付洋朕

TARGET ROLE

Agentic AI Engineer

联系方式

所在地	中国 · 苏州
电话	+86 18506200312
邮箱	fuyangzhen@gmail.com
个人主页	me.mintmind.io
LinkedIn	YangzhenFu
GitHub	0xmoat
Twitter/X	@moat_mintmind

核心技能

Agent 架构与编排

LangGraph · ReAct · Plan-and-Execute · StateGraph · DAG Execution · Tool Use · MCP · Multi-step Reasoning

记忆与知识系统

mem0 · Vector Retrieval (pgvector, ChromaDB) · Hybrid Search · RRF Reranking · Long-term Memory

LLM 工程

Claude / OpenAI API · Prompt Engineering · Context Engineering · Structured Output (Instructor) · Fine-tuning (LoRA via LLaMA Factory)

评估与可观测性

LLM-as-Judge · Langfuse Tracing · Prometheus · Grafana

Code Agent 实践

Claude Code · Cursor · Spec-Driven Development · Reusable Skills

后端

Python · C# · FastAPI · PostgreSQL · Docker

前端

React · TypeScript · Next.js · Tailwind CSS

教育背景

北京航空航天大学
软件可靠性工程 · 硕士
2015.09 – 2018.04

中国计量大学
质量管理工程 · 学士
2011.09 – 2015.06

个人总结

7 年微软软件工程师经验，深度参与 Teams 会议回顾等 AI 产品交付链路；积累将 LLM Agent 端到端集成进企业既有工作流的工程经验——牵头从 0 到 1 设计并落地内部事故值班 Agent — Incident Copilot，在企业真实场景下打通 ReAct + 工具调用 + 长期记忆的完整 Agent 闭环。独立交付个人 Agent 项目 JobHunter（已上线）：基于 LangGraph 实现 ReAct + Plan-and-Execute 双 StateGraph 架构，工程化 Plan 子图的可执行 DAG 转换、长任务三层熔断与跨通道控制；配套 mem0 跨会话记忆、Prometheus + Langfuse 全链路可观测性、7 维度 LLM-as-Judge 评估流水线。在 Agent 记忆、知识库、工具调用、任务规划、评估闭环五大核心模块具备从设计到落地的完整经验；熟悉 Agent Harness 工程理念，持续跟踪并实践 SOTA Agent 工程范式并应用于自身项目。

工作经历

铭想智能 (MintMind) — *2025.10 – 2026.05 · 苏州*

创始团队成员

- 主导 GEO（Generative Engine Optimization）技术方案：搭建 AI 搜索可见度评估指标体系，量化品牌在 DeepSeek / Doubao 等主流 AI 搜索引擎答案中的引用率与排序位次。
- 累计服务 20+ 家企业客户、覆盖 30+ 款产品，交付从可见度评估到优化策略迭代的完整闭环；负责技术方案设计与交付。

微软 (Microsoft) — *2018.05 – 2025.09 · 苏州*

软件工程师

- **Incident Copilot — 内部事故值班 Agent（主导设计与实现）**
  - 主导设计并独立交付内部事故值班 Agent，将 LLM 驱动的 Agent 端到端集成进既有事故值班 toolchain，被值班轮换团队采纳为日常工具，将重复事故模式的平均分诊耗时降低 ~50%。
  - 设计 ReAct 动态工具决策架构：LLM 自主选择 `search_docs` / `get_incident_summary` / `get_incident_detail` 三个工具的调用顺序与时机，多轮累积上下文直至信息充足后再推理根因。
  - 基于 ChromaDB 构建工程文档知识库，`search_docs` 由 LLM 自主构造 query 检索，支持多路向量检索 + RRF 重排序。
  - 集成 mem0 沉淀长期事故记忆：每次 Incident 关闭后自动从日志与讨论评论中提炼结构化记忆（metadata / 根因 / 涉及团队 / 关联文档），下次会话作为背景上下文召回。
  - 将 Ragas 评估流水线集成进 CI：覆盖 Faithfulness / Answer Relevancy / Context Precision × 50 个回归测试用例；PR 触发评估，指标低于绝对阈值或相对 main baseline 回退 >5% 即自动阻断代码合并，防止 Prompt / 工具 / 检索策略改动引入静默的产品能力回退。
  - 通过 MCP（SSE Transport）+ Azure AD 鉴权对外暴露检索能力；团队成员在 Code Copilot / Cursor 提问即由 Agent 自动检索历史故障根因分析，覆盖开发与值班排障场景。
- **Meeting Content Platform — 跨数据中心分布式会议内容中台**
  - 承担跨多数据中心分布式会议内容中台核心模块开发，统一承载会议笔记、录音摘要、聊天分享文件等会议工件。
  - 为全球 Teams 前端及多个 AI 下游业务（含会议回顾）提供 metadata 读写能力，日均承载亿级 metadata 请求，作为 Teams AI 内容生成链路的数据底座。
- **Teams Meeting Recap 前端**
  - 主导 AI 生成会议产出的结构化呈现，将会议纪要、话题、任务、关联人物与文档以分层卡片化页面交付给最终用户。
- **Carina QnA Studio — 微软 AI 可定制问答机器人平台**
  - 设计并实现基于 ElasticSearch 的 NLP 问答引擎，支持关键词 + 语义匹配的多路混合检索 + RRF 重排序——在 LLM 普及之前即落地了与当下 RAG 同源的混合检索范式，支撑微软内部 20+ 业务团队接入、100+ 定制化 Bot 上线。
  - 沉淀通用 Bot 开发框架与组件库，提供低代码体验，支撑企业客户快速搭建定制化知识搜索 Bot。
- **O365 控制板 — Outlook 内部调试平台**
  - 设计并实现可定制告警系统 + 自适应错误分类的实时日志加载项，将生产问题平均检测耗时降低 ~50%、减少 ~40% 手动排查工作量，加速历史趋势分析。

个人项目

JobHunter — *2025.10 – 至今*

AI 驱动的求职全生命周期管理 Agent

独立设计与交付（已上线）→ jobhunter.mintmind.io

背景：求职过程涉及职位搜索、JD 分析、简历定制、求职信生成、申请追踪等多环节，信息散落于 LinkedIn / 招聘网站 / 个人笔记，求职者需在多平台反复切换且难以基于历史画像沉淀个性化策略。

目标：构建一个能同时承载即时对话与多步批处理任务、具备跨会话长期记忆、可量化评估的 AI Agent，端到端覆盖求职全流程。

过程：

- 设计 LangGraph 双 StateGraph 架构：对话通道走 ReAct 即时响应；批处理通道由 Planner / Executor / Replanner 协同完成多步任务。
- 工程化 P&E 子图：将 LLM 生成的 plan 自动转为可执行 DAG（含校验 / 自动修复 / 拓扑分波 / 不可调度时串行降级）。
- 实现长任务三层熔断：step timeout、bounded recursion、语义级 loop detection，保障批处理稳定性。
- 设计跨通道工具调用：让 ReAct 自主决定何时把多步任务让渡给 P&E 子图，避免硬编码路由。
- 设计可扩展的 Tool 注册与编排架构，已集成 14 个领域工具覆盖情报获取（职位检索 / JD 差距分析）、文档生成（定制简历 + PDF / 求职信）、申请追踪、跨通道控制 4 大场景；新工具通过统一接口注册后即可被 Agent 自主发现与调用。
- 职位检索做 SEO 落地页过滤 + Instructor 结构化 rerank，避免脏数据进入 Agent context。
- 集成 mem0 + pgvector 实现跨会话用户画像持久化与语义检索（mem0 search P50 ~300ms）。
- 针对双通道延迟差异采用差异化记忆策略：对话通道每轮动态检索，批处理通道入口一次性预加载。
- 构建全链路可观测性：Langfuse 追踪节点 span，Prometheus 埋点 TTFT / TPOT / E2E latency / 工具与 mem0 耗时，Grafana 呈现 P50 / P99 曲线。
- 搭建 7 维度 LLM-as-Judge 评估流水线（51 cases × 5 通用维度 + P&E 专项 plan_quality / replan_decision），其中 tool_appropriateness 采用双轨评分（ReAct 用 Jaccard 惩罚冗余调用，P&E 用 recall 容忍并行扩展）。

结果：通过 trace 数据分析定位评估口径偏差，迭代 judge prompt 完成质量基线校准——relevancy 均分自 0.585 提升至 0.830，pass rate 自 57% 提升至 83%；mem0 search P50 延迟稳定在 ~300ms；系统已上线，前后端 SSE 流式链路在生产稳定运行。