

[姓名]

AI 应用 / 全栈 / 本地 AI 多媒体工程师 · 大数据专业

手机: [待补充] 邮箱: [待补充] 城市: [待补充] GitHub / 作品集: [待补充]

在线项目: [Lark 多机器人控制台](#) · [Radar Cloud Lite](#) · [Work 工单系统](#)

01 个人概况

大数据专业背景，方向覆盖 AI 应用工程、全栈开发、本地 AI 多媒体与网站部署。具备从需求拆解、架构设计、接口与数据状态实现，到测试、发布、域名配置和线上验证的端到端项目实践。重点实践包括 Lark 多机器人协作系统、工业电商售后工单后端、本地视频超分、抽帧补帧与 AI 剪辑、真人配音与声音转换、Godot 4 游戏开发技术及数据雷达。能够把模型、脚本、网页、服务和运维配置整理为可运行、可验证、可回滚的项目。

AI 与 Agent PyTorch · LLM / Agent · Ollama · LM Studio	编程与数据 Python · Pandas · Hadoop / Kettle · MySQL
系统与部署 Linux · Docker · Nginx · systemd	SEO 与增长 长尾词 · KGR · Semrush · GSC / GA4

02 核心能力与 AI 学习投入

- 海外 AI 与自费学习：精通海外主流 AI 产品和开发工具的使用 workflow；个人自费使用正规 Plus、Team 与 K12 教育类订阅持续学习，不依赖第三方接口中转，熟悉长上下文、文件分析、编码、测试、图像与文档 workflow。
- 编程、数据与深度学习：以 Python 为主，使用 NumPy、Pandas、Matplotlib 处理与分析数据；具备 C/C++、Java、Lua、自动化脚本及数据结构与算法基础，熟悉 PyTorch、TensorFlow 与 MobileNetV2 量化网络应用。
- LLM、Agent 与工程化：使用 ChatGLM/GLM-4、Ollama、LM Studio、DeepSeek 与 OpenAI-compatible API，实践 Prompt Engineering、结构化输出、Agent 任务编排与工具调用；了解 OpenClaw，精通可复用 Skill 编写并熟练对接 MCP JSON-RPC。
- 大数据、数据库与平台交付：熟悉 Hadoop (HDFS/Hive)、Kettle ETL、MySQL CRUD、索引优化和主从复制配置维护；可在 CentOS/Ubuntu 上完成 Docker/Compose、Nginx/OpenResty、systemd、DNS/TLS、日志、健康检查与哈希回滚。
- SEO 与增长：熟悉长尾关键词挖掘与关键词聚类，能够按业务词、产品词、问题词及搜索意图拆分主题，并结合 KGR、搜索量、竞争度和 Semrush 竞品数据筛选内容机会；完成 Title/Meta/H1、URL 与站内结构、内链、Canonical、robots.txt、Sitemap 和索引覆盖优化，通过 GSC/GA4 复盘展示、点击、CTR 与落地页表现；推动部分相关长尾关键词进入搜索结果首页并取得靠前位置。

个人 AI 学习用量 (原始截图)



统计区间 2026-06-17—2026-08-19: 183.7K 请求、22.7B Token、成功率 79%、平均延迟 16.38s；面板估算价值 \$2,815.80 (仅为估算价值，不是实际账单)。

03 重点项目经历

Lark 多机器人控制台 / 企业 AI 协作系统 | 独立设计与全栈开发

Node.js · ES Modules · Lark OpenAPI · WebSocket · Interactive Card · DeepSeek API 在线地址: work.lsyapi.xyz/lark/

面向企业岗位协作演示, 从 0 到 1 搭建本地 Lark/飞书多机器人控制台, 统一编排自建应用机器人和 Webhook 机器人, 覆盖消息事件、卡片内容、AI 托管、定时任务、演示脚本与用量统计。

- 多机器人配置与路由: 以本地 JSON 保存机器人、群聊、模板与调度配置, 统一展示名称、Open ID、连接状态和职责; 收到消息后按 mention、机器人身份和关键词规则分流。
- 事件驱动消息链路: 通过 Lark WebSocket 订阅 im.message.receive_v1, 完成 @ 识别、发送、引用回复、Reaction 等待胶囊和事件回调; 使用 botKey + messageId 占位去重, 避免多事件或多机器人重复回复。
- 业务卡片与自动化: 支持 Lark 原生 table 元素、Markdown 卡片、自定义 JSON 卡片, 以及“正文—表格—正文”组合卡片; 提供模板库、DIY 编辑、实时预览、关键词触发、定时推送和 Webhook 批量管理。
- 界面与展示: 在不改变消息链路的前提下完成 iOS 风格 Liquid Glass、背景模糊、胶囊按钮/状态卡与响应式布局; 公开版本采用只读演示模式并挂载到 Work 站点 /lark/ 目录。
- AI 托管与质量控制: 接入 DeepSeek Chat Completions, 为不同岗位设置公司上下文、人设和输出规范; 对套话、占位符、过期日期、结构异常和无效表格做质量检查, 失败时自动重试并回退本地模板。
- 额度、演示与交付: 用长连接事件替代消息轮询, 记录发送、回复、关键词、AI 调用和 Token 用量; 制作脱敏发布包和只读公开演示, 公网禁用写操作与 /api/*, 敏感配置不进入前端。
- 目录化部署: 为统一 Work 子域名配置 DNS 与 Nginx/OpenResty 路由, Lark 使用 /lark/ 独立目录; 静态演示、Radar、工单后端和管理面板保持边界隔离。

工业电商售后工单系统 / 可体验后端 Demo | 独立设计、开发与部署

Python · SQLite WAL · REST API · HTML/CSS/JavaScript · Nginx/OpenResty · Ubuntu 在线地址:

work.lsyapi.xyz/workorder/

面向电商与工业售后场景搭建轻量工单系统, 不绑定具体现场行业; 公开站点可直接创建、审核、分派、处理和关闭工单, 后端状态与事件记录实时持久化。

- 后端与数据模型: 使用 Python 标准库 ThreadingHTTPServer 提供 REST API, 以 SQLite WAL 保存工单、状态、优先级、负责人、SLA 与事件时间线; 提供健康检查、统计、列表、详情和事件查询接口。
- 工单闭环: 实现 review、open、assign、queue、resolve 等状态动作, 前端每次操作都回写后端并生成事件记录; 支持待审核筛选、优先级展示和处理时长统计。
- 可体验前端: 使用原生 HTML/CSS/JavaScript 构建响应式控制台, API 采用相对路径, 既能本地运行也能挂载在统一 Work 站点的 /workorder/ 目录, 不新增项目子域名。
- 部署与验证: 在 Ubuntu 上以 systemd 守护后端, 由 Nginx/OpenResty 将 /workorder/ 前缀代理到服务; 完成桌面/移动端截图验收、4 项 unittest 回归测试, 并对 250 条公开演示容量设置上限。

本地 AI 视频生产与智能剪辑工具链 | 本地多媒体 / 工程化整合

Python · Node.js · TypeScript · Real-ESRGAN · FFmpeg · RIFE NCNN Vulkan · WebGPU · HyperFrames 在线地址:

work.lsyapi.xyz/videos/

将画质增强、抽帧补帧、多轨剪辑、AI 字幕与无旁白演示视频生成组织为本地工具链; 素材留在本机处理, 从网页交互、任务状态到 MP4/WebM/GIF 成片形成完整链路。

- 超分链路: 针对真人素材优先让 Real-ESRGAN 按原生 4× 倍率恢复细节, 再高质量回缩至 2× 或 1080p, 减少非原生倍率造成的接缝; 同时提供快速预览和动漫优化模式。
- 输出与性能: 支持 HEVC 10-bit 和 H.264, 音频轨直接复制; 围绕 RTX 4060 Laptop GPU 管理显存、临时帧、失败清理和任务状态, 兼顾画质、兼容性与文件体积。
- 抽帧补帧系统: 以 Node.js 网页控制台整合 FFmpeg/ffprobe 与 RIFE NCNN Vulkan, 支持区间拆帧、图片替帧、关键帧补画、AI 平滑慢放、双视频插入/替换、边界截图和多画幅时间轴导出; 生成前后检查帧率、音画时长与浏览器兼容性。

- AI 剪辑与视频生成：本地搭建 React/TypeScript 多轨剪辑台，覆盖视频/音频/文字/蒙版/Lottie、关键帧、转场、自动节拍与字幕转写；使用 HyperFrames 以 HTML/CSS/GSAP 编排确定性动效，逐帧捕获并通过 FFmpeg 输出 MP4/WebM，本次工单演示明确采用无旁白方案并挂载在统一 Work 站点的 /videos/ 目录。

04 独立 AI 配音系统、Godot 4 技术与其他项目

本地 AI 真人配音与声音转换系统 | 本地工作台 / 多引擎集成

Python • Flask/Waitress • FFmpeg • CosyVoice 3 • Seed-VC • RVC • GPT-SoVITS • Piper

构建仅监听本机的网页配音台，将文本转语音、零样本音色复刻、RVC/Seed-VC 声音转换、脚本清洗、时长适配和视频混音组织为一条可试听、可回退、可出片的工作流；该能力为独立项目，不参与上述无旁白工单演示。

- 多引擎与角色控制：以 CosyVoice 3 为本地主引擎，接入 Seed-VC、RVC、GPT-SoVITS、Piper；各引擎使用独立环境，并支持零样本参考音色、情绪、语态、语速与多语言角色试听。
- GPU 与离线运行：针对 8GB 显存设计单任务串行队列，在大型模型之间按需拉起、停止和释放；本地模型使用离线缓存并校验关键资产。
- 视频成片：可导入 TXT/SRT/VTT/ASS，清洗时间轴与标签并估算配音时长；输出 48 kHz PCM WAV 编辑母版、MP3 试听和混音后 MP4，支持人声响度统一、音乐闪避、峰值限制与淡出。
- 质量与来源管理：维护参考音频、模型、训练数据和许可说明，提供试听缓存、质量报告、完整性检查与输出安全测试；仅使用本人、公开许可或明确获得授权的声音素材。

Godot 4 游戏开发技术实践 | Godot 4 / GDScript • 技术实践 (在研)

Scene/Node • Resource • Autoload • Signals • World Chunk • CharacterBody • 状态机 • 存档 • OpenAI-compatible API 在线地址: work.lsyapi.xyz/game/

- 引擎架构：使用 Godot 4 / GDScript，掌握 Scene/Node 组合式场景组织、PackedScene 实例化、Resource 数据驱动配置、Autoload 全局服务与类型化 Signals 事件解耦。
- 世界与角色：实践 2.5D World Chunk 分块、固定种子程序化生成、区块缓存与按需加载；使用 CharacterBody、碰撞层、导航查询和有限状态机组织角色移动与状态切换。
- 数据与接口：实现存档序列化与状态恢复，并封装 OpenAI-compatible HTTP API，支持结构化 JSON 校验、超时、重试和本地回退；当前仍处于技术验证阶段，尚未完成最终可执行版本导出。

其他可展示项目

项目	说明
Radar Cloud Lite	数据目录、周期索引、Web 控制台与认证边界；完成 Ubuntu、Nginx、systemd、日志轮转、robots.txt 与 sitemap.xml 部署，并在统一 Work 站点保留独立展示目录。 在线站点
响应式活动 H5	使用 React 19、Next.js 16、TypeScript、vinext 与 Drizzle 搭建全栈 H5，完成构建、渲染测试及 Cloudflare D1/R2 绑定预留。

05 技术栈与工作方式

编程	Python (NumPy / Pandas / Matplotlib) ; C / C++; Java; Lua; Node.js / TypeScript; React / Next.js; PowerShell / Shell; Godot / GDScript
AI / Agent	PyTorch; TensorFlow; MobileNetV2 量化网络; ChatGLM / GLM-4; Ollama; LM Studio; Prompt Engineering; OpenClaw; Agent 任务编排与工具调用
数据	MySQL (CRUD / 索引优化 / 主从复制) ; Hadoop (HDFS / Hive) ; Kettle ETL; SQLite; JSON
系统	Linux (CentOS / Ubuntu) ; Docker / Compose; Nginx / OpenResty; systemd; HTTP API; WebSocket; TLS / DNS
SEO	长尾关键词; 关键词聚类; KGR; 搜索意图; Semrush; 站内结构 / Meta / 内链; GSC; GA4; Sitemap / 索引覆盖