



李欣龙

AI 工程师

✉ 3277017758@qq.com

🔗 <https://github.com/YS-BW>

☎ 13289486468

🏠 陕西理工大学

教育经历

陕西理工大学

人工智能 · 本科

2023/09 - 至今

- 主修课程：机器学习，深度学习，自然语言处理，计算机视觉，数据结构，操作系统
- 竞赛奖项：移动杯智能体开发大赛一等奖

工作经验

四川乐途智行科技

AI 工程师

2026/01 - 2026/04

- 构建实时语音 Agent，基于 LiveKit、WebRTC、RAG 与全双工语音交互完成实时对话、打断控制，做到 500ms 首字超低延迟。
- 配合 Completion API 向 Chat API 迁移，TTFT 降低 35%，Token 成本降低 40%，缓存命中率提升至 80%。
- 参与构建业务产品库，推动保险产品、竞品信息及业务资料融入智能体知识库，增强智能体进行竞品分析的功能。
- 参与保险、旅游等场景下 Agent 应用的迭代优化，围绕多轮对话、信息追问与结果输出等链路开展验证和测试。

项目经历

Nomi | 协作养成型个人 AI 助手

独立完成

2026/04 - 2026/06

- 项目概述：**
基于终端的个人 AI 助手，支持 CLI、微信 Channel 和 API 接入，全渠道消息统一处理，共享长期记忆、用户画像、特定规则与全局任务推送机制，实现跨平台了解与陪伴用户。
- 项目亮点：**
 - 上下文管理：**整合长期记忆、近期历史、Skill 摘要与任务规则，构建多轮对话上下文，并结合压缩策略有效控制长会话输入规模。
 - 记忆机制：**采用三层记忆设计，将长期记忆沉淀、历史归档与长会话压缩整理拆分为独立环节，平衡上下文连续性与 token 开销。
 - 多 Agent 协作：**支持跨实例建立好友关系与协作网络，实现 Agent 之间的消息互通、任务转达与社交化协同。
 - 工具系统：**构建统一工具注册与调度框架，集成 Shell、Web Search、文件系统及 MCP 等能力，并设计 Skill 按需加载机制，提升复杂任务场景下的执行能力与扩展性。
 - 任务调度：**实现定时任务触发与统一调度机制，支持提醒、延迟执行等自动化场景。
 - Channel：**支持微信、飞书等外部渠道统一接入 Agent 主链路，实现远程交互与跨平台控制。

- **项目概述：**

基于 LiveKit 与 LightRAG 构建低延迟实时语音 RAG 应用，支持用户通过 WebUI 上传和管理个人知识库，并在语音对话中实时检索知识库生成回答，适用于个人知识管理、学习资料问答和实时语音助手场景。

- **技术栈：** FastAPI、LiveKit Agents、LightRAG、Qwen、MiniMax TTS、豆包 ASR、SQLite、Docker、Next.js

- **项目亮点：**

- **实时可打断语音交互：** 基于 LiveKit Agents 搭建 ASR、LLM、TTS 流式 Pipeline，支持语音实时转录、流式生成、TTS 播放和用户随时打断，优化实时对话体验。
- **个人知识库 RAG 闭环：** 集成 LightRAG 实现文档上传、解析、索引、删除、状态管理和检索增强问答，支持展示命中文档、片段内容、检索耗时等回答依据。
- **单知识库隔离检索：** 设计多知识库管理能力，每个知识库独立保存原文件、元数据和索引；语音会话开始前锁定单个知识库，避免跨库检索干扰并保证查询速度稳定。
- **多模型多音色配置：** 支持通过 WebUI 配置语音 LLM、ASR、TTS provider、模型和音色，用户挂断重连后自动生效，便于在成本、延迟和效果之间灵活调优。
- **上下文与记忆管理：** 将角色设定、知识库概览、历史会话摘要和 RAG 工具规则渲染为固定 Session System Prompt，减少每轮动态拼接成本，提升上下文稳定性。
- **统一 API 与容器化部署：** 使用 FastAPI 封装知识库、会话状态、RAG 查询、模型配置和运行态监控接口，并通过 Docker Compose 部署前端、后端、RAG 服务、LiveKit Server 和 Agent。

- **项目成果：**

- 实现端到端实时语音 RAG 闭环，覆盖知识库管理、实时语音对话、检索增强生成、回答依据追踪和公网部署。
- 将无 RAG 场景端到端首响延迟优化至约 **900ms**，支持实时打断和连续多轮语音交互。
- 在启用知识库检索的场景下，端到端首响延迟稳定在约 **1500ms**，兼顾检索准确性与实时语音体验。

专业技能

- **Agent 开发：** 熟悉智能体系统的核心模块设计与落地，包括上下文管理、记忆机制、工具调用、任务调度及子 Agent 协作；掌握 LangChain、LangGraph 等框架，并对 OpenClaw、Harness 等 Agent 系统有实践与理解。
- **LLM 工程化：** 具备多轮对话应用开发、模型调用链路调试、RAG 全流程实现、知识库与产品库构建及实时语音 Agent 原型开发经验，能够独立推进大模型应用的原型验证与功能落地。
- **编程语言：** 熟练使用 Python，具备 AI 应用、后端服务与数据处理开发经验；掌握 Java，具备并发、IO 与后端项目开发能力。
- **AI 开发工具：** 熟练使用 Codex，Claude Code、Cursor 等工具辅助代码开发、问题排查与原型验证。