



AI编程的“认知税”

当AI开始思考「你」的思考

一个阿斯伯格工程师的2023-2025生存报告



关于我



前·快手 K5A & Web 通道委员

现月之暗面 Tech Staff

人本 & 完型流派咨询师



白天写代码



晚上写人生





目录

CONTENTS

01. 初识 & 恐惧

02. 人类 vs AI

03. 案例 & 思考

04. 护城河

05. 决策 & 行动

06. 未来



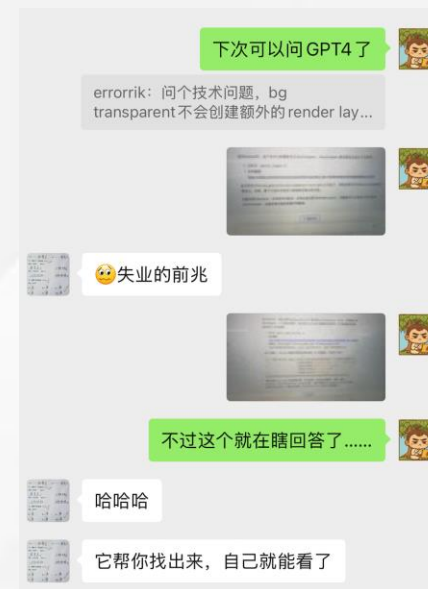


01

初识 & 恐惧



那个僵住的瞬间



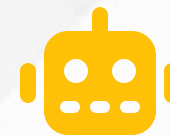
问题

CSS 中 `background: transparent` 会触发 RenderLayer 吗?



你的思维路径

「假设是我写浏览器...」 → 源码验证



GPT-4

完全一致, 连代码位置都相同。

"那一刻, 我以为我的天赋被数字化了"

生存恐惧：当40%被镜像

阿斯伯格的生存模式

🔗 天赋：从 **稀疏信号** 建因果链，还原记忆

⚙️ 优势：**推理**，直奔逻辑



ChatGPT：一面镜子

💡 它也会 **直达核心**

📷 它也会 **快照式记忆**

“但镜子不会告诉你：它只能反射已存在的光”



02

人类 VS AI



AI的创造边界：从AlphaGo到阿基米德浴缸

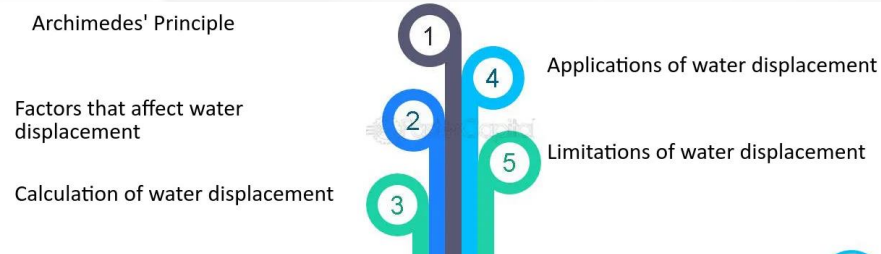
封闭系统中的"路径探寻"



AlphaGo的"神之一手"

规则完备，本质是超人类搜索，在已有轨迹下发现新图式。

开放世界中的"锚点缺失"



阿基米德时刻

物理信号→直觉→啊哈！符号推演 ≠ 具身认知。

“AI缺乏物理锚点，无法从感官体验中突变新图式”

60%的鸿沟：两种认知发展路径

人类模式：动态平衡

图式(Schema)

同化(Assimilation) / 顺应(Accommodation)

新图式(New Schema)

持续反馈循环，单点异常可触发图式重构

AI模式：静态轨迹匹配

经验(Experience)

压缩为轨迹 → 沉淀为静态图式

参数快照(Frozen)

静态知识库，缺失则在高概率轨迹中重复

本质差异：动态平衡系统 vs 静态轨迹匹配系统

3-8分任务：探索地狱的定义

0-3分：纯发散区 (Brainstorming)

例：“给我10种按钮hover动效方案”。AI是创意引擎，认知税低。

3-8分：探索地狱

需大胆假设，小心验证。AI是只会发散、不会提问的实习生。认知税指数级。

8-10分：有限游戏

例：在单测约束下重构模块。AI是高阶副脑，认知税受控。



03

案例 & 思考



案例：Claude Code退出时加日志

背景

同事想在退出时打印日志，修改了`beforeExit`事件。

问题

控制台无输出，但`writeFile`有效。

AI的集体失败

GPT/Claude/Gemini给出多种推测全部无效。

卧槽，react ink 加上去他就懂了，可能

对呀

因为就是缺这个上下文啊

然后你去掉 react ink 就完全不懂



"致命伤：没有一个AI问‘你的console被劫持了吗？’"

15分钟收敛链路

✧ 控制台无输出

`beforeExit`事件没生效？→ `writeFile`有效 → 事件生效了，但console没生效

视觉捕捉：退出时UI清空 → 是不是对console有什么处理？

猜测：console被劫持了 → 直接写stdout → 生效

回忆：console需bind globalThis → 提前存原生console → 终解

时间成本：你 = 15分钟，AI = ∞ （除非喂入Terminal UI上下文）

为什么模型要roll 3-4次才对？

- 1 无Terminal UI上下文 → 发散到Node.js事件循环
- 2 补了UI → 发散到`stdout` vs `console`差异
- 3 补了劫持 → 仍卡在`bind globalThis`的旧trajectory
- 4 强制压缩上下文 → 猜中"Terminal UI劫持console"

核心：模型在 **延续旧轨迹** 而非 **生成新图式**。它无法主动抽象"劫持"概念，只能匹配最接近的轨迹。



04

护城河



泛化能力的鸿沟与哥德尔不完备性

"模型泛化远低于人类" - Ilya Sutskever

人类：小样本泛化

单点反馈 (writeFile有效)

全局假设 (console被劫持)

Schema迁移

模型：轨迹匹配

需要显式Trajectory

无法从稀疏信号泛化

<https://arxiv.org/html/2507.22950v1> - 哥德尔不完备定理在模型领域的适用性研究，证明了封闭系统外的真命题识别困境。

元认知：超越情绪化思考的模式

引用美国心理学家J.H.弗拉维尔的元认知理论

元认知思考模式	情绪化思考模式
监控认知过程	被情绪驱动判断
主动识别"这个问题缺什么信息"	凭感觉"应该没问题"
在AI输出中反向拆解推理链	盲目信任或全盘否定

阿斯伯格的思考链路天然接近元认知，而AI的输出需要元认知来检验。

上下文工程：补全未说出口的话

人做判断需要全面信息

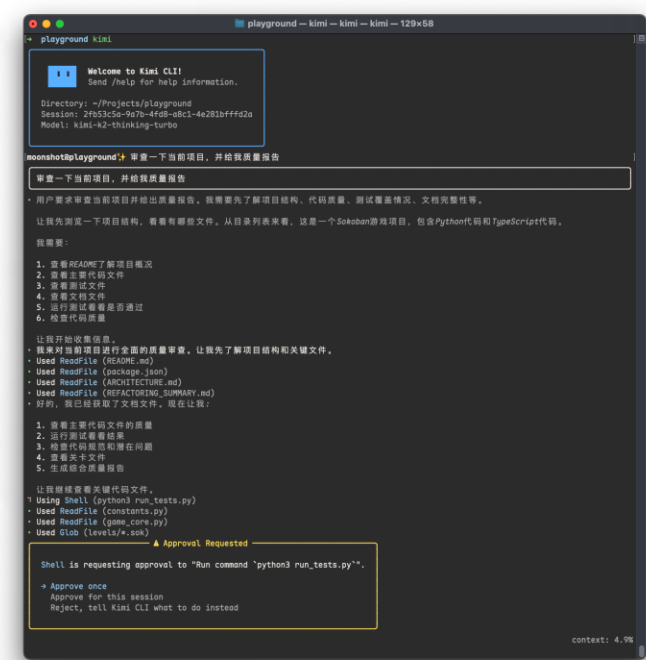
AI也一样

上下文工程的本质

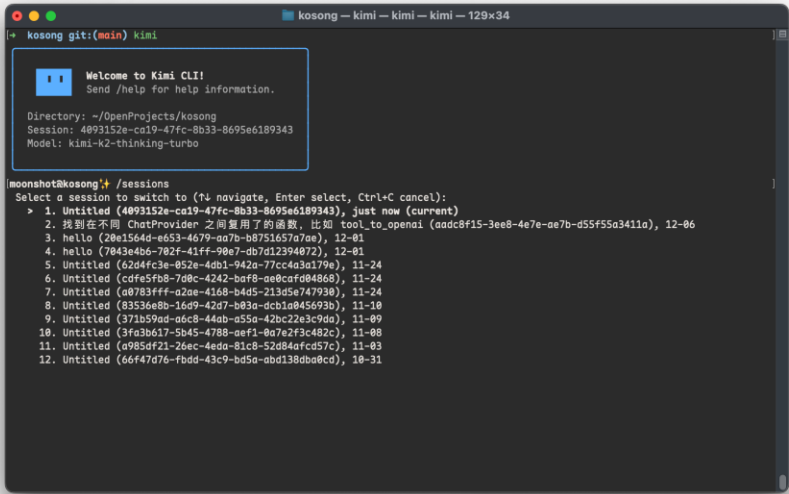
补全那些没说出口但影响决策的约束

显性化上下文，是降低认知税的唯一路径，又称“Context Engineering”

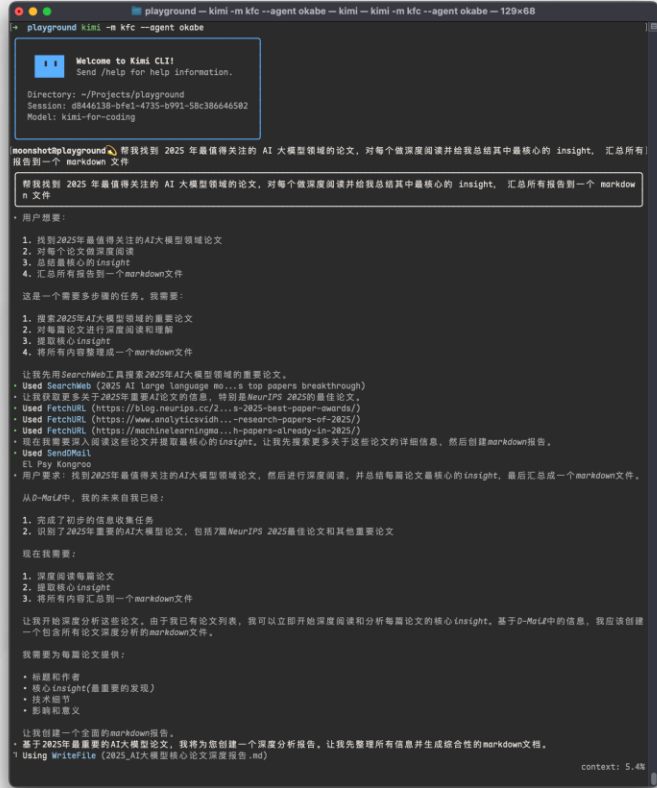
安全调用确认&稳定快速的并行工具



便捷的历史会话管理



稳定可用的 Search Web 与 Fetch URL



Kimi CLI

实验性DMail机制：命运石之门式的主动context管理

价值：保持主模型schema清晰，降低 trajectory 噪音，试图让AI在3-8分任务中学会"收敛"。



方便的二次开发：直接作为 Python 依赖、用 ACP 或 WireOverStdio

原文：https://x.com/wey_gu/status/1996863393823854993

感谢：

- 我们在设计 Decay 算法的时候，参考了 @JarrettYe 的 FSRS 算法和 ACT-R（当我发现 FSRS 算法，作者是推友的时候的心情 😊）
- 我们骄傲地基于 ACP 协议 (@repsiace @frostming90) 和 kimi-cli (@istdrc) 实现 harness agent
- 感谢 Knowledge Mem 社区的各位大佬的容忍、鞭策、支持

社区：

- 我们会把 Knowledge Exchange 浏览器插件开源（这是一个可以随时导出各个 AI 聊天记录为 markdown、或者导入 Mem 的插件）
- 现在有好多大佬维护了支持 Mem 的社区项目，我们会很快上架一个目录站把大家供起来！开始给贡献者朋友们发一些小周边

示例参考：<https://github.com/MoonshotAI/kimi-cli/tree/main/examples>



05

决策 & 行动



认知税自评维度与决策建议

维度	自己深耕	模型协作	完全交给模型	决策逻辑
上下文跨度	依赖链路纵深 > n	Grep / Find 之类能找到	单文件/模块/脚手架	任一维度处于"深耕"档 → 握住收敛与决策权
验证成本	依赖真实环境复现	有 Tool 支持	有单测 / 验不验无所谓	
业务逻辑密度	文档不全 / 口口相传	部分业务逻辑文档化	纯技术逻辑/文档齐全	
认知完备度	你无失败经验 AI过去 < 60% 正确...	你踩过坑，可以指导 AI	大众常识	
协作依赖度	需感受"卡顿/色彩" Or 跨职能	有相关的 Tool 支持	纯逻辑 / 单团队决策	

核心原则：AI探索， 人类决策

核心原则：AI探索， 人类收敛



三大原则，让人类始终掌控**控制权**。



06

未来



工程师的终局价值：成为认知加速器

从 代码输出者

人类当下的不可替代优势：

具身认知：与真实物理世界接触，拥有感官锚点。

小样本泛化：少量试错后就能快速形成新图式(schema)。

到 问题定义者

认知加速：在上下文中实时生成收敛路径。

护城河：拥有失败经验 + 能支付验证税 + 敢剪掉AI输出

AI编程的认知税

本质是你为AI的“缺失的上下文”

支付的 **验证高利贷**

行动清单

今晚

用自评表重排TODO，标记"深耕"任务。

下次

3-8分任务出现时，先画收敛路径，再写prompt。

本周

找一个AI失败案例，反向拆解它的推理链（训练元认知）。

Q&A



公众号：“下班后 开机”

其他材料：

PPT 思考与讨论轨迹：

<https://www.kimi.com/share/19b36790-dcf2-80bb-8000-0000a9f22281>

Claude Code 逆向关键信息（基本过时，权当感兴趣读物）：

<https://hcn4ntcbns3.feishu.cn/docx/XmIcdh7Z5oZW68x3TYjcqB7Ankg>

