**0907-MBA分享**

**MBA社群AI分享完整方案**

**开场调查问卷设计**

**现场快速调研（举手投票）**

1. **使用频率**：每天用AI的举手 → 每周用的 → 偶尔用的 → 没用过的
2. **使用深度**：只用ChatGPT聊天的 → 会写复杂Prompt的 → 用过API/插件的 → 自己开发过AI应用的
3. **使用场景**：工作提效 → 学习研究 → 创意娱乐 → 商业决策
4. **认知边界**：觉得AI很神秘的 → 了解基本原理的 → 关注商业机会的

"根据调研结果，我今天会重点讲解适合大家层次的内容，让每个人都能带走3个立即可用的AI技巧。"

**第一部分：AI深度使用者的进阶秘籍**

**🌟 AI 核心概念小词典（专业解释 + 通俗解释版）**

**🔢 Token**

* **专业解释**：Token 是大语言模型处理文本的最小单元，可以是一个字、一个词、或一个子词片段。模型在训练和推理时，会把输入文本拆分为一连串的 tokens，再逐个预测下一个 token。它同时也是计费和计算的基本单位。



**🔥 Temperature**

* **专业解释**：生成模型在采样下一个 token（词）的概率分布时，Temperature 控制分布的平滑程度。数值低时（接近 0）模型更倾向选择概率最高的词；数值高时（接近 1 甚至 >1）则会增加随机性，让低概率词也更容易被选中。
* **通俗解释**：也就是AI的"创意度"，0=严谨，1=天马行空

**🎲 Top-K**

* **专业解释**：在生成时，只考虑概率最高的 **前 K 个候选词**，其余词的概率直接丢弃，再在这 K 个中按概率重新采样。
* **通俗解释**：想象你在点餐：菜单上 100 道菜，AI 只看前 K 道热门菜，然后随机点一道。
* K 越小 → 越保守（只吃最火爆的菜）。
* K 越大 → 越有探索性（敢点冷门菜）。

**🌀 幻觉（Hallucination）**

* **专业解释**：在大语言模型生成过程中，由于缺乏足够的训练样本或推理链路错误，模型可能输出与事实不符、逻辑错误或凭空捏造的信息。这类现象被称为 **幻觉**。它是当前生成式 AI 的主要挑战之一，尤其在需要高准确性的场景（医疗、法规、科研）中影响较大。
* **通俗解释**：AI 有时候会“一本正经地胡说八道”。捏造一条不存在的法规、论文和“事实”。是需要人为干预的重灾区。

**✍️ Prompt**

* **专业解释**：指令提示，输入给大语言模型的自然语言或结构化文本，用来引导模型输出特定风格、格式或内容的结果。
* **通俗解释**：就是你和 AI 聊天的“话术”。说清楚了，AI 才能干活。比如：“帮我写个情书，200 字，押韵。”

**💻 Vibe Coding**

* **专业解释**：一种轻量化、实验性强的编程方式，强调快速原型、氛围感和与 AI 协作的创造性开发过程。
* **通俗解释**：氛围感编程。用prompt调用AI输出编程的效果，但是不一定要人实际做coding的内容。

**📑 Content Engineering（上下文工程）**

* **专业解释**：在 AI 提示设计中，除了 Prompt，还要补充背景资料、任务需求、输入输出示例等上下文，帮助模型更准确理解目标。
* **通俗解释**：就是“喂料”，先给 AI 背景知识（比如需求文档、产品介绍）

**🧠 RAG（Retrieval-Augmented Generation）**

* **专业解释**：一种“检索增强生成”方法。先从外部知识库检索相关文档，再把这些文档作为上下文输入模型，以减少幻觉、增加可追溯性。
* **通俗解释**：就像给 AI 装上一个“随身百科全书”。它不会只靠记忆瞎编，而是先查资料再回答。

**🎯 Fine-Tuning（模型微调）**

* **专业解释**：在预训练模型的基础上，用精心准备的小规模领域数据（训练集/验证集）进行二次训练，使模型参数更贴近特定任务或行业需求。
* **通俗解释**：大模型像个通才学生，啥都懂点；微调就像开小灶补课，让它在一个领域（比如医疗、法律、金融）变成专家。

👌 MCP 我也帮你加上，延续之前的「专业解释 + 通俗解释」风格：

**🛠️ MCP（Model Context Protocol）**

* **专业解释**：MCP 是 OpenAI 提出的 **模型上下文协议**，用于标准化 AI 模型和外部工具/数据源的交互方式。通过 MCP，开发者可以把数据库、API、文件系统等“接到”模型的上下文里，让模型能像调用插件一样使用这些资源。
* **通俗解释**：就像给 AI 插 USB 外设：
* 想查资料 → 插个“数据库 U 盘”
* 想调 GitHub → 插个“代码库 U 盘”
* 想读文件 → 插个“文件夹 U 盘”  
   AI 不再是闭门造车，而是能随时访问你接上的工具。

|  |
| --- |
| **Takeaway**：学会这10个词，你就能和技术团队/投资人专业对话 |

**1.2 工具矩阵，覆盖工作全场景**

**分类展示**：

* **基础对话**：ChatBox（多模型切换，本地化）

1. **Chat GPT**

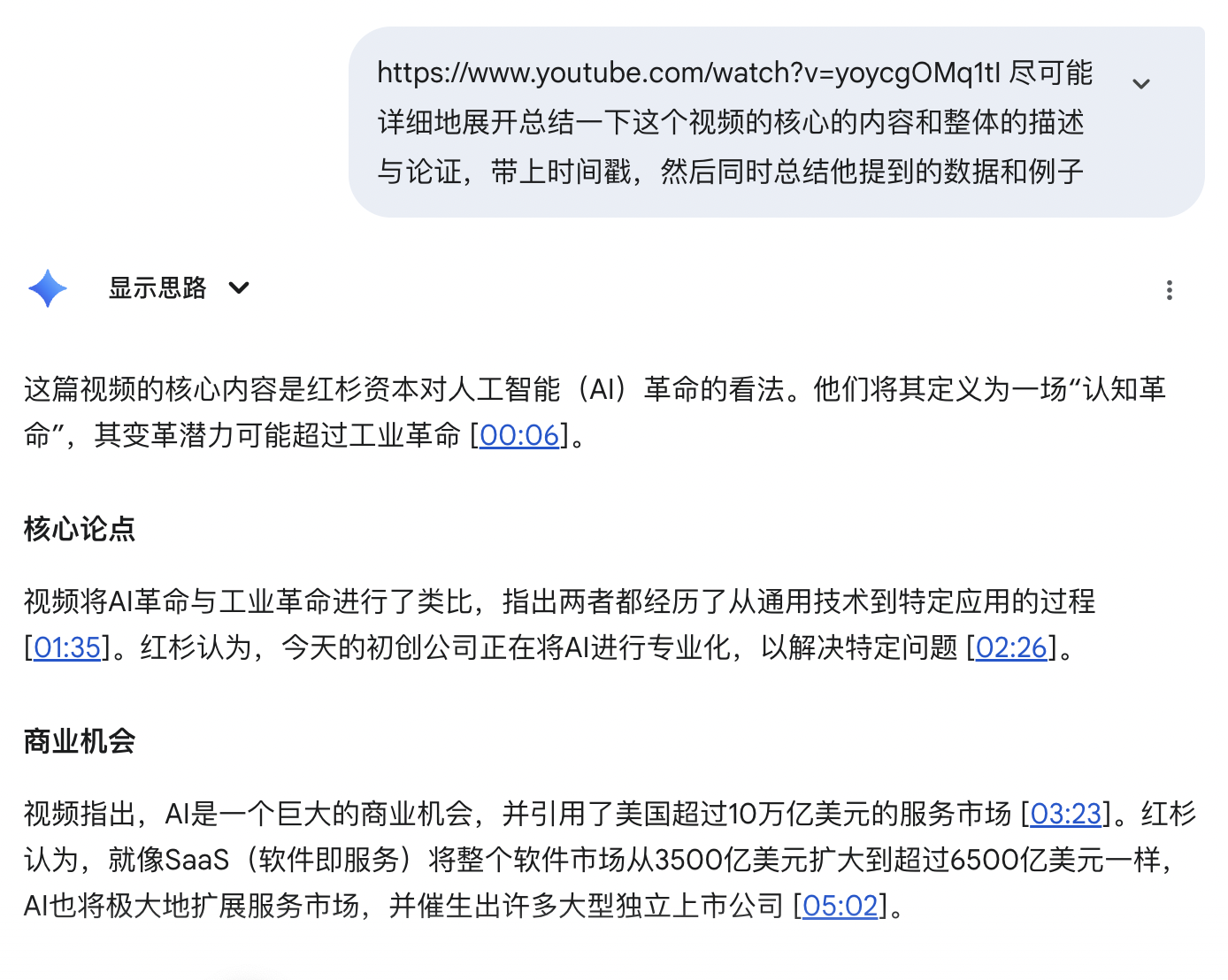
Memory Seeting（感知人为对话偏好和项目进度、日常所有问题都会先问一遍它，也方便后期回忆调用）



1. Claude Code

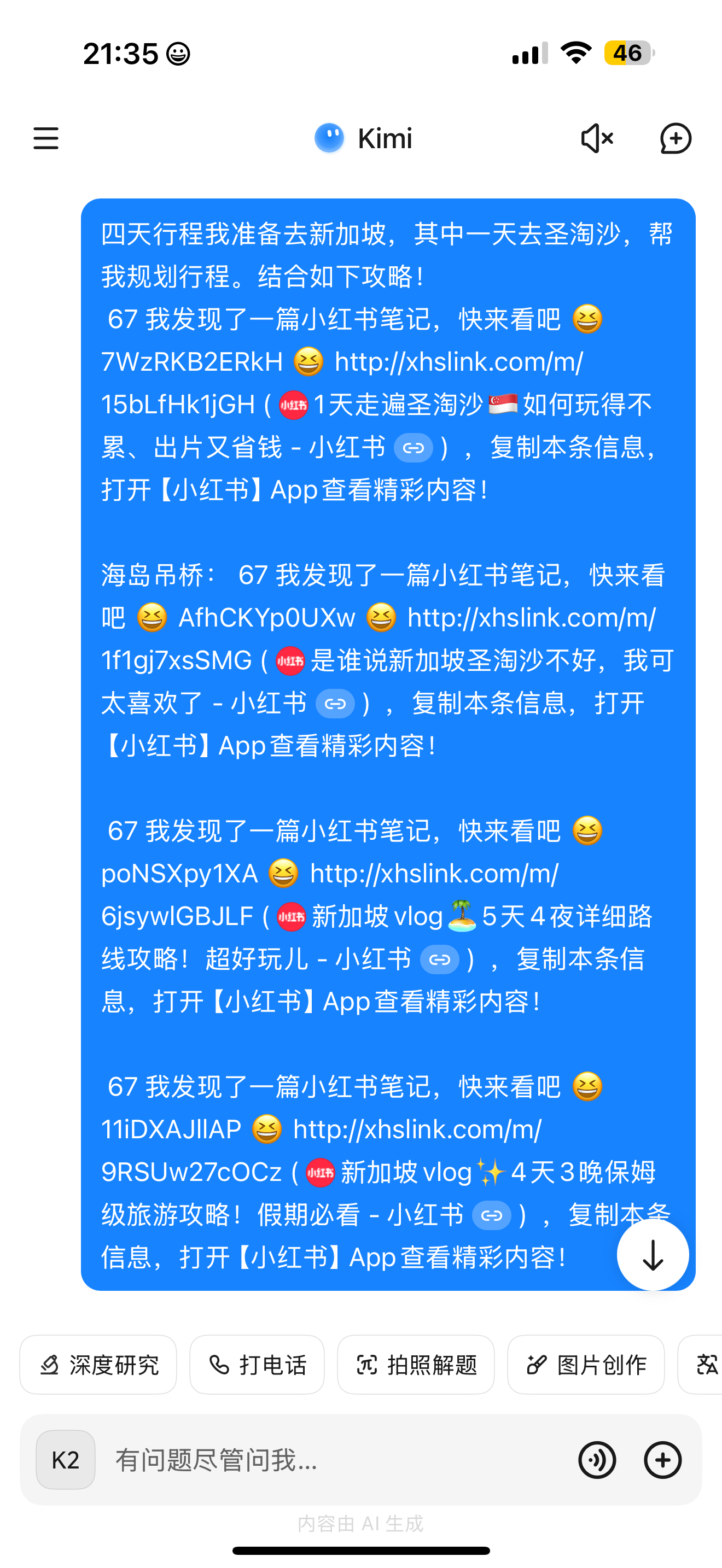
Prof级别的报告撰写+Coding Manager

1. Gemini



1. Kimi

扒链接



💡：https://github.com/Jianan-Huang0609/KnowVis

1. DeepSeek
2. Qwen

* **内容创作**：
* NanoBanana、RoboNeo、Snipd
* **设计工作**：Lovable V0 Bolt Figma（低代码界面设计）

**1.3 AI 可以是最懂你的Mentor**

**实用场景**：

因为大量的交互，AI记住你的工作背景、偏好、专业领域、目标、能力乃至性格抱负。

* 效率工具+知识学习工具
* 每周工作复盘，AI提供改进建议
* 个人学习计划，AI推荐资源和路径
* 决策支持，基于你的价值观和目标
* 情感复盘，正式自己的负面情绪，并行动消解。

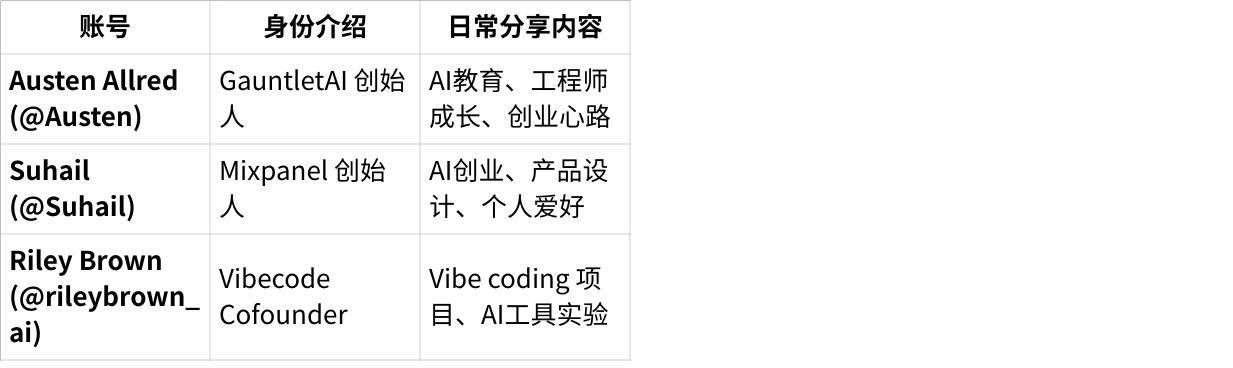
**1.4 信息源升级，构建AI雷达**

🔬 AI研究机构 & 科学家



**点击图片可查看完整电子表格**

🚀 AI创业者 / 产品实验



**点击图片可查看完整电子表格**

💰 投资人 / VC / 孵化



**点击图片可查看完整电子表格**

🛠️ 产品人 / 创业者



**点击图片可查看完整电子表格**

📰 媒体 / 播客



**点击图片可查看完整电子表格**

🌱 社区 / 创业氛围



**点击图片可查看完整电子表格**

**第二部分：从工具使用到产品开发**

**2.1 为什么要自己开发AI应用？**

**商业逻辑**：

* 通用工具 → 行业定制 → 商业机会
* 成本优势：API调用比雇人便宜90%
* 速度优势：想法到原型，从6个月→48小时

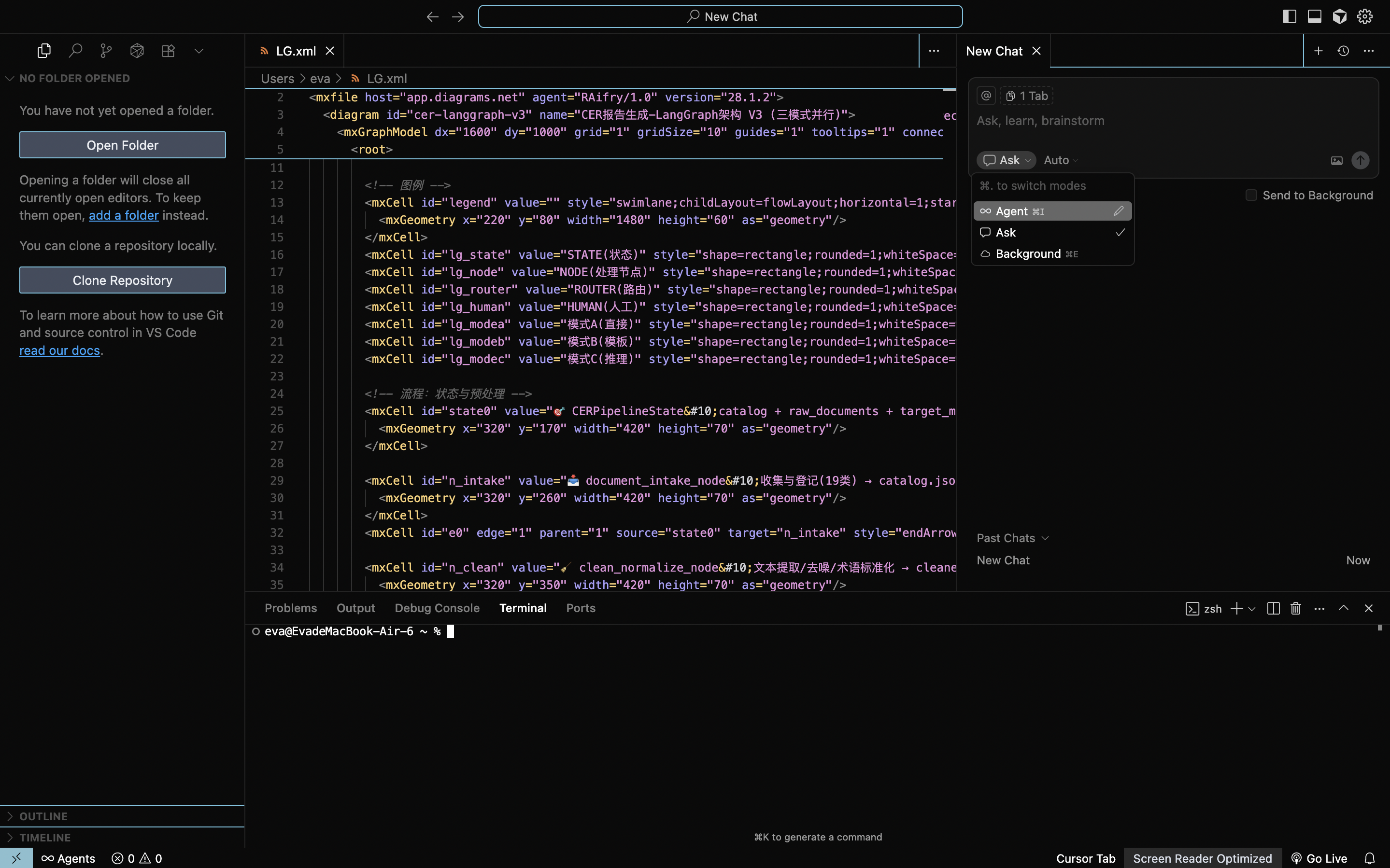
**类比**：就像从用Excel→写宏→开发专业软件

**个人角度**：

* AI+灵感+品味+痛点=最完美的个人表达
* 成为创造者，把AI设计变成生活艺术
* 成为社交名片，更好地链接同频社群

**2.2 IDE选择：Cursor深度解析（3分钟）**

* **谁适合用**：有编程需求但不想深入学代码的人
* **能干什么**：自动化脚本、静态页面、集成网站、插件工具、小程序、App等；
* **怎么用**：自然语言描述需求 → AI生成代码 → 迭代优化
* **进阶使用**：Rules设定+MCP工具箱+周期开发意识+上下文工程



**2.3 观点结合**

**IDE展会观察（3分钟）**

**链接🔗：**https://github.com/Jianan-Huang0609/Cursor\_MeetsUp

* **蓬勃又野蛮生长的VB圈**：开发者数量指数级上升，随之而来的IDE需求、VB Hunt需求、产、展、销闭环还没跑通；
* **个人机会**：信息差+执行力+真实需求验证+个人品味+影响力

**现场分享**：展会上最让我印象深刻的3个产品

**WAIC展会观察**

**链接🔗：**https://jianan-huang0609.github.io/WAIC\_Site/

* **多模态AI**：文本+图像+语音一体化
* **Agent生态**：AI助手从单一功能向复合能力发展
* **行业落地**：从通用工具向垂直应用深化

**现场分享**：展会上最让我印象深刻的3个产品

**2.4 项目展示（3分钟）**

**项目展示🔗**：

1. **RAifry医疗注册助手**：B2C Tool+ B2B SaaS，提效70% https://github.com/Jianan-Huang0609/PoC\_Structure
2. **可视化笔记：**个人需求驱动的产品探索 https://github.com/Jianan-Huang0609/KnowVis
3. **VB Assistant:** Vibe Coding helps Vibe Coder https://jianan-huang0609.github.io/VB\_Assistant/

**2.5开放讨论：**

* 你的行业有哪些AI化机会？
* 对我的项目有什么建议？
* 有投资或合作意向吗？

**第三部分：学习路径**

**3.1 学习路径**

|  |
| --- |
| 理论打底：   * 首先了解深度学习-deep learning（前馈神经网络、梯度传播）； * 其次理解语言传播的演进（LLM-RNN-LSTM-Transformer自注意力、多头注意力的机制、位置编码、残差连接、前馈网络）； * 最后是模型训练（Layer Norm KV-Cache）；分词（BPE、WordPiece、SentencePiece）   模型认知：   * BERT、GPT、LAMA模型本身的结构差异和设计理念； * （目前行业都在用的decoder-only，就具备着推理能力高、部署简单、延迟低）   训练和对齐   * 学习如何采集数据、清洗数据、构造好的结构化数据； * 训练阶段了解学习率调度、优化器选择、混合精度训练、梯度累积； * 微调阶段：LoRA、QLoRA、Prompt Tuning;目标是能完整执行一轮的SFT * 对齐阶段：RLHF DPO   工程化（能部署并跑通，并且得到比较好的效率）   * 掌握模型压缩（蒸馏、量化、剪枝） ； * 学会分布式训练（数据并行、模型并行、流水线并行、ZeRo技术优化方式） * 推理部署：KV缓存优化、动态batch调整、服务化封装、容灾体系； * 前沿应用：构建RAG系统、打造Agent系统、拓展多模态能力（如Clip、Blip处理图文的能力） * 评估优化：内容理解、幻觉识别、有害内容的过滤、隐私保护；合规方面（模型的版本管理机制、接入用户反馈等）   推荐：Youtube、论文、博客综合学习 |

**3.2 开发路径**

|  |
| --- |
| 明白啦 🙌 你是想分享 **V1–V5 技术演进 & 资源**，但避免让别人对你的实际产品规划一眼看穿。换句话说，要做一份 **去产品化 / 去场景化** 的「技术路径分享」，更像是 **行业趋势盘点 + 学习路线梳理**。我帮你改写成一份 **脱敏版**：  **🌐 AI 系统能力演进 V1–V5 （技术趋势 & 学习资源）**  这里的 V1–V5 并非具体产品路线，而是一个通用的 **AI 系统演进框架**。它展示了从「简单可行性验证」到「复杂多智能体协作」再到「自学习」的逐步升级路径。  **V1 — 框架生成 & 单点验证**  **目标**：验证可行性，证明 AI 能够生成文档/任务框架，并在单个模块内完成一致性或正确性检查。 **技术路径**：   * API 调用（GPT、Gemini 等大模型） * Prompt 链 & 规则匹配 * OCR 支持（文本提取） * 多 Prompt 分链路管理   **参考资源**：   * **RapidOCR** — 高效 OCR 工具 * **Bedrock Prompt Chaining 示例** — 多链路管理范例   **V2 — 全流程扩展 & 轻量 RAG**  **目标**：扩展到更复杂的任务范围，引入检索增强生成（RAG），提升检索与生成的准确性。 **技术路径**：   * 文本清洗与标准化、切块向量化 * 稀疏检索（BM25）+ 向量检索（FAISS/Chroma/Pinecone）混合 * ReRank（Cross-Encoder / Cohere / BGE） * 自动评估（RAGTruth 等）   **参考资源**：   * **RAGFlow** — 开源 RAG 引擎 * **RAGTruth** — 事实标注数据集 * **ARES** — 自动评估框架   **V3 — Planner + 小模型优化**  **目标**：实现任务拆解、模块化执行；在特定子任务中使用轻量小模型进行优化。 **技术路径**：   * Planner / 任务图（LangGraph 等） * 子任务级 LoRA / PEFT 微调 * 示例驱动优化、数据对优化   **参考资源**：   * **LangGraph** — 有状态 Agent 编排框架 * **Open Deep Research** — 多 Agent 深度研究框架 * **DeepAgents** — 自主 Agent 框架 * **MetaGPT** — 多角色协作框架 * **LLM Agent Papers** — 论文合集   **V4 — 多 Agent Pipeline**  **目标**：形成可用的多智能体流水线，支持跨模块协作、自动报告生成等复杂工作流。 **技术路径**：   * 多 Agent 协作：Retriever / Extractor / Checker / Composer * 审核 Agent（Auditor）+ 人机反馈闭环 * 子 Agent 与消息流转机制   **参考资源**：   * **deepagents** — 子代理与虚拟文件系统 * **MetaGPT** — 虚拟团队多 Agent 协作 * **open\_deep\_research** — 自学习雏形   **V5 — RLHF + 自学习**  **目标**：模拟专家角色，具备自学习与迭代优化能力。 **技术路径**：   * RLHF（强化学习 + 人类反馈） * Learner + Auditor 闭环优化 * 知识图谱与奖励模型结合   **参考资源**：   * **awesome-RLHF** — RLHF 学习资源 * **awesome-foundation-agents** — 多 Agent 系统资料 * **LLM Agent Papers** — Agent 必读论文合集   **📚 学习建议**   1. **V1–V2** 阶段：适合入门者，重点掌握 Prompt 设计、OCR、RAG 基础。 2. **V3–V4** 阶段：面向进阶者，重点研究任务规划、多 Agent 框架。 3. **V5** 阶段：属于前沿探索，需要结合 RLHF、奖励建模和知识图谱。 |