



PILOTDECK

—— 面向生产力的智能体操作系统

—— *An Attempt at a Productivity-Oriented Agent Operating System*

启元实验室 大模型中心 副研究员

清华大学自然语言处理（THUNLP）实验室 客座研究员

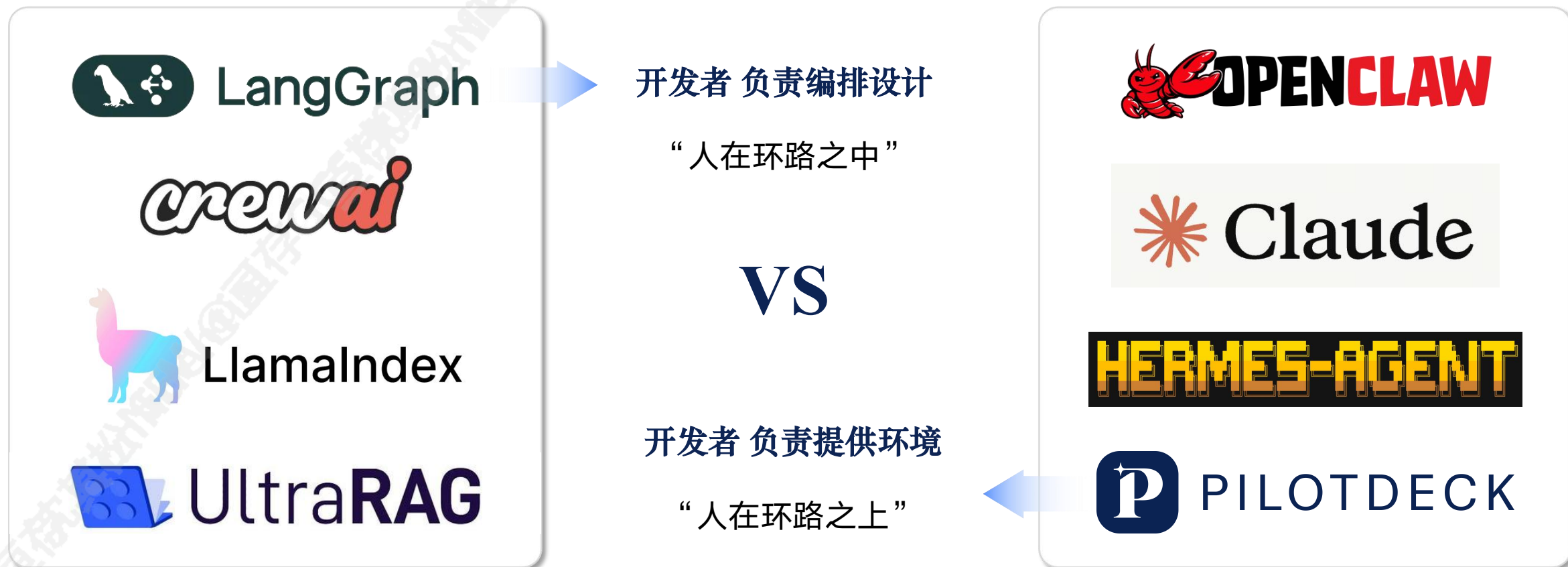
闫宇坤

2026年7月

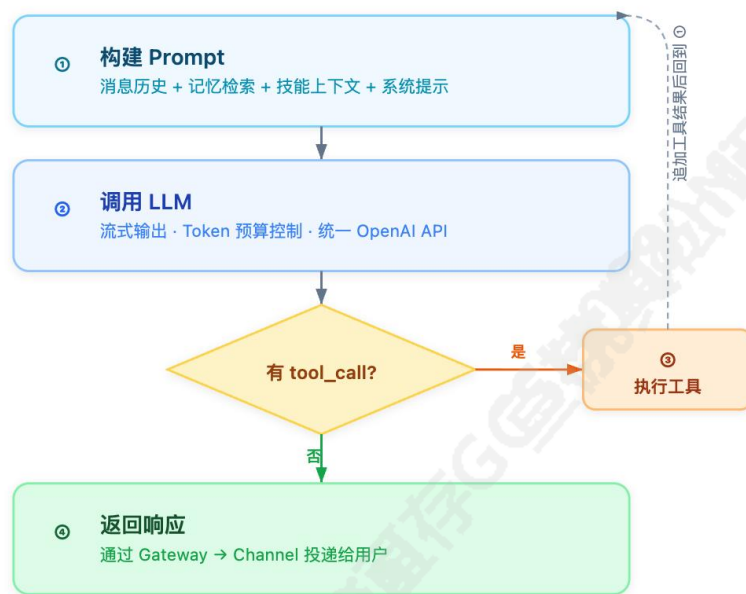
智能体技术已经历“推理能力增强——工具能力提升——复杂问题拆解”等发展阶段，正在迈向通用能力更强的智能体操作系统



FrameWork 的核心是灵活编排智能体工作流，AgentOS 的核心是定义标准化智能体工作流，并提供良好的“生产环境”



围绕核心Agent-Loop循环，AgentOS从控制、调度、执行、通信多个层面创建工作环境



核心循环 Agent-Loop



引擎 VS 车辆



AgentOS架构（注：不同项目之间有细微差异）



生产力提升的第一性原理 —— 提供好的生产关系。PilotDeck的设计思路就是给大模型提供最适合其工作的“生产环境”与“执行模式”，解决现有系统的瓶颈问题。

瓶颈一：多项目并行的记忆污染

现有 AgentOS 普遍使用全局记忆池，多项目并行时上下文互相污染。

瓶颈二：不加区分的 Token 成本浪费

当前 AgentOS 对简单任务和复杂任务使用同等昂贵的模型配置。

瓶颈三：问答范式的能动性缺失

现有 Agent 本质上是“问答机器”——用户不问，它不动。

当前 AgentOS 面临的部分挑战



Workspace 独立工作舱

项目制白盒记忆

多模型智能路由

常驻式主动执行

PilotDeck的核心设计



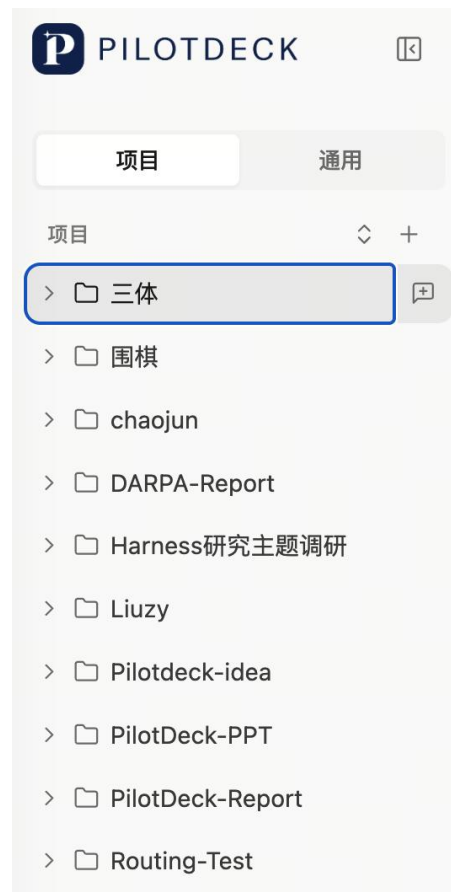
Pilotdeck 的基础核心设计是引入“工作空间”概念，给执行不同任务的“智能体”独立的生产环境，实现多任务的高效并行推进。



仅文件系统隔离



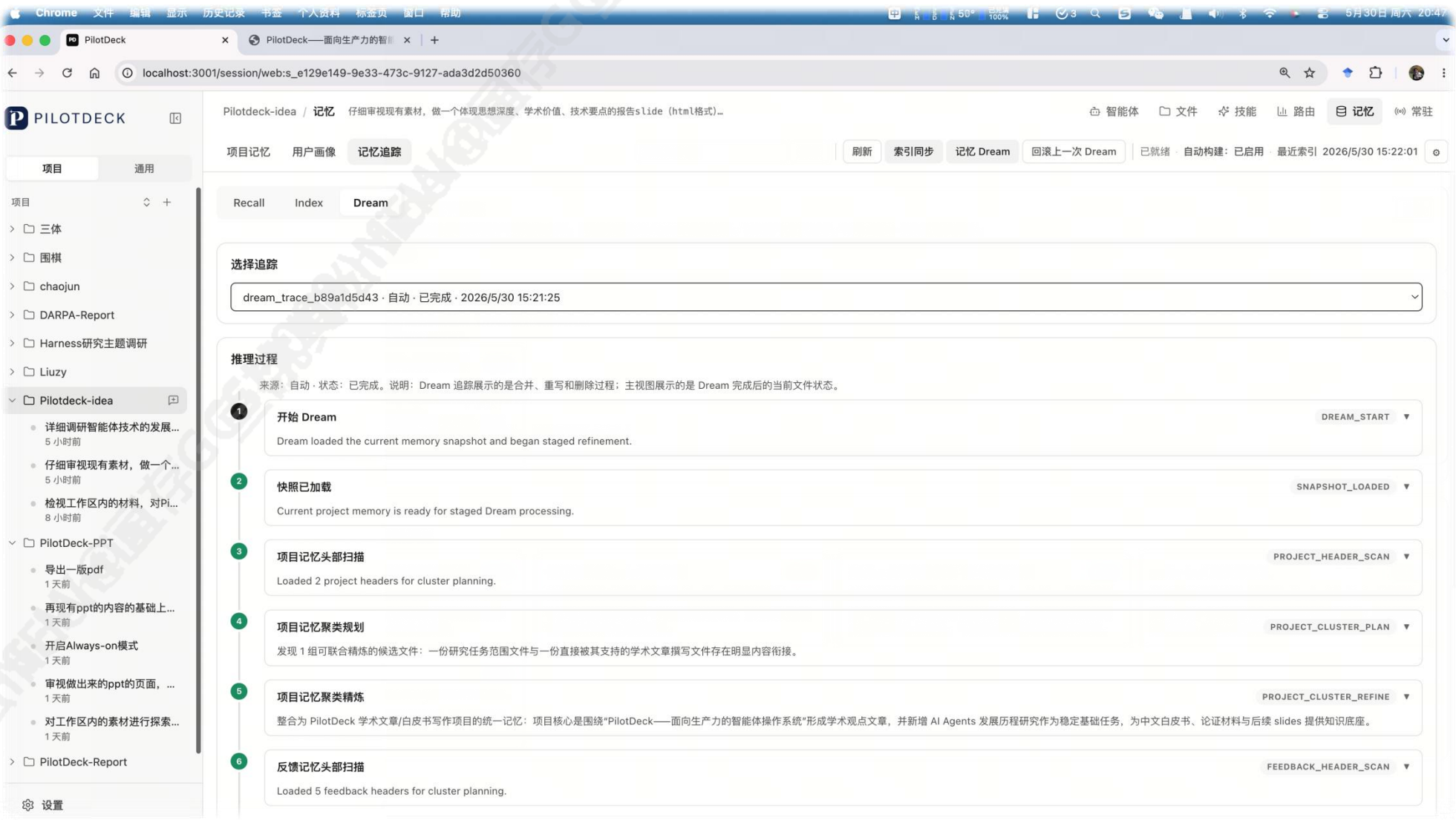
生产环境隔离



Workspace-Oriented

- **文件隔离**：基于 Git worktree 实现物理隔离，每个 Workspace 独立工作目录，支持一键回滚
- **记忆隔离**：按项目标识分区存储，检索时只在当前 Workspace 范围内搜索，跨项目按需召回
- **技能隔离**：基于项目级配置加载不同技能集





Project Memory + Feedback Memory



PilotDeck 的记忆系统包含心跳索引 → 推理检索 → Dream Review 三阶段功能过程

索引构建阶段

- 按批处理未索引会话记录
- 项目记忆（项目信息）、用户画像（偏好风格）、反馈（评价纠正）+文件记忆（变更记录）
- 持久化到 SQLite，按项目标识分区

推理检索阶段

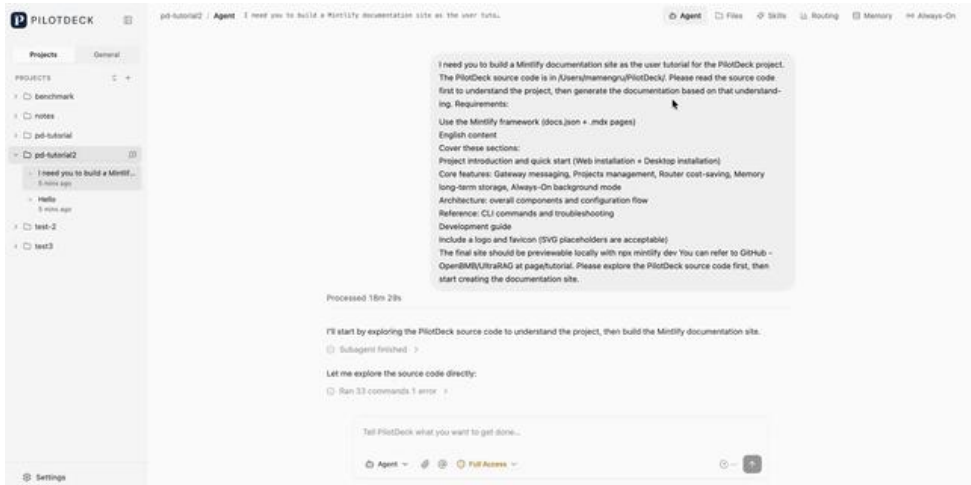
- 查询归一化 + 30 秒 TTL 召回缓存
- 第一轮推理：LLM 判断查询是否与已有记忆相关
- 第二轮推理：选择最相关的记忆条目（默认 5 条）

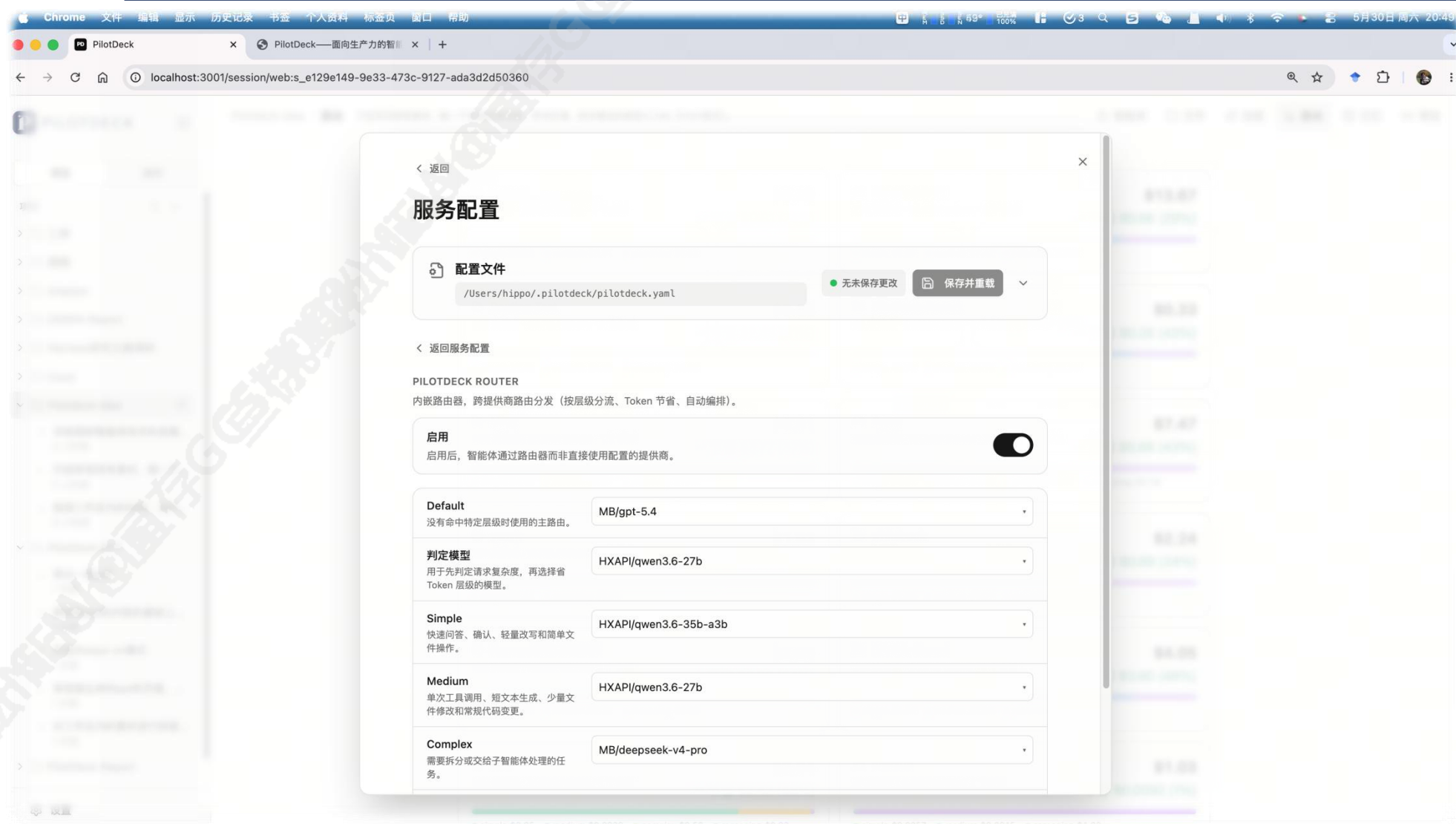
Dream Review阶段

- LLM 聚类 → 合并去重 → 冲突解决 → 过时清理 → 画像更新

维度	传统 Agent（黑盒）	PilotDeck（白盒）
可见性	看不到 AI 记住了什么	随时查看，按项目/类型/时间筛选
可控性	写入后无法修改	手动编辑、删除、标记
可追溯	无法定位原因	生成→抽取→存储→使用全链路可查
可回滚	压缩后无法恢复	Dream 整理后支持一键回滚

黑盒记忆与白盒记忆对比





Model Routing

当前 Agent 系统的 Token 成本主要来源：对简单任务和复杂任务不加区分地使用旗舰模型。智能路由的目标是：在维持输出质量的前提下，通过任务复杂度自动分级，将简单任务降级到轻量模型。

决策时机

- 在子任务阶段进行决策：避免KV-cache hit机制失效（多种LLM-Router通病）

决策机制

- 显式指定 → 子代理选择 → 默认智能路由
- Judge 模式：Routing Prompt 模型进行五级难度分类

兜底策略

- 粘性路由：同一会话内保持模型选择一致性
- 自动编排：复杂任务拆分多 Agent 协作（主强子弱）
- 路由压缩：四级上下文压缩基础上添加一层Routing后压缩判别

方案	模型编排	费用	相对倍率
开启智能路由	主 Opus 4.5 + 子 Sonnet 4.5	\$2.83	1.1×
不开智能路由	全 Opus 4.5	\$12.58	5.0×
单体大模型	单体 Opus 4.5 长 ReAct	\$12.20	4.8×

社交媒体内容创作实测节约75+%费用

配置	质量得分	成本
MiniMax-M2.7 单 Agent	37.1	\$1.90
Claude Sonnet 4.6 单 Agent	69.1	\$18.36
主 Sonnet 4.6 + 子 MiniMax-M2.7	70.6	\$3.15

WildClawBench（部分）复杂任务实测效果对比



PILOTDECK

ProjectsGeneral

PROJECTS

1-奇思妙想

1-小学奥数

1-wang zhe rong yao

2-mylife

2-voiceinput

4399

白皮书

测试

测试

测试1

测试3

测试6

好困

会议纪要

剧本杀

摩托车竞赛

内部测试

图表制作

小说创作系统

Aeneas

ai-marketing

ai-media

ai2ppt

Settings

general / Always-On

DashboardPlans & Cron Jobs

Back to Plans & Cron Jobs

Add Session Split Recommendations to DM Manual Section 5

小说创作系统completed

WORKSPACE	EXECUTION SESSION	OUTCOME
snapshot-copy	—	—

Plan

Add Session Split Recommendations to DM Manual Section 5

Always-On Discovery Plan id: plan_6151c924-66ac-42f4-8619-93dabd80be20 sourceRunId: 6151c924-66ac-42f4-8619-93dabd80be20 createdAt: 2026-05-24T01:12:15.456Z projectRoot: /home/liyishan/内部测试 dedupeKey: naruto-trpg-dm-manual-session-split

Summary

The DM manual's chapter 5 (节奏控制) has time tables but lacks the 方案A/B/C Session切分建议 from the time test report, leaving DMs with no structured multi-session planning guide.

Rationale

The time test report section 6 "Session切分建议" provides three concrete multi-session plans: 方案A (standard group, 3 sessions), 方案B (novice group, 5 sessions), and 方案C (side-quest Danzo session guidance). The DM manual's chapter 5 currently has sections 5.1 (time-per-act table), 5.2 (combat rhythm), and 5.3 (plot rhythm), but the word "Session" does not appear anywhere in the file. DMs must separately consult the time test report to know how to cut a 7-11 hour campaign into manageable 2-hour sessions, which breaks the single-document DM workflow. Adding a "### 5.4 Session切分建议" subsection to the DM manual directly after 5.3 mirrors the structure used in the time test report and ensures DMs have all operational guidance in one place.

Context Signals

- grep -in "session" 火影忍者_木叶暗部日志_DM指引手册.md | wc -l returns 0 — the word "Session" never appears in the DM manual.
- 火影忍者_木叶暗部日志_时间测试报告.md lines 271-297 contain the full "Session切分建议" section with 方案A/B/C tables, confirmed by grep -n "Session切分" 火影忍者_木叶暗部日志_时间测试报告.md returning line 271.
- The DM manual has only three subsections under chapter 5 (5.1-5.3); grep -n "### 5\." 火影忍者_木叶暗部日志_DM指引手册.md shows 5.1 各幕时间控制, 5.2 战斗节奏控制, 5.3 剧情节奏控制 — no 5.4 exists.
- The 5.1 table already references "主线合计 ~11h / ~7h 30m / ~6h", making it natural to follow with a session-splitting subsection that uses those totals.
- 方案B item #1 matches time test recommendation #7: "序章单独作为第1次Session: 新手序章约2h, 配合规则讲解刚好装满一次聚会。" — neither is implemented in the DM manual.

Execution Started - 内部测试05:32

Always-on



构建 Agent 的“问答机器”之外的自主服务范式：用户离开后工作继续推进、持续监控外部信息源、主动发现任务。

调度层

- 定时触发 + 信号触发
- 门控评估 (8道) - 多层防无效并发

执行层

- 分布式锁 (防文件修改冲突)
- Gateway启动服务 -> 结果磁盘持久化

隔离层

- 优先使用Git Worktree进行保护
- Git Worktree不可用时使用 clonefile/reflink等机制进行文件复制
- 产出结果被批准后方可影响原始工作区

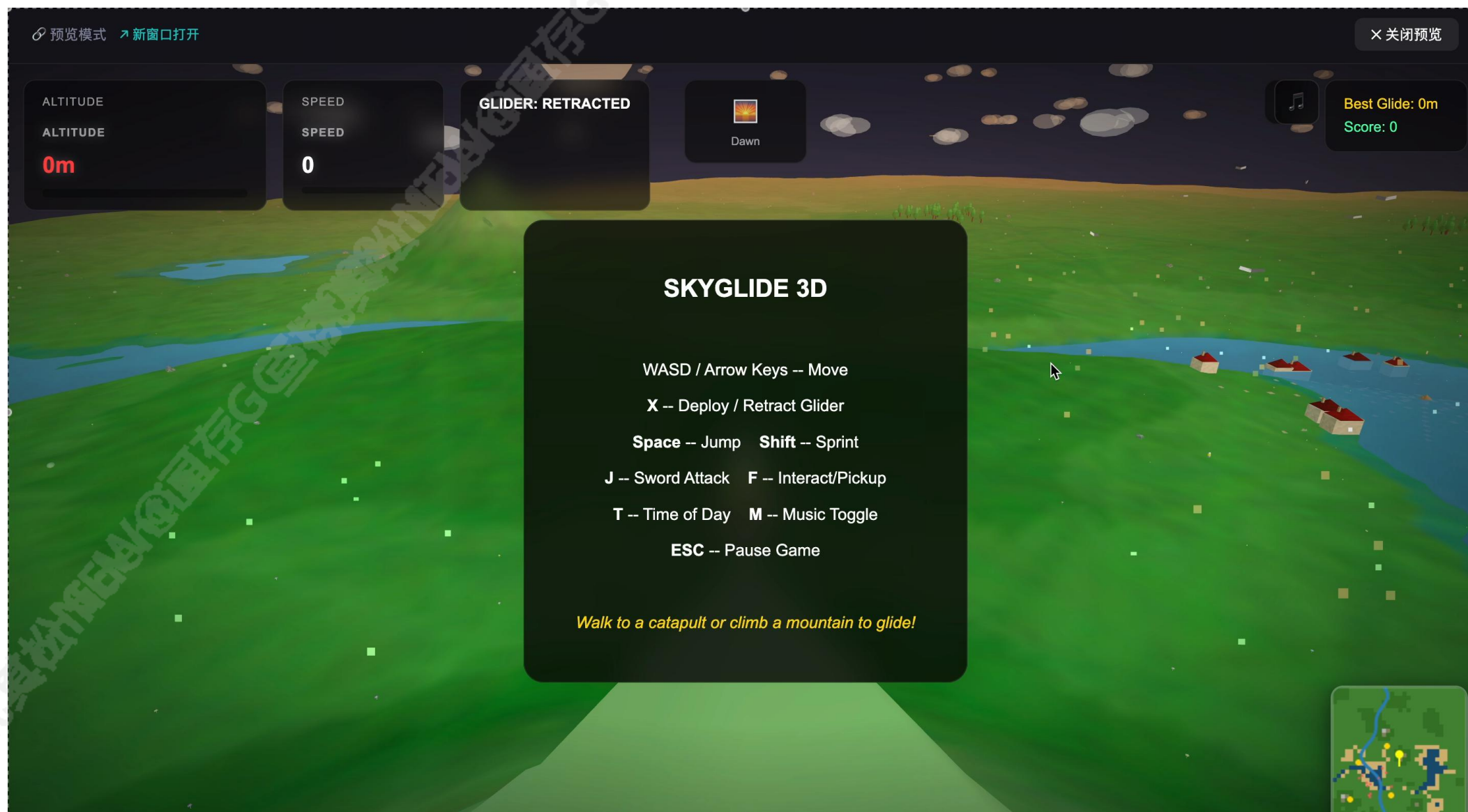
维度	传统 Cron Jobs	PilotDeck Always-on
任务发现	用户预定义固定任务	Agent 主动发现潜在任务
条件评估	无条件执行	门控评估，避免无效运行
成果落地	部分	自动落地为文件
进度汇报	无	结构化报告 + 推送通知
工作隔离	无	Git worktree 隔离

对比传统定时任务

工具	功能
AlwaysOnDiscoveryPlanTool	创建/管理发现计划
AlwaysOnReportTool	生成运行报告
AlwaysOnWorkspaceTool	管理工作空间
AlwaysOnChatHistoryTool	读取历史聊天记录

专属工具列表





20B级别模型构造“塞尔达”世界



行业现状

各有所长,各有盲区

Claude Code
Cursor
Trae Solo

深度集成 IDE
专攻编程

2024 年开始,各种 AI Agent 工具像潮水一样涌过来。
Claude Code 帮你写代码,Cursor 帮你改 Bug,飞书机器人帮你回消息。

感谢聆听！

欢迎大家关注 PilotDeck，合作共建！

我们也在进行Memory、Harness等方向的研究工作，欢迎投递科研实习简历！



个人邮箱地址：
yanyk.thu@gmail.com



项目仓库



开发者社群

<https://github.com/OpenBMB/PilotDeck>

