(E)和危害的可控性(C)。在

参数赋值设计中,ASIL定级方法充分考虑了汽车领域应用场景的独特性,使定级结果更具适用性,例如以驾驶员的可控性来评估危害可控性。在完整性推导阶段,ASIL定级方法采用三元素矩阵划分方式,ASIL等级随3个参数等级的递增而逐级升高。

轨道交通领域功能安全相关的标准中并未明确推荐SIL定级方法,但产业界和学术界均对此展开了广泛研究。欧洲轨道交通技术项目 MODSafe 提出了一个半定量的SIL定级方法——EPC方法^[15]。该方法中,安全功能失效导致后果的严重程度(THR)作为原始风险数值,而风险群体成员(如乘客)暴露的可能性(E)、减少事故的可能性(P)和减轻后果的可能性(C)3个参数分别对应其他情况或措施对原始风险的降低效果。通过将THR值除以E、P、C3个参数的值,可计算安全相关系统所需达到的风险降低能力,进而对应至SIL等级。EPC方法属于半定量方法。

在学术界,张冬梅^[16]、杨娟^[17]、李葛亮^[18]、白春光^[19]等提出了类似的方法。俞杭华^[20]和刘亚妮^[21]等对不同SIL定级方法在轨道交通领域的适用情况进行了分析,其中后者还通过实际案例验证了EPC方法与风险图方法定级结果的一致性。

以上几个领域的安全完整性定级方法在参数选择上具有较高的相似性,均考虑了危险事件的后果、人员暴露于危险区域的频率/时间,以及避免危险事件的可能性3个参数。然而,现有方法在评估危险事件后果时,均从对人身安全构成威胁的角度出发,而网络运行事故在绝大多数情况下并不会危及人身安全,而是影响其上承载的业务,进而波及社会生产生活。由于影响对象不同,现有的安全完整性定级方法无法直接应用于信息通信领域,因此有必要设计针对网络运行安全的、具有领域特色的安全完整性等级确定方法。

2 NSIL定级方法设计

基于上述对工业控制、机械、汽车等领域安全完整性定级方法的分析,定级方法的设计需从3个层面展开:首先选择合适的定级参数,然后根据各领域特点确定各参数的赋值,最后明确不同参数赋值所对应的安全完整性等级,即完整性推导规则。NSIL定级方法也将遵循这一框架进行设计。

2.1 定级参数选择

从各领域的安全完整性定级方法可以看出,安全完整性的定级通常需考虑设备或系统故障或失效时可能造成的危害后果严重性、影响对象暴露于危险的可能性、避免危险事件

发生的可能性,以及减轻后果的可能性等因素。

在其他领域中,危险事件的影响客体是人体,危害后果为人身伤亡;而在信息通信领域,危险事件的影响客体通常并非人体,所造成的危害后果体现为业务性能下降或中断后对社会、经济等方面产生的影响。由于信息通信网络上承载的业务类型多样,不同类型的业务对社会、经济的影响程度各异,因此需要结合业务类型与故障危害类型来评估受影响业务对社会的危害程度,同时根据目标业务的用户范围评估危险事件可能的影响范围。基于此,NSIL定级方法可将危害后果严重性因素细化为业务重要性(I)、故障危害类型(T)和业务影响范围(S)3个参数。

对于影响对象暴露在危险中的可能性因素,由于依托信息通信网络运行的业务均不间断持续运行,网络发生故障时其上业务必然受到影响,无须分级别讨论,因此在NSIL定级方法中不再将该因素纳入考虑范围。

对于避免危险事件发生的可能性和减轻危害后果的可能性因素,信息通信网络建设中通常会部署各类措施以预防危险事件发生或减轻其后果,例如通过异地备份帮助业务从机房故障中快速恢复。然而,信息通信领域中许多措施既能一定程度上避免危险事件发生,也能减轻危害后果。为防止重复考虑这些措施而影响完整性推导结果的合理性,我们将上述两个因素合并为一个因素进行考虑,并根据具有这两类效果的相互独立措施的数量对该因素进行分级。

综上所述,在网络运行安全完整性等级(NSIL)定级方法中,将考虑以下两类共4个参数:

- 1) 危害后果严重程度(C),细分为3个子参数:业务重要性(I)、业务影响范围(S)、故障危害类型(T);
- 2) 避免危险和减轻后果的可能性(P)。

2.2 参数赋值设计

2.2.1 业务重要性(I)赋值设计

信息通信网络承载种类丰富的业务。当发生网络故障造成各类业务中断或者业务性能劣化时,不同业务对社会和经济发展产生的影响亦有所不同。因此,需要结合业务本身的属性,完成业务重要性的等级赋值。本文参考《电信网络运行监督管理办法》^[22]中对网络运行事故的分级规则,设计业务重要性(I)的赋值规则,具体如表1所示。

2.2.2 业务影响范围(S)赋值设计

业务影响范围可从两个维度衡量:一是根据行政区域范围衡量,适用于直接搭载在运营商网络上运行的业务,如基础电信业务(语音通话等)及区域性党政机关业务等;二是

根据影响用户数量衡量,适用于运行于上层互联网的业务,如社交应用、电商服务平台等。这类业务通常部署在互联网数据中心,可能面向全国提供服务,但其实际影响范围仅限于使用该类业务的目标用户。不同业务的用户数量差异悬殊,从几万到上亿不等,此时若采用行政区域范围衡量,易产生较大偏差,因此宜使用影响用户数量作为衡量依据。

本文参考《电信网络运行监督管理办法》中对网络运行事故的分级规则,确定行政区域范围的划分方法;依据《互联网平台分类分级指南(征求意见稿)》^[23],确定影响用户范围的划分标准。在用户数量分级上,本文将互联网超级平台用户数量(5亿)的十分之一作为影响用户范围的最高等级,后续依次递减。最终设计业务影响范围(S)的赋值规则如表2所示。

由于互联网业务广泛采用异地备份,单点故障未必会导致其原有影响范围内的全部用户受损,因此,在明确知晓业务已实现异地备份的情况下,可将业务影响范围等级下调一个级别。

2.2.3 故障危害类型(T)赋值设计

信息通信网络设备或系统的故障或失效对业务造成的危害类型相对集中,主要可从响应性能下降和业务中断两个方面加以考察。

然而,并非所有业务都对性能下降敏感。本文从网络运行事故所引发的社会、经济影响角度出发,仅关注性能下降

表1 业务重要性(I)参数赋值规则

赋值	说明
I3	党政领导机关业务,电力、燃气、水利、交通、公共服务、金融等基础民生业务
I2	固话及移动电话语音通信业务、卫星通信业务、国际通信业务
I1	短消息业务、数据业务及其他增值业务,一般行业业务(不属于I3范围的其他行业业务,如线上教育)

表2 业务影响范围(S)参数赋值规则

赋值	说明
S4	1)两个及以上省(或自治区、直辖市)的用户业务受到影响,或全国的用户业务受到影响 2)影响超过5 000万用户
S3	1)两个及以上地市的用户业务受到影响,或一个省(或自治区、直辖市)的用户受到影响 2)影响超过500万用户
S2	1)两个及以上县(区)的用户业务受到影响,或一个地市的用户受到影响 2)影响超过100万用户
S1	1)一个县(区)的用户业务受到影响 2)影响超过20万用户

超过50%、显著影响业务质量的情形。将性能下降比例阈值设为50%，是考虑到若存在一个负载均衡设备/系统或冗余备份设备/系统，当一个设备/系统因故障停止运行时，所能造成的最坏影响即为性能损失一半的情况。

对于业务中断，考虑到并非所有设备/系统故障都会立即引起中断，当故障经过一段时间后才导致中断时，系统或操作人员在中断发生前有机会发现异常并采取补救措施。因此，本文将中断类型分为延时中断和立即中断两种情况。此外，对于时效性要求较高的业务，显著的性能下降也可按延时中断处理。

另外，由于业务影响范围(S)的赋值设计中已纳入异地备份的修正因素，故在故障危害类型(T)的分级设计中不再重复考虑该因素，而是以业务无异地备份作为不同级别赋值的前置假设条件。

本文设计的故障危害类型(T)赋值规则如表3所示。

2.2.4 避免危险和限制伤害的可能性(P)分级设计

信息通信网络建设中通常会部署各类措施以预防危险事件发生或减轻其后果，相互独立的安全措施数量直接影响这些措施综合产生的避免或减轻效果。因此，本文根据具备明确效果的独立安全措施数量来确定避免危险和减轻后果的可能性(P)，并设计其赋值规则如表4所示。该设计与轨道交通领域EPC方法的理念相符。

表3 故障危害类型(T)参数赋值规则

赋值	说明
T3	在不考虑其他保护措施(例如业务异地备份)存在的前提下,故障发生后立刻造成业务中断(或无法正常提供服务)
T2	1)在不考虑其他保护措施存在的前提下,故障发生一段时间后才可能造成业务中断(或无法正常提供服务) 2)高时效要求的业务,其性能显著下降,显著影响业务质量
T1	性能下降超过50%,显著影响业务质量

表4 避免危险和限制伤害的可能性(P)参数赋值规则

赋值	描述	判定
P1	不可能	没有额外的措施来减轻危害变成事故的概率或避免风险业务可能遭受某种危害后果的伤害
P2	有一定可能	只有一种措施能明确地降低危害演变成事故的概率或避免风险业务遭受某种危害后果的伤害
P3	有较大可能	有两种及以上独立的措施能明确地降低危害演变成事故的概率或避免风险业务遭受某种危害后果的伤害

2.3 完整性推导

在完整性推导阶段，本文首先根据业务重要性(I)、业务影响范围(S)和故障危害类型(T)推导出危害后果严重程度(C)，再结合避免危险和减轻后果的可能性(P)推导出最终的NSIL等级。NSIL与SIL等级保持一一对应关系。

2.3.1 危害后果严重程度(C)推导

危害后果严重程度(C)参考《电信网络运行监督管理办法》划分为C1至C4共4个等级；同时，为提升场景覆盖能力，增设C0等级以涵盖可能造成较低危害的情形。各等级的具体描述如表5所示。

在根据业务重要性(I)、业务影响范围(S)和危害类型(T)推导危害后果严重程度(C)时，本文采用三元素矩阵的推导方式，以尽可能保证推导结果符合实际情况。相应推导规则如表6所示。

鉴于同一信息通信网络上可能同时运行多个重要业务，一个设备或系统故障可能同时影响多个业务，因此有必要对该场景下的危害后果严重程度进行修正，需综合考虑受影响业务的数量及其各自的重要程度等级，以判定最终等级。

假设同时影响的业务数量为 n ，危害后果严重程度等级

表5 危害后果严重程度(C)等级描述

危害后果严重程度等级(C)	描述
C0	对公共利益、社会经济和秩序造成较低危害
C1	对公共利益、社会经济和秩序造成一般危害
C2	对公共利益、社会经济和秩序造成较大危害
C3	对公共利益、社会经济和秩序造成重大危害
C4	对公共利益、社会经济和秩序造成特别重大危害

表6 危害后果严重程度(C)推导规则

业务重要性(I)	业务影响范围(S)	故障危害类型(T)		
		T1	T2	T3
I1	S1	--	--	C1
	S2	--	--	C1
	S3	--	C1	C2
	S4	C1	C2	C3
I2	S1	--	--	C1
	S2	--	C1	C2
	S3	C1	C2	C3
	S4	C2	C3	C4
I3	S1	--	C1	C2
	S2	C1	C2	C3
	S3	C2	C3	C4
	S4	C3	C4	C4

最高的业务等级为 C_x (其中 x 可选值为 1、2、3、4, 为危害后果严重程度 C 的具体等级数值, 下文 y 、 z 、 m 同), 次高为 C_y , 第三高为 C_z , 整体严重程度等级为 C_m , 其中需满足 $1 \leq z \leq y \leq x \leq 4$, 则修正规则如表 7 所示。

2.3.2 NSIL 推导

在获得危害后果严重程度 (C) 的等级后, 即可结合避免危险和减轻后果的可能性 (P) 的赋值推导出最终的 NSIL 等级, 具体推导规则如表 8 所示, 其中 NSIL 4 为最高等级。

NSIL 1—NSIL 4 可对应 IEC 61508 中 SIL 1—SIL 4, 适用相关等级的安全完整性要求。NSIL 0 可进行质量管理和基本的安全维护要求。

3 NSIL 定级方法验证

本文以某数据中心空调子系统为例, 结合 NSIL 定级方法与 IEC 61508 中的安全完整性要求, 对该系统进行安全分析, 识别其中的安全隐患, 从而验证 NSIL 定级方法的合理性与必要性。

根据 Uptime 发布的 2024 年数据中心运行事故原因统计报告^[24], 空调系统故障已成为网络运行事故的第二大诱因, 且易造成大面积影响。对空调系统开展安全完整性定级与要求制定, 对于提升数据中心网络运行安全能力具有重要意义。然而, 其他领域的安全完整性定级方法仅从人身安全角度出发, 难以反映空调系统故障对数据中心网络运行安全能力的影响。因此, 有必要采用本文提出的 NSIL 定级方法对该数据中心的空调子系统进行分析。

该数据中心承载了多个知名互联网业务, 如电商平台、

社交平台等。数据中心中央空调配备 8 台冷机, 其中 2 台既可用于制冷, 也可用于蓄冷。冷冻水系统管道与蓄冷池管道在与两用冷机连接时存在一段共用管道, 需要通过几个阀门的启闭来控制当前水系统与两用冷机的连接状态。该架构的简化示意图如图 1 所示, 其中 F1、F2、F3、F4 为控制冷冻水系统管道与蓄冷池管道隔离的关键阀门, 图中箭头标注了各管道的水流方向。空调 BA 系统在下发制冷或蓄冷指令时, 会相应调整阀门的启闭状态, 以使管道连接满足当前工作模式的需求。

在该数据中心空调子系统中存在一个安全隐患: 若阀门因故障未按 BA 系统指令正确启闭, 可能导致冷冻水系统管道与蓄冷池管道相连, 造成冷冻水流失、空调系统停机、机房温度升高, 最终引发服务器停机。因此, 需实现一项安全功能 (目标安全功能), 即在启动制冷或蓄冷模式前及运行期间, 对各个关键阀门的启闭状态进行检查, 以确保系统始终运行在正确状态。由于制冷、蓄冷模式由 BA 系统触发, 且 BA 系统可查看并控制各阀门状态, 因此由 BA 系统实现该安全功能的执行逻辑, 将阀门状态检查纳入制冷、蓄冷模式的启动流程中。

下面本文将使用所提方法分析 BA 系统的目标安全功能需达到的 NSIL 等级, 并对照 IEC 61508 中对应 SIL 等级的软件安全要求, 检验 BA 系统在设计实现过程中是否已应用相应的技术和措施。

3.1 NSIL 定级推导

1) 危害后果严重程度 (C) 推导

表 7 危害后果严重程度(C)修正规则		
参数取值		C_m 取值
$n = 2$	$x < 4$ 且 $x = y$	$C_m = C(x + 1)$
$3 \leq n < 10$	$x < 4$ 且 $x = y$, 或 $x < 4$ 且 $y = z = (x - 1)$	
$n \geq 10$	$x < 4$	
其他情况		$C_m = C_x$

表 8 NSIL 推导规则			
危害后果严重程度等级(C)	避免危险和减轻后果的可能性(P)		
	P3	P2	P1
C0	NSIL 0	NSIL 0	NSIL 0
C1	NSIL 0	NSIL 0	NSIL 1
C2	NSIL 0	NSIL 1	NSIL 2
C3	NSIL 1	NSIL 2	NSIL 3
C4	NSIL 2	NSIL 3	NSIL 4

NSIL: 网络运行安全完整性等级

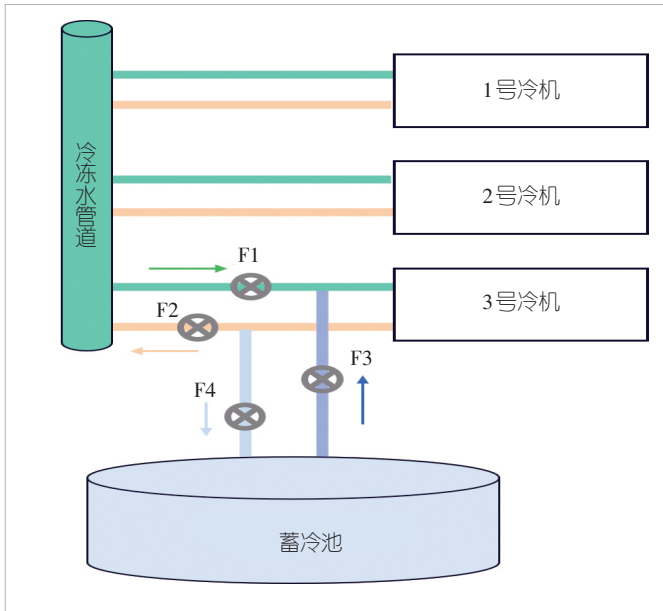


图1 某数据中心空调系统部分架构图

该数据中心承载多个业务,其中最为关键的包括一个知名电商平台和两个知名社交平台。各业务对应的危害后果严重程度推导分别如下:

(1) 知名电商平台业务的危害后果严重程度推导

在业务重要性方面,该电商业务属于一般行业业务,依据表1取值为I1。在业务影响范围方面,该电商平台在本地数据中心承载的业务为全国性业务,且用户数量超过8 000万,依据表2取值为S4。在故障危害类型方面,该安全隐患导致空调系统停机,一段时间后机房温度升高,服务器因过热宕机,进而引发业务中断,依据表3取值为T2。根据表6, I1、S4、T2组合对应的危害后果严重程度为C2,即前述故障对该知名电商平台业务造成的危害后果严重程度为C2。

(2) 知名社交平台业务的危害后果严重程度推导

两个社交平台业务同属一家公司,情况相似,此处以其中一个为例进行推导。

在业务重要性方面,社交业务属于一般行业业务,依据表1取值为I1。在业务影响范围方面,该社交平台在本地数据中心承载的业务为全国性业务,用户数量超过1亿,依据表2取值为S4;同时,该公司的社交平台业务均存在异地备份,因此将影响范围等级修正为S3。在故障危害类型方面,与电商平台案例一致,依据表3取值为T2。根据表6, I1、S3、T2组合对应的危害后果严重程度为C1。另一社交平台业务情况相同,危害后果严重程度同为C1。

(3) 总体危害后果严重程度推导

由于存在1个C2等级的业务和2个C1等级的业务,因此根据表7中所示的修正规则, $n = 3$, $x = 2$, $y = z = 1$, $C_m = C(x + 1) = C3$ 。即该数据中心空调子系统BA系统的总体危害后果严重程度(C)至少为C3。

2) 避免危险和限制伤害的可能性(P)判定

该数据中心配有冰块降温措施,可在空调系统报警后延长服务器过热宕机前的响应时间,有利于及时排查故障,提高阻止事故发生的可能性,降低危害演变为事故的概率。因此,依据表4,避免危险和减轻后果的可能性等级确定为P2。

3) NSIL判定

根据表8, C3、P2对应的NSIL等级为NSIL 2,即该数据中心空调子系统BA系统的目标安全功能需至少达到NSIL 2等级要求。

3.2 安全分析

IEC 61508-3^[25]对软件安全生命周期各环节应采取的技术和措施进行了详细介绍。受篇幅所限,本文以软件模块测试环节中的相关要求为分析对象。

在本案例中,BA系统需实现的安全功能与异常检测相关,因此选取SIL 2(对应NSIL 2)动态分析和测试中的“根据错误推测执行测试用例”要求作为切入点。根据IEC 61508-3及IEC 61508-7^[26]中的相关解释,该要求在测试用例设计时需考虑哪些特殊值或值的组合容易出错,并据此设计测试用例,以检验系统是否能正确处理这些异常情形。

对应本案例,根据图1所示的架构图,BA系统在实现功能时需考虑F1、F2、F3、F4这4个阀门的不同启闭组合,识别其中会导致不安全状态的组合,并在启动制冷、蓄冷模式前检查是否存在此类组合,进行相应处理;在产品测试验证环节,设计的测试用例也应覆盖所有不安全状态组合,以检验BA系统是否能正确处理。

经分析,在实际实现中,BA系统在启动蓄冷模式前会首先检查F1、F2阀门状态,确保两个阀门均关闭再启动蓄冷模式。然而,由于编程疏忽,BA系统实际实现的逻辑却变为:仅当F1、F2中有一个处于关闭状态时,即判断为安全状态,从而允许启动蓄冷模式。同时,在测试验证阶段也未考虑所有异常情况的检测,导致该安全隐患未被发现和治理。在某次BA系统启动蓄冷模式时,F1阀门意外处于开启状态,BA系统未能检测到该异常,正常执行蓄冷操作。蓄冷命令下达后,冷冻水系统与蓄冷池系统连通,导致冷冻水流失,空调系统停机,服务器过热宕机,进而造成该数据中心承载的业务中断,产生了严重的社会影响。

由此可见,在信息通信领域引入安全完整性体系,明确各安全功能的安全完整性等级及对应等级所应采取的技术和措施,将有助于提前发现安全隐患,提高信息通信网络运行环境的安全性,降低网络运行事故发生的概率。

4 结束语

本文分析了在信息通信领域引入安全完整性体系的必要性。通过分析过程工业、机械、汽车、轨道交通等领域的安全完整性定级方法,并结合信息通信领域自身特点,提出了NSIL定级方法。结合实际案例,验证了在信息通信领域引入安全完整性体系的必要性以及本文所提NSIL定级方法的合理性。

NSIL定级方法的提出仅为安全完整性体系在信息通信领域应用的第一步。文中的案例分析借鉴了IEC 61508中的安全完整性要求,但该标准主要面向工业控制领域,面对的系统相对简单且封闭,部分技术和措施并不完全适用于信息通信领域的设备和系统。因此,后续仍需结合信息通信领域的特色,制定网络运行安全完整性体系标准,明确各NSIL等级对应的技术和措施要求。

参考文献

- [1] IEC 61508-4-2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - part 4: definitions and abbreviations [S]
- [2] GB/T 20438.4-2017 电气/电子/可编程电子安全相关系统的功能安全 第4部分:定义和缩略语 [S]
- [3] GB/T 25069-2022 信息安全技术 术语 [S]
- [4] IEC 61508-5-2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - part 5: examples of methods for the determination of safety integrity levels [S]
- [5] GB/T 20438.5-2017 电气/电子/可编程电子安全相关系统的功能安全 第5部分:确定安全完整性等级的方法示例 [S]
- [6] 周荣义, 钟岸, 任竟丹, 等. 安全系统安全完整性等级确定方法比较研究 [J]. 中国安全生产科学技术, 2014, 10(3): 67-73. DOI: 10.11731/j.issn.1673-193x.2014.03.011
- [7] IEC 61511-3-2016 Functional safety - safety instrumented systems for the process industry sector - part 3: guidance for the determination of the required safety integrity levels [S]
- [8] GB/T 21109.3-2007 过程工业领域安全仪表系统的功能安全 第3部分:确定要求的安全完整性等级的指南 [S]
- [9] IEC 62061 CORR 1-2015 Safety of machinery - functional safety of safety-related electrical, electronic and programmable electronic control systems (edition 1.2) [S]
- [10] GB 28526-2012 机械电气安全 安全相关电气、电子和可编程电子控制系统的功能安全 [S]
- [11] ISO 13849-1-2015 Safety of machinery - safety-related parts of control systems - part 1: general principles for design (third edition) [S]
- [12] GB/T 16855.1-2018 机械安全 控制系统安全相关部件 第1部分:设计通则 [S]
- [13] ISO 26262-3-2018 Road vehicles - functional safety - part 3: concept phase (second edition) [S]
- [14] GB/T 34590.3-2022 道路车辆 功能安全 第3部分:概念阶段 [S]
- [15] European Commission. Analysis of safety requirements for MODsafe continuous safety measures and functions [EB/OL]. (2011-01-21) [2025-03-20]. http://www.modsafe.eu/fileadmin/documents/deliverables/DEL_D4_2_UITP_WP4_110121_V2_0.pdf
- [16] 张冬梅, 刘伟. 基于风险矩阵的轨道交通车辆安全完整性等级分析 [J]. 城市轨道交通研究, 2020, 23(8): 62-65. DOI: 10.16037/j.1007-869x.2020.08.015
- [17] 杨娟, 张小林. 基于EPC因子的有轨电车信号系统安全完善度等级(SIL)评估 [J]. 城市轨道交通研究, 2018, 21(4): 57-60+65. DOI: 10.16037/j.1007-869x.2018.04.014
- [18] 李葛亮, 刘明, 吕远斌. 城市轨道交通列车关键子系统的安全完整性等级分析 [J]. 城市轨道交通研究, 2021, 24(7): 98-101. DOI: 10.16037/j.1007-869x.2021.07.021
- [19] 白春光. 基于安全完整性级别分析的全自动驾驶地铁列车安全功能设计 [J]. 铁道技术监督, 2021, 49(4): 53-57
- [20] 俞杭华, 胡宗福, 薛小平. 城市轨道交通信号系统安全完整性研究 [J]. 大连交通大学学报, 2008, 29(6): 34-39
- [21] 刘亚妮, 王大平, 李梁, 等. 轨道车辆常见安全完整性等级确定方法研究 [J]. 电力机车与城轨车辆, 2023, 46(2): 108-112. DOI: 10.16212/j.cnki.1672-1187.2023.02.023
- [22] 电信管理局. 关于印发《电信网络运行监督管理办法》的通知 [EB/OL]. (2009-05-12) [2025-11-20]. https://www.miit.gov.cn/jgsj/xgj/gzzd/art/2020/art_01e36db302234d8b8b460e92a9889540.html
- [23] 网络交易监管司. 互联网平台分类分级指南(征求意见稿) [EB/OL]. (2021-10-29) [2025-11-28]. https://www.samr.gov.cn/cms_files/filemanager/1647978232/attach/20233/P020211029613134705.doc
- [24] Uptime Institute. Annual outage analysis 2025 [EB/OL]. (2025-05-06) [2025-11-20]. <https://uptimeinstitute.com/resources/research-and-reports/annual-outage-analysis-2025>
- [25] IEC 61508-3-2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - part 3: software requirements [S]
- [26] IEC 61508-7-2010 Functional safety of electrical/electronic/programmable electronic safety-related systems - part 7: overview of techniques and measures [S]

作者简介



路晔绵, 中国信息通信研究院高级工程师; 主要从事智能终端/信息通信设备网络安全、网络运行安全等方面的研究和测试评估工作; 参与制定国家标准2项, 牵头制定通信行业标准和团体标准14项, 参与制定行标和团标25项, 发表论文6篇, 拥有专利2项、软著1项。



郑梦青, 中国信息通信研究院工程师; 主要从事网络运行安全、网络安全、信息通信网络检测认证相关研究; 牵头制定通信行业标准和团体标准4项, 发表论文3篇, 拥有专利1项。



吴荣春, 中国信息通信研究院工程师; 主要从事网络安全研究和攻防测试评估相关工作; 参与多项安全领域国家重点研究项目, 在设备安全检测、网络安全评估、渗透攻防测试等领域拥有丰富经验; 参与编写国家标准5项、电力行业标准2项、通信行业团体标准6项, 牵头完成重点重大课题10余项, 发表论文6篇, 拥有专利2项。



张治兵(通信作者), 中国信息通信研究院泰尔系统实验室副总工, 高级工程师, 同时担任电信终端产业协会(TAF)网络设备技术与安全技术组组长、中国通信标准化协会(CCSA)网络关键设备工作组组长; 长期从事通信安全领域研究工作, 在网络运行安全与通信安全生产、网络与信息安全、通信产品检验检测等方面有多项研究成果; 牵头研究制定国家标准5项、通信行业标准和团体标准32项, 拥有发明专利4项, 发表论文15篇, 出版专著2项。



王晓颖, 中国信息通信研究院工程师; 主要从事信息通信行业安全生产、网络运行安全等研究工作; 参与通信网络相关科研课题5项, 发表论文2篇。

AM-WiGeR: 一种基于注意力机制增强残差网络的 Wi-Fi 手势识别方法



AM-WiGeR: A Wi-Fi Gesture Recognition Method Based on Attention Mechanism Enhanced Residual Network

李业深/Li Yeshen, 姚可欣/Yao Kexin, 熊轲/Xiong Ke

(北京交通大学, 中国 北京 100044)
(Beijing Jiaotong University, Beijing 100044, China)

DOI: 10.12142/ZTETJ.202604012

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20260902.0959.002>

网络出版日期: 2026-08-24

收稿日期: 2026-03-06

摘要: 提出了一种基于注意力机制增强残差网络的 Wi-Fi 手势识别方法。该方法从 Wi-Fi 设备采集的信道状态信息 (CSI) 中提取多普勒频移 (DFS) 作为特征源, 通过共轭矩阵相乘、天线对选择及相位偏移消除等方法, 从中提取能有效表征手势信息的向量参数, 并进一步通过短时傅里叶变换获得对应的 DFS 频谱图特征。然后, 以 DFS 频谱图特征作为 AM-RES 模型的输入, 进行人体手势识别。实验结果表明, 相较于现有方法, 所提 AM-WiGeR 方法对不同朝向手势识别的平均准确率达到 96.22%, 取得了良好的手势识别效果。

关键词: 手势识别; CSI; DFS; 注意力机制

Abstract: A Wi-Fi gesture recognition method based on Attention Mechanism Enhanced Residual Network (AM-RES) is proposed, termed AM-WiGeR (AM-RES based Wi-Fi Gesture Recognition). Channel state information (CSI) collected from widely deployed Wi-Fi devices is utilized, from which Doppler frequency shift (DFS) is extracted as the feature source. Through conjugate matrix multiplication, antenna pair selection, and phase offset elimination, vector parameters that can effectively characterize gesture information are extracted from the DFS, and the corresponding DFS spectrogram features are further obtained via short-time Fourier transform. The DFS spectrogram features are then fed into the AM-RES model for human gesture recognition. Experimental results demonstrate that, compared with existing methods, the proposed AM-WiGeR method achieves an average recognition accuracy of 96.22% for gestures performed in different orientations, indicating satisfactory gesture recognition performance.

Keywords: gesture recognition; CSI; DFS; attention mechanism

引用格式: 李业深, 姚可欣, 熊轲. AM-WiGeR: 一种基于注意力机制增强残差网络的 Wi-Fi 手势识别方法 [J]. 中兴通讯技术, 2026, 32(4): 79-87. DOI: 10.12142/ZTETJ.202604012

Citation: Li Y S, Yao K X, Xiong K. AM-WiGeR: A Wi-Fi gesture recognition method based on attention mechanism enhanced residual network [J]. ZTE technology journal, 2026, 32(4): 79-87. DOI: 10.12142/ZTETJ.202604012

新一代信息技术的持续演进, 推动了各类智能应用的发展, 这对人体动作及行为识别的需求日趋迫切。在诸多新兴人机交互技术中, 手势动作识别技术能够为用户提供和谐、简便、友好的操作体验^[1], 已成为当前主流的人机交互方式之一, 在智能家居、智慧商场等领域具有广阔的应用前景。

传统的手势识别方法可分为 3 类: 基于计算机视觉的方法、基于穿戴式传感器的方法和基于红外感知的方法^[2]。这 3 种技术虽各有优势, 但也均存在局限性。例如, 基于计算机视觉的方法受光照和遮挡影响较大, 且存在较高的用户隐私

泄露风险; 基于穿戴式传感器的方法要求用户持续佩戴传感器设备, 安装与穿戴不够便捷, 影响了用户的使用体验; 基于红外感知的方法需要布设专业精密设备, 增加了应用成本, 同时其感知范围有限, 难以满足大规模应用的需求^[3]。

通感一体化技术近年来发展迅速, 无线信号因同时具备通信与感知能力而受到极大关注^[4-8]。利用无线信号实现对人体动作的非接触式感知, 正逐步取代传统识别技术^[9-11], 并已引起广泛关注。

现有基于无线信号的动作识别方法, 按信号类型的不同,

可分为基于超宽带 (UWB)、基于射频识别 (RFID) 和基于 Wi-Fi 的方法等^[12]。基于 UWB 的方法需要部署大量较为昂贵的 UWB 设备, 部署成本相对较高; 基于 RFID 的方法要求识别目标随身携带标签, 而标签具有一定体积, 使用不够便捷, 且相对容易损坏; 相较而言, 基于 Wi-Fi 的动作识别方法可利用现有已广泛部署的 Wi-Fi 无线路由器进行工作, 无须额外布设专用设备; 采用非接触感知方式实现感知, 用户体验良好; 相较于基于计算机视觉的方法, 不受光照等环境条件影响, 且无须采集人体图像数据, 对用户隐私的泄露风险较低。因此, 基于 Wi-Fi 的动作识别已发展成为新的研究热点^[13]。

Wi-Fi 信号传播对环境感知灵敏, 人体动作会改变 Wi-Fi 信号的直射、反射和散射路径。当 Wi-Fi 覆盖区域内的人体手势等动作发生变化时, Wi-Fi 接收器 (如电脑、手机等) 所接收信号的强度、到达角度等参数也会发生相应变化。因此, 通过分析接收端数据变化与人体动作变化之间的作用规律, 即可实现对人体动作的识别。目前, 常见的基于 Wi-Fi 的动作识别方法通常基于接收信号强度指示 (RSSI) 或信道状态信息 (CSI)。RSSI 记录了接收端无线电信号的平均功率, 能够提供关于通信链路的相对粗粒度特征^[14]; CSI 则记录了每个 Wi-Fi 子载波信道的幅度和相位信息, 能够提供关于通信链路的相对细粒度特征^[15]。2010 年, Halperin 等通过改造 Wi-Fi 设备固件, 实现了对 IEEE 802.11n 标准无线网卡上 CSI 的获取, 从而推动了基于 CSI 的人体动作识别研究^[16-18]。

传统基于 CSI 的手势识别通常采用机器学习方法, 对从 CSI 中提取的统计特征或物理特征进行分类^[9], 从而实现对人体手势的识别。例如, 文献[20]从 CSI 中提取幅值、相位和信号到达角度 3 种信息作为特征, 利用 K 近邻 (KNN) 方法进行分类, 实验结果表明该方法对 5 种动作的识别准确率达到 92.6%。传统基于 CSI 的手势识别方法的不足之处在于, 需要手动设计和提取结构化数据特征作为机器学习算法的输入, 而这类方法通常采用信号变换等传统手段提取易于表示的统计特征, 导致所提取特征的维度和适用范围较为有限。深度学习是一种具有强大特征表达能力的机器学习方法, 其输入可以直接为非结构化数据。深度学习能够通过神经网络模型自动提取输入数据的特征, 从而实现更高效、更精确的识别效果。因此, 近期越来越多的研究开始利用深度学习^[21]开展 Wi-Fi 人体手势识别研究。文献[22]通过设计手势速度关系模型, 利用卷积神经网络 (CNN) 提取空间特征, 并利用门控循环单元 (GRU) 提取时间特征, 将提取到的动作速度信息作为特征输入神经网络模型, 其识别准确率可达 92.7%。文献[23]则以 CSI 相位差矩阵作为特征输入, 采用双向门控循环网络从前后两个方向对相位差序列进行自动学习和特征提取,

实现对 4 种手势的识别, 其准确率达 94%。

综合上述相关研究发现可知, 现有利用 CSI 进行手势识别的研究虽已取得大量有价值的成果, 但仍存在若干亟待解决的问题。例如, 利用 CSI 的物理特征和统计特征表征动作时, 无法有效消除环境因素的影响, 导致特征难以准确表示手势信息; 再如, 现有深度学习模型在训练时利用输入特征的全局信息, 难以有效聚焦于与手势动作相关的特征, 从而导致识别效果不佳^[24]。

针对上述问题, 本文首先提出了一种基于注意力机制增强残差网络 (AM-RES) 模型。该模型采用具有深层网络结构且不易退化的残差网络作为骨干网络, 同时引入注意力机制以自动捕捉与手势动作相关的特征, 从而实现更高的识别准确率。在此基础上, 提出了一种基于 AM-RES 的手势识别方法, 即 AM-WiGeR。该方法利用现有广泛部署的 Wi-Fi 设备采集 CSI 数据, 通过共轭乘法、天线对选择和相位偏移消除等手段, 从中提取与动作相关性更高的多普勒频移 (DFS) 信息, 并进一步通过短时傅里叶变换得到对应的 DFS 频谱图特征。然后, 以 DFS 频谱图特征作为输入, 利用所提出的 AM-RES 模型进行人体手势识别。最后, 在自测数据集上对所提方法进行实验验证。该方法能够取得良好的手势动作识别效果, 对不同朝向手势识别的平均准确率达到 96.22%。

1 基于 CSI 手势识别原理

在室内环境中, 人的手动动作会引起 Wi-Fi 信号传播路径和传播特性的改变, 具体表现为接收信号中提取的 CSI 特征发生相应变化。不同的手动动作会对无线信道的 CSI 产生不同的影响。因此, 可利用观测到的 CSI 特征来分析引起该 Wi-Fi 信号变化的人体动作, 从而实现基于 Wi-Fi 信号的手势动作识别。本节将介绍 CSI 的基本概念及 Wi-Fi 信号的传播模型。

1.1 CSI

在无线通信系统中, CSI 是通信链路的信道属性, 能够描述信道衰落、散射和传播距离等因素的综合影响。Wi-Fi 采用多输入多输出 (MIMO) 和正交频分复用 (OFDM) 技术来发送和接收数据包, CSI 可在不同天线对和子载波上测量获得。在 t 时刻第 k 个子载波的无线信道可定义为:

$$\begin{bmatrix} Y_1[k, t] \\ \vdots \\ Y_n[k, t] \end{bmatrix} = \begin{bmatrix} H_{11}[k, t] & \cdots & H_{1m}[k, t] \\ \vdots & \ddots & \vdots \\ H_{n1}[k, t] & \cdots & H_{nm}[k, t] \end{bmatrix} \begin{bmatrix} X_1[k, t] \\ \vdots \\ X_m[k, t] \end{bmatrix} \quad (1),$$

其中, X 为发射端信号, Y 为接收端信号, H 是表示 CSI 信息的信道矩阵, m 表示发射天线数量, n 表示接收天线的数量。

对于每个子载波其CSI值是一个复数,实部表示CSI幅度信息,虚部表示相位信息。因此,当一个人在Wi-Fi信号的传播环境中做出动作时,会对接收端的信号特性产生一定影响,并反映在CSI的幅值和相位信息中。

1.2 Wi-Fi信号传播模型

为了有效地描述由手势动作引起的Wi-Fi信号变化,接下来我们将介绍Wi-Fi信号如何传播以及动作影响无线信道的原因。如图1所示,手势动作识别系统由发射端(X)、接收端(Y)和用户组成。

Wi-Fi信号在室内传播时,由于受到墙壁、地面、人体等的阻碍,发生反射、散射、衍射等现象,这使得接收端收到来自不同路径的多个信号,称为多径效应。在多径效应下接收端的CSI表示为:

$$h(f, t) = \sum_{i=1}^L A_i(t) e^{-j2\pi f \tau_i(t)} \quad (2),$$

其中, f 是信号的原始载波频率, L 表示总的路径条数, $A_i(t)$ 是第 i 条路径的幅度衰减, $\tau_i(t)$ 是第 i 条路径的传播延迟。

多普勒效应是指当发射端和接收端之间存在相对运动时,接收端接收到信号的频率与发射端发射信号的频率不同,两者之差称为多普勒频移。当人体执行手势动作时,会间接改变Wi-Fi信号发射端与接收端之间的相对距离,从而产生多普勒效应。DFS的大小与路径长度变化速率相关,可表示为:

$$f_D = f \frac{v_{\text{path}}}{c} \quad (3),$$

其中, v_{path} 是路径长度变化的速度, c 是Wi-Fi信号的传播速度。

因此,CSI可以用DFS表示为:

$$h(f, t) = \sum_{i=1}^L A_i(t) e^{-j2\pi f \left(\frac{v_i t}{c} \right)} = \sum_{i=1}^L A_i(t) e^{-j2\pi f_{D(i)} t} \quad (4),$$

其中, v_i 表示第 i 条路径长度变化速度, $f_{D(i)}$ 表示第 i 条路径下由动作产生的多普勒频移。如图1所示,在多径效应下,Wi-Fi信号到达接收端的路径可划分为静态路径和动态路径。静态路径是指由LoS路径以及静止物体反射的路径,其DFS值为0。动态路径是指由手势动作反射的路径,动作会引起路径长度随时间发生变化,导致DFS值不为0。因此,CSI值可以分为静态分量和动态分量:

$$h(f, t) = h_s(f, t) + \sum_{i \in G} A_i(t) e^{-j2\pi f_{D(i)} t} \quad (5),$$

其中, $h_s(f, t)$ 为静态分量, G 表示动态路径。因此,可以从CSI中提取动态分量,去除环境干扰信息,并通过时频分析进一步提取出DFS频谱图特征,更有效地表示手势动作。

2 AM-WiGeR 方法

本文提出的AM-WiGeR手势识别方法具有以下两个特点:在特征选取方面,提取DFS频谱图特征,能够降低环境因素的干扰,更有效地表征手势动作;在深度学习模型方面,提出结合注意力机制的残差网络模型AM-RES进行手势识别,从而有效提升识别准确率。

如图2所示,所提AM-WiGeR方法由4个功能部分组成,分别为数据采集、数据预处理、特征提取和动作识别。数据采集部分利用Wi-Fi收发设备采集CSI数据;数据预处理部分通过对原始CSI数据进行滤波器去噪、相位偏移消除及主成分分析,获取预处理后的CSI;特征提取部分采用静态分量消除、天线对选择和短时傅里叶变换等方法提取DFS频谱图;动作识别部分以DFS频谱图作为输入,利用AM-RES实现手势识别。本节将介绍这4个功能部分的工作原理与具体实现。

2.1 数据采集

所提AM-WiGeR方法的第1部分是数据采集模块,负责CSI数据的获取,涉及的硬件装置包括1个Wi-Fi发射端和1个Wi-Fi接收端。需要说明的是,随着Wi-Fi采用MIMO和OFDM技术,几乎在所有支持IEEE 802.11n标准的无线网卡上都能获取到CSI。因此,Wi-Fi手势识别的发射端可采用常见的支持IEEE 802.11n标准的路由器,接收端则为配有多根天线的支持IEEE 802.11n标准的无线网卡。相较于传统基于单天线单链路的手势识别方法,采用多根天线接收CSI,所采集的数据能够更完整地保留手势特征。

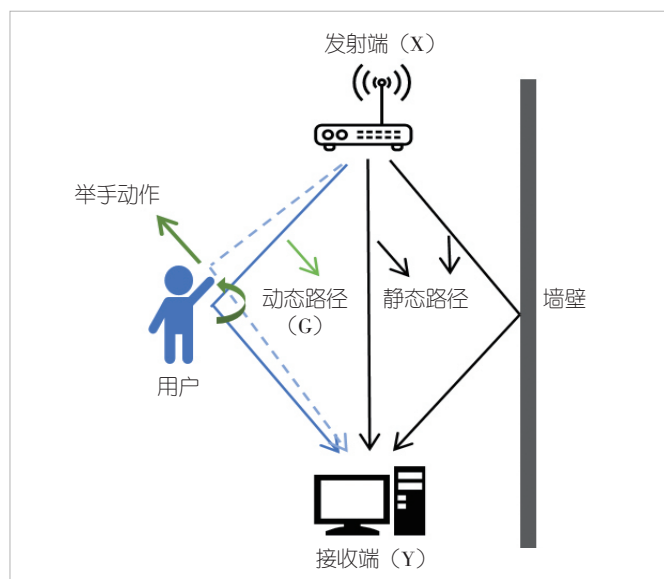


图1 室内环境Wi-Fi信号传播示意图

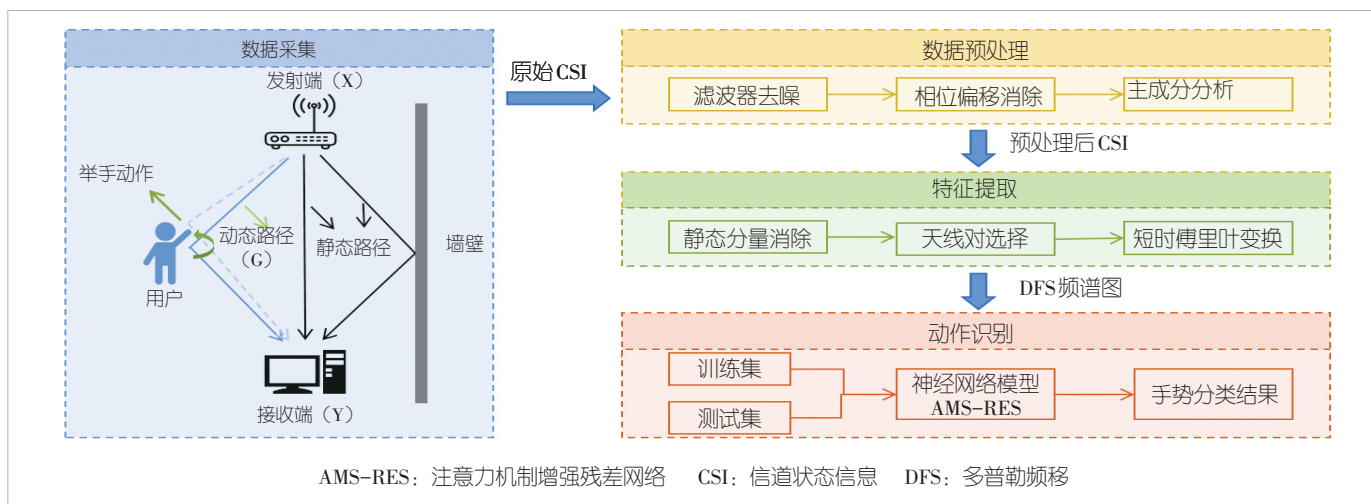


图2 AM-WiGeR整体结构图

2.2 数据预处理

1) 滤波器去噪

室内环境中存在与动作无关的背景噪声及由于干扰导致的异常值,因此采集到的原始CSI数据无法直接用于动作识别,需先进行去噪处理。巴特沃斯滤波器能够平滑通频带内的频率曲线,且设计简单、性能良好,相较于其他方法,可有效消除背景噪声和异常值,同时保留手势相关信息。因此,所提AM-WiGeR方法采用巴特沃斯滤波器对数据进行去噪处理。图3给出了两根天线 r_1 、 r_2 上第15个子载波的原始CSI数据及经巴特沃斯滤波处理后的CSI数据。可以看出,经滤波处理后,波形更加平滑,噪声得到有效抑制。

2) 相位偏移消除

由于Wi-Fi收发设备之间的时钟不同步,CSI的相位存在时变随机偏移,不能直接用于DFS特征的提取。含有相位偏移的CSI可以表示为:

$$h(f, t) = e^{-j\theta_{\text{offset}}} \left(h_s(f, t) + \sum_{i \in G} A_i(t) e^{-j2\pi f D_i(t)} \right) \quad (6).$$

由于接收端的网卡配有多根天线,这些天线的相位偏移相同,所以通过对接收端的两根天线的CSI矩阵做共轭乘法,可有效消除相位偏移。

3) 主成分分析

IEEE 802.11n标准采用多载波传输方式,接收端可收集包含多个子载波的CSI。由于CSI数据维度较高,易导致特征信息冗余,不利于后续特征提取,因此需对数据进行降维处理。主成分分析(PCA)能够在保留绝大部分有效信息的前提下对数据进行降维,提升对样本中有效信息的处理效率。因此,所提AM-WiGeR方法采用PCA分解得到第1主成分。由于第1

主成分包含了由动作引起的主要功率变化的大部分信息,可将其用于后续DFS特征的提取。

2.3 特征提取

经过上述步骤得到的数据仍包含与手势无关的冗余信息,

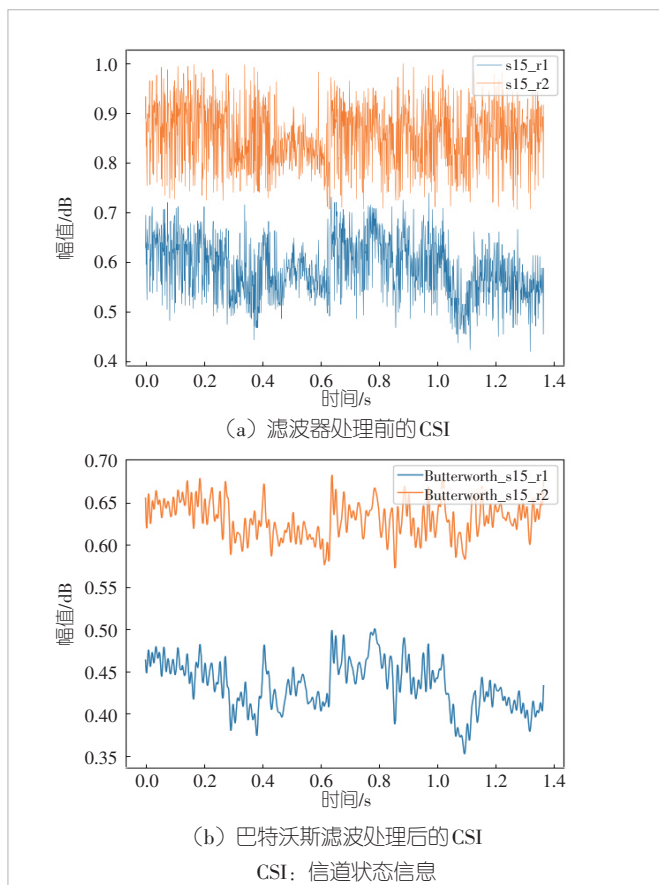


图3 巴特沃斯滤波器去噪前后CSI波形图

需进一步进行特征提取, 以利于后续手势识别。特征提取是指将原始信息转换为新的、更有效的数据描述形式, 这些形式更易于被动作识别模型有效利用。首先, 对预处理后的 CSI 进行静态分量消除和动态分量放大, 以提取 DFS 信息; 然后, 利用短时傅里叶变换将 DFS 特征信息可视化, 生成 DFS 频谱图, 作为深度学习算法的稳定输入特征。下面将详细介绍特征提取步骤。

1) 静态分量消除

由公式 (5) 知, CSI 可表示成静态分量和动态分量的和。由于 DFS 只存在于 CSI 的动态分量中, 因此需要对 CSI 进行分解, 消除静态分量, 在动态分量中提取 DFS 特征。

首先分解预处理后的 CSI 信息为静态分量和动态分量:

$$h_{cm}(f, t) = h_1(f, t)h_2(f, t)^* = \quad (7),$$

$$(h_{1,s}(f, t) + \sum_{i \in G_1} A_i(t)e^{-j2\pi f_{D(i)}})(h_{2,s}(f, t)^* + \sum_{k \in G_2} A_k(t)e^{j2\pi f_{D(k)}}) = h_{1,s}(f, t)h_{2,s}(f, t)^* + \quad (8),$$

$$h_{1,s}(f, t) \sum_{k \in G_2} A_k(t)e^{j2\pi f_{D(k)}} + \quad (9),$$

$$h_{2,s}(f, t)^* \sum_{i \in G_1} A_i(t)e^{-j2\pi f_{D(i)}} + \quad (10),$$

$$\sum_{i \in G_1, k \in G_2} A_i(t)A_k(t)e^{j2\pi t(f_{D(k)} - f_{D(i)})} \quad (11),$$

其中, 公式 (11) 为 2 根天线的动态分量之积, 其值非常小, 可忽略; 公式 (9) 和式 (10) 为 1 根天线的静态分量与另 1 根天线的动态分量之积, 其中包含所需的 DFS 信息; 公式 (8) 为 2 根天线的静态分量之积。静态分量中不含 DFS 信息, 但含有直接路径信号, 导致其功率较高。通过从预处理后的 CSI 中减去其平均值来消除静态分量, 可避免静态分量对 DFS 估计的干扰。

2) 天线对选择

公式 (9) 和 (10) 所表示的动态分量中包含所需的 DFS 信息, 但由于 2 根相邻天线具有相似的多径, 导致所包含的 DFS 信息幅值相近但方向相反。若直接利用上述 2 个动态分量计算 DFS, 会导致两者相互抵消。因此, 为从动态分量中有效提取 DFS, 需设计天线对选择策略, 以放大其中 1 个动态分量 (公式 (9)) 并缩小另 1 个动态分量 (公式 (10))。由文献[26]可知, CSI 均值较大的天线具有较高的静态分量, CSI 方差较大的天线具有较高的动态分量。因此, 在选择用于共轭矩阵相乘的 2 根天线时, 第 1 根天线选择均值与方差之比最大

的天线 $\text{Max}(\frac{E(h_i)}{\text{var}(h_i)})$, 第 2 根天线选择均值与方差之比最小

的天线 $\text{Min}(\frac{E(h_i)}{\text{var}(h_i)})$ 。通过上述天线选择步骤, 即可获得

DFS 向量参数。

3) 短时傅里叶变换

经过上述步骤, 从 CSI 中得到的 DFS 特征为大小为 $m \times n \times k \times t$ 的 4 维向量, 其中 m 代表发射天线数, n 代表接收天线数, k 代表子载波数, t 代表时间戳。为提取能够更直观体现手势信息的特征, 对 DFS 向量参数进行短时傅里叶变换 (STFT), 并生成 DFS 频谱图。传统的快速傅里叶变换 (FFT) 仅能反映信号在频域的特性, 无法在时域对信号进行分析。而 STFT 能够从时间和频率两个维度描述手势对 DFS 的影响。STFT 通过在时间轴上移动窗函数, 选用高斯窗函数截取信号片段, 并对每个片段进行 FFT 操作, 从而获得 DFS 特征, 即 DFS 频谱图。图 4 给出了两种不同动作的 DFS 频谱图, 其中, 图 4 (a) 为推拉手势对应的 DFS 频谱图, 图 4 (b) 为画圈手势对应的 DFS 频谱图。由图 4 可知, 两种动作下 DFS 的变化存在明显差异。因此, DFS 频谱图可作为动作识别的数据特征, 以用于区分不同手势。

2.4 动作识别

1) 注意力机制

动作识别部分是 AM-WiGeR 的核心模块。目前大多数动作识别方法采用深度学习技术, 通过神经网络模型自动提取输入数据特征, 实现手势的分类识别。然而, 现有神经网络模型大多利用输入特征的全局信息进行训练, 难以有效聚焦于与手势相关的特征信息。为此, 本文提出了基于注意力机制增强的残差网络模型 AM-RES。该模型采用经典的残差网络 ResNet-18 作为主干网络, 如图 5 所示, 其扩展部分在于引入了注意力机制, 能够依据权重值判断各信息的重要程度, 自动提取与手势相关的特征, 并实现手势识别。

在神经网络的学习过程中, 注意力机制能够对当前输入特征信息的重要性进行评估, 得到对应的权重值, 并在后续网络学习过程中依据权重值判断各信息的重要程度, 从而决定保留或丢弃相应信息。图 6 给出了注意力机制的基本结构。对于输入的图像特征, 注意力机制生成对应的注意力特征:

$$W = f(P_f, C) \# (2 - 7) \quad (12),$$

$$C' = C \otimes W \quad (13),$$

其中, P_f 是参数矩阵, 通过一系列函数 f 的作用得到注意力权

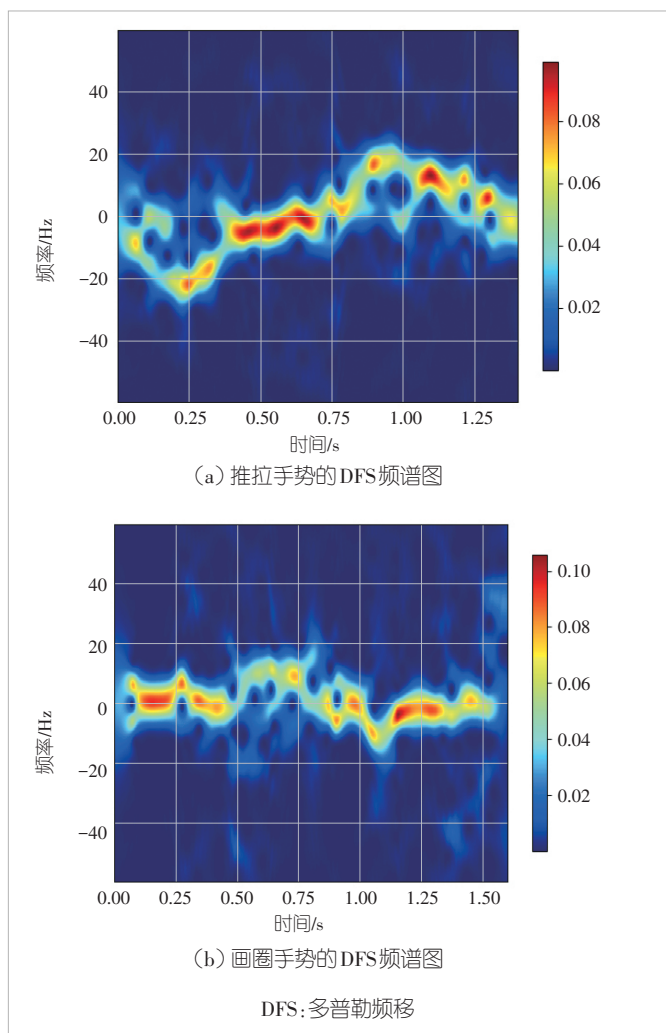


图4 推拉和画圈手势的DFS频谱图

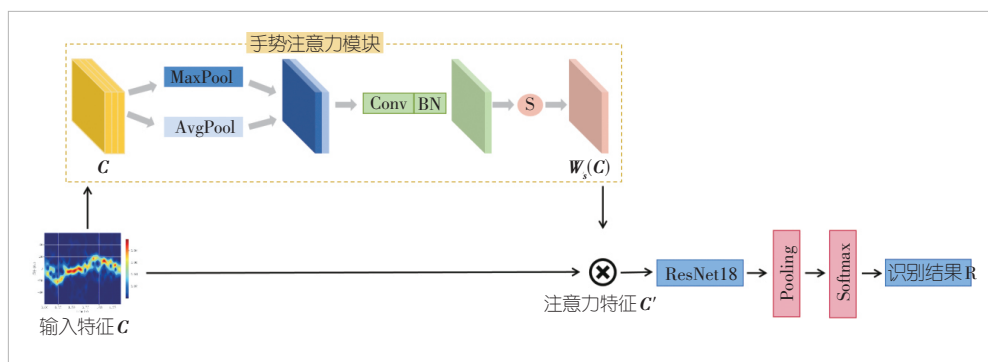


图5 本文提出的AM-RES手势识别模型网络架构

值矩阵 W , W 中的值对应不同像素在 C 中的权重。权重越大, 对应的像素越重要。然后将 C 与 W 按元素乘, 得到注意力特征 C' 。

2) AM-RES模型

模型输入特征为DFS频谱图 C , 本文引入的注意力模块为

C 生成注意力权值矩阵 $W_s(C)$, 其计算公式为:

$$W_s(C) = \text{sigmoid}\left(\text{BN}\left(\text{Conv}\left(\text{cat}\left(\text{AvgPool}(C), \text{MaxPool}(C)\right)\right)\right)\right)(14),$$

其中, Conv表示卷积运算, BN表示批归一化运算, AvgPool和MaxPool分别代表平均值池化和最大值池化操作。DFS频谱图 C 首先通过组合池化层进行特征降维。组合池化层包括最大值池化与平均值池化, 结合二者优势同时关注重要的局部特征和全局特征, 丰富了特征信息。然后进行卷积运算、批归一化运算, 通过Sigmoid激活函数计算出 $W_s(C)$ 后, 将其与 C 相乘以进行自适应特征优化, 得到注意力特征 C' 。

$$C' = W_s(C) \otimes C \quad (15),$$

然后通过预训练的ResNet18对注意力特征进行处理, ResNet18采用残差网络(ResNets)结构, 如图7所示。

在不同卷积层之间添加跨层连接, 使前面卷积层的数据可越过中间层直接传入更深的卷积层, 从而有效缓解梯度消失问题, 并提取更深层的特征信息。最后, 通过池化层和Softmax层获得识别结果, 实现对6种手势的分类识别。训练过程中, 采用交叉熵损失函数对模型分类结果与真实标签进行计算, 得到分类损失值。神经网络基于反向传播算法更新参数, 并持续迭代直至模型收敛。通过上述方式, 注意力模块能够自适应地学习如何生成准确的注意力特征, 促使网络更多地关注与手势相关的信息。同时, 预训练的ResNet-18网络可有效提取深层特征, 从而获得更优的分类效果。

3 实验设置与结果分析

3.1 实验环境及设置

本文实验在北京交通大学网络实验室完成, 实验场景示意图及实际部署情况如图8所示。实验室面积约为 $6\text{ m} \times 3.5\text{ m}$, 室内摆放有桌子、椅子, 以及显示器、电脑等实验设备若干。实验硬件设备具体包括Wi-Fi信号发送端和接收端, 其中发射端为支持IEEE 802.11n协议的TP-LINK TL-WR886N型号、带有3根天线的路由器, 接收端为配备3根天线的Intel 5300网卡的主机, 操作系统为Ubuntu 16.04。Wi-Fi发射端与接收端的间距为2 m, 距离地面的高度均为0.8 m, 实验人员坐在收发链路中间执行预设

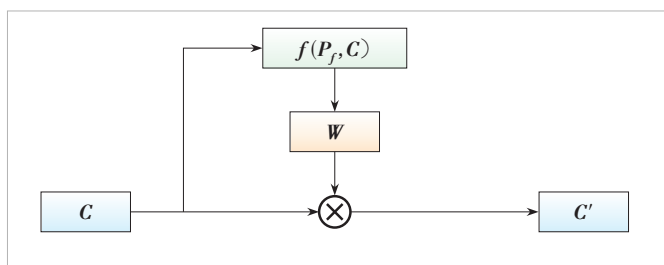


图6 注意力机制的基本结构

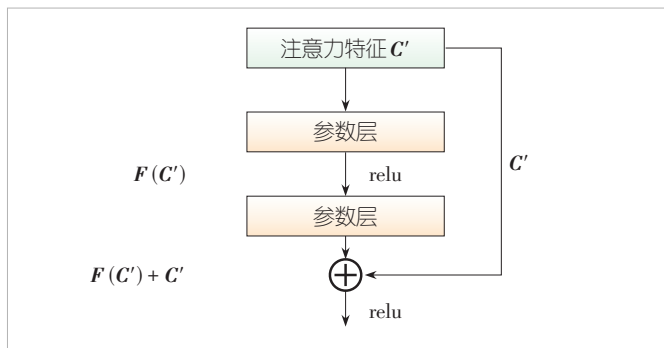


图7 残差结构示意图

手势动作。

3.2 实验数据采集

实验采集了室内环境下实验人员执行不同手势动作时接收端的CSI数据。通过Halperin等设计的CSI Tools软件采集接收端CSI数据，采样频率设置为1 kHz。实验采集的手势包括6种常用的人机交互动作，分别为拍手、举手、平扫、推拉、画圈和交叉。在采集数据时，为避免人体位于收发链路之间因朝向不同而产生的差异，实验人员分别朝向自身的正前方、右前方和左前方3个方向执行手势动作。每个动作执行时间为2s，每种动作各执行900次，最终收集5 400组数据。在深度学习模型训练过程中，80%的数据为训练集，20%为测试集。

3.3 实验结果与分析

本文使用评价指标分为识别准确率和混淆矩阵。

1) 准确率验证与对比实验

(1) 模型准确率验证

在手势识别问题中，手势识别准确率，即分类正确的样本数占总样本数的比例，是最重要的评判指标之一，能有效反映模型的识别效果。图9给出了本文所提AM-WiGeR方法在训练和测试过程中对6种手势识别的平均准确率，其中横轴表示迭代轮次，纵轴表示准确率。最终，模型在测试集上的平均识别准确率达到96.22%。

(2) 不同特征对识别准确率影响

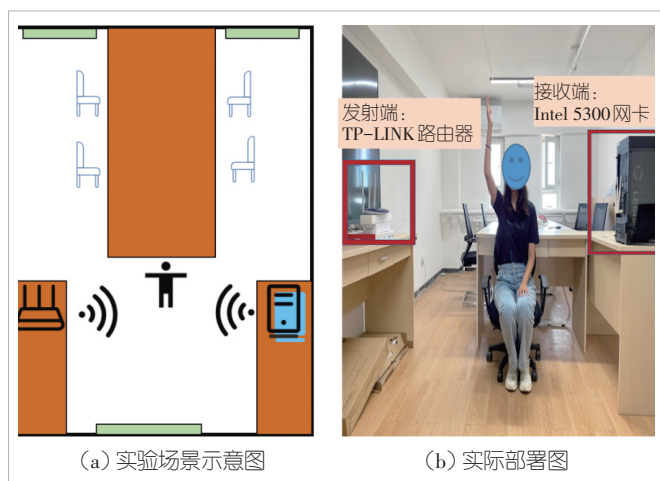


图8 Wi-Fi手势识别实验场景

为探索不同特征对识别准确率的影响，分别将CSI幅值、CSI相位、CSI幅值与相位的融合信息以及所提AM-WiGeR方法提取的DFS频谱图作为模型输入，并对比其识别准确率，结果如图10所示。

由图10所示的实验结果可知，采用DFS频谱图作为特征时识别准确率最高，这验证了所提DFS特征提取方法的有效性。采用CSI相位特征时识别效果最差，这是由于CSI相位存在固有偏移，仅依靠相位信息难以获得理想的识别效果。与CSI幅值和相位信息相比，DFS能够消除环境中静态路径的干扰，同时有效保留动态路径的CSI信息。因此，采用DFS频谱图作为输入特征时，手势识别准确率更高。

(3) 不同识别算法的准确率比较

本文还对比了不同识别算法的效果。除本文所提AM-RES模型外，还对比了未引入注意力机制的ResNet-18模型，以及文献[23]中WiHGR算法和文献[22]中Widar3.0算法。将这

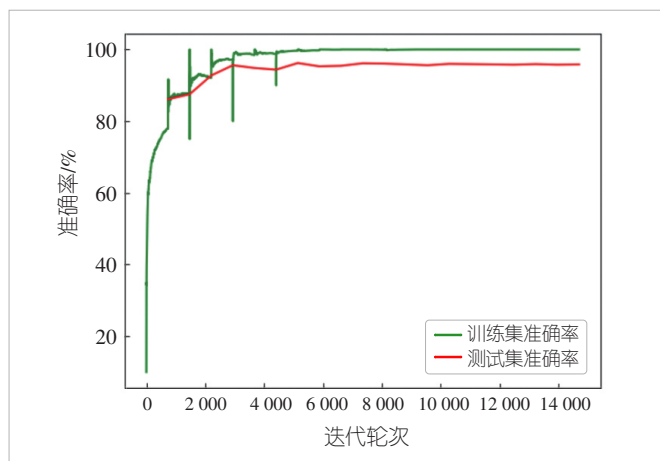


图9 AM-WiGeR方法识别手势的平均准确率

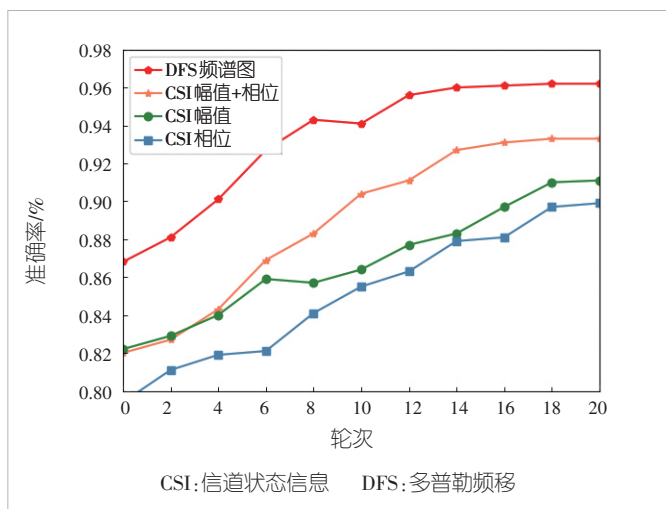


图10 不同特征的识别准确率

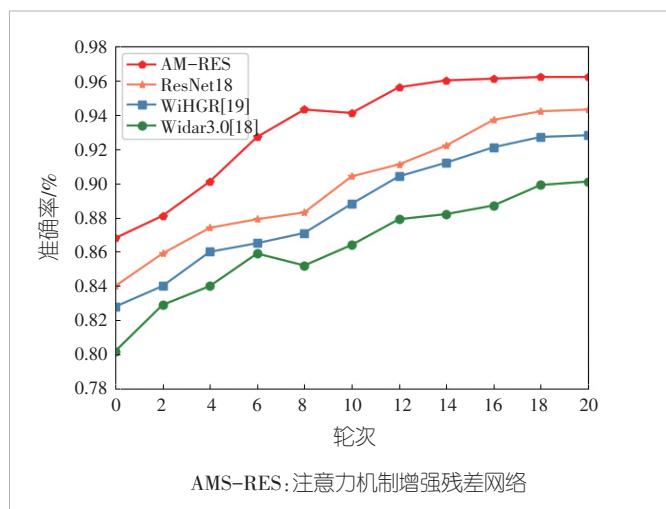


图11 不同识别方法下准确率效果

4种算法在相同数据集上进行实验,并对比其识别准确率,结果如图11所示。

由图11所示结果可知,相较于其他算法,本文所提AM-RES算法的识别准确率显著提升,且模型收敛速度更快,验证了所提算法的有效性。文献[22]中Widar3.0算法的识别准确率为89.7%。文献[23]中WiHGR算法采用双向循环网络,从前两个方向对特征序列进行学习,但该网络结构导致模型收敛速度较慢,训练耗时较长。本文所提AM-RES模型引入注意力机制,依据权重值判断各信息的重要程度,提取DFS频谱图的深层特征,相较于未引入注意力机制的ResNet-18模型,识别准确率有显著提升。同时,采用预训练的ResNet-18网络作为模型主干,其参数量相对较小,训练时间较短^[24-25]。

2) 混淆矩阵性能实验

动作识别的目的在于尽可能准确地区分各类动作,区分度越高,识别效果越佳。混淆矩阵以矩阵形式统计不同类别数据的实际类别与模型预测类别。其中,行表示样本的实际标签,列表示模型的预测类别。对角线元素代表各类手势动作的识别准确率,对角线元素数值越大,表明该手势的区分度越高,非对角线元素则表示识别错误的概率。图12给出了采用所提AM-WiGeR方法在自测数据集上进行实验所得的混淆矩阵。图12表明,AM-WiGeR对交叉手势的识别准确率最高,达到98%,对其余5种手势的识别准确率均在94%以上。

4 结束语

在手势识别领域,传统识别技术存在易受障碍物影响、易泄露隐私、部署困难等问题,因此,利用已广泛部署的Wi-Fi设备实现手势动作的非接触式感知尤为重要。针对当前基于Wi-Fi信号的手势识别研究中,表征能力强的信号特征难

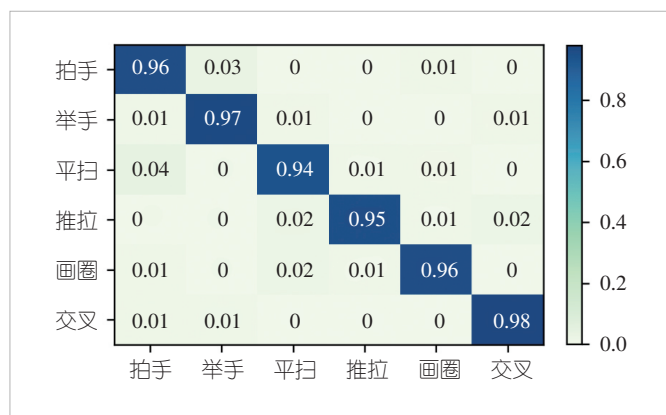


图12 6种手势识别结果的混淆矩阵

以提取、模型识别准确率提升困难等问题,本文提出了一种基于注意力机制增强残差网络模型的手势识别方法AM-WiGeR。该方法利用广泛部署的Wi-Fi设备采集CSI,从中提取DFS频谱图作为输入特征,并利用所提AM-RES模型进行分类识别。实验结果表明,相较于现有方法,所提AM-WiGeR方法有效提升了手势识别的准确率,对不同朝向手势识别的平均准确率达到96.22%。

参考文献

- [1] Zou H, Zhou Y X, Yang J F, et al. WiFi-enabled device-free gesture recognition for smart home automation [C]//Proceedings of IEEE 14th International Conference on Control and Automation (ICCA). IEEE, 2018: 476-481. DOI: 10.1109/ICCA.2018.8444331
- [2] 鲁勇, 吕绍和, 王晓东, 等. 基于WiFi信号的人体行为感知技术研究综述 [J]. 计算机学报, 2019, 42(2): 1-21. DOI: 10.11897/SP. J.1016.2019.00231
- [3] Khan R Z, Ibraheem N A. Hand gesture recognition: a literature review [J]. International journal of artificial intelligence & applications(IJAIA), 2012, 3(4): 161. DOI: 10.5121/ijaia.2012.3412
- [4] Venkatnarayan R H, Mahmood S, Shahzad M. WiFi based multi-user

- gesture recognition [J]. IEEE transactions on mobile computing, 2021, 20(3): 1242–1256. DOI: 10.1109/TMC.2019.2954891
- [5] 吴彤, 李业深, 黄镇煌, 等. 一种行人遮挡下的UWB非视距传播识别方法[J]. 物联网学报, 2023, 7(4): 63–71. DOI: 10.11959/j. issn. 2096–3750.2023.00348
- [6] Liu Z H, Li H, Gao B, et al. DQ-WiAPDoM: a DQN-based AP deployment optimization method for Wi-Fi FTM positioning [C]// Proceedings of 2nd International Conference on Smart Technologies and Systems for Next Generation Computing (ICSTSN). IEEE, 2023: 1–5. DOI: 10.1109/ICSTSN57873.2023.10151521
- [7] Wu T, Li H, Gao B, et al. NM-LUD: a NLOS mitigation method with low-cost UWB devices for indoor positioning [C]// Proceedings of 8th International Conference on Computer and Communication Systems (ICCCS). IEEE, 2023: 204–208. DOI: 10.1109/ICCCS57501.2023.10151336
- [8] Kexin Yao, Han Li, Ming Liu, et al. ARES-WiGR: An Attention-enhanced ResNet based Wi-Fi Gesture Recognition [C]// Proceedings of IEEE INFOCOM. IEEE, 2023
- [9] Gao Q H, Wang J, Ma X R, et al. CSI-based device-free wireless localization and activity recognition using radio image features [J]. IEEE transactions on vehicular technology, 2017, 66(11): 10346–10356. DOI: 10.1109/TVT.2017.2737553
- [10] Zhang J, Wu F X, Wei B, et al. Data augmentation and dense-LSTM for human activity recognition using WiFi signal [J]. IEEE Internet of Things journal, 2021, 8(6): 4628–4641. DOI: 10.1109/JIOT.2020.3026732
- [11] 周祥, 常俊, 王颖颖, 等. 基于信道状态信息的无接触人体行为识别方法[J]. 现代电子技术, 2021, 44(17): 80–85. DOI: 10.16652/j. issn. 1004–373x.2021.17.015
- [12] Ma Y S, Zhou G, Wang S Q. WiFi sensing with channel state information: a survey [J]. ACM computing surveys, 2020, 52(3): 1–36. DOI: 10.1145/3310194
- [13] Zhang J, Wei B, Wu F X, et al. Gate-ID: WiFi-based human identification irrespective of walking directions in smart home [J]. IEEE Internet of Things journal, 2021, 8(9): 7610–7624. DOI: 10.1109/JIOT.2020.3040782
- [14] Yang Z, Zhou Z M, Liu Y H. From RSSI to CSI: Indoor localization via channel response [J]. ACM computing surveys, 2013, 46(2): 1–32. DOI: 10.1145/2543581.2543592
- [15] Farid Z, Nordin R, Ismail M. Recent advances in wireless indoor localization techniques and system [J]. Journal of computer networks and communications, 2013, 2013: 185138. DOI: 10.1155/2013/185138
- [16] Wang F X, Gong W, Liu J C. On spatial diversity in WiFi-based human activity recognition: a deep learning-based approach [J]. IEEE Internet of Things journal, 2019, 6(2): 2035–2047. DOI: 10.1109/JIOT.2018.2871445
- [17] Zhu A C, Tang Z H, Wang Z X, et al. Wi-ATCN: attentional temporal convolutional network for human action prediction using WiFi channel state information [J]. IEEE journal of selected topics in signal processing, 2022, 16(4): 804–816. DOI: 10.1109/JSTSP.2022.3163858
- [18] Zhang X, Tang C P, Yin K, et al. WiFi-based cross-domain gesture recognition via modified prototypical networks [J]. IEEE Internet of Things journal, 2022, 9(11): 8584–8596. DOI: 10.1109/JIOT.2021.3114309
- [19] Feng C H, Arshad S, Zhou S W, et al. Wi-multi: a three-phase system for multiple human activity recognition with commercial WiFi devices [J]. IEEE Internet of Things journal, 2019, 6(4): 7293–7304. DOI: 10.1109/JIOT.2019.2915989
- [20] Zhu D L, Pang N, Li G, et al. WiseFi: activity localization and recognition on commodity off-the-shelf WiFi devices [C]// Proceedings of IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS). IEEE, 2016: 562–569. DOI: 10.1109/HPCC-SmartCity-DSS.2016.0085
- [21] Nirmal I, Khamis A, Hassan M, et al. Deep learning for radio-based human sensing: recent advances and future directions [J]. IEEE communications surveys & tutorials, 2021, 23(2): 995–1019. DOI: 10.1109/COMST.2021.3058333
- [22] Zhang Y, Zheng Y, Qian K, et al. WIdar3.0: zero-effort cross-domain gesture recognition with Wi-Fi [J]. IEEE transactions on pattern analysis and machine intelligence, 2022, 44(11): 8671–8688. DOI: 10.1109/TPAMI.2021.3105387
- [23] Meng W, Chen X C, Cui W, et al. WiHGR: a robust WiFi-based human gesture recognition system via sparse recovery and modified attention-based BGRU [J]. IEEE Internet of Things journal, 2022, 9(12): 10272–10282. DOI: 10.1109/JIOT.2021.3122435
- [24] Pu Q F, Gupta S, Gollakota S, et al. Whole-home gesture recognition using wireless signals [C]// Proceedings of the 19th Annual International Conference on Mobile Computing & Networking – MobiCom '13. ACM, 2013: 27. DOI: 10.1145/2500423.2500436
- [25] Qian K, Wu C S, Yang Z, et al. WIdar: decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi [C]// Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing. ACM, 2017: 1–10. DOI: 10.1145/3084041.3084067

作者简介



李业深, 北京交通大学计算机科学与技术学院在读博士研究生; 主要研究方向为人工智能、无线感知、夹持天线等; 已发表论文9篇。



姚可欣, 北京交通大学计算机科学与技术学院在读硕士研究生; 主要研究方向为人工智能、无线感知等; 已发表论文2篇。



熊轲, 北京交通大学计算机科学与技术学院教授、博士生导师, 副院长; 主要研究方向为人工智能+5G/6G网络、无线大数据分析处理、AI赋能的移动网络优化设计、绿色智慧物联网、网络大数据分析、云计算边缘计算、无线感知等; 已发表论文200余篇。

B5G 高铁网络演进方案研究



Research on Evolution of B5G High-Speed Railway Networks

孙普/Sun Pu

(中国电信股份有限公司河南分公司, 中国 郑州 450016)
(China Telecom Henan Branch, Zhengzhou 450016, China)

DOI: 10.12142/ZTETJ.202604013

网络出版地址: <https://link.cnki.net/urlid/34.1228.TN.20241212.1527.006>

网络出版日期: 2024-12-13

收稿日期: 2024-10-15

摘要: 快速增长的高铁业务规模给高铁网络容量及用户感知带来了巨大的挑战, 高铁场景下需要考虑以用户为中心进一步提升高铁用户业务感知和体验。提出了 B5G 演进的高铁无蜂窝组网方案。该方案采用无蜂窝网络架构, 解决切换、重叠干扰等带来的用户感知下降问题, 构建以用户为中心的网络, 保障了高铁用户体验。

关键词: 高铁; 5G; 无蜂窝; 用户感知

Abstract: The rapidly growing service requirements of high-speed railway users have brought huge challenges to the high-speed railway network capacity and user experience. In the high-speed railway scenario, user-centered services need to be considered to further improve the user experience of high-speed railway users. A cell-free networking solution for high-speed railways in the B5G evolution is proposed, which uses the cell-free network architecture to solve the problem of decreased user experience caused by handover and overlapping interference, build a user-centered network, and ensure high-speed railway user experience.

Keywords: high-speed train; B5G; cell-free; user experience

引用格式: 孙普. B5G 高铁网络演进方案研究 [J]. 中兴通讯技术, 2026, 32(4): 88-94. DOI: 10.12142/ZTETJ.202604013

Citation: Sun P. Research on evolution of B5G high-speed railway networks [J]. ZTE technology journal, 2026, 32(4): 88-94. DOI: 10.12142/ZTETJ.202604013

当前, 中国 4G 网络已全面覆盖城乡地区, 5G 网络覆盖正逐步从“市市通”向“县县通”延伸, 并持续向乡镇、行政村等纵深推进。5G 网络建设进入后半程, 正由规模建设转向精准规划, 聚焦重点场景的网络覆盖及“补盲”“补热”工作。2024 年 1 月, 工业和信息化部、国家发展改革委等 11 部门联合印发《关于开展“信号升格”专项行动的通知》(简称《专项行动》), 提出重点场景网络建设要求, 加快推进移动网络深度覆盖, 提升网络质量, 优化用户感知。运营商已将重点场景覆盖达标率纳入重点关注指标。《专项行动》提出, 到 2024 年底, 超过 8 万个重点场所实现移动网络深度覆盖, 25 000 km 铁路、350 000 km 公路和 150 条地铁线路实现移动网络连续覆盖。其中, 高铁 5G 网络覆盖已成为当前无线网络覆盖的关注重点。

2023 年, 国家铁路完成旅客发送量 36.8 亿人次, 高峰日发送旅客突破 2 000 万人次, 全年及高峰日发送量均创历史新高。高铁业务规模的快速增长, 给高铁网络容量及用户感知带来巨大挑战。本文针对 5G 网络建设后期高铁网络面临的挑战开展研究, 并提出一种高铁无蜂窝组网方案, 旨在

突破小区边界约束, 尽可能保障高铁用户的稳态业务感知, 提升用户体验。

1 高铁 5G 网络面临的挑战

网络建设初期, 高铁网络普遍面临三大挑战: 车体穿透损耗大、多普勒频偏、高速移动带来的频繁切换^[1]。对此, 可通过多样化的设备选型、不同频段组合及软件增强功能等方式进行优化。随着高铁网络建设不断完善, 用户需求持续提升, 现阶段高铁 5G 网络面临新的挑战: 一方面, 高价值用户数量增加对网络性能提出了更高要求; 另一方面, 视频类业务时长占比提升也对用户感知保障提出了新的诉求。

1.1 高铁场景高价值用户不断增多

高铁已成为中国重要的铁路运输方式之一。截至 2023 年底, 中国铁路营业里程已达 154 000 km, 其中高铁里程超过三分之一, 达到 37 000 km。铁路网络的不断扩展, 为国家交通运输体系的完善提供了有力支撑。伴随经济快速发展, 高铁日益成为公务、商务等高价值人群的出行首选。铁路移动通信网络的持续完善, 提供了连续的网络服务, 为出

行途中处理工作、保持联络提供了重要保障。根据中国某运营商的相关统计数据（如表1所示），高铁用户的每用户月均流量（DOU）约为全网用户的2倍，高铁场景的感知与体验保障已成为运营商的关注重点^[2]。

由上述统计可以看出，高铁用户多为高DOU、高每用户每月平均收入（ARPU）值用户，用户价值高，品牌影响大。因此，在高铁场景下需持续提升用户体验，保障用户感知。

1.2 高铁5G网络用户体验仍有提升空间

根据中国某省份运营商高铁数据业务类型分析（见表2）可知，视频类业务的用户时长占比最高。其中，视频直播类业务时长占比达36.85%，视频点播类业务时长占比为10.75%。整体来看，视频类业务在高铁用户使用中的合计占比接近50%。

随着移动互联网的快速发展，高铁场景下的用户业务类型日益丰富。其中，视频通话、视频直播类业务已成为移动通信的主流应用。面向未来，多样化的业务类型和不断增长的业务负荷，将对网络性能及用户感知提出更高要求。

1.3 大容量和稳态感知对高铁组网方案提出新要求

在传统的4G和5G高铁网络覆盖中，高铁沿线的射频拉远单元（RRU）仍按照独立小区或合并小区的方式进行组网，如图1所示。网络覆盖仍以RRU单节点独立处理为主，当用户终端在高铁沿线移动时，仍会发生频繁的小区切换。

随着容量需求的不断增长，采用U6G与高频针对高铁场景进行覆盖的研究日益增多。3GPP R18进一步讨论了高频在高铁场景中的应用。采用U6G与高频进行高铁覆盖时，站间距可能进一步缩小，多站点间的协作将更具意义。此外，面向未来的高铁网络还需考虑进一步减少小区间切换，可引入无小区边界的无蜂窝网络架构，从而为用户提供更加稳定的感知体验。

1.4 高铁场景无线终端频繁切换导致感知下降明显

高铁无线覆盖网络具有典型的线性组网特征，高铁列车沿固定线路运行，沿途5G站点分布在高铁线路两侧。按列车运行速度300 km/h、单个物理站点覆盖约300 m估算，列车通过一个站点的时间约为4 s，这意味着高铁上的终端大约每4 s发生一次小区间切换。小区间切

表1 用户DOU、ARPU对比

指标	京雄高铁	京张高铁	全网
DOU/GB	22	20	11
ARPU/元	44	37	28

ARPU：每用户每月平均收入 DOU：每用户月均流量

表2 高铁用户主要业务分类及使用时长占比

业务分类	高铁用户时长占比/%
视频直播(抖音、快手)	36.85
即时通信(微信、QQ)	34.10
网页浏览(网页、今日头条、微博)	10.36
视频点播(腾讯视频、哔哩哔哩)	10.75
购物(淘宝、京东)	7.95

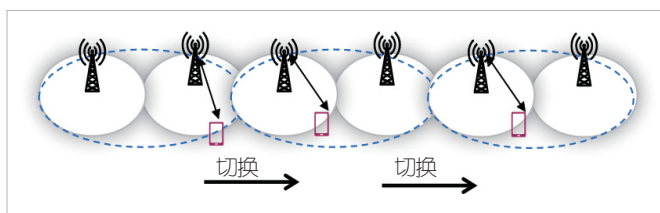


图1 传统高铁组网方案

换会造成数十毫秒的业务短时中断，而切换前后的业务准备阶段，更会导致业务速率明显下降。

高铁沿线终端的频繁切换会显著影响用户体验。在高铁场景下，对比无蜂窝（切换次数减少85%）与较多切换两种场景下的下行业务速率。结果表明，较多切换时终端下行业务平均速率下降约25%，如图2所示。由此可见，通过无蜂窝网络架构减少小区切换次数，能够有效改善业务体验。

2 无蜂窝技术概述

当前移动通信组网普遍采用蜂窝组网方案，该方式可按

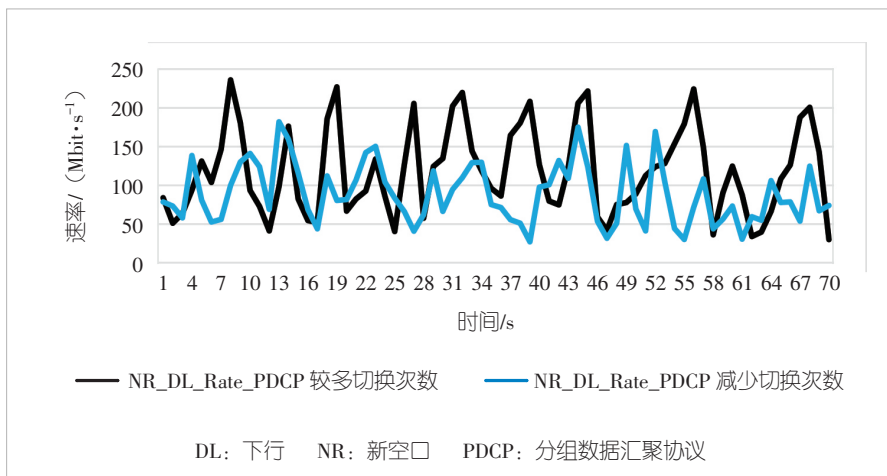


图2 无蜂窝减少切换次数和较多切换次数的下行业务速率对比

需实现局部区域覆盖，具备较强的可扩展性。蜂窝组网具有明确的小区边界划分，用户在小区间移动时需要切换，从而造成业务的短时中断。在同频组网场景下，小区边界的存在还会带来重叠干扰，进一步导致边缘用户性能下降。

6G 对更高频段、更高速率及更高可靠性提出了要求，在多天技术演进方面也明确了超大规模多输入多输出 (MIMO) 的发展方向。传统蜂窝组网方案难以有效保障小区边缘的可靠性，在构建以用户为中心的网络方面亦存在诸多局限。无蜂窝技术^[3-5]已被列为面向 6G 的关键技术之一，该技术不仅能够真正实现以用户为中心的网络架构，为边缘用户提供更可靠的性能保障，还可通过分布式天线面构建超大规模 MIMO 系统。

无蜂窝技术的核心在于打破传统小区边界，为移动用户提供稳定的体验，同时还可通过多收发节点 (TRP) 间的协作构建大规模 MIMO 系统^[6-9]。

无蜂窝网络的典型组网架构由一个中央处理单元 (CPU) 连接大量无线接入点 (AP) 构成，如图 3 所示。其中，CPU 负责所有节点的信道状态评估、资源分配及数据收发决策，AP 则主要承担射频前端处理。AP 通常配置较少的天线数，多个 AP 可联合构成大规模天线系统为同一用户提供服务，也可通过空分方式为不同用户分别提供服务。该架构真正实现了以用户为中心的网络形态^[10]，突破了传统蜂窝网络以小区为边界的局限性，有助于缓解切换和重叠干扰等因素导致的用户感知下降问题。

当前无蜂窝技术的研究多聚焦于宏覆盖典型场景，如城区商圈、大型体育场馆等，而对高铁等特殊场景的关注相对较少。然而，从打破小区边界的初衷出发，高铁场景对无蜂

窝技术的需求实际上更为迫切。

3 高铁无蜂窝组网方案研究

本文将介绍面向高铁场景的无蜂窝组网方案。该方案有别于传统蜂窝网络架构，能够实现以用户为中心的网络形态，突破传统独立小区的边界限制，通过无蜂窝化组网有效缓解切换、重叠干扰等因素导致的用户感知下降问题，从而显著提升高铁用户的业务体验。

3.1 高铁无蜂窝组网方案

高铁场景中的信号发射单元通常采用铁路两侧交错分布的部署方式，以保证车厢两侧用户的覆盖均衡，避免单侧弱覆盖问题。随着 5G-A 与 6G 的持续演进，U6G 与高频段在高铁场景中的应用愈发受到关注。一方面，收发节点的部署间距可能进一步缩小；另一方面，如何突破传统小区边界，联合多个 AP 为用户提供连续服务，使得无蜂窝组网在高铁场景中的研究价值日益凸显。

无蜂窝网络的网络架构可以很好地解决高铁在发射站点间频繁切换的问题。高铁无蜂窝网络可以采用 Radio Stripes 的模式，其中的 CPU 作为集中点，连接 L 个 AP 条带，每个条带上的 AP 采用级联的方式，如图 4 所示。

L 个 AP 按照一定顺序在条带上依次部署，形成一种空间上离散串行的 AP 分布结构。各 AP 通过链式 (条带式) 结构完成顺序连接，并在同一管道 (条带) 内统一实现前向回传与供电^[11]。场景内的用户可与一个或多个 AP 进行通信，从而实现无蜂窝式的通信方式。

需要注意的是，现有单条带模式存在一定缺陷：当条带上某处发生断裂导致通信中断时，其后端所有 AP 均将停止工作，带来较大的不稳定性，系统鲁棒性较低。对此，一种直接的改进方式是采用更多条带 (可选用更小规模)，用户可与一个或多个条带上的 AP 进行通信，从而获得更高的鲁棒性与灵活性。另一种方案是采用环形组网模式，将条带两

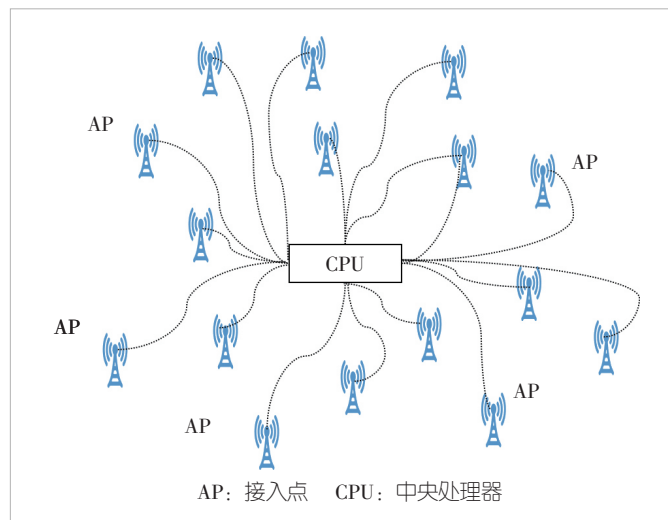


图3 无蜂窝典型组网示意图

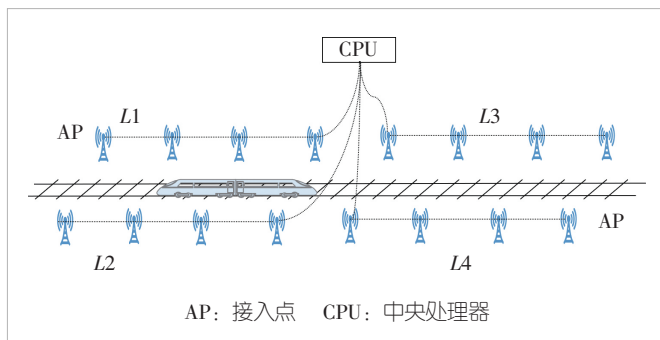


图4 高铁无蜂窝条带组网

端的AP均与CPU对接。

此外,通过对多个条带进行地理上的延伸部署,可进一步扩展系统通信覆盖范围,提升用户服务质量。这种多条带联合的工作模式可称为混合模式,即星式与星式、星式与条带式、条带式与条带式的组合形式(其中星式可视作条带式的特例,此时条带上仅包含一个AP)^[12],如图5所示。

3.2 系统模型与传输方案

这里考虑Radio Stripes混合模式的系统模型。该系统共服务 K 个单天线用户。假设系统场景中有 N 个条带(对于图3,可分别表示为 $N=2,3,4$),其集合表示为 $N=\{1,\dots,N\}$,而对于其中的第 n 个条带,其包含 L_n 个AP,由此形成AP个数集合 $L=\{L_1,\dots,L_n,\dots,L_N\}$,并得到总AP数 $L=\sum_{n=1}^N L_n$ 。每个AP有 M 根天线,于是,总天线数为 ML 。需要注意的是,当 $L_n \geq 2$,其结构表示为条带;当 $L_n = 1$,其条带退化为星式结构,即可呈现为混合的结构模式。

对于第 n 个条带,其中的第 l_n 个AP第 m 根天线与第 k 个用户的无线信道(信道增益)可以表示为 $h_{n,l_n,m,k} \in \mathbb{C}$,则第 l_n 个AP与第 k 个用户的无线信道向量可以表示为 $\mathbf{h}_{n,l_n,k} = [h_{n,l_n,1,k}, \dots, h_{n,l_n,M,k}]^H \in \mathbb{C}^{M \times 1}$ 。于是,对于来自 K 个用户的上行信号 $\{\mathbf{x}_k\}_{k=1}^K$,其中 $\mathbf{x}_k = \sqrt{p_k} \mathbf{s}_k \in \mathbb{C}^{1 \times \tau_u}$, $E\{\mathbf{x}_k \mathbf{x}_k^H / \tau_u\} = p_k$, p_k 为用户的上行发送功率, τ_u 为上行传输符号数,则第 n 个条带上的第 l_n 个AP的联合接收信号 $\mathbf{Y}_{n,l_n} \in \mathbb{C}^{M \times \tau_u}$ 表示为求和的形式:

$$\mathbf{Y}_{n,l_n} = \sum_{k=1}^K \mathbf{h}_{n,l_n,k} \mathbf{x}_k + \bar{\mathbf{N}}_{n,l_n} \quad (1),$$

其中, $\bar{\mathbf{N}}_{n,l_n} \in \mathbb{C}^{M \times \tau_u}$ 表示为接收噪声,其各元素建模为独立同分布的复高斯变量 $CN(0, \sigma_n^2)$ 。

3.2.1 分布式处理

各AP对接收信号进行处理,并通过条带反馈至CPU, CPU对所有接收到的处理信号进行最终合并,从而完成上行数据的整体处理流程。具体而言,令第 l_n 个AP对于第 k 个用户的上行检测向量为 $\mathbf{w}_{n,l_n,k}^U \in \mathbb{C}^{1 \times M}$,则最终在CPU上对于第 k 个用户的合并符号表示为求和的形式:

$$\hat{\mathbf{x}}_k = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \mathbf{w}_{n,l_n,k}^U \mathbf{Y}_{n,l_n} \quad (2).$$

现将上述内容改写成组合联立的形式(对于所有用户)。

令 $\mathbf{X} = [\mathbf{x}_1^H, \dots, \mathbf{x}_K^H]^H \in \mathbb{C}^{K \times \tau_u}$, $\mathbf{H}_{n,l_n} = [\mathbf{h}_{n,l_n,1}, \dots, \mathbf{h}_{n,l_n,K}] \in \mathbb{C}^{M \times K}$,则存在:

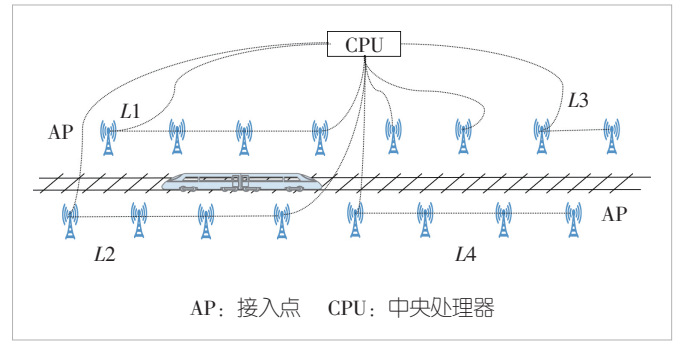


图5 高铁无线蜂窝混合模式组网

$$\mathbf{Y}_{n,l_n} = \mathbf{H}_{n,l_n} \mathbf{X} + \bar{\mathbf{N}}_{n,l_n} \quad (3).$$

令 $\mathbf{W}_{n,l_n}^U = [\mathbf{w}_{n,l_n,1}^U, \dots, \mathbf{w}_{n,l_n,K}^U]^H \in \mathbb{C}^{K \times M}$,则组合后的合并符号 $\hat{\mathbf{X}} = [\hat{\mathbf{x}}_1^H, \dots, \hat{\mathbf{x}}_K^H]^H \in \mathbb{C}^{K \times \tau_u}$ 表示为:

$$\hat{\mathbf{X}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n} \quad (4).$$

当系统采用MRC检测,则各AP采用本地信道状态信息(CSI)对接收信号进行处理,并通过条带进行叠加反馈, CPU对所有接收到的处理信号进行最终合并操作,从而完成整体的上行数据处理。具体地,检测向量(矩阵)可以表示为:

$$\mathbf{w}_{n,l_n,k}^U = \sqrt{p_k} \mathbf{h}_{n,l_n,k}^H, \mathbf{W}_{n,l_n}^U = [\sqrt{p_1} \mathbf{h}_{n,l_n,1}^H, \dots, \sqrt{p_K} \mathbf{h}_{n,l_n,K}^H]^H = \bar{\mathbf{H}}_{n,l_n}^H \quad (5).$$

此时的合并符号即为 $\hat{\mathbf{X}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \bar{\mathbf{H}}_{n,l_n}^H \mathbf{Y}_{n,l_n}$ 。由于每个条带都能对其AP所反馈的信号进行叠加操作,即实现 $[\hat{\mathbf{x}}_{1(n,l_n)}^H, \dots, \hat{\mathbf{x}}_{K(n,l_n)}^H]^H = \sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n} \in \mathbb{C}^{K \times \tau_u}$,则每个条带的反馈量(回传信息维度)为 $K\tau_u$,进而系统的整体反馈量为 $NK\tau_u$ 。当AP获得非理想的信道信息时,相应的向量(矩阵)可修改为 $\hat{\mathbf{h}}_{n,l_n,k}$ ($\hat{\mathbf{H}}_{n,l_n}$)。MRC检测的流程如下:

- 1) 条带上的各AP接收到上行信号 $\mathbf{Y}_{n,l_n}$;
- 2) 各AP计算 $\mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$ ($\bar{\mathbf{H}}_{n,l_n}^H \mathbf{Y}_{n,l_n}$);
- 3) 各条带对处理信号进行(顺序)叠加: $\sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$,并反馈至CPU;
- 4) CPU对各条带反馈的信号进行叠加: $\hat{\mathbf{X}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$,以完成上行数据处理。

3.2.2 协作处理

协作处理可简单描述为 AP 与 CPU 间的协作式处理。对于上行链路, AP 首先进行信号处理(一阶段处理),将处理后的信息通过条带反馈给 CPU,再由 CPU 完成后续处理(二阶段处理),从而在减轻前传负载的同时获得较高的性能增益。具体而言, AP 需要获取信道信息,并将相应结果用于空间合并下的本地数据估计(检测);随后,估计后的数据与信道相关信息通过条带从 AP 传送至 CPU, CPU 据此进行进一步的空间合并处理,以完成上行数据整体处理流程。下面给出方案描述:

1) 系统采用 Q-MMSE 检测^[13], 检测流程如图 6 所示。

该检测方案具体包括 5 个步骤:

(1) 条带上的各 AP 接收到上行信号 Y_{n,l_n} ;

(2) 各 AP 计算 $\mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$ ($\bar{\mathbf{H}}_{n,l_n}^H \mathbf{Y}_{n,l_n}$) (一类前传信息: 可理解为 MRC 结果, 其隐含信道相关信息);

(3) 各条带对处理信号进行(顺序)叠加: $\sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$,

并反馈至 CPU;

(4) CPU 对各条带反馈的信号进行叠加: $\hat{\mathbf{X}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$;

(5) CPU 对叠加信号进行处理, 以完成上行数据处理。

可以看到, 系统进行了两次信号叠加操作: 条带内叠加和条带间叠加(可理解为条带间的加权系数恒为 1)。

CPU 对叠加信号进行处理, 得到信道相关信息并进行最终合并, 其流程如下:

(1) 对叠加信号进行相关操作: $\hat{\mathbf{C}} = \hat{\mathbf{X}} \hat{\mathbf{X}}^H / \tau_u$;

(2) SVD: $\hat{\mathbf{C}} = \mathbf{U} \mathbf{\Omega} \mathbf{U}^H$, 其中的奇异值 $\{\omega_l\}_{l=1}^d$, $\mathbf{\Omega} = \omega_1 \mathbf{I}_{n_1} \oplus \cdots \oplus \omega_d \mathbf{I}_{n_d}$;

(3) 变换操作: $\hat{\mathbf{X}}_{Q-MMSE} = \mathbf{U} (\mu_1 \mathbf{I}_{n_1} \oplus \cdots \oplus \mu_d \mathbf{I}_{n_d}) \mathbf{U}^H \hat{\mathbf{X}}$, $\mu_l =$

$$\frac{2}{\sqrt{\sigma_u^4 + 4\omega_l + \sigma_u^2}}。$$

为了表述清晰, 现将上述内容进一步改写成组合的形式(对于所有条带)。令 $\mathbf{Y}_n = [\mathbf{Y}_{n,1}^H, \cdots, \mathbf{Y}_{n,L_n}^H]^H \in C^{ML_n \times \tau_u}$, $\bar{\mathbf{H}}_n = [\bar{\mathbf{H}}_{n,1}^H, \cdots, \bar{\mathbf{H}}_{n,L_n}^H]^H \in C^{ML_n \times K}$, 进一步地, $\mathbf{Y} = [\mathbf{Y}_1^H, \cdots, \mathbf{Y}_N^H]^H \in C^{ML \times \tau_u}$, $\bar{\mathbf{H}} = [\bar{\mathbf{H}}_1^H, \cdots, \bar{\mathbf{H}}_N^H]^H \in C^{ML \times K}$, 那么 $\hat{\mathbf{X}} = \bar{\mathbf{H}}^H \mathbf{Y}$, $\hat{\mathbf{X}}_{Q-MMSE} = \mathbf{U} (\mu_1 \mathbf{I}_{n_1} \oplus \cdots \oplus \mu_d \mathbf{I}_{n_d}) \mathbf{U}^H \bar{\mathbf{H}}^H \mathbf{Y}$ 。

对于此方案, 每个条带的反馈量为 $K\tau_u$ ($\sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$ 的维度为 $K \times \tau_u$), 进而系统的整体反馈量为 $NK\tau_u$, 这与 MRC 方案的结果相同。

2) 系统采用 C-RZF 检测^[14], 则流程如图 7 所示。

该检测方案具体也包括 5 个步骤:

(1) 条带上的各 AP 接收到上行信号 Y_{n,l_n} ;

(2) 各 AP 计算 $\mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$ ($\bar{\mathbf{H}}_{n,l_n}^H \mathbf{Y}_{n,l_n}$) 与 $\bar{\mathbf{H}}_{n,l_n}^H \bar{\mathbf{H}}_{n,l_n}$ (两类前传信息);

(3) 各条带对处理信号进行(顺序)叠加: $\sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$, 对信道相关矩阵进行叠加: $\sum_{l_n=1}^{L_n} \bar{\mathbf{H}}_{n,l_n}^H \bar{\mathbf{H}}_{n,l_n}$, 并将两类信息反馈至 CPU;

(4) CPU 对各条带反馈的信息进行叠加: $\hat{\mathbf{X}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \mathbf{W}_{n,l_n}^U \mathbf{Y}_{n,l_n}$, $\bar{\mathbf{H}}^H \bar{\mathbf{H}} = \sum_{n=1}^N \sum_{l_n=1}^{L_n} \bar{\mathbf{H}}_{n,l_n}^H \bar{\mathbf{H}}_{n,l_n}$;

(5) CPU 对叠加信息进行处理: $\hat{\mathbf{X}}_{C-RZF} = (\bar{\mathbf{H}}^H \bar{\mathbf{H}} + \mathbf{A} \mathbf{I}_K)^{-1} \hat{\mathbf{X}}$, 其中有 $\mathbf{A} = \sigma_u^2 \mathbf{I}_K$ 或 $\mathbf{A} = \text{diag}(a_1, \cdots, a_K) + \sigma_u^2 \mathbf{I}_K$, 以完成上行数据处理。

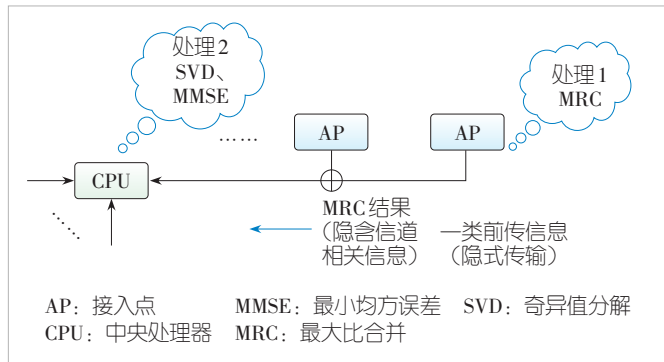


图6 Q-MMSE检测流程

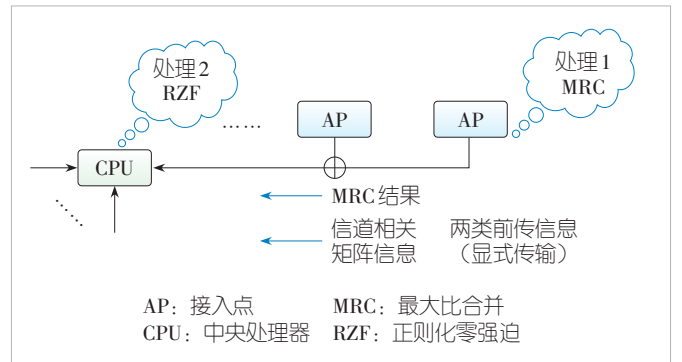


图7 C-RZF检测流程

在上述步骤中, $a_k = \frac{1}{ML} \text{tr} \left(\sum_{i \neq k} \left(\frac{\tau_u}{\sigma_u^2} \mathbf{I}_{ML} + \frac{\mathbf{R}_i^{-1}}{p_i} \right)^{-1} \right)$, $\mathbf{R}_i = \mathbf{R}_{i,1,1} \oplus \cdots \oplus \mathbf{R}_{i,n,l_n} \oplus \cdots \oplus \mathbf{R}_{i,N,L_N}$, $\mathbf{R}_{i,n,l_n} \in C^{M \times M}$ 表示第 n 个条带中的第 l_n 个 AP 与第 i 个用户的空间相关矩阵 (CPU 已知的信道统计信息), 并可假设 $\mathbf{h}_{n,l_n,i} \sim \text{CN}(\mathbf{0}, \mathbf{R}_{i,n,l_n})$ 。对于此方案, 每个条带需要反馈两类信息, 其反馈量为 $K\tau_u + \frac{K(K+1)}{2}$ (注意, 信道相关矩阵的维度为 $K \times K$, 但只需反馈其上三角元素), 进而系统的整体反馈量为 $NK\tau_u + \frac{NK(K+1)}{2}$ 。

3.3 高铁快速移动对信道估计的影响

在高铁场景下, 由于列车移动速度较高, 无线信道随时间快速衰落, 多普勒频移较为显著, 对数据解调性能造成严重影响。高速移动引起的信道快速变化主要体现在两个方面: 一是符号间产生相位偏移; 二是破坏子载波间的正交性, 引入载波间干扰 (ICI)。

5G 根据移动速度适配附加导频数量, 通过多个导频联合进行频偏跟踪与信道估计, 能够更有效地跟踪高速场景下的多普勒频偏变化, 并获取信道的时变特性, 从而提升解调性能。

在 CDL-C 信道模型下, 图 8 仿真表明, 在间隔 0.4 ms 的两个符号上, 信道时变特性使得两个时刻的误差矢量幅度 (EVM) 差异与用户设备 (UE) 移动速度强相关。具体而言, UE 以 300 km/h 移动时, 两个符号间的信道差异显著大于 120 km/h 时的情况。

在无蜂窝组网模式下, 各节点独立进行信道估计与补偿

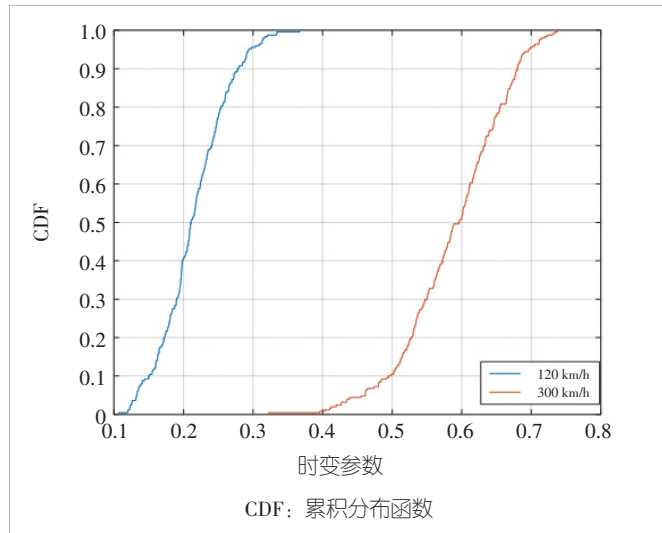


图8 高铁时变特性在不同速度下的差异

处理, 即各节点在接收数据合并之前, 已完成信道插值估计和频偏补偿。该处理方式不会影响多节点数据合并增益的获取。

3.4 高铁下行数据分布式处理的分析

考虑到高铁场景中高速运动带来的频偏影响, 若各 AP 节点间的频偏与相位未能有效对齐, 则在联合处理时会导致系统性能下降。图 9 仿真对比了单个 8T (single_8T) 节点与两个分布式 8T 节点 (cell_free_double_8T) 在无蜂窝组网下为用户发送数据的性能。结果表明, 若未经频偏与相位对齐处理, 采用无蜂窝节点联合发送的方式反而会造成性能损失。

考虑各节点间的频偏处理, 一种可行的方式是在各 AP 处独立进行频偏补偿, 统一向零频偏靠拢, 随后将处理后的信息汇聚至 CPU 进行后续处理。同时, 不同 AP 的权值也需进行补偿对齐, 经过权值拉齐后, 信号可实现有效增强。

$$\begin{aligned} r(f) &= \left(\sum_i h_i(f) w_i(f) s(f) \right) + n(f) = \\ &= \left(\sum_i A_i(f) e^{j\varphi(f)} s(f) \right) + n(f) = \\ &= \left(\sum_i A_i(f) \right) e^{j\varphi(f)} s(f) + n(f) \end{aligned} \quad (6)$$

$$\text{SNR}(f) = \frac{\left| \sum_i A_i(f) \right|^2}{\sigma_n^2} \quad (7)$$

其中, $h_i(f)$ 代表第 i 个 AP 的频域信道, $w_i(f)$ 代表第 i 个 AP 的频域发送权值, $A_i(f)$ 代表第 i 个 AP 的等效信道 $h_i(f)w_i(f)$ 的幅度, $\varphi(f)$ 代表第 i 个 AP 的等效信道

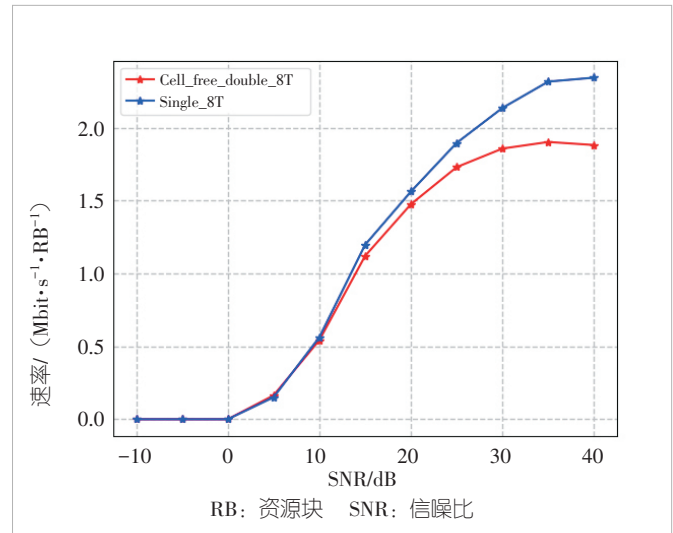


图9 无蜂窝和单节点发送性能对比

$h_i(f)w_i(f)$ 的相位。

不同 AP 的空口信道 $h_i(f) = \alpha_i(f)e^{j\psi_i(f)}$, 其权值 $w_i(f) = \rho_i(f)e^{j\delta_i(f)}$ 。在使用权值补偿后, $\psi_i(f) + \delta_i(f) = \psi_j(f) + \delta_j(f) = \varphi(f)$, 即不同 AP 在使用权值补偿后的等效信道相位均拉齐至 $\varphi(f)$ 。

4 结束语

中国持续推进“八纵八横”高速铁路网及重点客运普速铁路的移动网络信号连续覆盖, 以支撑旅客通话及综合信息服务需求。本文提出的高铁无蜂窝组网方案, 采用无蜂窝网络架构, 实现了从“以网络为中心”向“以用户为中心”的转变。未来, 随着业务的快速发展和高铁无线网络覆盖的持续升级, 该方案能够有效保障用户体验, 助力运营商构建高质量无线网络覆盖体系。

参考文献

- [1] 王佐, 李晓萌, 董瑶, 等. 高铁 5G NR2.1 覆盖方案和特性参数验证 [J]. 通信与信息技术, 2024, (2): 74–78+82
- [2] 李新, 王强. 高铁 5G 覆盖网络建设挑战及应对 [J]. 电信快报, 2024 (1): 1–4
- [3] 胡彦丰, 王东明, 梁楚龙, 等. 无蜂窝大规模 MIMO 中的大规模随机接入 [J]. 中兴通讯技术, 2024, 30(1): 26–32. DOI: 10.12142/ZTETJ.202401006
- [4] 吴越, 王东明, 尤肖虎. 面向 6G 的无蜂窝无线接入网技术 [J]. 中兴通讯技术, 2024, 30(4): 14–25. DOI: 10.12142/ZTETJ.202404004
- [5] 章嘉懿. 无蜂窝大规模 MIMO 系统研究进展与发展趋势 [J]. 重庆邮电大学学报(自然科学版), 2019, 31(3): 285–292
- [6] Ammar H A, Adve R, Shahbazpanahi S, et al. User-centric cell-free massive MIMO networks: a survey of opportunities, challenges and solutions [J]. IEEE communications surveys & tutorials, 2022, 24(1): 611–652. DOI: 10.1109/comst.2021.3135119
- [7] Wang R, Shen M, He Y, et al. Performance of cell-free massive MIMO with joint user clustering and access point selection [J]. IEEE access, 2021, 9: 40860–40870. DOI: 10.1109/

ACCESS.2021.3056051

- [8] Ngo H Q, Ashikhmin A, Yang H, et al. Cell-free massive MIMO versus small cells [J]. IEEE transactions on wireless communications, 2017, 16(3): 1834–1850. DOI: 10.1109/TWC.2017.2655515
- [9] 赵亚军, 郁光辉, 徐汉青. 6G 移动通信网络: 愿景、挑战与关键技术 [J]. 中国科学: 信息科学, 2019, 49(8): 963–987
- [10] Kassam J, Castanheira D, Silva A, et al. A review on cell-free massive MIMO systems [J]. Electronics, 2023, 12(4): 1001. DOI: 10.3390/electronics12041001
- [11] Shaik Z H, Bjornson E, Larsson E G. Cell-free massive MIMO with radio stripes and sequential uplink processing [C]// Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2020: 1–6. DOI: 10.1109/iccworkshops49005.2020.9145164
- [12] Björnson E, Sanguinetti L. Making cell-free massive MIMO competitive with MMSE processing and centralized implementation [J]. IEEE transactions on wireless communications, 2020, 19(1): 77–90. DOI: 10.1109/TWC.2019.2941478
- [13] Yuan Z F, Ma Y H, Yu G H. LMMSE processing for cell-free massive MIMO with radio stripes and MRC fronthaul [C]//2022 IEEE Wireless Communications and Networking Conference (WCNC). IEEE, 2022: 1336–1340. DOI: 10.1109/WCNC51071.2022.9771691
- [14] Ma Y H, Yuan Z F, Yu G H, et al. Cooperative scheme for cell-free massive MIMO with radio stripes [C]//Proceedings of IEEE International Conference on Communications Workshops (ICC Workshops). IEEE, 2021: 1–6. DOI: 10.1109/iccworkshops50388.2021.9473493

作者简介



孙普, 中国电信股份有限公司河南分公司副总经理; 研究方向为无线通信、新型基础设施规划及企业数字化转型。

中兴通讯技术杂志社

促进产学研合作青年专家委员会

主 任 陈 为 (北京交通大学)

副主任 秦晓琦 (北京邮电大学) 卢 丹 (中兴通讯股份有限公司)

委 员

曹 进	西安电子科技大学	史颖欢	南京大学
陈 力	中国科学技术大学	唐万恺	东南大学
陈 为	北京交通大学	王景璟	北京航空航天大学
陈琪美	武汉大学	王兴刚	华中科技大学
陈舒怡	哈尔滨工业大学	王勇强	天津大学
陈思衡	上海交通大学	温森文	华南理工大学
高 镇	北京理工大学	吴泳澎	上海交通大学
官 科	北京交通大学	武庆庆	上海交通大学
韩 充	上海交通大学	夏文超	南京邮电大学
韩凯峰	中国信息通信研究院	向路平	南京大学
何 姿	南京理工大学	徐梦炜	北京邮电大学
侯天为	北京邮电大学	徐天衡	中国科学院上海高等研究院
胡 杰	电子科技大学	杨川川	北京大学
黄 晨	紫金山实验室	叶迎晖	西安邮电大学
霍佳皓	北京科技大学	尹海帆	华中科技大学
李 昂	西安交通大学	游昌盛	南方科技大学
李 礼	中国科学技术大学	于季弘	北京理工大学
刘 凡	东南大学	张 娇	北京邮电大学
刘俊宇	西安电子科技大学	张宇超	北京邮电大学
卢 丹	中兴通讯股份有限公司	章嘉懿	北京交通大学
陆游游	清华大学	赵毅哲	电子科技大学
梅渭东	电子科技大学	赵昱达	浙江大学
宁兆龙	重庆邮电大学	赵中原	北京邮电大学
潘存华	东南大学	周 伊	西南交通大学
祁 亮	上海交通大学	朱秉诚	东南大学
秦晓琦	北京邮电大学	朱光旭	深圳市大数据研究院
秦志金	清华大学	朱政宇	郑州大学
史 瑶	哈尔滨工业大学 (深圳)		

刊物相关信息



投稿须知



投稿平台



过刊下载



论文索引与
引用指南

办刊宗旨：

以人为本，荟萃通信技术领域精英
迎接挑战，把握世界通信技术动态
立即行动，求解通信发展疑难课题
励精图治，促进民族信息产业崛起

产业顾问：

段向阳、高 音、胡留军、华新海、刘新阳、
史伟强、屠要峰、王会涛、熊先奎、许 进、
闫新成、赵亚军、朱晓光

双月刊 1995 年创刊
第 32 卷 总第 190 期
2026 年 8 月 第 4 期

主管：安徽出版集团有限责任公司
主办：时代出版传媒股份有限公司
深圳航天广宇工业有限公司
出版：安徽科学技术出版社
编辑、发行：中兴通讯技术杂志社

总编辑：王喜瑜
主编：陶善勇
执行主编：吴玲、黄新明
副主编：卢丹
编辑部主任：王萍萍
责任编辑：王镇、徐烨
编辑：杨广西、朱莉、任溪溪
设计排版：徐莹
发行：王萍萍
编务：王坤

《中兴通讯技术》编辑部
地址：合肥市金寨路 329 号凯旋大厦 1201 室
邮编：230061
网址：tech.zte.com.cn
投稿平台：tech.zte.com.cn/submission
电子信箱：magazine@zte.com.cn
电话：(0551) 65533356

发行方式：自办发行
印刷：安徽添锦印刷科技有限公司
出版日期：2026 年 8 月 31 日
中国标准连续出版物号：ISSN 1009-6868
CN 34-1228/TN
定价：每册 20.00 元