2026 • 初夏
James Cheng

理解 LLM 的三条根本限制

写给 AI Agent时代的入门用户：从 context 到 feedback 再到共识塌缩，那些进步、失败、积累方向的底层逻辑。

“不是 model 变得更聪明让你更厉害，是用context 让 model 真正成为远超均值的水平。”

AGItopia社区

WHY WE ARE HERE

你是不是觉得，AI 的概念太多了？

每两周一个新名词。这只是过去 18 个月里被反复讨论过的部分概念：

Cursor | Claude Code | vibe coding | RAG | GraphRAG | LangChain | LangGraph | Multi-Agent | RLHF

Harness | MCP | Skills | CLI Agent | TUI | IDE Agent | Tool Use | ReAct | Reflexion | Long Context

Context Engineering | Memory | Mem0 | Fine-tuning | LoRA | In-Context Learning | Few-Shot | Chain-of-Thought

Agent Workflow | Co-Work | Agentic UI | OpenClaw | Hermès Agent | 低代码 | 无代码 | Eval Harness | 还有更多.....

THE REAL QUESTION

跟得越紧，越抓不住主线

三个困惑，每一个都让人焦虑：

01 **哪些是真的范式革命，哪些只是包装新瓶？**

每个新概念都有自己的故事，但它们到底在解决什么问题，谁能说清楚。

02 **我应该积累什么，才不会被下一波技术淘汰？**

学的工具下个月就过时了。背后有没有一种“不会过时”的能力？

03 **我应该怎么和 AI 合作，而不只是低水平地用它？**

用得越深，越觉得 AI 给的不是“对的答案”、“深刻的答案”，而是“平庸的标准答案”。怎么办？

你需要的不是更多的概念，是一套能解释关键概念的、持续有影响力的，**底层原理。**

THE ANSWER

所有这些进步，都收敛到三条根本限制

基于 Transformer 架构的 LLM，有三条几乎无法跨越的内在限制。理解它们，就理解了所有问题为什么发生、所有进步在解决什么、以及未来该往哪里积累。

A

Context 有限性 与 有效性

1M 是宣传，可用是 200-300K。视野**“两头清晰、中间模糊”**。

B

Feedback in the Loop

模型不知道自己写得对不对，必须有**明确的评价标准，和外部反馈**才能闭环。

C

共识 / 均值塌缩

默认输出是**“聪明的平庸”**，需要**准确的语境、超越均值的设定**拉出均值。

接下来的内容围绕这三条展开：直观对比 → 抽象规律 → 你应该怎么做。

PART 01

三条根本限制：是什么，为什么会有

从 Transformer 的内在结构，看清模型的能力边界。这些边界不会因为模型变大而消失。

THE ANSWER

所有这些进步，都收敛到三条根本限制

基于 Transformer 架构的 LLM，有三条几乎无法跨越的内在限制。理解它们，就理解了所有问题为什么发生、所有进步在解决什么、以及未来该往哪里积累。

A

Context 有限性 与 有效性

1M 是宣传，可用是 200-300K。视野**“两头清晰、中间模糊”**。

B

Feedback in the Loop

模型不知道自己写得对不对，必须有**明确的评价标准，和外部反馈**才能闭环。

C

共识 / 均值塌缩

默认输出是**“聪明的平庸”**，需要**准确的语境、超越均值的设定**拉出均值。

接下来的内容围绕这三条展开：直观对比 → 抽象规律 → 你应该怎么做。

同一份文档，两种处理方式，结论完全不同

SETUP　把 50 份月度会议纪要拼成 300 页文档，让 AI 总结“过去一年公司战略变化的关键转折”。

APPROACH 1 • 一次塞进去

把 300 页文档整个贴给模型作为prompt

*“**公司战略呈现持续优化趋势**”。Q1 关注产品竞争力，Q4 强调数据驱动，体现管理层对市场变化的敏锐感知，整体方向稳健。*

APPROACH 2 • 分层处理

先按月做摘要，再基于 50 份摘要做总结

*“**关键转折在 6 月：放弃 to-C 增长，转向企业客户**。8 月小程序失败实验后，资源进一步**收紧到付费 SaaS**。**年底付费 ARR +47%，DAU -12%——一次主动的牺牲**。”*

WHY • 为什么差距这么大

6月、8月的关键转折恰好落在文档中段。 Lost in the middle：开头/结尾召回 ~90%，中段掉到 ~75%。模型不会告诉你“中段没看清”，它会基于看到的部分编一个合理结论——所以你以为它读懂了，其实它在凑。

模型的 context window 不是均匀的内存——是一个“两头清晰、中间模糊”的视野

1M	200–300K	85–95%	76–82%
宣传 context window	实际有效区间	开头 / 结尾召回准确率	中段召回准确率

为什么：Transformer 的 attention 是 O(n²)。为跑得动长 context，工程上做了稀疏 attention、滑动窗口、位置编码插值——每个妥协都有代价。再加上训练数据“短文本占主流、长文本里高质量样本稀缺”，模型对中段的理解能力天然弱于两端。

Sources: Liu et al., Stanford 2023 • BenchLM 2026 长上下文评测 • NIAH / LongBench v2 / RULER

错的不是塞得不够多，是塞错了方式

FAIL

把 300 页 Python 教程整本扔进去

“基于这本书，给我设计一份 4 周学习计划”——AI 给出的计划和直接搜“Python 4 周学习”几乎没区别。

为什么 → 中段那些“作者特别强调的概念”它根本没真的读到，最后给出的还是世界平均答案。

WIN

拆章节后让 AI 逐章蒸馏

先让 AI 读目录，再让它逐章节做“这章作者的独特观点是什么”，最后基于章节摘要做学习计划。

为什么 → 每段的有效注意力都被充分利用，输出针对这本书的视角而非互联网均值。

200 个文件代码库一次贴：Claude 改前几个文件 OK，后续函数重名、参数冲突、引用幽灵类。

先梳理结构再单模块改：用“项目梳理 + 依赖图”形成知识图谱，采用Graph RAG，每次只读取相关文件，改完代码跑测试再下一个。

THE ANSWER

所有这些进步，都收敛到三条根本限制

基于 Transformer 架构的 LLM，有三条几乎无法跨越的内在限制。理解它们，就理解了所有问题为什么发生、所有进步在解决什么、以及未来该往哪里积累。

A

Context 有限性 与 有效性

1M 是宣传，可用是 200-300K。视野**“两头清晰、中间模糊”**。

B

Feedback in the Loop

模型不知道自己写得对不对，必须有**明确的评价标准，和外部反馈**才能闭环。

C

共识 / 均值塌缩

默认输出是**“聪明的平庸”**，需要**准确的语境、超越均值的设定**拉出均值。

接下来的内容围绕这三条展开：直观对比 → 抽象规律 → 你应该怎么做。

同一段代码，两种处理方式，30 分钟 vs 5 分钟

SETUP　用 VBA 写“自动生成数据透视表”宏。AI 给了 50 行代码，复制运行，报错 **运行时错误 1004：对象不支持此属性**。

APPROACH 1 • 直接信任

不告诉 AI 报错，自己换 prompt 重问

“请重新写一个 VBA 宏...” → *新代码* → *不同错误*

“能不能用 PowerQuery 替代...” → *又一段* → *又一个错*

3 轮过去，结论：“AI 写 VBA 不行”。

总耗时 30 分钟，问题没解决。

APPROACH 2 • 报错喂回去

把错误原文 + 一句话 paste 给 AI

“这段代码报错：运行时错误 1004：对象不支持此属性。在第 12 行。基于这个错误改。”

AI 给修复版 → *跑通。*

总耗时 5 分钟。

WHY • 为什么报错喂回去就解决了

LLM 单次推理是 next-token prediction，**没有“我刚才写的代码跑通了吗”这个概念**。训练 loss 是 token-level，不是任务完成度。SWE-bench 数据：单次 LLM 30-40%，加 agent harness 跃升到 70%+——差距全在闭环上。

LLM 没有“完成感”——它不知道自己写得对不对

为什么

- 单次推理 = 从前到后吐 token，没有“回头看”
- 没有 self、没有内省、没有“我刚才写得好不好”的判断
- 训练目标是 token-level loss，不是“任务完成度”
- RLHF 学到的是“风格人类喜欢”，不是“任务真的解决了”

INDUSTRY SHIFT

2025 是模型年，

2026 是 harness 年。

模型本身的进步在放缓，但 agent 的实际可用性还有数量级的提升空间——提升点几乎全部在 LLM 外的“反馈/评估/状态管理”基础设施。

含义：单次推理的 LLM 不能独立完成需要“自我校正”的任务。要让它干成一件事，必须在它外面建一个反馈闭环——**清晰的评估保准，以及让输出能被外部力量评价，再来一轮。**

完成任务 = 多次输出 + 外部验证 + 迭代

FAIL

PPT 画完直接交

配色乱、文字溢出、有些图根本没显示，风格与模板不统一，风格不专业，布局不合理

为什么 → 没有明确的风格、模板指导，没有给出评估标准，而且AI 整个流程没“看到”自己输出。

WIN

设定标准并让AI自己审视

“麦肯锡级别”的商业顶级专业化的PPT排版与审美 + PPT制作后，用截图方式让AI自我审查

为什么 → 把外部世界的信号（标准 + 结果）注入回模型，让模型自我判断、优化建立闭环。

项目做完直接提交代码：字段、接口问题，依赖混乱，无法跑通

测试驱动开发：每做完一个相对独立的功能，让模型进行自我审查，并进行测试，看是否跑通 + 搭建部署审查机器人，生产环节若出现问题，自我修复，或自动回滚

THE ANSWER

所有这些进步，都收敛到三条根本限制

基于 Transformer 架构的 LLM，有三条几乎无法跨越的内在限制。理解它们，就理解了所有问题为什么发生、所有进步在解决什么、以及未来该往哪里积累。

A

Context 有限性 与 有效性

1M 是宣传，可用是 200-300K。视野**“两头清晰、中间模糊”**。

B

Feedback in the Loop

模型不知道自己写得对不对，必须有**明确的评价标准，和外部反馈**才能闭环。

C

共识 / 均值塌缩

默认输出是**“聪明的平庸”**，需要**准确的语境、超越均值的设定**拉出均值。

接下来的内容围绕这三条展开：直观对比 → 抽象规律 → 你应该怎么做。

同一个任务，两种处理方式，输出像两个人写的

SETUP　让 AI 帮你写一封求职信，应聘某互联网公司的产品经理岗位。

APPROACH 1 • 默认 prompt

“帮我写一封求职信”

*“**我对贵公司的产品理念深表认同**，希望能加入这样一个充满创新精神的团队。我具备扎实的产品思维和数据分析能力，曾在某互联网公司参与过多个用户增长项目，能够为团队带来积极价值...”*

APPROACH 2 • 注入三件具体的事

① 你是谁（你做过的产品+数字） ② 欣赏对方的具体决策 ③ 最有信心和最焦虑的

*“**贵公司去年把首页推荐流改成** explore-only*的决策，是我读过最有信念感的产品判断之一——你们要的不是 DAU，而是**用户好奇心被激活的那一秒**。我做过类似的事：放弃 CTR 优化，换**首次到第二次互动转化率**。三个月 DAU 跌 7%，七日留存涨 23%...”*

WHY • 为什么差距这么大

next-token = **高概率 = 共识**，模型输出向中位数塌缩。但 PNAS Nexus 2026 显示：注入哪怕一两句个人 context，输出多样性立即提升 40-60%——**少量高密度的“你的语境”，胜过一万字泛泛要求**。

LLM 的默认输出是“聪明的平庸”——它在向均值塌缩

> “硅基大脑似乎绝对客观，但最终只能抵达聪明的平庸。”
>
> —— yage.ai • Context Infrastructure

训练目标

next-token = 高概率词 = 共识。模型学的是“全人类文本的统计平均”。

RLHF 偏好

标注员喜欢“看起来正确、不冒犯、覆盖各方”——二次平均。

采样策略

偏向高概率token——三次平均。结果：往中心塌缩。

实证：LLM 之间的输出相似性 > 人类之间的相似性（PNAS Nexus 2026）。所有 LLM 都有这个问题，且使用 LLM 协作创作的人越多，同质化反而被进一步放大（diversity gap widens at scale）。

少量高密度的“你的语境”，能把 AI 从均值里拉出来

FAIL

AI的公众号文章撰写

“反共识观点”“各种片面的常见引爆词”“各种分层级的Bullet point”——读者一眼能看出是 AI 写的，且审美疲劳。

为什么 → AI 在从全网公众号文章均值里抽样，且保留常见的文书梳理习惯，没有锚点，输出只能是“中位数”。

商业赛道“5 条机会 + 5 条风险”两面派分析：什么都说 = 什么都没说。

WIN

注入“三个维度的参考”

① 你是谁（你偏好什么，你的输出逻辑，你的认知公理）；② 你期待的结果是什么（目标受众，结果衡量标准）；③ 操作指导（过去失败经验，可用的工具与技能，参考文档体系）。

为什么 → **少量、远超均值、高密度、个人化的 context**，足以把输出从均值拽出来。

强制取立场：“我是一名红杉资本的顶尖投资人，你有我的全部投资偏好和投资逻辑的记录，如果你是我，你会因为什么决定不投？” 把 AI 从中立观察者拉到具有超越平均水平的判断者的程度。

PART 02

影响 + 演进：过去三年，未来三年

vibe coding 翻车的方式 · 历史进步的内在逻辑 · 下一阶段瓶颈的转移

vibe coder 的所有翻车，大体上都能归到这三条

下次代码出错前，先用这张表做诊断。

CRASH 1 • LIMIT A

项目越大
AI 越糊涂

表面是 context 不够，本质是有效 context 不够、不准确。AI要不读不到，要不即使读了，也没法在中段模糊的视野里整合所有信息。

CRASH 2 • LIMIT B

AI 写完就交
代码一跑就崩

代码领域有明确的结果衡量标准，但AI 不知道自己写的代码能不能跑。此外，代码领域暂时没有明确的过程衡量标准。

CRASH 3 • LIMIT C

代码“很标准”
但没灵魂

AI 默认会写出“互联网上 60 分代码的中位数”。除非主动注入你的语境，AI 永远在写“通用代码”，不是“你的代码”。

过去三年所有重大进步，都在攻坚这三条

LAW	主战场	代表性技术 / 框架
A	**让 context 更“可用”**	Flash Attention • Ring Attention • YaRN • Mamba • RWKV • RAG • GraphRAG • MemO • **分层记忆... ...**
B	**让 LLM 能“自我校正”**	ReAct • Reflexion • Self-Refine • LangGraph • AutoGen • CrewAI • Tool Use • MCP • **Harness... ...**
C	**让输出“有你的味道”**	System Prompt • Persona • Few-shot • Fine-tuning • LoRA • RAG • Skills • **Context Infrastructure... ...**

关键观察：每一个新概念都明确指向其中一条，三条之间几乎没有交集。所以你**回头看任何“新名词”时，先问：它在解决 A、B、还是 C？这个问题答出来了，那个概念也就掌握了。**

Note：**当然，除此之外还有很多落地实践中的其他问题，**例如算力调用渠道（比如im），权限，资源调度等，**但并非本质问题。**

瓶颈正在转移：
从“模型聪不聪明”到“context 够不够好”

2022-2025中 • 模型叙事

模型能力是瓶颈

模型性能逐年翻倍，智能水平逐渐提高，瓶颈在转移到 memory bandwidth、cache 设计、数据局部性。

→

2025年中至今 • 模型使用系统

使用系统配套

GPT-5、Claude 4.7、Gemini 3 之间，水平差异在缩小，基于LLM的模型智能逐渐遇到天花板。瓶颈正在转移到“你能注入多少高质量的个人/组织上下文”。下一阶段的赢家不再是“接了哪个模型”，**而是“context infrastructure 怎么样”，未来，进一步的矛盾将出现在组织的组织方式是否是AI native的**

未来三年的工程主战场：① 更聪明地用 context（A） • ② 建更可靠的 harness（B） • ③ 沉淀更深的AI友好的个人/组织的组织方式（C）

PART 03

你应该怎么做：实操 + 积累

把认知翻译成今天就能用的工作方式 • 把工作方式翻译成长期复利的个人资产。

今天就能用的 5 条原则 - 挑战自己，诘问自己

01

我有没有关注到context的管理？是应该用subagent，还是新开对话，还是rewind，还是继续推进？

Lost in the middle 的直接应用。长 prompt 写完之后，在末尾把最关键的指令再重复一遍——啰嗦但效果立竿见影。

02

我有没有提供标准并建立闭环？有没有自己手动贴给模型它自己跑的结果？有没有、在什么环节提供了人工判断？

代码：让它跑起来，报错喂回去。文档：让它自己读自己。图表：截图给它看。分析：让它列 3 个反方观点。

03

我有没有主动注入“我的语境”？语境是否足够准确，简洁，清晰？Context是否超过了平均水平？

每次开始新会话花 30 秒想一下：哪些“只有我知道”的事情？公司命名习惯、老板口头禅、客户禁忌、你的判断偏好。注入了，AI 才开始为你工作。

04

我该关注今天的这个新产品/新技术吗？解决的是哪个问题，有什么优势？我有没有抓到本质，有没有对下一个趋势的思考判断？

看到新闻后，追问本质，同时需要注意，今天所有的讨论是围绕LLM来的，架构的进步，落地的摩擦管理，是讨论范畴之外的进步力量。

05

我有没有把积累系统化、自动化，形成沉淀？我有没有基于项目的实操手感，而不是简单复制？

积累自己的语境与基础设施并沉淀在系统中，AI使用是一门手艺，手艺要通过亲身实践才能建立优势。

你应该积累的，是 AI 永远拿不到的“你的语境”

L1 • OBSERVER

原材料

Agent任务seesion、日常聊天记录、邮件、纪要、草稿、截图、复盘、读后感、对话片段。关键是不要丢。这些是 AI 永远拿不到的“你的训练数据”。

James今天发现了Lark-CLI的工具，并有机融入到目前的context infrastructure中，开放了15个非敏感权限，进行了五项潜在使用场景的探索，其中三项具有实际意义

L2 • REFLECTOR

模式

定期翻看原始材料，问自己：在反复想什么？犯什么错？有什么稳定偏好？把这些蒸馏成“行为规律”——比原始材料更精炼，但还没上升到判断准则。

James持续关注工具层的新晋高星仓库，并思考如何融入自己的infra中，形成了筛选、测试、融入、生产使用的基本流程，并在过程中迭代经验

L3 • AXIOM

公理

跨场景稳定的判断准则。“在 X 类问题上我倾向先做减法。”“面对 Y 类对手我的优势是 Z。” 别人写的拿来用不了——它从你的经历里长出来。

James认为被动工具使用者 vs 主动工具构建者，决定了 AI 价值捕获能力。护城河不在工具本身，而在于对工具的态度——是否愿意在工具不足时进行改造、组合、扩展。

这三层正好打中三条限制。L3 是密度最高的 context（攻击 A：几十个 token 浓缩你成千上万次经验）；你的判断准则是反馈标准（攻击 B：没有 L3，反馈闭环就是空的）；你的公理是把 AI 从均值拽出来的最强工具（攻击 C：直击 LLM 核心弱点）。

未来不属于“会用 AI 的人”

这个门槛会很快消失。

未来属于
“*有自己沉淀*”
“*有AI合作手感*”的人。

谢谢

分享人 • James Cheng

2026 年 初夏

AGItopia社区