

<div>宋飒</div> <div><div><div><div><div></div></div><div>Beijing</div></div><div><div><div></div></div><div>15083129061</div></div><div><div><div></div></div><div>shemol106@gmail.com</div></div><div><div><div></div></div><div>https://shemol.tech</div></div></div></div>		
基本信息	<div><div><div></div></div>SherlockShemol</div>	
教育经历	北京邮电大学 通信工程 85.01/100(29.34%)	2020年9月 - 2024年7月 本科
	北京邮电大学 信息与通信工程	2024年9月 - 至今 研究生
专业技能	<div><div><div><div><div></div></div><div>热衷尝试各种AI产品，包括Z-Code，Alma，Cursor，Antigravity，v0，Lovable，Cline，Readever等等</div></div><div><div><div></div></div><div>关注Claude，OpenAI等技术博客，关注Andrew Karpathy，Lee Robinson等人的推文和博客文章。</div></div><div><div><div></div></div><div>熟悉 HTTP/HTTPS 协议及常用数据结构与算法</div></div><div><div><div></div></div><div>熟悉掌握 HTML5、CSS3 及 ES6+ 语法</div></div><div><div><div></div></div><div>熟悉使用 TypeScript 进行强类型编程，提升代码健壮性</div></div><div><div><div></div></div><div>熟悉 React 框架及 Hooks 编程模式，了解常用的状态管理方案</div></div><div><div><div></div></div><div>熟悉 Webpack/Vite 等前端构建工具的基本配置</div></div><div><div><div></div></div><div>熟悉使用Go进行开源项目开发，熟悉相关数据类型和基本的并发编程</div></div><div><div><div></div></div><div>熟悉 Go 语言及并发编程，具备 Kubernetes Operator 开发经验</div></div><div><div><div></div></div><div>熟练掌握 Docker 容器化技术与 Linux 系统配置，能够独立完成应用的容器化部署与环境搭建</div></div><div><div><div></div></div><div>熟练掌握Git分布式版本控制系统，具备开源社区协作经验，能够高效完成分支策略设计、代码冲突解决、Pull Request提交</div></div></div></div>	
项目经历	Agora：分布式协议 Agent 测试平台	2025年3月 - 至今
	<div>技术栈：Python、asyncio、gRPC、Event-Driven Architecture、Prompt Engineering</div> <div>项目简介：设计并实现了一套分布式系统测试平台。核心创新在于将 LLM 作为协议决策引擎，取代传统硬编码状态机。系统采用两层架构：上层 Orion 层提供智能客户端、故障注入和行为验证；下层 Constellation 层实现基于 LLM 决策的协议 Agent （Raft/PBFT）。</div> <div><div><div><div><div></div></div><div>LLM-Native 协议决策引擎：将 Raft/PBFT 协议的决策逻辑完全交给 LLM。Agent 通过构造 STATE（角色/term/日志状态）+ TRACE（近期事件历史）的结构化 Prompt，让 LLM 输出 JSON 格式决策（action + params），再由确定性的 Handler 执行。实现了 "感知→推理→执行" 的 AI Agent 闭环。</div></div><div><div><div><div></div></div><div>Constellation 统一框架：设计 BaseProtocolAgent 抽象基类，封装事件系统（EventSystem）、状态管理（StateManager）、网络层（NetworkLayer）、定时器（TimerSystem）等通用组件。新协议只需实现 get_protocol_rules() 和 make_fallback_decision() 等抽象方法，大幅降低协议开发成本。</div></div><div><div><div><div></div></div><div>安全回退与可解释性：当 LLM 输出非法 JSON 或违反协议安全时，自动切换到纯规则 Fallback 策略，确保一致性安全优先于 LLM 表达。通过 STATE/TRACE 机制保留完整决策链路，支持问题追溯和调试。</div></div><div><div><div><div></div></div><div>智能测试 Orion 层：Client Agent 由 LLM 驱动，根据响应状态（ok/redirect/error）智能选择 send/retry/success/fail 动作；Injector 支持网络分区、延迟注入、状态篡改等多种故障场景；Checker 实时验证系统行为是否符合协议不变量。</div></div></div></div></div></div></div>	
	Consen：Multi-Agent 分布式协议代码自动生成与验证系统	2025年11月 - 至今
	<div>技术栈：Python、Multi-Agent、asyncio、Prompt Engineering</div> <div>项目简介：构建了一套 Multi-Agent 协作系统，通过 LLM 驱动的红蓝对抗机制实现分布式共识协议（Raft/EPaxos）代码的自动化生成、测试与修复。系统包含三个核心 Agent：Orchestrator（编排智能体）、Coder（代码生成）、Checker（对抗测试），形成 PLAN→BUILD→TEST→FIX→STABLE 的完整开发闭环。</div> <div><div><div><div><div></div></div><div>Multi-Agent 协作架构：Orchestrator Agent 作为编排者，通过 JSON 决策协议协调 subAgent：Coder Agent 和 Checker Agent，实现 PLAN→BUILD→TEST→FIX 生命周期的全自动流转。使用无状态 Prompt 设计，每轮决策构建完整上下文，确保 Agent 决策的一致性。</div></div><div><div><div><div></div></div><div>LLM 驱动的代码生成：Coder Agent 支持 Plan/Build/Fix 三种模式：Plan 模式解析协议规范自动生成实现计划；Build 模式按步骤增量构建代码；Fix 模式结合 Failure Log 和协议规范定位并修复 Bug。通过 Chain-of-Thought 提示工程，实现 THINK→CODE 的结构化输出。</div></div><div><div><div><div></div></div><div>红队对抗测试系统：Checker Agent 作为 LLM 驱动的红队攻击者，根据协议规范和源代码自动生成攻击方案，通过 DropRule/DelayRule/MutateRule 进行故障注入，探测 Safety（一致性违规）和 Liveness（活性故障）Bug。支持 CFT/BFT 两种故障模型。</div></div><div><div><div><div></div></div><div>经验驱动的测试优化：实现 Tests Memory 模块，持久化存储成功的攻击模式并基于相似度检索复用；实现 Bug Pattern Loader，从历史 Bug 模式库中检索相关案例注入 Prompt，提升测试覆盖率和 Bug 发现效率。</div></div></div></div></div></div></div>	
获奖情况	2020-2021学年校级二等奖学金	2021年9月
	2022-2023学年校级三等奖学金	2023年9月
	挑战杯”首都大学生课外学术科技作品竞赛二等奖	2023年7月
	2024开源之夏成功结项	2024年11月
	2024年校级一等奖学金	2024年11月
	2025 CloudWeGo黑客松大赛决赛第二名	2025年4月
	2025年校级一等奖学金	2025年11月
开源经历	KubeEdge-Sedna：联合推理与联邦学习控制器优化	
	<div>PR链接1：https://github.com/kubeedge/sedna/pull/446</div> <div>PR链接2：https://github.com/kubeedge/sedna/pull/445</div> <div>PR链接3：https://github.com/kubeedge/sedna/pull/438</div> <div>PR链接4：https://github.com/kubeedge/sedna/pull/437</div> <div>minionS：添加docker容器化支持；PDF文件处理支持Windows；为远端客户端提供DeepSeek API 支持</div> <div>PR链接：https://github.com/HazyResearch/minions/pull/54</div> <div>PR链接：https://github.com/HazyResearch/minions/pull/47</div> <div>PR链接1：https://github.com/HazyResearch/minions/pull/16</div> <div>PR链接2：https://github.com/HazyResearch/minions/pull/40</div> <div>lmp：为eBPF运行脚本提供Fedora依赖安装支持；实现自动检测KVM BTF信息，不存在时回退到vmlinux</div> <div>PR链接：https://github.com/linuxkerneltravel/lmp/pull/976</div> <div>Dify：修复前端思维链渲染 Bug；修复高负载下内存泄漏；为 Avatar、Chip、Badge 组件添加单元测试</div> <div>PR链接1：https://github.com/langgenius/dify/pull/27776</div> <div>PR链接2：https://github.com/langgenius/dify/pull/30236</div> <div>PR链接3：https://github.com/langgenius/dify/pull/30201</div> <div>PR链接4：https://github.com/langgenius/dify/pull/30119</div> <div>PR链接5：https://github.com/langgenius/dify/pull/30096</div> <div>Cherry Studio：修复 API Key 空格截断问题；使用 Map 优化模型状态查找性能；修复全局记忆设置提交失败；修复自定义端点后缀问题</div> <div>PR链接1：https://github.com/CherryHQ/cherry-studio/pull/10751</div> <div>PR链接2：https://github.com/CherryHQ/cherry-studio/pull/12161</div> <div>PR链接3：https://github.com/CherryHQ/cherry-studio/pull/12147</div> <div>PR链接4：https://github.com/CherryHQ/cherry-studio/pull/12163</div>	
语言水平	<div>中文</div> <div>母语水平</div> <div>英语</div> <div>六级 527</div>	