

郑 培桢

AI Infra 算法研究员 | 大模型训推与数据基础设施

180-7276-9066 pzheng46@qq.com 上海

github.com/pzheng460 pzheng460.github.io

教育背景

2024.05 | **佐治亚理工学院 • 计算机学院**
计算机科学与工程 • 硕士，GPA：3.83/4.0

2022.06 | **浙江大学 • ZJU-UIUC 学院**
土木工程 • 学士，GPA：3.70/4.0

工作经历

至今 | **华为技术有限公司 · 2012 实验室 · 理论研究部 · 计算数学组（樊玉伟）**
2025.02 | AI Infra 算法研究员；职级：15 级；**首年绩效 A+**。

核心项目

至今 2026.06 | **APIE / RSI Infra：RSI 大模型推理框架与 Agent 自动化优化**（独立开发）

- **AI Native 推理框架**：面向 RSI 的全新 LLM 推理框架，配套优化 Skill 与 Benchmark，分离稳定运行时与模型适配代码，缩小 Agent 的理解和修改范围；提供 Agent 可直接调用的统一工具接口，输出结构化执行结果与诊断信息。通过运行状态采集/回放、算子至端到端分层精度验证及源码/环境追踪，支持 Agent 定位错误、复现实验与验证优化；以 Issue 沉淀假设和证据，PR 审查与 CI 验证约束改动合入，形成可追踪的自迭代闭环。
- **融合图与异步投机流水**：将 target 验证与 draft 生成融合为同一张执行图，减少两模型交替执行的调度与算子启动开销；复用固定地址 CPU/NPU 缓冲区、增量上传 block-table，提前异步回传验证结果，使设备侧草稿生成与主机调度、输入准备重叠，减少数据搬运与串行等待。
- **并行化 Tokenizer**：以 Rust 有界线程池实现 LoPT 算法（见下），为各 worker 配置独立 BPE cache，消除 Python 多进程 IPC 与共享缓存锁竞争；DeepSeek-V4 Tokenizer 在 16 线程、21.8K-token 输入下，相比原始实现耗时由 **13.51 ms** 降至 **9.79 ms**。

2026.06 2025.08 | **openPangu-v2 预训练架构选型与 MTP 推理优化**（深度参与）

- **MTP 架构选型**：开展结构选型、接收率优化与理论分析，支撑三头 MTP 与 2048 滑窗方案落地：通过多 token 草稿提高访存带宽受限场景的解码效率，以局部注意力保留高质量预测所需上下文，控制长上下文下 MTP 的计算与 KV Cache 开销；开发快速推理验证工具，加速先导实验与候选方案迭代。
- **Multi-MTP 全图推理**：补齐框架侧多 MTP 层调用逻辑，结合 TND 输入、动态序列边界与 draft padding，在 P/D 分离部署下，使单步输入仅 1–3 个 token 的 Decode 请求仍走静态图，避免退出图执行或触发重编译；支撑 openPangu-2.0 整网优化，在 A3 单机、128K 输入/1K 输出、单并发下 Flash / Pro 的 TPOT 达到 **5.63 / 9.55 ms**，Pro 在 P/D 分离、128K 输入/8K 输出、每 NPU batch 24 下达到 **1,326 tokens/s/NPU**（TPOT 18.1 ms）。

核心项目（续）

至今 2026.08 **openPangu-v3 代际 Data Claw Dev 平台 Agent 自动化数据处理管线**（独立开发）

- **arXiv 论文预训练数据处理**：针对复杂模板兼容、公式结构保真与长尾耗时，独立实现主文档定位、依赖展开、宏规范化、公式占位/恢复及图文引用重写；以 PDF 回退与超时分流控制单篇处理成本。建立循环式开发机制：Agent 与人工每日标注错误，Agent 定时归集错误样本、自动改进解析逻辑，持续覆盖长尾模板及公式、图表解析问题。
- **GitHub 大规模分布式预训练数据处理**：基于 Data-Juicer 构建 files / commits / PRs 分布式处理管线，将逐样本清洗与跨样本去重拆为并行处理和聚合阶段。采用 commit-SHA、内容指纹精确去重与 MinHashLSH 候选检索，减少重复处理及近重复样本的比较开销，结合向量语义去重处理语义冗余；复用清洗阶段的语言语法校验结果进行质量评分，避免重复校验。按语言与质量分桶加权采样，以仓库级上下文组织样本并贪心打包，结合评测去污染与 loss-mask，兼顾处理效率和训练样本质量。

2025.08 2025.02 **DeepSeek-V3 昇腾推理性能优化**（深度参与）

- **LoPT 无损并行 Tokenizer**：在 vllm-ascend（ascend-vllm/omni-infer）实现重叠分块、字符位置匹配与动态扩块，解决并行 Tokenizer 的边界错配。论文基准中，DeepSeek-V3 Tokenizer 在 LongBenchV2、32 进程配置下耗时由 **622.4 ms** 降至 **108.2 ms**，与串行输出完全一致；成果发表于 ACL 2026 主会。
- **FlashComm 1/2/3 通信与多流优化适配**：将 Norm、量化和 QKV 降维移至 ReduceScatter 与 AllGather 之间，以投影前低维 All2All 替代投影后 ReduceScatter，通过权重冗余降低通信量；以多流重叠 Gating、共享专家计算与激活值通信。FlashComm 1/2 在并发 8、输入 2K 下将 Prefill 时延由 **951 ms** 降至 **699/631 ms**；FlashComm3 在高并发场景下使 Decode 吞吐提升 **25%–30%**，共享专家 TP8 方案节省单机约 **2.08 GB** 内存。

论文成果

LoPT: Lossless Parallel Tokenization Acceleration for Long Context Inference of Large Language Model

Wei Shao, Zheng Lingchao, Pengyu Wang, **Peizhen Zheng**, Li Jun, Yuwei Fan

ACL 2026 · Main Conference, Long Paper

专业技能

计算平台	GPU / NPU 开发，PyTorch、torch_npu、Triton；昇腾适配与性能优化。
推理系统	Continuous Batching、Paged KV Cache、P/D 分离、MTP、融合图与异步流水；vllm-ascend（ascend-vllm/omni-infer）。
模型训练	理解大模型训练与参数分片原理，了解 Megatron、PyTorch FSDP。
分布式并行	DP / TP / PP / CP / SP / EP 并行、集合通信与计算通信重叠；FlashComm 适配。
开发与数据	Python、C++、Rust、Git；多线程、分布式数据管线、文档解析与代码去重。
Agent 开发	熟练使用 Claude Code / Codex / DeepSeek Harness，结合 Skill 与 Issue/PR 开发。
英语	TOEFL：107；CET-6。