

张阔涛

电话: 19157690185 | 邮箱: zht5277865@gmail.com



教育背景

东南大学 (985)
杭州电子科技大学

电子信息 | 硕士
智能科学与技术 | 本科

2024.09 - 至今
2020.09 - 2024.06

实习经历

中国科学院深圳先进技术研究院

AI 研发实习生

2025.07 - 2026.08

实习工作:

NeurolmgAgent 影像数据分析智能体平台

项目简介: 面向临床与科研用户, 提供自然语言驱动的磁共振影像数据处理与分析能力, 降低复杂分析流程的使用门槛。

【多 Agent 协同架构】: 基于 LangGraph 构建 Profiler、Planner、Executor、Reporter 多 Agent 系统, 完成数据探查与工具编排; 自动解析任务依赖并生成支持并行调度、动态重规划的 DAG, 内部 Benchmark 规划准确率达 92%。

【自适应容错与失败恢复】: 针对长链路任务设计节点级 Checkpoint、断点续跑与 Reflect-and-Retry 机制, 关键步骤异常自动修复成功率达 78%, 将任务失败后的平均恢复时间由 2.5 小时缩短至 5 分钟以内。

【全链路可追溯】: 构建覆盖 Agent 决策、工具调用及节点产物的全链路追溯机制, 支持执行路径回放与分析结果复现。

主导研发高性能多模态脑机接口实时计算平台 (发明专利一篇)

项目简介: 面向脑功能实时监测研究, 从 0 到 1 主导系统架构与核心算法研发, 自研覆盖异构数据同步采集、跨模态对齐与融合、实时状态解码及反馈控制的实时端到端处理链路。基于 CUDA 重构高吞吐图像预处理流程, 较 CPU 基线提速约 50 倍, 端到端处理与推理延迟 P95 < 50 ms。平台已投入科研使用, 围绕核心算法申请 1 项发明专利。

项目经历

Witness: 面向仓库级 PR 自动修复的证据驱动 Agent Runtime

github.com/CodeNoob-SEU/Witness

项目简介: 面向仓库级代码自动修复中的代码检索、长上下文管理与补丁安全验证问题, 自研事件溯源 Agent Runtime 及 Issue-to-PR 修复应用, 实现 GitHub Issue 到 Docker 沙箱验证 Pull Request 的端到端自动化。

技术栈: Python + asyncio + MCP + Docker + OpenTelemetry

主要工作:

- 自研可观测、可追溯 Agent Runtime: 自研事件溯源 Agent Runtime, 以追加式事件日志统一记录模型决策、工具调用与执行结果, 运行状态均可从日志重建; 打通 Session Resume、模型/工具轨迹 Replay 与 Trace/成本统计, 使仓库级长任务在进程中中断后可续跑、历史执行可复现、问题链路可审计。
- 确定性淘汰优先的分层上下文治理: 面向仓库级长任务, 构建“事件驱动淘汰—生成式压缩—硬预算兜底”三级机制。基于 Agent 事件流, 以零模型调用无损淘汰被修改、重读、重跑或成功重试取代的过期工具观测; 仍超预算时, 再触发可持续化生成式压缩。A/B 实测末步输入 Token 降低 34%, 会话成本降低 24%。
- Runtime Debugging & PR Evidence: 将调试能力封装为 MCP 工具集成到 Agent 工具体系中, 支持断点控制、可疑栈帧跳转和帧内变量探测, 诊断静态仓库检索难以解释的隐式运行时缺陷; 基于事件日志确定性回放原始调试观测, 直接生成 PR 根因说明与证据时间线, 无需额外模型归纳, 实现“诊断即取证、取证即评审”, 减轻 Reviewer 的认知负担。

FixMate | 家电售后智能服务平台

github.com/CodeNoob-SEU/FixMate

项目简介: 面向家电售后场景的 AI Agent 服务平台, 集智能排障、可修复性判定、国补以旧换新名额领取与工单预约于一体; 用户以自然语言报修即可获得排障方案, 并一键完成权益申领与维修建单。

技术栈: Spring Boot + Spring AI + MyBatis Plus + PostgreSQL + Redis + Lua + RabbitMQ

主要工作:

- 多路召回与重排优化: 针对口语化故障描述召回率低的问题, 先经 Rewrite 归一化为标准故障术语, 再通过 向量+BM25 双路召回与 RRF+Rerank 融合重排; 在 97 条评测中, Recall@5 较纯 BM25 由 45.3% 提升至 92.5%, 且支持原文溯源。重复查询则直接复用缓存的 Embedding 与 Rerank 结果, P95 从 325ms 降至 118ms。
- 高并发写入一致性保障: 基于 Spring AI Tool Calling 实现 Agent 故障判定与国补资格核验后的自动核销建单。设计“预占—确认”两段式核销: 预占阶段通过 Redis Lua 原子扣减并基于请求 ID 幂等, 有效防止超发与重复核销; 确认后由 RabbitMQ 异步削峰落库。压测 2000 QPS / p99 45ms 下, 未发生名额超发与重复核销。
- 全链路追踪与成本归因: 以 request_id 串联检索、推理、工具调用至落库全阶段轨迹, 记录入出参、token、耗时与成本, 存入 PostgreSQL 并通过 Spring Boot Actuator 暴露延迟与成本指标; 基于该轨迹, 可按 request_id 分钟级回放定位事故, 按请求与工单归因成本, 并沉淀为评测集。

专业技能

- Agent 应用落地: 具有 Agent/Workflow 编排实战经验, 熟悉 ReAct、多轮状态管理及 MCP 工具封装与接入; 熟练使用 LangGraph、LangSmith 等 Agent 开发与可观测性框架; 具备一定的 RAG 链路构建与优化经验。
- 全栈开发经验: 具备 Java/Python 服务端开发经验, 熟悉 Spring Boot、MyBatis Plus 等开发框架; 掌握 PostgreSQL、Redis、RabbitMQ 等数据存储与中间件的应用, 能够完成接口开发、缓存优化及高并发场景下的一致性处理。