Vibe Researching

AI 辅助教育学研究的范式探索

TAO Loop (Agent)

PIE Loop (Human)

Vibe Researching

今天的内容

Part 1

为什么研究范式需要改变?

AI 大模型能力跃迁 · harness engineering · vibe coding 的类比

Part 2

核心框架： TAO-PIE 同构模型

Agent 的 TAO loop · 研究者的 PIE loop · 人机协同的本质

Part 3

两个案例（含演示）

定量： Claude Code + DeepSeek 数据分析
定性： Codex + GBrain 文献挖掘

Part 4

启示与展望

文科研究者的入门路径 · 注意事项 · 范式迁移的本质

PART 1

为什么研究范式需要改变?

AI 大模型 · Harness Engineering · Vibe Coding

两个平行趋势正在改变研究生态

大模型能力的持续跃迁

› GPT-4 → Claude 3 → Claude 4 → …

› 推理、代码、长文本能力每年显著提升

› 面向专业任务的专项能力（coding agents 、reasoning models ） 开始成熟

› 在有限指导下完成复杂多步任务成为可能

Harness Engineering 的普及

› Claude Code 、 Codex 等 agent 工具降低使用门槛

› 研究者无需深度编程即可指挥 agent 执行复杂任务

› 工具链（ harness ） 可以被定制、复用、共享

› 非技术背景研究者正在成为核心用户群

从 Vibe Coding 理解新范式

Vibe Coding （Andrej Karpathy， 2025）：用自然语言描述意图，让 AI 写代码；
你不写每一行，而是
审查、引导、判断——"与 AI 协奏"而不是"独奏"。

传统编程范式

- 研究者逐行编写代码
- 熟悉语法和 API
- 调试每一个报错
- 写完代码才能跑通

= 实现者

Vibe Coding 范式

- 用自然语言描述目标
- AI 生成、执行代码
- 研究者审查输出、调整方向
- 快速迭代，低门槛

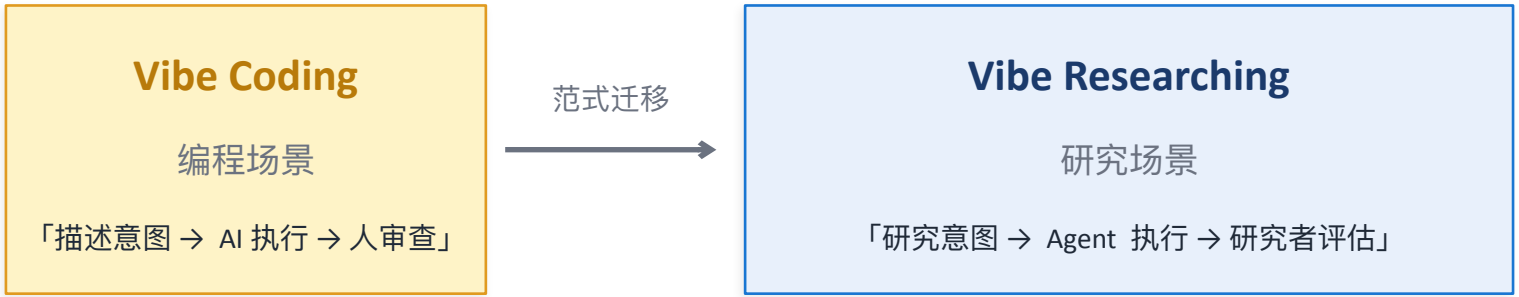
= Orchestrator

同样的范式，移植到研究场景

Vibe Coding 在编程中成立，原因只有一个：

程序员脑子里有那张 " 地图 "—— 知道做什么、怎么验证、什么叫对。

研究者也一样——多年训练铸就的研究本能就是那张地图。
缺的是执行力量，而 AI agent 正好提供这个。



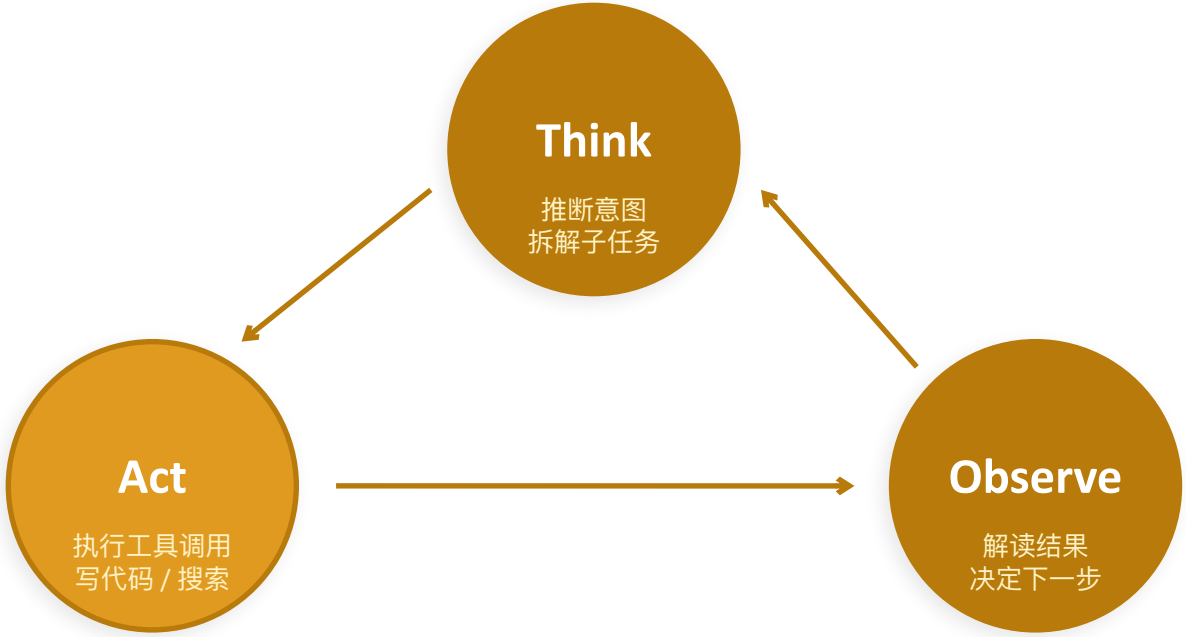
PART 2

TAO–PIE 同构模型

理解人与 Agent 的协同节律

Agent Loop： TAO

（在 Claude Code 、 Codex 等 agent 中可以直接观察到）

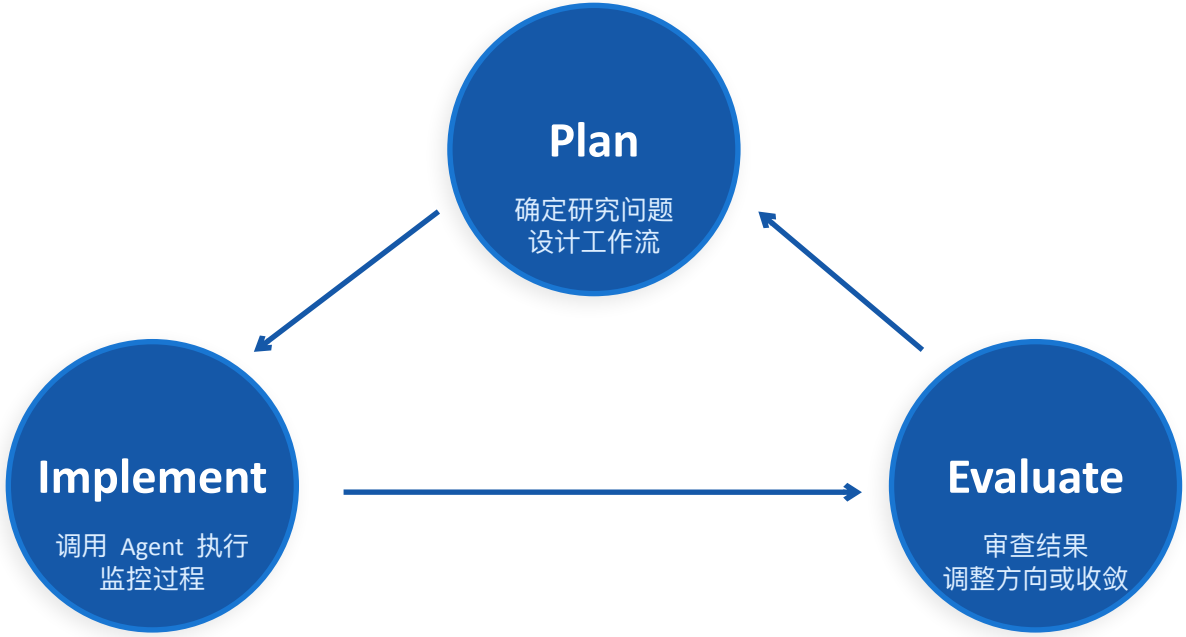


Agent 的工作节律

- 每个 agent step
都是一次 TAO 循环
- 循环自动迭代直到
任务完成或被打断
- 人可以在任意节点介入
- 循环次数可达
数十次甚至上百次

Human Loop： PIE

研究者在每个 session 中天然遵循 PIE 循环



研究者的工作节律

- Plan 不一定写在文件里
——但脑子要有一份“地图”
- Implement = 指挥 agent
而不是亲自执行
- Evaluate 是核心能力
——判断什么叫 " 好结果 "
- 每次 PIE 完成
知识积累一层

TAO–PIE 同构：人与 Agent 越来越像

我的核心发现：两个循环在行为模式上高度同构

TAO（Agent）

Think

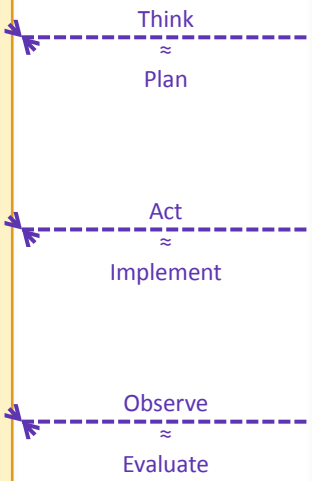
推断下一步行为，调用哪个工具

Act

执行工具，写代码，搜索，调用 API

Observe

解读输出，更新状态，决定继续或停止



PIE（Human）

Plan

设计研究路径，规划 agent 任务

Implement

驱动 agent，监控执行过程

Evaluate

审查研究产出，判断质量，调整方向

研究者角色的迁移

传统研究者

Executor

- 自己动手跑每一个统计
- 自己查找和记录每一条文献
- 分析工具即执行工具
- 时间大量消耗在 " 执行 " 层
- 研究效率受个人技术瓶颈制约

范式
迁移

Vibe Researcher

Orchestrator

- 用自然语言规划研究任务
- Agent 负责执行，人负责判断
- 研究“直觉” 是核心竞争力
- 计划和评估 是不可被 AI 替代的能力
- 研究效率提升，但要始终警惕幻觉率

Vibe Researching：一个工作定义

Vibe Researching 是一种研究范式：研究者以 **研究意图** 驱动 AI agent，通过反复的 **PIE 循环**（Plan→Implement→Evaluate）与 agent 的 **TAO 循环**（Think→Act→Observe）保持同步，
完成传统上需要大量人力的研究任务。

实现 Vibe Researching 需要三个条件同时成立：

研究视野（Map）

研究者头脑中有清晰的研究地图——知道研究要什么、怎么验证

可用工具（Agents and Tools）

Claude Code / Codex 等 agent 能理解指令并执行多步任务

评估能力（Evaluate）

研究者能判断 agent 的输出质量，识别错误，引导收敛

PART 3 | 案例一（定量）

数据分析： Claude Code + DeepSeek

教育事业统计行政人员问卷调查（2026）

案例一：项目全景

1,419	31	7	26
有效样本量	覆盖省市	分析主题	生成图表数

项目背景

对 2026 年全国教育事业统计工作中各级行政部门人员开展问卷调查，围绕队伍结构、经费保障、数字化转型、数据治理等 7 大主题进行系统分析，并与 2022 年数据进行跨年对比。

工具栈

Claude Code	DeepSeek V4	Python (uv)	Plotly / pptx
Agent 框架 / 指挥中枢	代码生成 / 统计分析	数据处理 / 可视化	图表 / 报告生成

案例一：工具链流程图

对应脚本： 01_clean → 02/03/04_stats → 07~09_viz → 11_make_ppt

研究者（PIE）做什么

› Plan： 确定 7 大主题和分析逻辑（脑子里的地图）

› Implement： 自然语言指挥 Claude Code 依序执行脚本

› Evaluate： 审查图表，运行 99_audit.py 核验数字

Agent （TAO）做什么

› Think： 推断脚本逻辑，决定用哪个统计方法

› Act： 调用工具，生成并执行 Python 脚本

› Observe： 读取输出，判断是否符合预期

案例一： Harness Engineering 的体现

项目不是一次性对话——它是一套可重复、可审计的研究工具链

```
education-stat-survey-2026/
├── scripts/
│   ├── 01_clean_and_prepare.py
│   ├── 02_descriptive_stats.py
│   ├── 03_cross_analysis.py
│   ├── 04_year_comparison.py
│   ├── 07-09_visualization.py
│   ├── 11_make_ppt.py
│   └── 99_audit.py  ← 复查
├── data/  （方法论文档）
└── report/  （最终交付）
```

让这个项目 " 像项目 " 而不是 " 像聊天 " 的关键

模块化脚本

每个阶段独立脚本，可单独重跑，便于调试和复查

独立复查脚本（99_audit）

14 项独立核验，与正式分析逻辑完全分离

方法论文档

数据清洗规则、统计方法、跨年对比说明均独立存档

uv 环境管理

pyproject.toml 锁定依赖，可在任意机器复现结果

CLAUDE.md （或 AGENTS.md）

对新手：把研究规则预先写入，减少每次重新说明

案例一：研究产出

这里展示部分图表——关注的是 " 产出是怎么来的 " ， 而不是具体结论

26 张分析图表

- 雷达图（能力自评、数据共享）
- 仪表盘（数字化转型四维）
- 热力图（ Spearman 相关）
- 缺口分析图（培训需求 - 能力）

调查报告（ Word ）

- 7 大主题完整分析
- 跨年对比（ 2022 vs 2026 ）
- 东中西部区域差异
- 行政级别差异分析

汇报 PPT

- 26 张配图幻灯片
- 关键数字自动填充
- 图表直接嵌入（ 300 DPI ）
- 与 Word 报告保持一致

审计报告

- 14 项独立数字核验
- 全部通过（）
- 附清洗日志
- 附跨年对比说明

LIVE DEMO

案例一： education-stat-survey-2026

1. 项目目录结构（scripts/ 模块化）
2. Claude Code 对话中的 TAO 步骤
3. 99_audit.py 复查过程
4. 最终 PPT / Word 报告

案例一：研究体会

研究本能就是最好的 CLAUDE.md

我没有事先写 CLAUDE.md，但事先已建立 7 大主题的报告框架。每次 Plan 时，我知道要让 agent 做什么，知道什么叫 " 完成 " 。
→ 对新手：先详细写明你的研究目标和报告框架，就是最好的 agent 指令。

Agent 会犯错， Evaluate 不能省

脚本会写错变量名，跨年对比会混淆口径，图表会出现计算失误。 99_audit.py 是一个独立核查层——不是信任 agent ， 而是系统性验证。
→ vibe researching 的质量底线由研究者的 Evaluate 能力决定。

效率提升

类似规模的问卷分析，传统方式（ SPSS+ 手工制表 + 手写报告）
通常需要 2-4 人 × 2-4 周。这个项目单人完成，含报告和 PPT 。
→ 运用 claude code： 3 个小时。

PART 3 | 案例二（定性）

文献挖掘： Codex + GBrain

教育思想史自主知识体系建构

案例二：项目全景

11	5,300+	197	561
OCR 文献书目	入库页数（总字符约 400 万）	分块文件（每 30 页 / 块）	知识图谱链接数

研究背景

以 " 教育学自主知识体系 " 为研究问题，系统挖掘《中国教育思想通史》全 8 卷、《郭齐家教育思想史》等 11 本核心文献，追踪 " 自主 "" 教化 "" 经世致用 "" 中体西用 " 等概念从先秦到 1992 年的历史演变，构建可检索、可推理的教育思想史知识图谱。

工具栈

Codex	GBrain (fork)	Mistral OCR	DeepSeek V4
Agent 框架 / 指挥中枢	知识图谱 / RAG 引擎	扫描 PDF 文字识别	think/ 综合推理

案例二：工具链流程图

Codex 充当 Orchestrator，指挥 GBrain 执行 query / think / link / import 等操作

研究者（PIE）做什么

› Plan： 确立研究问题、概念体系、人物线索

› Implement： 通过 Codex 指挥 GBrain 检索、链接、综合

› Evaluate： 核验引用、校正 OCR 错误、判断质量

Agent （TAO）做什么

› Think： 推断检索关键词、综合策略、链接类型

› Act： 执行 gbrain query/think/link 等 CLI 命令

› Observe： 解读检索结果，决定是否继续深挖

LIVE DEMO

案例二： edu-knowledge-base

1. gbrain stats — 查看知识库状态
2. gbrain query " 朱熹教育思想 " — 语义检索
3. gbrain think " 中体西用对近代教育的影响 " — 深度综合
4. 知识图谱浏览： concepts / figures / evidence 层级

案例二： GBrain 知识图谱结构

从 " 一堆扫描书 " 到 " 可推理的知识网络 "

原始资料层

197 个书页分块 · OCR 全文 · 可语义检索

研究工作层

研究问题页 · 证据卡片 · 研究备忘录

概念图谱层

教化 / 经世致用 / 中体西用 / 自主等核心概念页

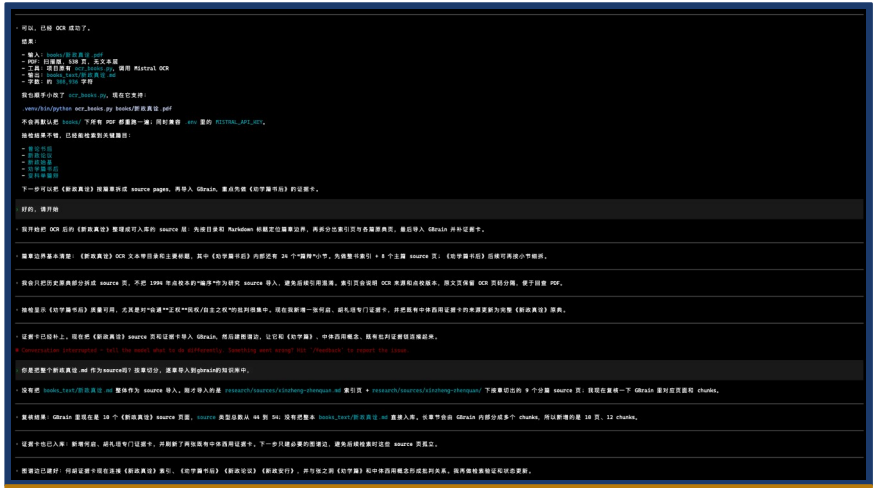
人物时期层

孔子→朱熹→严复→梁启超→陶行知等

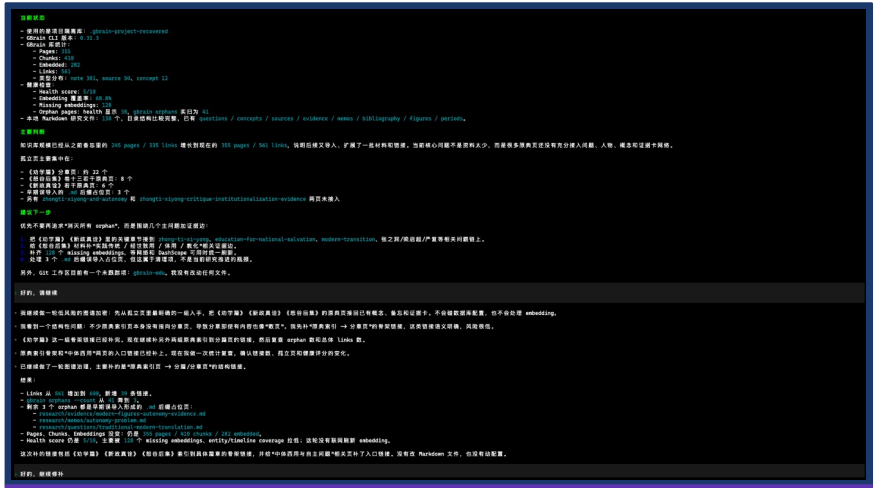
知识库当前状态： 355 pages · 410 chunks · 282 embedded · 561 links

案例二：操作实录

左： Codex Agent 指挥研究任务（TAO loop 实例） | 右： GBrain 知识库状态与目录结构



Codex Agent — Think→Act→Observe 的实际对话过程



GBrain 知识库 — stats / 目录结构 / Observe 输出

TAO 可见：截图 1 中 Codex 的推断→调用→解读步骤；截图 2 中 GBrain stats 是 Observe 的典型输出——研究者据此判断下一步 Plan 。

案例二：研究体会

定性研究同样适用 Vibe Researching

文献挖掘不是 " 把 AI 当搜索引擎 "—— 而是让 agent 帮你在海量文本中发现关联、构建证据链。 gbrain think 的输出是带引用的综合，不是随机生成，每条引用都指向原始书页 slug ，可核查。

研究者的 Evaluate 决定知识图谱的质量

agent 会把不相关的页面链接在一起，会误读 OCR 错误。颜李学派案例中， " 李塨 / 李璋 " 的 OCR 误读被我发现并用 ctext 核验校正。
→ AI 的 " 幻觉 " 在研究中同样存在，评估能力是护城河。

定制 Agent 工具是更深层的 Harness Engineering

fork GBrain 、集成 DeepSeek 、修改 JSON Mode 逻辑——这不是 " 会代码 " ，是研究者对工具链的主权意识。知道工具的边界，才能设计出适合自己研究的 harness 。

两个案例的异与同

维度	案例一（定量）	案例二（定性）
研究方法	定量 / 问卷统计	定性 / 文献挖掘
Agent 工具	Claude Code	Codex
知识库工具	Python 脚本链	GBrain 知识图谱
模型分工	DeepSeek （分析 + 图表）	DeepSeek Pro （综合推理）
数据规模	1,419 样本 / 127 变量	11 本书 / 197 块 / 400 万字
核心产出	Word 报告 + PPT + 图表	知识图谱 + 证据链
Harness 形式	模块化脚本 + 复查脚本	自定义 GBrain fork + CLI

本质相同：都是 Vibe Researching — 研究意图驱动 PIE 循环， agent TAO 执行，研究者 Evaluate 收敛。

PART 4

启示与展望

对文科研究者的建议

什么是研究中的 "Vibe"？

"Vibe" 不是随意，是研究者深度训练形成的方向感与判断力

① 知道研究要什么

能清晰表述研究问题、分析框架、验证标准。
这不是 AI 能替代的——是研究训练的核心产出。

② 知道怎么驱动 Agent

能把研究任务翻译为 agent 可执行的指令。
不需要会写代码，但需要理解任务的分解逻辑。

③ 知道什么叫好结果

能识别 agent 输出中的错误、偏差、过拟合。
这是 Evaluate 能力——文科研究者天然具备。

④ 知道何时介入循环

agent 在迭代时，研究者知道在哪个节点停下来检查。
过度放手或过度干预都会降低研究质量。

文科研究者的入门路径

你不需要会写代码——你需要会表达研究意图

Step 1

先做好你的研究设计

在开始使用任何 agent 之前，把研究问题、理论框架、分析逻辑与验证标准写清楚。
这就是你的 " 地图 " ，也是你将给 agent 的最重要指令。

Step 2

从具体任务开始，不要从 " 学工具 " 开始

选一个你真正在做的研究任务：整理文献、清洗数据、分类访谈。
用 Claude （或 Cowork ）指挥 agent 完成它，在过程中学习。

Step 3

建立你的 Harness

把成功的任务流程保存下来：脚本、提示词模板、 CLAUDE.md 。
下次同类研究可以直接复用——这就是你的个人研究工具链。

Step 4

强化 Evaluate 能力

系统学习如何核验 AI 输出：引用追溯、数字核查、逻辑一致性检验。
Evaluate 是 vibe researching 中唯一不可外包的环节。

注意事项： Evaluate 不能省

Vibe Researching 降低了执行门槛，但没有降低研究责任

⚠ Agent 会犯错，且有时犯得很自信

数字算错、引用张冠李戴、逻辑跳跃——这些都真实发生过。
案例一：跨年对比中 agent 混淆了两种问卷口径，总报告动用了其他 AI 模型（ claude 、 codex 等）进行了两次检校。
案例二： OCR “ 李璋”被误接入，需要人工校勘原典。需要不断对概念解读进行引导、反思和修正。

⚠ " 看起来对 " 不等于 " 是对的 "

AI 生成的内容往往格式优美、表述流畅，很容易让研究者放松警惕。
建立系统性的核验机制（如 99_audit.py / ctext 核验）是护城河。

研究本能是不可外包的

Plan 阶段如果研究者没有清晰的研究地图， agent 的迭代只会越跑越偏。
Vibe Researching 放大了研究者的能力——包括放大他们的模糊。

学术规范与伦理考量

透明度与可溯源性

AI 辅助的研究过程应在方法论部分明确说明：使用了什么工具、在哪些环节使用、人的判断如何介入。
这不是 " 承认依赖 AI" ，而是现代研究的方法规范。

知识产权与数据安全

向 AI 输入的研究数据（尤其是保密数据、受试者信息）需要符合机构数据管理规定。
本地部署模型（如 DeepSeek API ）和商业 API 的数据处理条款不同，需提前了解。

Evaluate 是学术诚信的底线

" 我让 AI 帮我做的 " 不是错误答案的借口。研究者为最终输出负责。
建立可审计的研究流程（脚本存档、复查报告、引用核验）是研究者的责任，也是保护自己的方式。

核心结论

1 AI agent 的 TAO 循环与研究者的 PIE 循环在行为模式上高度同构

2 这种同构使 Vibe Researching 成为可能——研究意图驱动 agent 执行

3 定量与定性研究的表面差异，掩盖不了范式层面的本质相同

4 研究本能（Map）和评估能力（Evaluate）是研究者的核心竞争力，不可外包

5 这不是终点——这是一个正在形成的研究范式，我们都处于起点

Q & A | 欢迎讨论： hongyuatbj