

许宏鑫

191-2921-2198 | xuhx56@mail2.sysu.edu.cn | [xhx1022.github.io](https://github.com/xhx1022)

教育经历

中山大学 | 计算机科学与技术, 计算机学院 | 学术型硕士研究生 2024.09—2027.06 (预计)
导师张献伟, 主要研究方向为 **MLSys**, 在大模型推理系统方面有一定的研究和工程经验。

华南理工大学 | 计算机科学与技术, 计算机学院 | 工学学士 2020.09—2024.06
GPA: 3.84/4.0(专业前 10%), 获校级学业奖学金、企业奖学金多次, 并保研至中山大学。

技术能力

- 编程语言: Python, C++, Shell
- 工具: Linux, Git, Docker, Nsight System
- 技术栈: 熟悉 Pytorch, SGLang, vLLM, slime
- 研究兴趣: 大模型推理调度优化、KV Cache 压缩、Agent 应用、RL 高效采样和 rollout

科研论文

- Hongxin Xu, Tianyu Guo, et al. *DynaPipe: Dynamic Layer Redistribution for Efficient Serving of LLMs with Pipeline Parallelism*. NeurIPS, 2025.
- Zejia Lin, Hongxin Xu, et al. *Boosting LLM Serving through Spatial-Temporal GPU Resource Sharing*. Preprint, arXiv, 2025.

实习经历

深圳商汤研究院 | 科研实习生 2025.08—至今

- 参与开发LightLLM, 主要学习多模态部署和请求调度优化工作。
- 调研并学习 slime 强化学习框架, 探索基于历史信息的高效采样策略以提升 rollout 的有效性, 并研究 partial rollout 在精度与速度间的平衡机制。

项目经历

基于动态层重分配的 LLM 高效流水线并行服务系统 | *NeurIPS* 一作 2025.03—2025.05

该项目聚焦于大模型推理中流水线并行的 **inter-stage** 不平衡问题, 系统通过实时预测计算与采样延迟, 动态调整各阶段的层分配, 有效缓解因尾部阶段采样开销造成的流水线气泡与阶段失衡, 显著提升硬件利用率。在多种负载下, 端到端推理延迟降低了 10% 至 49%, 优于现有主流推理框架。

- 延迟预测器**: 离线条件下通过 profile 数据进行建模, 实时根据负载预测前向计算和采样开销, 用于调度器决策;
- 气泡感知调度器**: 根据 stage 执行时间差异自适应调整层分配, 打破传统平均分配策略, 缓解流水线气泡现象;
- 迁移机制**: 支持推理过程中的非阻塞重分配, 异步迁移 KV Cache, 保持流水线运行连续性。

基于 GPU 空间-时间协同编排的高效大模型推理系统 | 在投论文二作 2024.11—2025.03

针对大模型推理中 Prefill 与 Decode 阶段在 GPU 上资源使用不均的问题, 设计并实现空间-时间协同编排机制, 提升 GPU 利用率与系统整体吞吐。

- 实现 Prefill 与 Decode 阶段的并发执行调度机制, 动态划分 SM 资源, 使两阶段可共享 GPU 计算能力。
- 提出即时资源重配置策略, 预配置多种 SM 分区方案, 支持毫秒级动态切换;
- 结合 GPU 映射技术实现 KV Cache 的零拷贝跨进程共享, 降低通信与状态迁移开销。

基于 SLO 满足率的混合负载调度优化 | 独立实现 2025.01—2025.02

在处理输入输出长度高度异构的大模型推理混合负载场景中, 传统调度策略难以兼顾吞吐与公平性。为此, 独立设计并实现一套以 SLO 满足率为核心优化目标的调度机制。

- 请求重排序策略**: 按去除 prefix 后的 prefill 长度重排序请求, 在不违反 SLO 的限制下, 优先满足短请求的执行。
- 设计窗口调度机制**: 仅对调度窗口内的请求进行排序, 避免长请求饥饿, 平衡响应公平性与整体效率。

多智能体间 KV Cache 复用优化 | 独立实现 2024.09—2024.11

在多智能体系统中, 针对一个智能体的输入往往包含其他智能体的输出而导致 Prefill 重复计算的问题, 提出基于部分 Token 重计算的 KV Cache 复用策略。在保证精度的前提下实现智能体间 KV Cache 共享, 显著降低推理延迟。

- 利用每一层中只有部分 token 的注意力分数较高, 提出“**逐层筛选 + 选择性重计算**”机制。
- 在每一层动态评估交叉注意力分数, 仅对关键 Token 在关键层进行重计算, 其余部分复用原有的 KV Cache。