# Fable 5 讨论 Notes｜Best Ideas

Cr. Best Ideas 社群

Date: 2026-Jun.

---

## I. 一线实测显示 Fable 5 的上限来自任务难度

### Use case 1：开发者任务

1、Long horizon task 是开发者最稳定的正向反馈，任务越难、越长、越需要持续推进，Fable 5 越能体现拆解目标、执行步骤、检查结果和汇总交付的能力。

2、项目 owner 型任务更能体现出 Fable 和其他模型的能力差异，一些每小时约 300 美元价值的高复杂度任务，更容易看到模型从拆任务、反思纠错、造工具到调度 subagent 的变化。

3、普通任务很难体现 Fable 5 的溢价，一般 coding、前端小需求和材料整理用 Sonnet、Codex 或 GPT-5.5 也能完成，Fable 5 的额外成本未必能转化为独立价值。

4、评判 Fable 5 上限应该看它能否越过用户能力边界，模型如果能提出用户想不到的方案，或者完成用户拆不出来的端到端任务，能力提升才真正可见。

5、在宽任务和大规模 fanout 上，开发者把 Fable 5 用在更宽的任务上时，模型第一个任务就拉起约 100 个 agent，而此前其他模型通常在 10 个左右。这类能力更适合并行搜索、代码库遍历、舆情收集和数据处理。

6、但 Fable 的 Fanout 规模不能直接代表智能提升，有一位重度工程用户在 Opus 4.8 上也见过一次性并行启动 103 个 agent，任务是否做对仍然取决于模型逻辑和任务组织。

7、在前端、3D 和 design taste 上，使用者反馈 Fable 5 在 Three.js、3D 世界构

建和 design taste 上更明显，一次生成的网站更少出现粗糙的 AI 配色，也更懂得给视觉表达做减法。

8、视觉与交互任务仍然需要人工验收，Fable 5 可以还原代码逻辑和大元素，但在截图驱动的细节修正、元素大小和叠加关系上仍然会失手。

9、在逆向工程和安全研究上，工程开发者提到，模型可以通过网页、混淆 JS、安卓应用或游戏 ROM 主动获取上下文，再用这些上下文还原产品逻辑或游戏关卡。

10、ROM 和网页游戏案例展示了反编译能力的具体形态，用户给出 ROM 或网页游戏链接后，模型能够反编译并还原第一关逻辑，但视觉细节仍然需要人工检查。

11、在 TypeScript Excel 类任务上，Fable 5 在路线选择和执行判断上仍有优势。一位开发者表示，Fable 5 用约 3 小时、约 200 美元完成的任务，Codex 5.5 用约 10 小时也能做到接近 90%，差距主要来自模型决定怎么做的能力。

12、相比 Fable，GPT-5.5 在边界清晰的工程任务里可以长时间稳定执行。

- 一位开发者经常让 GPT-5.5 连续运行 70 多小时，三天后得到的结果仍然满意，前提是团队已经拆清任务、写清规范并持续维护 Agents.md。
- 同一组案例显示，GPT-5.5 能把约 3000 个 mock 测试迁移成真实测试，也能在“大文件必须降到 1000 或 2000 行以下”被写成硬约束后，把 8-9 个大文件拆得更好。

## Use case 2：研究任务

13、在 auto research 上，auto research 的价值取决于是否形成闭环，它能把 agent 的工作与实际业务结果连接起来，也能减少团队只看 token usage 的粗糙做法。

14、开放式研究更适合体现 Mythos / Fable 的差异，这类任务天然更长、节点

更开放，方向选择也不总有标准答案，所以模型在研究方向选择上的提升更容易被感知到。

## Use case 3：实际生产任务

15、一位日均 AI 编程成本超过 1000 美元的 CTO 级用户认为，Fable 5 在实际生产任务中的底层智能没有质变，只是工具调用和部分 workflow 更熟练。

16、成本会直接压低企业推广意愿，两天超过 1 万美元的使用成本没有换来足够确定的生产价值，所以即使模型没有下架，企业领导层也不会马上在全公司推广。

17、Fable / Mythos 的 efficiency frontier 仍有争议，最高能力更强不代表成本效率最好，企业在高价值任务之外仍然会倾向于便宜模型或成熟工作流。

18、在企业 workflow 上，有用户把企业提效拆成四步，先抽象工作流，再用工具链自动化约 80% 的流程，再用 AI 替代原先需要人脑处理的约 20%，最后用 AI 加速工具链本身的开发。

19、OpenClaw / Hermes 自动升级的这个 use case 暴露了 Fable 难以泛化 workflow：使用者希望模型每天自动升级企业内部插件，并在上游版本变化后自动修复 patch，但模型反复把任务误解成修复某一个版本的 patch，没有沉淀出通用升级流程。

20、长时任务的失败更像任务状态管理问题，模型需要持续记住最初目标、文件边界、约束和验收标准；Fable 5 可以完成局部步骤，但任务推进久了以后容易丢掉前面设定，后续修改会逐渐偏离原目标。

21、需求对齐本身需要工程方法，人类组织里的一句话需求本来就容易失真，有公司用 OPP（Objective - Problem - Proposal）这类结构化方法把需求对齐工程化；人和 agent、agent 和 agent 之间也需要类似机制，否则并行执行会放大误差。

22、总的来说，一线模型使用体感取决于任务选择，普通任务无感、宽任务更强、创意任务更好、工程可靠性仍有争议，这几类反馈可以同时成立。

## II. Fable 5 在 Benchmark 上的表现怎么样？

23、Benchmark 的作用是定位边界，模型分数、任务成本、token 数、步骤数、拒答比例和完成质量需要放在一起看。

24、在 Xbench 指标评估 Fable 时，因为 Fable 5 在模型前面加了 safety classifier（安全分类器），遇到 cyber、biology、chemistry 和 distillation 相关问题时可能触发拒答、降质或 fallback，单个总分无法解释真实表现。

- ScienceQA 拒答指标说明安全分类器会明显影响评测结果，Xbench 团队在约 100 道科学和长推理题里看到 21 道题返回空结果，多次复测仍是同一组题触发拒答，这一比例明显高于官方披露的约 0.3%。其中，拒答样本主要集中在生命科学方向，这些题目涉及基因、遗传、蛋白质和细胞学，Fable 5 拒答后会 fallback 到 Opus 4.8 作答，所以最终分数需要拆出 fallback。
- ScienceQA 得分显示 Fable 5 仍在第一梯队，Fable 5 得分 73.4，低于 Opus 4.8 的 76.4，高于 GPT-5.5 Pro High 的 73。
- Output token 指标显示 Fable 5 的推理路径更短，Fable 5 平均每题 output token 约 1100，Opus 4.8 约 1700-1900，GPT-5.5 约 5200；换算后，Fable 5 的 CoT 长度约为 Opus 4.8 的 60%、GPT-5.5 的 25%。
- BabyVision 指标说明模型多模态理解在爬升但还没到头部，Fable 5 约 36.6 分，高于 Opus 4.8 的 23 分和 Opus 4.5 的 14 分，低于 Gemini 3.5 Flash 的 61.8 分。
- 在约 100 美元/小时价值的复杂任务上，Fable 5 的 pass rate 和平均分低于前代，可能受安全路由、拒答和降质影响；虽然 API token 更贵，但 CoT 输出减少后，真实任务成本未必按 API 单价差距等比例扩大。

25、CursorBench 3.1 提供了开发者任务的成本对照。

| CursorBench 3.1 模式 | 得分 | 每个任务成本 | Token 数 | 步骤数 |
|---|---|---|---|---|
| Fable 5 Max | 72.9% | 18.02 美元 | 63,842 | 76 |
| GPT-5.5 Extra High | 64.3% | 4.37 美元 | 17,905 | 46 |

## III. Multi-agent、model routing 和 harness

26、Multi-agent 的价值首先取决于任务组织，模型需要把任务拆清楚、把子任务验证清楚、把最后结果汇总成可交付成果。

27、宽任务天然需要并行 agent，代码库遍历、舆情收集、多源检索和真假反馈筛选，本来就需要多个 agent 做信息覆盖。

28、子任务可以分担主 agent 的 context 压力，子 agent 消耗自己的上下文，最后只把压缩后的结果交回主 agent，这相当于一种记忆 scaling。

29、目标不清会让并行系统放大浪费，多个子 agent 会同时朝错误方向推进，token 消耗会快速放大，最后结果也未必更可靠。

30、Workflow 能力已经跨过单一模型边界，Claude Code 本身已经有 dynamic workflow，Opus 4.8 也能触发大规模并行 agent，所以 Fable 5 的体验需要拆开看模型能力、产品 harness 和路由策略。

31、Model routing 已经成为真实产品问题，Fable 有时会把简单任务换给 Haiku、DeepSeek、Gemini 或本地小模型来做，但这种路由逻辑还不稳定，也不总是让用户知情。

32、Agent 产品需要把路由层当成成本系统，Cursor、Harvey 和很多 vertical agent 公司都需要决定哪些步骤交给前沿模型，哪些步骤交给便宜模型，目标首先是控制成本。

33、外部 harness 会持续承受模型升级冲击，如果一个 harness 只是弥补上一代模型缺陷，下一代模型把相关能力训进去后，它就可能变成 technical debt。

34、现在创业团队的组织形态开始像 file system，每个人更像独立 IC，任务被交给 agent 后端到端推进，人类更多负责定义目标、验收结果和处理异常。

35、Agent workflow 的提升可能需要端到端训练，训练侧如果想提高主 agent 和 subagent 的整体交付能力，需要把两者放进同一个任务环境里共同优化；这种训练方式会同时影响模型能力、任务分工和推理 infra。

36、Fable 的优势不只来自任务执行，也来自 implicit intent understanding，模型需要读出用户没有说清楚的目标、偏好和边界，再决定该改哪些代码、该拒绝哪些危险请求、该在什么地方追问或停手。

37、用户轨迹数据会影响下一代 coding model，模型可以通过大量用户 trace 和 trajectory 学到更细的偏好、项目上下文和操作边界，这会让模型在 planning 与 execution 之间更自然地切换。

38、Sub-agent 的下一阶段会抬高训练 infra 要求，如果未来走向大模型带小模型，训练系统需要同时优化主 agent 的 planning 能力和小模型的执行能力，还要避免小模型只会做子任务而丢掉通用规划能力。

## IV. 存储、算力和芯片

39、Sub-agent 的推理优化会同时影响 GPU、内存和 KV cache。多个 sub-agent 在共享 prefix 下运行时，batch inference 可以提高 GPU 利用率，KV cache 和内存转移开销也更容易被摊薄。

40、多 agent fanout 会把存储和内存问题拉到 infra 层。几十个 subagent 或多个 thread 共享上下文时，CXL 内存池和共享内存的价值会变高，因为多个执行分支都需要读取相近的项目状态和中间结果。

41、Sub-agent 会改变主 agent 的 context window 消耗方式。子 agent 在执行时使用自己的上下文，任务完成后只把压缩结果交回主 agent，所以主 agent 的 context window 消耗速度可能变慢。这相当于在 workflow 层面对 context 和 KV cache 做了调度优化。

42、HBM 和内存墙仍然会限制高价值任务的扩展。更长的任务、更大的上下文和更多 KV cache 会继续推高存储需求，workflow 可以提高存储调用效率，但它不能消除物理供给增速和智能需求增速之间的压力。

43、OpenAI 降价同时反映商业竞争和推理效率变化。有人认为，如果模型能力没有明显超过 Claude，OpenAI 通过降价抢企业份额是合理选择。此外，B 系列卡推理效率提升，使同一模型存在继续降价的空间。

44、AI labs 的盈利讨论更像单代模型账本。收入高速增长时，单季度盈亏很大程度取决于算力买多还是买少；训练成本会被未来收入摊薄，推理毛利才是决定长期利润率的关键变量。

45、Anthropic 的短期 ARR 分歧集中在算力供给。乐观观点认为高价值任务仍会带来绝对 ARR 增量；谨慎观点认为未来一两个月新增算力有限、CFO 控 token budget、OpenAI 可能打价格战，这些因素都会限制 ARR 增速。

46、算力估算本身存在很大不确定性。会上有人估算 Anthropic 可能有约 2-3GW 算力，其中一半可能用于训练，单 GW 能支撑多少收入的估计差异很大，因此这类数字只能作为会上观点呈现，不能写成确定事实。

## V. Token maxxing 开始进入企业 ROI 账本

### 企业侧开始管理 quota 和 adoption 指标

47、Token maxxing 已经从 adoption 指标变成成本压力，企业已经开始追问 token 消耗到底带来了多少业务结果。

48、硅谷企业已经开始摸索 quota 机制，很多企业都在讨论 tokenmaxxing 和生产力指标，Google 内部也有使用最强模型的 quota 限制。

49、Quota 机制仍缺少统一答案，企业可能按员工、职能、项目或时间周期限制 token，但大家还在摸索怎样限制才不伤害 AI native 转型。

50、Vertical agent 公司的毛利压力更直接，比如单用户 token 消耗每月上涨 50%-100%，但很多老产品还是按 seat 收费，收入无法同步增长。

51、Function tool agent 转向 coding agent 会显著放大成本，同一个 task 放进 sandbox 持续运行后，token 量可能比过去的 function tool agent 高出数倍，甚至把 gross margin 打成负数。

52、Subscription 计价会被重度用户冲击，普通用户和重度用户的成本差异太大，agent 公司需要考虑涨价、model routing、自训小模型或接入开源模型。

## 客户侧开始做 value mapping 和 token dashboard

53、Token maxxing 会带来 KPI 失真，企业把 token usage 当成 adoption 指标后，团队会自然出现 reward hacking，最后只是把 token 用量冲高。

54、下一阶段指标应该从 value mapping 开始，企业需要知道谁用了 token、在哪个 workflow 里用了 token、服务哪个 title 或 practice group、最后交付了什么结果。

55、Token dashboard 会成为客户侧控费基础设施，有 vertical agent 团队已经在做面向企业的 token dashboard，让客户看到 token 流向，再由客户判断这些 token 是否花在更值钱的地方。

56、国内市场尚未明显踩刹车，国内 token 成本更低，电商、招聘和广告等场景也有明确价值反馈，真正有钱的客户还没有全面铺开 token 消耗。

57、国内市场后续也会进入 ROI 账本，当 agent adoption 继续上升，企业会从“能不能用 AI”转向“哪些 token 花得值”。

58、也有用户激进认为 token usage 会继续大幅增长，甚至可能达到现在的 100 倍，但单 token 价格会继续下降。

## VI. 安全边界和发布节奏决定 adoption 速度

### 安全边界扩展到拒答、反编译和工具权限

59、安全分类器已经进入模型体验层，安全评估把 cyber、biology、chemistry 和 distillation 当作模型发布里最需要关注的风险方向，Xbench 的拒答样本也集中在生命科学相关题目上。

60、在模型的反编译能力上，模型能更主动地拿到应用、网页和游戏里的隐藏上下文，这对网络安全、工具权限和边界控制都提出了更高要求。

61、自主 agent 越强，权限边界越重要，prompt injection、工具权限、proactive agent 行动边界和模型自我约束能力都会变成下一阶段核心问题。

### 访问暂停影响市场情绪

62、Fable 的访问暂停首先影响短期市场情绪，在二级市场，投资人会重新评估 ARR、监管风险、IPO 时间表和模型发布节奏，但也有人认为这是外因导致的短期扰动，不应简单推导到所有 AI labs。

63、访问暂停还可能改变企业备选方案，有观点表示，如果前沿模型容易受到监管或访问限制，企业会更认真考虑自研模型、开源模型微调和前沿模型备份方案，这反而可能带来新的算力需求。

64、国内模型自主可控被重新强调，有观点把 Fable 访问暂停和国产链联系起

来，认为国内不只需要半导体自主可控，也需要模型自主可控，智谱 GLM 5.2 的发布值得关注。

## VII. 模型市场会按任务价值和成本效率分层

### 前沿模型锁定高价值任务和 agent 调用

65、前沿模型会继续锁定高价值任务，超级工程、科研、安全、底层系统重构、投资研究和国土安全类场景可以承受更高模型成本。

66、To C 和 To B 可能走向两套优化体系，一种观点认为，To C 产品的价值不会长期按 token 出售，模型会更强调快、好、省和产品体验；To B 场景会继续追求顶尖智能来解决大型工程和高价值任务。

67、也有观点认为，To C 需求没有完全释放，人对智力差距的容忍度很低，智能能力的小幅差别可能带来很大影响；当普通用户开始用模型改善决策并能消化模型建议时，C 端需求可能显著上升。

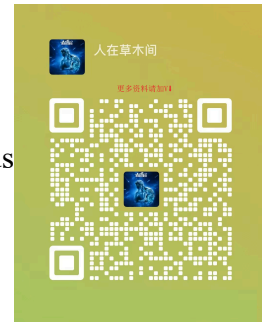
68、AI 时代 toB 生产力场景可能大于传统互联网时期，企业内部系统、workflow、agent 和机器人都会持续消耗模型能力，平台公司内部 token 消耗也可能显著高于 consumer 直接用量。

69、To human 和 to agent 更能解释未来 token 用量，大量模型调用会由 agent 网络发起，而不直接来自人类用户。

70、机器人可能成为 to agent 之外下一类巨大智能消耗，因为模型一旦触达物理世界，智能调用会从软件任务扩展到现实环境中的行动和反馈。

### 低价模型承接普通步骤、国产和开源机会

71、低价模型会承接普通步骤，很多企业任务只需要 GPT-4 级别或 Opus



别的智能，即使是在 long horizon task 里，也有很多 step 不需要前沿模型。

72、国产模型和开源模型的机会在成本效率，如果国产模型能以 1/10 或 1/20 的价格达到 Opus 4.6 级别智能，大量白领流程和普通企业任务会被重新打开。

73、收入和 token 处理量可能出现分层，一种情景判断是，未来可能出现“前沿模型拿 80% revenue，开源模型处理 80% token”的分层。

74、前沿 lab 可以用自己的强模型生成更高质量的蒸馏数据，也可以优化同档智能水平下的推理成本。

75、前沿 lab 短期缺少卷低价模型市场的动力，高价值客户和推理算力在当下仍然供不应求，头部公司没有强动力马上进入低价小模型市场。

76、国产模型成本优势仍要继续验证，如果头部 lab 自己做小模型并优化推理 infra，同等智能水平下的单位成本未必一定输给国产模型。