Andrej Karpathy 2025 ??????????????????Agent???

OpenAI 前创始成员安德烈 ·卡帕西（ Andrej Karpathy）在 2025年的全年演讲与公开访谈。他为我们揭示了 AI 从大语言模型到 Agent 演进的核心逻辑、工程现实与未来路径。

??????????????????????

首先，我们必须理解一个核心判断：当前的技术界正处在从代码编写向意图指引的不可逆转折点。

英语成为了最高效的编程语言。很多人把大语言模型仅仅当成是一个聊天机器人，这在卡帕西看来是一种严重的认知降维。大语言模型的本质是基于 Transformer 架构的新型计算平台，他将它定义为 " 大语言模型的操作系统 "。这个系统不仅重构了软件的生产方式，更是带来了前所未有的安全挑战与架构变革。

而这一切的根源，要从软件发展的三次相变说起。

??????????

卡帕西通过对 GitHub 代码版图与特斯拉自动驾驶架构的深度考古，提出了一个基于物理属性的软件演进坐标系。他认为，软件的演进并非连续的线性增长，而是经历了三次离散的物理相变，每一次跃迁都伴随着计算熵的指数级压缩，本质上是人类与硅基底交互界面的抽象层级跃迁。

????????1.0???2.0

软件 1.0是由人类程序员通过 C++、Python 等形式化语言编写的。它的物理特征是确定性与稀疏性。程序员必须将复杂问题分解为每一个微小的显式逻辑步骤，输入 A 经过固定的逻辑流必然会得到输出 B。

这种模式在处理高维、无序的现实世界数据时，不可避免地遭遇了 "复杂性这道墙 "。比如，人类根本无法手写出识别行人的所有边缘规则 —— 无论是行人的姿态、衣着、所处环境的变化，都能让传统代码顾此失彼。在 GitHub 的版图中，这代表了由 Python、Java Core 等旧大陆统治的疆域，它们的生产力完全受限于人类大脑对逻辑流的认知带宽。

软件 2.0 的出现，标志着计算范式的第一次黑盒化。程序不再是编写出来的，而是通过梯度下降在海量数据中生长出来的。它的产物从文本源代码转变为了二进制的神经网络权重。

卡帕西用特斯拉自动驾驶的架构演进作为最有力的工程确证：早期的车道保持功能依赖于数万行的 C++ 代码，需要工程师逐个定义各种路况下的决策逻辑，但是最终被 BEV Net 彻底取代。 BEV Net 是一个端到端的神经网络，它能够融合多模态的传感器数据，直接输出三维向量空间的预测结果，无需人类编写复杂的分支逻辑。

工程实践反复证明：在处理现实世界的长尾场景时， 权重优于代码 。因为优化算法能在高维空间中找到人类逻辑无法触达的最优解。

????????3.0???

如今，我们正处于软件 3.0的爆发期。随着 Transformer 架构的全面统治，软件的核心产物从神经网络权重进一步演变为了提示词。

大语言模型成为了一个通用的、可编程的神经网络。开发者无需定义模型的架构，也无需训练权重，只需要通过自然语言来描述意图、约束条件以及少样本示例，就能完成编程。

卡帕西给出了一个极具冲击力的对比：情感分类器的构建成本，从软件 1.0时代的几天、软件 2.0 时代的几周，直接坍缩为软件 3.0 时代的几秒。这不仅意味着英语成为了最热门的编程语言，更标志着计算资源的使用从训练专用转向了通用推理。

在这个范式下，技能金字塔被彻底倒置。曾经作为程序员核心竞争力的语法记忆、正则表达式运用等能力，如今变得无足轻重。取而代之的，是逻辑思维的清晰度和自然语言的精确表达能力 ——这些成为了定义资深工程师的新标准。

??????????????

理解了软件的三次相变，我们才能真正看懂大语言模型操作系统的本质。它不是一个简单的文本生成工具，而是一个具备完整冯 ·诺依曼特征的新型操作系统。

卡帕西通过建立大语言模型与传统操作系统的映射关系，为我们揭示了它的核心架构与潜在风险：

在这个架构隐喻中：

- **Transformer** 推理引擎 充当了 CPU，负责逻辑处理
- 上下文窗口 则是系统的内存，这是所有计算发生的物理场所，具有易失性与稀缺性

目前大语言模型面临的最大瓶颈，正是内存限制。无论模型在预训练中压缩了多少知识，它的推理能力始终受限于上下文窗口的带宽。

模型权重 则相当于只读的硬盘，存储了从互联网海量数据中压缩而来的模糊记忆。这种记忆是有损的 ——模型记得知识的概率分布，却无法像数据库一样精确检索，除非通过检索增强生成技术（ RAG ）通过外挂外设来弥补。

??????????????????????

更关键的是，大语言模型操作系统架构中还存在着一个致命的缺陷：那就是内核空间与用户空间的非隔离性。

在 Linux 或者 Windows 等传统操作系统中，Ring 0 级的内核权限会受到硬件指令集的物理保护，普通用户无法随意篡改系统核心逻辑。

但是在 LLM 操作系统中，系统提示词与用户输入本质上都是被拼接后送入 **Transformer** 的 **Token** 序列。这意味着，用户可以通过精心构造的提示词注入，在用户空间执行内核级攻击，覆盖或者绕过系统设定的安全红线。

卡帕西特别警告：这是软件 3.0 时代特有的安全漏洞。只要模型遵循指令跟随的概率机制，这个漏洞在逻辑上就无法根除，只能通过持续的对抗训练来提升防御能力，却无法从根本上消除。

?????????

在算力部署层面，卡帕西还观察到一个有趣的历史循环：

当前我们正处于类似 1960年代的分时共享大型机时代 —— 几十亿用户通过简单的终端（比如聊天界面）连接到云端的超级计算机，请求被排队、批处理。

但是随着硬件架构的演进，特别是 Apple Silicon 的普及，本地运行高性能大语言模型成为了可能。这背后的核心逻辑是：Transformer 的计算瓶颈在于内存带宽，而非纯粹的 Flops。

当本地设备拥有足够的统一内存时，我们将迎来 个人计算 **2.0**时代 —— 每个人都将拥有运行在本地、隐私化、零延迟的大语言模型操作系统。

这种架构回归不仅是算力成本的考量，更是实现 AI Agent 从工具向数字延伸转变的物理基础。未来的计算竞争，将不再仅仅是云端大模型的参数竞赛，而是谁能率先定义并且标准化大语言模型操作系统的文件系统与外设接口。

??Agent?????????

然而，大语言模型操作系统的崛起，并不意味着 Agent 已经成熟。卡帕西通过对当前 AI 技术的深度解剖，提出了 "锯齿状智能" 的关键概念，揭示了当前 AI 与生物智能的本质差异。

与生物智能通过亿万年进化获得的生存本能不同，基于大模型操作系统内核构建的 Agent，呈现出一种极端的锯齿状的能力分布。

AI??????????????

卡帕西严厉驳斥了将当前 AI 技术简单类比为生物大脑的仿生学观点。他指出：

- 生物智能 是进化的产物，它的核心算法被高度压缩在了微小的 DNA 双螺旋结构中，并且赋予了生物体强大的硬件本能。比如，斑马幼崽在出生几分钟后就可以奔跑并且跟随母体 ——这种复杂的运动控制与环境感知能力并不是通过后天的大规模数据训练习得的，而是预埋在基因中的生存硬编码。
- 大语言模型 是预训练的产物，它的本质是通过梯度下降算法，将 PB 级的人类互联网文本数据压缩进神经网络的权重矩阵中。这导致了 AI 并非是数字化的动物，而是基于人类社会数字化痕迹构建的 数字幽灵 ——一个纯粹的、离散的、概率性的精神实体。

????????

这种生成机制，导致了 AI 智能在能力谱系上的极端不连续性：

一方面，模型展现出了类似《雨人》中雷蒙德式的天才特征 ——能够精确回忆起晦涩的 Python 库函数文档、法律条文或者历史细节，这是模糊记忆在海量数据压缩后的直接体现。

另一方面，在处理简单的逻辑推理或算术问题时，模型却可能表现出令人咋舌的愚钝，甚至产生高置信度的幻觉。

卡帕西强调，这种矛盾并非是程序的 Bug，而是 自回归架构的物理必然。模型实际上从未进行过思考，而是在高维向量空间中进行概率预测。当训练数据中缺乏显式的因果推理路径时，模型只能通过概率统计来填补逻辑真空，从而导致了一本正经胡说八道的现象。

因此，产业界必须正视这种非生物性的智能特征：

- 在需要高容错与创造性的头脑风暴场景中，锯齿状智能的幻觉可能会成为创意的来源
- 但是在金融审计、医疗诊断或自动驾驶等逻辑严密的场景中，这种概率性的不确定性就构成了致命的可靠性短板

当前的 AI Agent，实际上是一个记忆力超群、但是缺乏常识校验机制的 "学者综合症患者"。它在没有外部工具辅助的情况下，无法形成闭环的逻辑链条。

??????????

除了 Agent 本身的特性限制，强化学习这个曾经被寄予厚望的技术路径，也面临着卡帕西所说的 "熵减陷阱"。

回顾过去十五年的 AI 发展史，卡帕西对被过度神话的强化学习范式进行了冷峻的技术考古。

在深度学习的早期阶段，业界曾经普遍认为强化学习是通向通用人工智能的圣杯。 OpenAI 甚至倾全力投入 Universe 项目，试图训练一个 Agent 能够像人类一样通过键盘和鼠标操作任意的网页与软件。

但是卡帕西复盘认为，这是一次典型的战略误判与时空错位。在当时缺乏强大表征能力的基础模型之前，直接在像素级或操作级进行强化学习探索，相当于让一个没有任何先验知识的盲人在迷宫中随机乱撞。由于环境反馈的奖励信号极其稀疏，Agent 即便消耗了巨量算力进行亿万次的试错，也难以习得有效的行为策略。

?????????

卡帕西举了"用吸管吸取监督信号"这个生动的比喻，揭示了当前强化学习在推理任务中的数学缺陷：

在处理一道复杂的数学题或者代码生成任务时，模型可能需要进行长达几千个 Token 的推理步骤，生成多种解题路径。然而，强化学习算法往往只能在最终得出答案时，获得一个单一的、二元的反馈信号（比如正确或是错误）。

这相当于将一比特的有效信息稀释并且广播到了整个漫长的推理链条中。这种做法导致估计器的方差极高，充满了统计噪声——模型可能会因为蒙对了一个答案而错误地强化了中间错误的推理步骤，或者因为最后一步的失误而否定了前面完美的逻辑推演。

相比之下，人类的学习从未依赖于成百上千次的盲目试错与稀疏奖励。人类是通过体验与复盘进行学习的：在一次失败后，大脑会进行深度的反思，识别逻辑链条中的具体断裂点，并且进行修正。

但是目前的 AI 训练架构中，完全缺失这种"系统二层面的反思机制"。

因此，卡帕西预测，未来的算法演进必须摆脱对简单强化学习的路径依赖，转向基于过程监督与内在反思的新范式——也就是让模型在推理的每一步都能获得密集的反馈信号，而非仅依赖最终结果的盲目奖惩。

?????????????

另一个潜在的致命风险，是"模型坍塌"的热力学定律。

随着 AI 生成内容在互联网上的泛滥，大语言模型正面临着内生性的退化风险。卡帕西从信息熵的角度指出，持续使用模型合成数据来进行自我训练，将会导致数据分布的极度窄化与多样性丧失。这与人类思维随年龄增长而陷入固化的过程，具有惊人的同构性。

?????????

针对当前通过合成数据解决数据荒的行业共识，卡帕西提出了"隐性坍塌"的理论预警：

虽然单个 AI 生成的样本在语法和逻辑上可能看起来完美无缺，但是在统计学层面，这些样本在所有可能的思想空间中只占据了一个极其微小的子空间。

以讲笑话为例，ChatGPT可能翻来覆去只能生成那几个经典的笑话模板，而无法像人类一样创造出无穷无尽的幽默变体。这种现象表明：AI生成的数据本质上是低熵的、同质化的。

如果将这些缺乏多样性的合成数据回填到下一代模型的训练集中，模型将陷入自我强化的循环，导致其泛化能力与创造性呈指数级的衰退，最终退化为只会重复标准答案的复读机。

??????????

卡帕西进一步将这个现象与人类认知的生命周期进行了深刻类比：

- 儿童的思维
  之所以充满创造力与非线性跳跃，是因为他们尚未过拟合于现实世界的规则，他们的思维模型保持着极高的熵值
- 成年人的思维往往陷入了既定的神经回路，学习新事物的效率下降，表现为思维模式的坍塌与固化

当前的大语言模型正处于一种危险的加速老化过程中：预训练阶段原本赋予了它理解世界的广阔可能性，但是过度依赖基于人类反馈的强化学习与合成数据微调，正在剥夺模型思想的丰富性。

?????????

解决这个问题的关键，在于如何在合成数据生成中引入有效的熵增机制。卡帕西认为，简单的随机化并不能解决问题，真正的出路在于构建能够像人类一样进行"白日梦"或"睡眠蒸馏"的架构。

在这个过程中，模型不是被动地预测下一个词，而是主动地对已有的知识进行重组、突变与逻辑自洽性校验，从而在不依赖外部输入的情况下，产生高质量、高熵值的新知识。

这是未来从单纯的文本生成向真正的认知核心进化的必经之路。

??Vibe Coding?????

尽管存在许多的技术挑战，但是软件3.0已经催生出了全新的开发范式。其中最具有代表性的，就是卡帕西提出的**Vibe Coding**（氛围编程）。这不仅是编程方式的改变，更是创造力经济学的重构。

通过自然语言的意图来取代语法细节，Vibe Coding实现了创造力的民主化，但是也引入了新的认知门槛。

Vibe Coding?????

Vibe Coding的核心理念，是让开发者忘记代码的存在。

在这个模式下，用户不再需要关注变量命名、内存管理或者语法结构，而是像产品经理一样，通过高层次的自然语言指令来描述功能需求、界面布局或者业务逻辑，然后将具体的代码生成全权委托给大模型操作系统。

卡帕西以他为孩子开发的 MenuGen 应用为例，生动展示了这一过程：一个没有任何前端开发经验的用户，只需要通过几轮对话，就能构建出一个能从照片中识别菜单且生成可视化图片的全功能 Web 应用。

这种范式的出现，意味着软件开发的边际成本在代码生成环节已经趋近于零。它彻底倒置了传统的程序员技能金字塔：曾经作为核心竞争力的精通 C++ 语法或者熟练掌握正则表达式，在 Vibe Coding 面前变得毫无价值。

取而代之的是：逻辑思维的清晰度、自然语言的精确表达、以及对系统模块的架构能力——这些成为了定义资深工程师的新标准。

?????????

对于像卡帕西这样拥有深厚计算机科学背景的专家，Vibe Coding 是效率的倍增器，让他能够以一人之力在几小时内完成过去需要团队几周的工作。

而对于完全没有技术背景的儿童或者行业专家，Vibe Coding 则更像是 赋能器，赋予了他们直接将创意转化为数字产品的上帝视角。

Vibe Coding ?????

但是卡帕西也敏锐地指出了 Vibe Coding 的黑盒陷阱：当用户习惯于直接接受 AI 生成的代码而不加审查时，代码库的复杂性并没有消失，只是被隐藏了。

一旦系统出现 Bug 或者需要进行深层的性能优化，不懂底层原理的 "氛围程序员 "将面临无法逾越的 "调试墙 "。

因此，未来的顶级开发者，将是那些既能利用 AI 来飞速构建原型，又能在必要时刻通过外科手术精准干预底层逻辑的 系统架构师 ——他们既懂 AI 的浪漫，也懂底层的现实。

?????????AGI???????

值得注意的是，在 AI 的众多应用领域中，编程成为了 AGI 渗透最深、产出价值最高的桥头堡。

从理论上看，AGI 应当首先席卷那些依赖知识检索与逻辑判断的领域，比如会计或者法律咨询。但是现实却呈现出戏剧性的偏差。卡帕西从信息模态与基础设施完备度的角度，深刻剖析了这个现象背后的必然性。

?????????

首先，代码本质上是高度结构化的文本。与自然语言的含混性不同，代码具有严格的语法逻辑和明确的执行结果，这与基于 Transformer 架构的文本预测模型具有天然的适配性。

其次，软件工程领域在过去数十年间积累了极其完善的数字化工具链：

- 我们有 IDE 用于即时反馈
- 有 Git 用于版本控制与差异比对
- 有 CI/CD 用于自动化测试

当 AI Agent 生成一段代码时，系统可以立即运行它，捕获错误信息，并且将错误回传给 Agent 进行自我修正。这种从生成到执行，再到反馈的闭环，在编程领域是毫秒级的；而在制作 PPT 或者法律咨询时，这个闭环要么不存在，要么极其漫长而且昂贵。

因此，卡帕西断言：编程是 **AI** 落地的完美风暴中心。目前的 AI 收入结构也证实了这一点——除去通用的聊天业务，绝大部分 API 调用与商业价值，都产生于代码生成与辅助开发工具。

这不是因为编程最简单，而是因为编程领域的数字化基础设施标准化程度最高，为 AI Agent 的介入铺平了道路。

这个现象也给其他行业带来了重要启示：如果想要复制编程领域的 AI 红利，首要任务不是训练更强的模型，而是构建类似于 Git 和 IDE 的标准化数字基础设施。

??????????????

但是 Vibe Coding 并没有解决软件开发的所有问题。随着核心业务逻辑的代码生成变得唾手可得，软件开发的瓶颈发生了转移——这就是卡帕西提出的集成之痛理论。

Vibe Coding 虽然解决了代码编写的问题，却无法自动解决 API 密钥管理、OAuth 认证、云端部署等胶水层的复杂性。这种复杂性的不可消除性，构成了新时代的技术护城河。

???????

卡帕西在构建 MenuGen 和 NanoChat 的过程中，亲身体验了这种集成之痛。他发现，尽管让大语言模型生成一个复杂的算法函数只需要几秒钟，但要将这个函数部署到互联网上供他人使用，却需要经历漫长而痛苦的胶水过程：

- 配置 Vercel 的部署环境
- 在 Clerk 中设置 Google 登录的 OAuth 回调地址
- 在 Stripe 中调试支付 Webhooks
- 处理不同云服务商之间不兼容的 API 接口文档等等

这些工作通常被称为 胶水代码 或者 配置地狱
。它们往往涉及跨平台的交互、敏感的权限管理，以及对物理世界状态的变更。由于这些操作往往发生在图形界面中，或者涉及到不可公开的私有密钥，目前的纯文本大模型操作系统难以直接介入。

??????

这背后遵循了 痛苦守恒定律 ： AI
消除了编写算法逻辑的痛苦，但是集成的痛苦并未消失，只是变得更加突出了。

这也揭示了软件 3.0
时代的新商业机会：谁能将这些复杂的胶水层抽象化、自动化，谁就掌握了新时代的流量入口。

未来的平台不应该只是提供 API 文档，而应提供 **MCP** 接口，让 AI Agent
能够像调用本地函数一样，自主完成云端服务的配置与集成。

在此之前，精通全栈集成与调试的工程师，依然拥有不可替代的非对称优势。

??????????????????

面对当前 AI 产业界盲目追求万亿参数的 Scaling Law 军备竞赛，卡帕西提出了一个极具颠覆性的终局预测：

通用人工智能的最终物理形态，并非是一个日益庞大的巨型怪兽，而是会经历一次剧烈的瘦身，最终演化为只需要十亿参数级别的纯粹认知核心。

这个预测不仅挑战了当下的主流技术路线，更揭示了智能与记忆分离的架构必然性。

??????????????

当前的大语言模型之所以体积庞大、动辄数万亿的参数，根本原因在于模型架构被迫承担了深度思考与海量记忆的双重职能 。

为了通过图灵测试，并且在各种基准测试中取得高分，模型不仅要学习推理的抽象逻辑，还要死记硬背互联网上浩如烟海的低质量事实。

而卡帕西深刻地指出： 互联网数据的极度低质与高噪声，是导致当前模型参数膨胀的罪魁祸首
。训练数据的信噪比极低，迫使模型必须构建巨大的参数规模来强行拟合这些冗余、琐碎且缺乏逻辑关联的信息。

这在信息论上是一种极大的浪费 ——相当于为了学会如何计算乘法，却背诵了整本九九乘法表的所有历史变体。

??????????????

未来的架构演进，将遵循严格的 认知与记忆分离原则 ：

??????????

所有的事实性知识、历史数据与专业文档，将被剥离出神经网络的权重，转而由检索增强系统（RAG）、向量数据库或者外部的知识图谱承担。这些外部存储系统擅长精确存储与检索，而且更新的成本极低，完全避免了大语言模型的幻觉问题。

???????????

在剥离了记忆负担之后，模型本身将退化为一个纯粹的认知核心。这个核心不存储具体的知识，只存储思考的算法——即如何分解问题、如何调用工具、如何验证假设，以及如何进行逻辑推演。

?????????

卡帕西大胆而且具体地预测：一个经过极致蒸馏、仅使用高质量教科书级数据训练的认知核心，它的参数量可能只需要十亿级别。

这个参数量级，大约相当于人脑前额叶皮层中负责核心执行功能的神经元突触规模。这个小巧而强大的引擎，将具备极高的推理密度与泛化能力。

它"知道自己不知道"——在面对2024年奥运会冠军是谁这类事实性问题时，它不会动用不可靠的模糊记忆去产生幻觉，而是能够生成正确的工具调用指令，主动去检索外部的可信数据源。

这种小参数、高智能、低延迟的架构，将彻底改变AI的部署成本结构，让它从昂贵的云端奢侈品，变为无处不在的基础计算单元。

????????2.0???

与认知核心的极简主义演进相呼应的，是个人计算**2.0**的回归。

卡帕西通过对半导体发展史与计算架构周期的宏观观察，判定我们正处于从云端分时共享向本地个人计算回归的历史拐点。这个转型并不是单纯的商业选择，而是由Transformer模型的计算物理学特性与Apple Silicon等新型硬件架构共同决定的物理必然。

Transformer?????

在目前的AI服务模式下，数十亿用户通过简单的聊天界面连接到OpenAI或者Google的集中式云端集群。这在计算架构上酷似20世纪60年代的大型机分时共享系统。在这种模式下，计算资源是高度集中的、昂贵的、高延迟的，而且用户必须将隐私数据上传到第三方服务器。

但是历史的车轮总是循环螺旋上升的。正如微处理器的出现终结了大型机时代，开启了个人电脑革命，AI领域也正在经历类似的个人计算2.0转型。

这个转型的物理基础，在于 **Transformer**

模型的计算特性是典型的内存带宽受限任务，而不是传统高性能计算的算力受限任务 。

在推理过程中，每生成一个 Token

，模型都需要将全部权重参数从显存搬运到计算单元进行一次矩阵乘法。这意味着，限制推理速度的核心瓶颈

不是 GPU 的计算核心数，而是 内存的传输带宽 。

Apple Silicon ???????

传统的 x86 架构在运行大模型时的效率极低，而 Apple Silicon 采用的 统一内存架构 彻底打通了 CPU 与 GPU

的内存隔阂。它提供了高达每秒几百 GB 的内存带宽，而且允许 CPU 和 GPU 直接访问同一块巨大的内存池。

这使得在单台消费级笔记本电脑上，流畅运行千亿参数的量化模型成为物理现实，而且能效比远超数据中心的

H100 集群。

?????????

基于这个现状，卡帕西描绘了未来的标准计算范式： 本地优先，云端兜底 。

每个人的设备上，都将常驻一个私有大语言模型操作系统。这个本地模型虽然可能只有 7B 或者 14B

参数，但是它完全掌握用户的个人上下文 ——它能读取你的本地文件、查看你的屏幕操作、索引你的邮件历史。

所有的基础推理、意图识别与日常任务处理都在本地完成。数据不出设备，既保证了隐私安全，又实现了零延

迟响应。

只有当遇到本地模型无法解决的极复杂问题时，本地大语言模型操作系统才会自动将任务路由到云端的超大模

型进行处理。

这种架构将彻底重构软件的分发与交互形式，使 AI 真正成为人类的数字延伸。

??????????????????????

架构的演进必然要求训练范式的革新。针对当前强化学习在解决数学证明、代码生成等复杂推理任务时的低效

与高噪声问题，卡帕西认为训练算法必须经历一次 类似从生物界的脊髓反射到人类大脑思考的质变 。

未来的训练将引入 系统二层面的反思机制 ，解决模型无法通过单次经验实现持续学习的根本缺陷。

??RLHF????

目前主流的 RLHF 或者 PPO

算法，在处理推理任务时会表现出极大的局限性。正如我们之前提到的，卡帕西将它比喻为 "

用吸管吸取监督信号 ":

模型在生成一段长达几千 Token 的推理链条后，往往只能获得一个单一的、二元的反馈信号。这导致了极高的方差——模型可能会因为蒙对了一个答案而错误地强化了中间一系列荒谬的推理步骤，或者因为最后一步的细微计算失误而全盘否定了前面完美的逻辑推演。

这种盲目的试错学习效率极低，需要天文数字般的样本量才能收敛。

???????

而人类的高级智能进化，并非通过在同一个坑里跌倒一万次，而是通过"吃一堑长一智"的反思与复盘。

当我们解决一个难题后，大脑会回溯整个思考过程，识别关键的转折点，强化正确的逻辑链路，修正错误的假设，并且将这个短期工作记忆固化为长期的突触权重。

但是目前的大语言模型训练流程，完全缺失了这个环节。模型在推理时的上下文是易失的——一旦对话窗口关闭，所有新获得的经验都会随风而散，模型重置为出厂状态，无法实现持续学习。

?????????

未来的训练范式革新将集中在两个核心的方向上：

????????????

模型应当具备在后台运行"梦境进程"的能力。在闲暇时段，模型会对白天产生的高质量交互数据进行深度的思考、重构与合成。

它会尝试用更优的路径去解决之前遇到的问题，生成高质量的合成数据，然后利用这些数据对自身权重进行微调。这相当于赋予了 AI "睡眠"的能力，让它能从短期经验中提炼出长期智慧，避免灾难性遗忘。

????????????

训练将不再仅依赖于最终的结果，而是转向对推理链条中的每一个中间步骤进行评价。

这需要构建更精细、更强大的奖励模型，甚至利用强模型来逐行审查弱模型的思考过程。通过密集的反馈信号，模型将学会如何思考，而不仅仅是如何输出答案，从而真正突破莫拉维克悖论中高维认知的最后一道防线，实现从概率拟合向逻辑推演的跃迁。

???Agent?????????vs??????

聊完了技术架构与训练范式，我们最关心的问题来了：Agent 到底什么时候才能真正落地呢？

针对 2024年被鼓吹为"Agent 元年"的市场躁动，卡帕西给出了极其冷静的修正：这并不是元年，而是 **Agent** 未来十年的开端。

他通过 "钢铁侠战衣 "与 "钢铁侠机器人 "的隐喻，划定了未来十年 AI 落地的两条截然不同的战略象限。

???Agent???????

卡帕西指出，当前业界对全自主 Agent
的预期严重超前于技术现实。一个能够像人类员工一样独立完成长期规划、工具调用而且零失误的 AI Agent
，在智能水平、多模态感知与持续学习能力上，仍然存在着根本性的断裂。

为了厘清发展的路径，他构建了一个二元战略模型：

????????????

这是短期内的最优解与唯一可行的商业模式。在这个模式下，AI
不追求完全的自主性，而是紧密地包裹在人类意识周围，作为 副驾驶 存在。

它的核心逻辑是通过 人机回路机制 ——人类负责高阶的意图指引与最终的决策审核，AI
负责底层的逻辑填充与代码生成。

Cursor 与 Perplexity
的成功，正是这条路径的最佳验证。它们并没有试图取代程序员或者研究员，而是赋予了人类极高密度的信息
检索与代码生成能力，让人类成为超级个体。

卡帕西断言，任何试图跳过这个阶段、直接构建全自主 Agent
的尝试，都将在可靠性的泥潭中通过燃烧资本而失败。

????????????

它指的是 AI 脱离人类的实时监控，独立在数字或物理世界中执行长周期的复杂任务。

但是目前这条路径面临着难以逾越的瓶颈：模型缺乏系统二的 "慢思考 "
能力与自我纠错机制。在解决从未见过的边缘情况时，它的概率性的本质导致它极其容易脱轨。

因此，在模型能够通过反思性训练实现自我迭代之前，全自主 Agent
只能在极低风险的沙盒环境中运行，无法应用于现实世界的复杂场景。

??????????????????

除了技术瓶颈，工程化落地还面临着 演示与产品的鸿沟 。

基于在特斯拉自动驾驶团队长达五年的经历，卡帕西提出了 AI 工程化领域最残酷的定律： 九的行军（ **March of 9s**）。

这个法则从数学上粉碎了 "演示即产品 "的幻想，揭示了从 90% 的演示成功率迈向 99.999%
的产品可靠性之间，横亘着指数级上升的工程难度。

????

在软件 2.0
时代，构建一个令人惊艳的演示是简单的。只要模型在某一次运行中成功，它就可以被录制下来并且在社交媒体上获得病毒式的传播。

然而，构建一个可交付的商业产品，要求模型在所有情况下都必须可靠运行。卡帕西将这个过程量化为九的行军 ：

- 第一个九（ **90%**）：早期的 Waymo 演示或者现在的 Agent Demo，看起来很美，但是在现实中每十次操作就会失败一次，完全不可用
- 第二个九（ **99%**）：需要几倍的数据清洗与边缘场景的挖掘
- 第三个九（ **99.9%**）：模型开始遭遇长尾分布的残酷打击，解决这 0.9% 的问题所需要的算力与数据，往往会超过前 99% 的总和
- 第五个九（ **99.999%**）：这是自动驾驶与金融级 AI 的准入门槛。在这个阶段，错误往往由极度罕见的物理现象或对抗性样本触发

卡帕西警告，当前的软件 3.0应用大多还停留在 第一个九的狂欢中。许多初创公司低估了尾部风险的破坏力。对于容错率为零的领域， AI 在未来很长一段时间内将长期处于辅助确认的地位。

那些试图忽略 "九的行军法则 "、直接将概率模型部署于安全关键系统的行为，不仅是不负责任的，更在工程逻辑上是注定无法收敛的。

????????????

回到我们之前提到的模型坍塌风险，这个问题在 Agent 的长期演进中显得尤为关键。

随着大语言模型生成的合成数据开始充斥互联网， AI 面临着一种类似热力学 "热寂 "的终极风险。卡帕西从信息熵的角度论证：如果缺乏人类新数据的注入，封闭的 AI 自我训练循环将导致智能的退化。

?????????

模型坍塌的物理机制在于： AI 生成的数据，本质上是训练数据分布的 平均值或模态 。

虽然单个样本看起来语法完美、逻辑通顺，但是从统计学上看，这些数据的方差与熵远低于真实的人类数据。 AI 倾向于输出安全、平庸而且高概率的回答，而过滤掉了人类思维中那些疯狂、边缘但是极具创造性的低概率样本。

如果 GPT-5 使用 GPT-4 生成的数据训练， GPT-6 使用 GPT-5 的数据训练，就会导致数据分布的尾部彻底消失，模型将收敛到一个极其狭窄的智力子空间中。

卡帕西将它比作 "哈布斯堡王朝的下巴 "——因为基因多样性的丧失而导致了畸形。

????????

在这个推演下，人类产生的数据成为了对抗 AI 熵减的 唯一负熵源 。

人类独特的不连续思维与非理性创造，是维持数字宇宙多样性的关键燃料。因此， AI
不会在一个封闭的数字真空中通过自我博弈无限进化成神。相反，
人机共生不仅是伦理选择，更是维持智能系统热力学平衡的物理必要 。

未来的数据工厂，将是人类负责生产高熵的灵感， AI 负责将它扩展为低熵的产品，二者缺一不可。

??????????????????????????

面对这场不可逆的技术革命，我们该如何重构基础设施、教育体系以及人类自身的定位呢？卡帕西给出了一个
清晰的应对策略。

??????????????Agent??

互联网必须经历一次从为人设计向为 Agent 设计的底层协议重构。

当前的互联网基础设施，充斥着 CSS 样式、弹窗广告与 JavaScript 动态加载的 HTML 页面 ——它们对于 AI
Agent 而言是极度低效的噪音。为了适应大语言模型操作系统的感知模式，新的标准与协议正在崛起。

llms.txt??

它被定义为 "AI 时代的站点地图 "。

在当前的 Web 架构中， robots.txt 协议主要用来告诉爬虫什么不能抓取 ——
这是一种防御性的、做减法的逻辑。

而 llms.txt 是一个位于网站根目录的纯文本文件，专门为大语言模型操作系统的 "视觉皮层 "
设计。它剥离了所有为人眼设计的 UI 噪音，仅仅保留了核心的、高信噪比的知识拓扑与语义链接。

对于企业而言， llms.txt 协议是 SEO 向 AIO（ AI Optimization
）转型的关键抓手。如果一个文档库无法被大语言模型操作系统高效地解析，那么在未来的代码生成与智能问
答中，该产品将在认知层被彻底隐形。

这不仅是格式的变更，更是信息分发权力的转移 ——从争夺人类的眼球转向争夺模型的上下文窗口。

MCP??

如果说 llms.txt 协议解决了 Agent 的阅读问题，那么 MCP 的目的是为了解决 Agent 的行动难题。

当前的痛点在于：Agent 在尝试调用外部工具时，往往面临着接口不标准、鉴权复杂、参数幻觉等问题。每个 SaaS 平台都需要为不同的模型编写特定的插件，这导致了极高的集成熵。

而 MCP 标准会将互联网服务封装为一组标准化的、具有明确类型定义的工具描述符，让 Agent 能够像调用本地 Python 函数一样，通过 JSON Schema 精确地调用云端 API。同时，它允许 Agent 实时查询系统的当前状态，从而在规划阶段将物理世界的约束纳入考量。

卡帕西推演，未来的互联网将分裂为两个平行的维度：

- 表层网络 为人类设计，强调视觉体验、情感交互与多媒体呈现
- 深层网络 为 AI 设计，由 llms.txt、MCP 接口与向量索引构成 —— 这是一个纯粹的、高速的、无摩擦的数据交换网络

人类在这个网络中是旁观者，而数以亿计的 Agent 将在此以光速进行信息套利与任务协作。

??????

此外，鉴于大语言模型操作系统的内存具有稀缺性与昂贵性，如何将现实世界的海量信息高效地加载到内存中，成为了新的工程挑战。

卡帕西特别提到了 Gitingest 这类 上下文构建器 工具的兴起。它们充当了现实世界与模型内存之间的预处理器或编译器。

例如，Gitingest 能够将一个包含数万个文件的复杂 GitHub 仓库，通过树状结构分析、去重、压缩，最终编译为一个单一的、对提示词友好的文本流。

如果说大语言模型是 CPU，上下文窗口是缓存，那么 上下文构建器就是内存控制器 —— 它决定了哪些信息有资格进入思考的核心区域，直接决定了推理的质量与成本。

??????????????

面对 AI 对知识获取方式的降维打击，卡帕西提出，教育体系必须经历一次从知识灌输向思维训练的彻底重构。未来的教育不再是填充水桶，而是点燃火焰。

???????????????

AI 导师的核心价值，不在于解答问题，而在于提问。

卡帕西理想中的 AI 教育，不是 ChatGPT 式的直接给出答案，而是模拟一位耐心的苏格拉底。AI 通过多轮对话，精确诊断学生的 最近发展区（也就是学生刚好能理解但是又具有挑战性的认知边界），然后根据学生的反馈，实时生成个性化的知识坡道。

如果学生不懂导数，AI 不会背诵定义，而是会退回到速度与加速度的物理直觉；如果学生理解了，AI 会立即拆除脚手架，推向更抽象的极限概念。

这种自适应的颗粒度，是人类大班教学所无法实现的。

??????????????

卡帕西极力推崇 第一性原理的教学法 ，并且以自己的 micrograd 项目为范本。

主流的深度学习教材，往往充斥着 PyTorch 的复杂 API、CUDA 优化与工程细节 ——
这些在他看来都是噪音。而 micrograd 仅用一百行的 Python
代码，就从零构建了一个支持反向传播的神经网络引擎。

教育的任务，就是从复杂的现实信号中提取出 "基本频率 "——也就是事物最底层的、不变的因果逻辑。

一旦学生掌握了基本频率，所有的工程变体都只是这个核心逻辑的展开。

在 AI
能够生成完美但是虚假内容的时代，唯有掌握了第一性原理的学生，才具备逻辑校验的能力。他们不需要记忆
公式，因为他们能随时推导公式。这种推导能力，是人类在认知层面上对抗 AI 幻觉的最后一道防线。

???AGI???????

最后，也是最核心的一点：人类在 AGI 时代的终极定位是什么呢?

当逻辑、编码、甚至艺术创作都可以被算法低成本的生成时，人类的价值究竟何在呢? 卡帕西从技术哲学的高
度给出了答案。

他认为，人类的核心价值在于:

1. ???????

虽然 AI 在执行层面已经通过软件 2.0和 3.0实现了对人类的超越，但是在意图的生成上， AI 依然是空心的。

- AI 可以写出完美的代码，但是它不知道为什么要写这个软件
- AI 可以生成绝美的画作，但是它没有想要表达某种情感的冲动

这意味着，未来的稀缺资源不再是解决问题的能力，而是 定义问题的能力
。人类应当放弃在工具层面的内卷，转而专注于磨练提出好问题、定义高价值目标以及审美判断的能力。

2. ?????????

同时，模型坍塌理论告诉我们: AI 倾向于收敛到概率分布的中心，生成同质化、低熵值的内容。

为了维持整个智能系统的热力学平衡，人类必须成为 负熵的提供者
。这意味着我们应当鼓励那些非理性的、直觉的、甚至是疯狂的思维跳跃 ——
这些在统计学上被视为离群点的数据，正是 AI 无法通过自我迭代产生的创新火种。

最强的智能形式，不是纯粹的硅基 AGI，而是 高熵的人类意图加上低熵的 AI 执行的共生体
。在这个共生体中:

- 人类负责在可能性的荒原中插旗
- AI 负责在旗帜下铺设高速公路

??

最后，我们总结一下卡帕西的核心思想：

他并非在预测 AGI 的速成，而是揭示了计算范式从确定性代码向概率性意图转移的物理必然。

软件 3.0的确立与大语言模型操作系统架构的成型，标志着 AI
已经从单一的工具演变为新的底层计算平台。但是工程的现实表明，当前模型的锯齿状智能与全自主 Agent
所需的可靠性之间，仍然存在显著断层。

九的行军法则 告诉我们：在解决反思性训练与多模态交互之前，
钢铁侠战衣是短期内唯一可行的商业落地路径 ，而钢铁侠机器人则属于长期的研发愿景。

在基础设施层面，随着 llms.txt 与 MCP
协议的推进，互联网正在经历从人读向机读的标准化重构。同时，硬件架构将迎来从云端集群向 Apple Silicon
统一内存的回归，配合十亿参数的认知核心，实现本地化的低延迟推理。

在这个进程中，人类的角色将被严格定义为系统的负熵源与意图定义者。面对合成数据可能引发的模型坍塌风险，人类所提供的非连续性思维与基于第一性原理的逻辑校验，构成了维持智能系统热力学平衡的必要条件。

这不是一场科幻叙事中的技术突袭，而是一场基于物理规律的技术迭代 ——一场漫长但是终将到来的革命。

感谢收看本期视频，我们下期再见！

Revision #2
Created 2026-01-17 14:02:18 UTC by Colin
Updated 2026-01-17 14:02:59 UTC by Colin