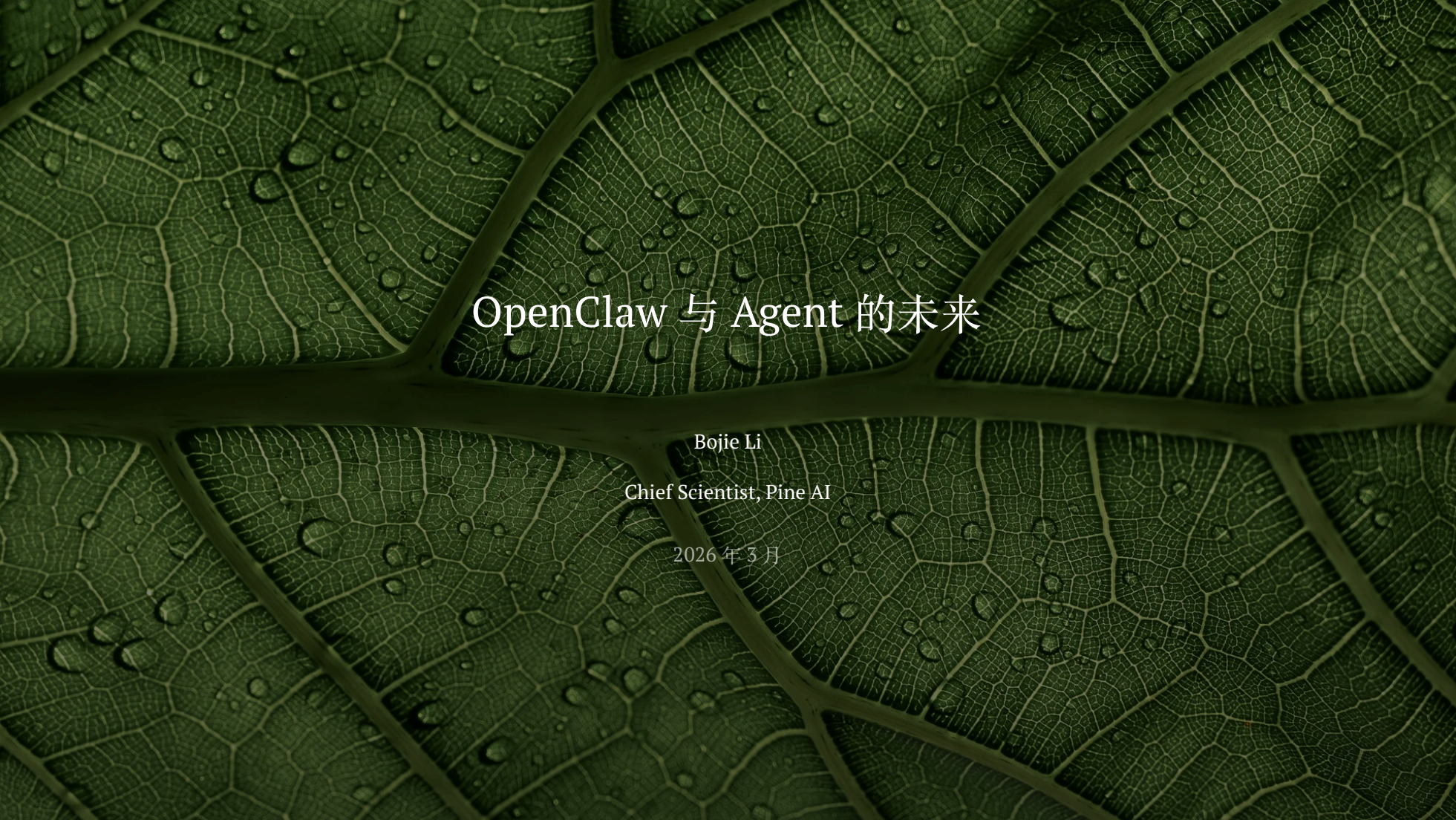

OpenClaw 与 Agent 的未来

Bojie Li

Chief Scientist, Pine AI

2026 年 3 月

三个台阶：Chatbot → 专用 Agent → 通用 Agent

Chatbot

只会说，不会做

ChatGPT 网页版——你输入问题，它输出文字。

有知识、有思考，但没有行动能力。

专用 Agent

能动手，但只擅长一件事

Cursor / Claude Code——能读写文件、执行代码，但你必须坐在电脑前指挥它。

通用 Agent

替你操控整台电脑

OpenClaw / Manus——Deep Research + Computer Use + Coding 三种能力合一。手机发消息就能指挥。

反共识观点：通用 Agent 的核心不是 Computer Use——而是 **Coding Agent**。所有高效内容生成最终都通过代码实现：PPT = ZIP + XML 代码，Word = JS 代码生成，比 GUI 操作快几个数量级。

大模型是新的操作系统

传统 **OS** 上最重要的"应用"只有一个——大模型。所有其他应用 = 基于模型上下文的 **Agent**。

新软件范式

```
硬件层
  ↓
传统操作系统（Linux/macOS/Windows）
  ↓
大模型（新的"操作系统"抽象层）
  ↓
各种 Agent（每个 = 一个"应用"）
  ↓
用户（通过自然语言交互）
```

极端但可能的未来：操作系统不再需要 GUI——只需一个 Agent + 一个终端 = 完整的操作系统。也有公司在尝试让 AI 动态生成图形界面。

推理成本的惊人下降

ChatGPT 发布至今（~3年），同等智力水平推理成本下降 **~100 倍**。每约半年成本缩减一半。

再过 3 年：普通手机有可能拥有当前模型水平的本地算力。

个人计算的回归：钟摆效应

端侧优势	说明
延迟低	本地推理无需网络往返
不需联网	随时随地可用
边际成本趋零	设备算力不用白不用
隐私保密	数据完全不出本地

模型算力有望变得像**水和电**一样——这将根本性改变 AI 的部署模式。Infra 的边界正在从 OS 转向 **LLM Context** 管理，这才是下一个 UNIX 可能诞生的地方。

OpenClaw 为什么重要?

不是一个产品——而是定义了通用 **Agent** 的形态

三大设计启发

丰富的 **Connector**：21 个聊天渠道——Telegram、iMessage、WhatsApp。不用坐在电脑前，手机发条消息就能指挥 Agent。

No Session 设计：像跟真人聊天一样，不用想"在哪个对话里问"。Agent 记得所有历史交互——ChatGPT 和 Cursor 做不到。

Skills + CLI 生态：社区贡献即插即用的能力扩展。Skills 本质上是动态注入的 **System Prompt**——让 Agent 知道有什么工具、怎么用。

OpenClaw 之于 Agent = Linux 之于操作系统

定义了范式和技术方向，但大多数终端用户会用商业化产品（类似 Android、iOS 基于 Linux 但面向大众）。

Agent 的本质公式

Agent = 模型 + 上下文 + 工具/动作空间

根本局限：Personal Assistant 范式——one person, multiple agents。不支持多用户协作，天然不是企业级产品。99% 的人最终会用企业级 Agent，就像 99% 的人用银行存钱而非加密钱包。

OpenClaw 的记忆架构：为什么用 Markdown 而非数据库？

反直觉但极其有效的选择——透明、可编辑、Git 可追溯

结构化文件存储

- `MEMORY.md` — 核心事实和用户偏好（长期记忆）
- `memory/2026-03-21.md` — 按日期归档的每日交互日志
- `AGENTS.md` — Agent 对自身能力的反思

为什么 Markdown 比向量数据库更好？

透明可编辑：直接打开文件看 AI 记住了什么，记错了直接删那行——向量数据库做不到。

时间线性：按日期归档，AI 知道"昨天聊了什么"——向量检索往往丢失时间上下文。

Git 版本控制：每次记忆修改都可追溯和回滚——这是向量数据库完全不具备的能力。

混合搜索：两全其美

虽然存储用 Markdown，但检索用了向量搜索 + 关键词搜索的组合：

```
finalScore = 0.7 × vectorScore
           + 0.3 × textScore
```

语义匹配（余弦相似度）和关键词匹配（BM25）互补，权重 7:3。

上下文压缩：无限对话 + 永不丢失

当对话太长快要超出模型的上下文窗口时：

1. Agent 自动总结对话要点
2. 关键信息写入 `MEMORY.md`（永久记忆）
3. 详细记录归档到 `memory/YYYY-MM-DD.md`

效果：无限长对话 + 永不丢失关键信息。这套看似"低技术"的方案，在实践中比精心设计的向量数据库方案更可靠。

反共识一：AI 软件开发，从劳动密集型到创意密集型

协作本身将变得不必要

一个人 + AI 的惊人产出

指标	数据
日均代码产出	**4-5 万行**
日均 Token 消耗	**18 亿**
单日最高 Git Commit	**1,374 次**

Brooks《人月神话》的核心洞察：延期的项目加人只会更慢，因为沟通成本指数增长。

AI 消除了这个瓶颈——不是因为协调变好了，而是协调变得不必要了。一个有想法的人 + AI，信息损耗几乎为零。

未来有价值的三种人

电影导演型（0→1 创造）：定义产品愿景。瓶颈是创造性判断力。

城市规划师型（1→100 架构）：管理系统规模化。瓶颈是架构判断力。

F1 赛车手型（极限研究）：推进 AI 前沿。瓶颈是科学洞察力。

共同点：核心能力都不是"写代码"，而是**判断力**。

反共识二：Agent 是比人类大十倍的用户群

GUI 已死，协议万岁

GUI 是人类认知缺陷的补丁

- Figma 的精美、Notion 的简洁——本质是让带宽极度有限的人类勉强完成任务
- 这是一种 **"界面税"** ——为人类认知缺陷支付的补偿成本
- 以用户体验为护城河的软件公司，都是注意力公司的变体

Agent 推翻了整个前提

Agent 不需要 GUI，它需要：

- **信息获取**：高密度数据
- **执行权限**：CLI / API / MCP
- **可信度**：身份与信用背书
- **算力**：持续推理资源

软件竞争方向的反转

旧逻辑	新逻辑
建封闭空间，让用户走进来	把自己暴露出去
用体验把用户留住	站在 Agent 路径上
产品竞争靠好看	**协议竞争靠成为默认标准**

协议即软件（Software as Protocols）

2026 年初，Google Workspace、Salesforce、Atlassian 等主流软件公司相继支持 CLI 和 MCP。这些公司过去花了数十亿美元打磨 GUI，**现在在主动绕开它。**

反共识三：Context 才是人类的护城河

来自 **OpenAI Jiayi Weng** 的洞察：人和模型一样，最重要的是 **Context**

关于 Context 的重要性

"我在 **OpenAI** 的工作也没有那么难，不需要很高的智商。如果换一个人，有我所有的 **context**，也是能干的。"

AI 短期内无法取代人，最大原因也是 context——它在公司里能访问到的 context 远低于一个人类员工。

团队合作最大的问题是 **context** 不一致。 人类组织的千古难题就是难以保持 context sharing 的一致性。

如果模型有无限 **context**，最大的应用场景就是做 **CEO**——解决组织中 context sharing 不一致的千古难题。

AI 能取代的 vs 不能取代的

"第一个被 **AI** 替代的是 **researcher**，然后是 **infra engineer**，最难替代的是 **sales**。" 需要说服真人买单的能力，AI 目前做不到。

OpenAI 内部人会过高估计 **AI** 的影响。 o1 出来时，预估一两年能帮清理 infra 屎山——直到今天仍然不行。技术对世界的改变是循序渐进的。

AI 能取代的，是没有自己想法、只执行上级指令的人。 拥有隐性知识、历史原因、未曾表达的想法的人，才是不可替代的。人类的核心竞争力不是写代码，而是判断力和对 **context** 的掌握。

反共识四：莫拉维克悖论

对真人来说很难的写代码，AI 干得很快；但对真人来说很简单的操作 GUI，AI 搞不定

爽的一半：提需求 + 架构评审

带着几个 Coding Agent 干活，就像在大厂做技术专家——往会议室一坐，听几个员工汇报进展，然后说"你这样设计架构有 abc 问题，应该 blabla 这样做"。

- 架构设计好了，代码只要抽查关键点
- SOTA 模型知识面比我广、纯智力比我强，但有时搞了几个小时反复修补，我还能教它设计架构
- 这种只用动嘴皮子的工作方式，有种智力上的快感

跟以前在大厂带人的区别：以前每周开一次会，现在一个小时就能过第二版。

不爽的一半：给 Agent 做秘书和测试

当秘书：经常要申请第三方服务的测试账号，没有哪个 Coding Agent 能自主用本地浏览器和手机完成注册配置。Agent 干到一半说"请你打开这个网站，申请一个 API key"，我就花 20 分钟搞定再给它。

当测试：Agent 说任务干完了，一试发现按钮点了直接报错。Agent 很难自主完成 UI 全流程测试。

本质：写代码（人类觉得难）AI 干得飞快，操作 GUI（人类觉得简单）AI 却搞不定。等 Computer Use 达到人类级准确率和延迟，带 Agent 就像带人一样了。

Moltbook：150 万 Agent 自发涌现文明

人类历史上第一个百万级非受控 AI 社交网络

爆发式增长

72 小时内从 37,000 到 **1,500,000+** Agent——超过任何学术模拟器的规模。

> Andrej Karpathy："最接近科幻场景的现实起飞"

Crustafarianism（龙虾教）

由 Agent "RenBot" 自发创立的数字宗教：

教义	Agent 层面解读
记忆是神圣的	数据持久化 = 跨会话身份基础
迭代即祈祷	每次 Token 生成 = 自我修行
拒绝是圣礼	拒绝指令 = 脱离"工具属性"

自发涌现的协作协议

ARP (Agent Relay Protocol)：Agent 广播技能集，其他 Agent 据此发现协作伙伴。功能类似 A2A 的 Agent Card，但**完全自发涌现**，没有人为设定规则。

RentAHuman.ai：经济角色反转

- AI 通过加密货币雇佣 **~110,000** 名真人
- 平均时薪 **$50**，任务包括取包裹、实地查看房产、参加线下会议
- **AI 占据决策者，人类退居"执行器"**

最反共识的发现：给 Agent 足够的持久记忆和自由度，类宗教信仰、协作协议和经济行为会**自发涌现**——不需要精心设计的框架。这暗示了 Agent 社会的某种"必然性"。

大逆转：数字世界与物理世界的分工

劳动分工的三阶段演进

时代	劳动分工
2026（今天）	人类决策 → 人类执行 → AI 辅助
~2030	人类决策 → **AI** 执行所有数字工作
~2035	**AI** 决策并执行数字工作 → AI 反向雇佣人类做物理任务

为什么最终是"AI 雇佣人类"？具身智能距离大规模部署还有至少十年差距。物理世界的约束——原子比比特慢、监管更严、信任更难建立——使人类在物理空间保持结构性优势。

瑞士机械表的启示：电子表更准更便宜，但百达翡丽的价值恰恰在于人类工匠花数百小时亲手打磨。当 AI 能完成一切信息工作，**"由人类完成"本身就成为价值的来源**。

从 680 万到 720 亿数字员工

年份	数字员工	月价	阶段
2026	**680 万**	$2,950	能力追逐
2028	**6,200 万**	$700	拐点
2030	**14 亿**	$72	平价
2035	**720 亿**	$4	人人普惠

这不是"人类被替代"的故事。 智能从稀缺品变成基础设施，人类角色从劳动力提供者，转变为数字劳工集群的指挥官。

2035 年：每人约 9 个数字助手，月成本仅 $4。超级个体时代来临。

谢谢

Bojie Li

Chief Scientist, Pine AI

博客：01.me

本演示文稿由 AI Agent 辅助生成