

万字赏析 DeepSeek 创造之美：DeepSeek R1 是怎样炼成的？ | 荐读

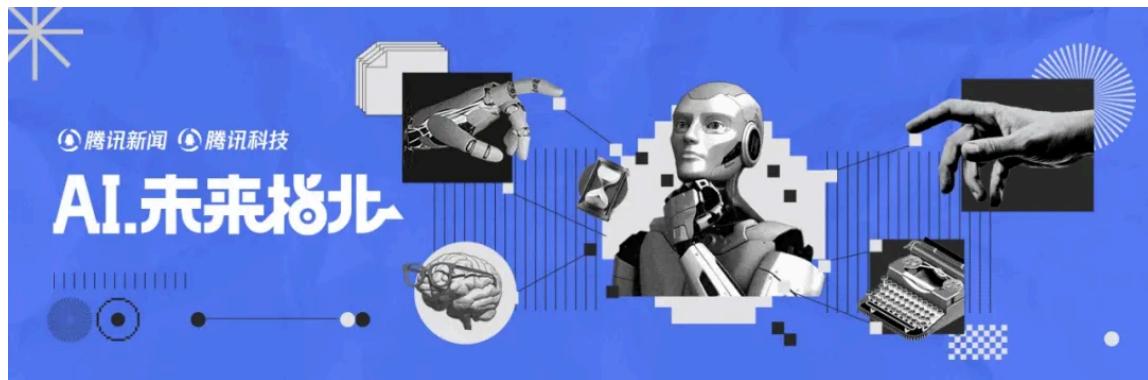
腾讯科技 2025年02月15日 20:57 北京

以下文章来源于真格基金，作者与你同在的



真格基金

专注早期投资，欢迎投递商业计划书至 dream@zhenfund.com



【编者按】当一项技术成为现象级话题，信息过载往往带来认知负担。DeepSeek的迅速崛起催生出海量解读，但真正具备信息增量的内容可能不足千分之一。腾讯科技以“价值密度”为筛，从全球视野筛选出5-10篇兼具技术纵深感与行业前瞻性的深度解析，为关注AGI进程的读者过滤噪音、提炼精华。

作者 Monica.im产品合伙人 张涛

1

最好的致敬是学习

今年春节，我在上海过年，没有回重庆，就通过视频给爸妈拜年。我给我妈说新年快乐时，听到我爸在旁边喊道：「你快问一下张涛，那个梁文锋是不是真的那么牛逼啊？」

今年的 DeepSeek 和 R1 话题真的是破圈的程度非常高，甚至像重庆这样的二线城市的老头老太太们都在关注这些话题，且真心关心它背后的原理到底是什么。

首先我们回顾一下这些发生的事情，理清时间线，确保大家对这个事情有共同的认知。

发生了什么？

去年 11 月 20 日，DeepSeek 在官方 Twitter 上发布了 R1 Lite Preview。当时发布的 R1 Lite Preview，实话说，离现在的影响力连 1% 都谈不上，可能只有万分之一。只有去年 11 月 o1 发布后，有一些人试图复现 o1，这时他们可能对这个 R1 Lite Preview 感兴趣，甚至有人基于它进行一些蒸馏和 SFT 的工作。但这些工作在学术界内部并未出圈。

接着到 12 月 26 日，发布了 DeepSeek V3。相比 R1 Lite Preview，它的影响力就更大了一些。稍后我会举一个例子证明，至少在学术界，它是有出圈的。

第三个时间点是 1 月 15 日，DeepSeek 发布了他们的 APP。当时如果大家仔细看，会发现 15 号发布的 APP 中，已经有了 DeepThink 模式。

但是 DeepThink 这个模式一直无人在意，国内没有，国外也没有。如果大家能回到 15 号的语境下，其实可以理解为什么。当时不仅是美国，包括我们在内，大家关注的新闻基本只有一个——特朗普即将登基。公众的注意力还更多集中在这些政治事件上。直到 20 号，R1 才正式发布，一方面是相关论文公开，另一方面是模型权重的开源。

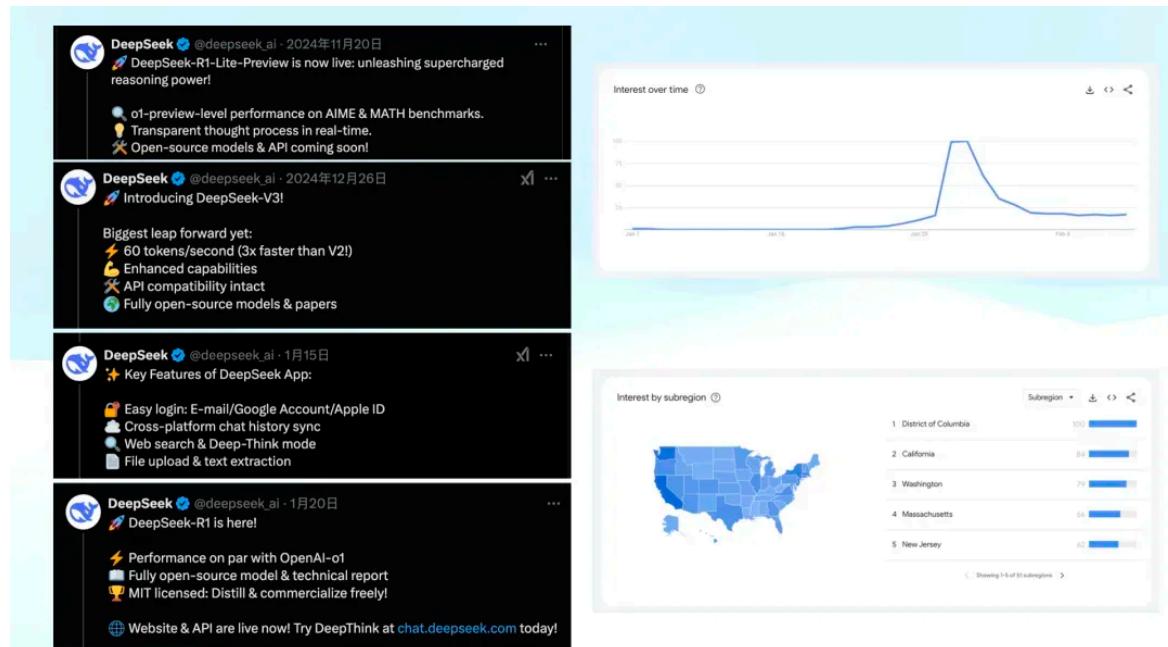
从时间线来看，R1 最早的苗头实际上在去年 11 月份就已经出现，并非一夜之间爆发的。在这个过程中，还有几个关键节点需要关注，包括 V3 的重要性——这是我们今天讨论的核心话题之一。

接下来，我给大家看一个有趣的现象。在 Google 上搜索 DeepSeek 这个关键词，可以看到其关注度的起点是在 1 月 20 号，也就是 R1 发布之后。随着学术界开始小范围讨论，20 号到 24 号、27 号之间热度逐渐升温，直到 27 号，英伟达及一众美国 AI 概念股「砸出巨坑」，DeepSeek 的搜索量也随之达到顶峰。即使热度在一周后有所回落，相比 20 号之前接近 0% 的状态，现在仍然维持在 20% 左右。这说明，尽管流量有所回调，但关注度并未完全消退。

接下来是一个有趣的话题，不知道大家能不能猜到：**在美国，按行政区域划分，哪个地区对 DeepSeek 关注度最高？**

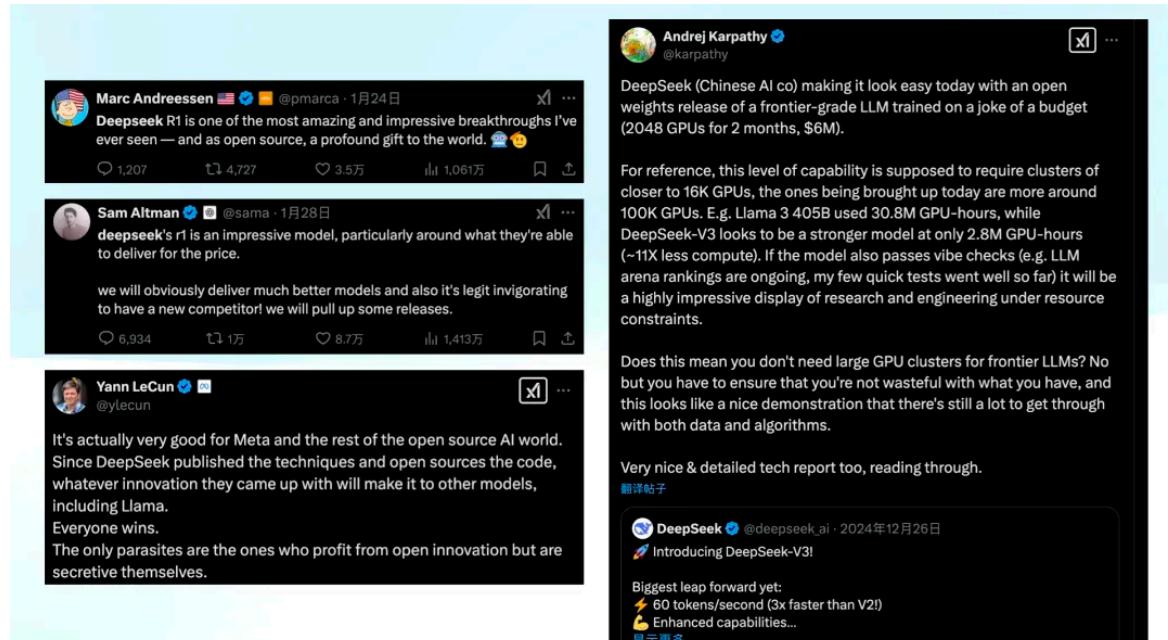
我当时在看数据时觉得非常有意思。本以为会是加州，毕竟 AI 相关研究人员主要集中在那，但实际上，最高关注度出现在华盛顿特区。可以想象，在 27 号市场震荡后，华盛顿的一众政客疯狂在 Google 上搜索 DeepSeek 试图搞清楚 DeepSeek 到底是个啥？

之后的排名则较为正常：加州、华盛顿州这些传统 IT 公司和 AI 研究机构集中的地区关注度较高。但 DeepSeek 这么高的关注度确实值得一提。



前面我们讨论的是发布方的反应，现在来看美国社会中精英 KOL 们的反馈。大家可能还记得，我之前提到，12 月 26 号 V3 发布时，**相比 R1 Lite Preview，这次在学术圈真正「破圈」了。**

为什么这么说？可以看这张图，右侧是 Andrej Karpathy 的 Twitter。当天，他发布了一条非常长的推文，详细介绍了 V3，并评价其为「very nice and detailed tech report（非常出色且详细的技术报告）」。可以确定的是，12 月 26 号 V3 发布时，它已获得美国主流学术圈的认可，只是当时很多人尚未意识到 V3 的更深层次价值。



我们再回到春节期间的「炸街」时刻。

第一次让我意识到美国舆论开始转变的节点是什么？当时，我们都在各种群聊里，应该有不少人看到了。那天，我特别兴奋地转发了 Marc Andreessen 的 Twitter。大家知道，他通常对中国科技持激进甚至有时候是轻蔑态度。

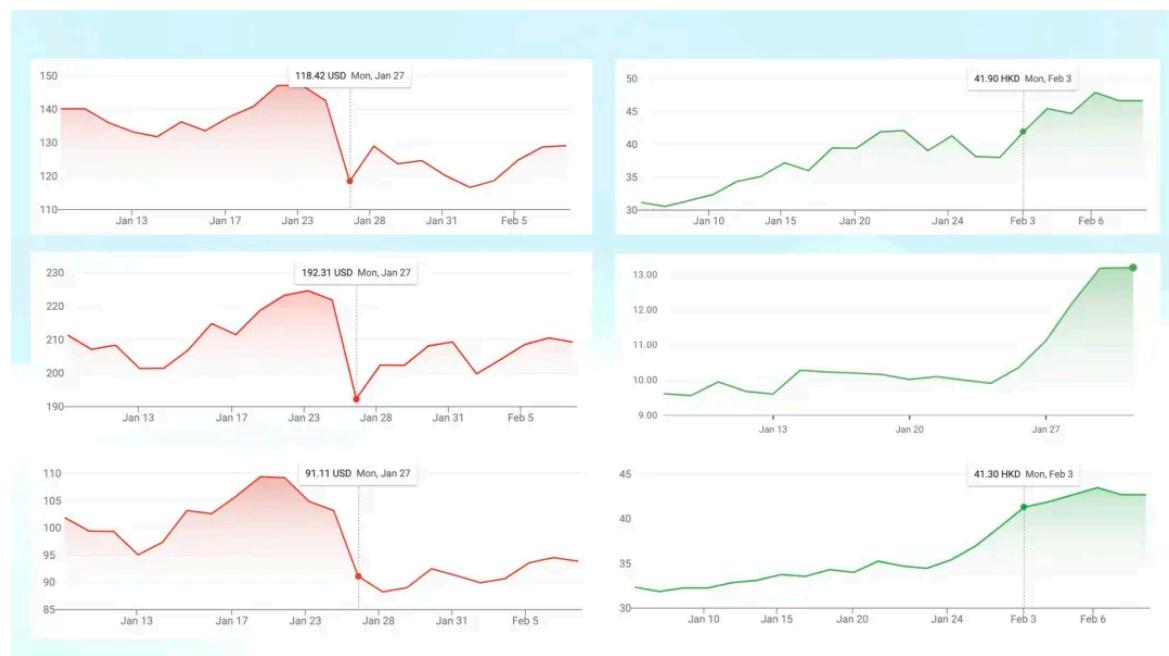
24 号时，他开始连发推文，惊叹这是什么东西？太炸裂了吧。之前他会批注比如，这太厉害了，但请注意，我说它好，不代表我很高兴，我是觉得它很危险。但仅仅一天后，他的语气彻底改变了。这条推文没有任何负面情绪，而是完全正面的表达。

到了 28 号，Sam Altman 也不得不出面表态，尽管说得别别扭扭的，比如暗示「其实我是想开源的，但组织上不允许」之类的托词。杨立昆也承认 R1 的影响力和研究质量，不过试图将话题引向「开源」的胜利，而非某个国家的胜利。

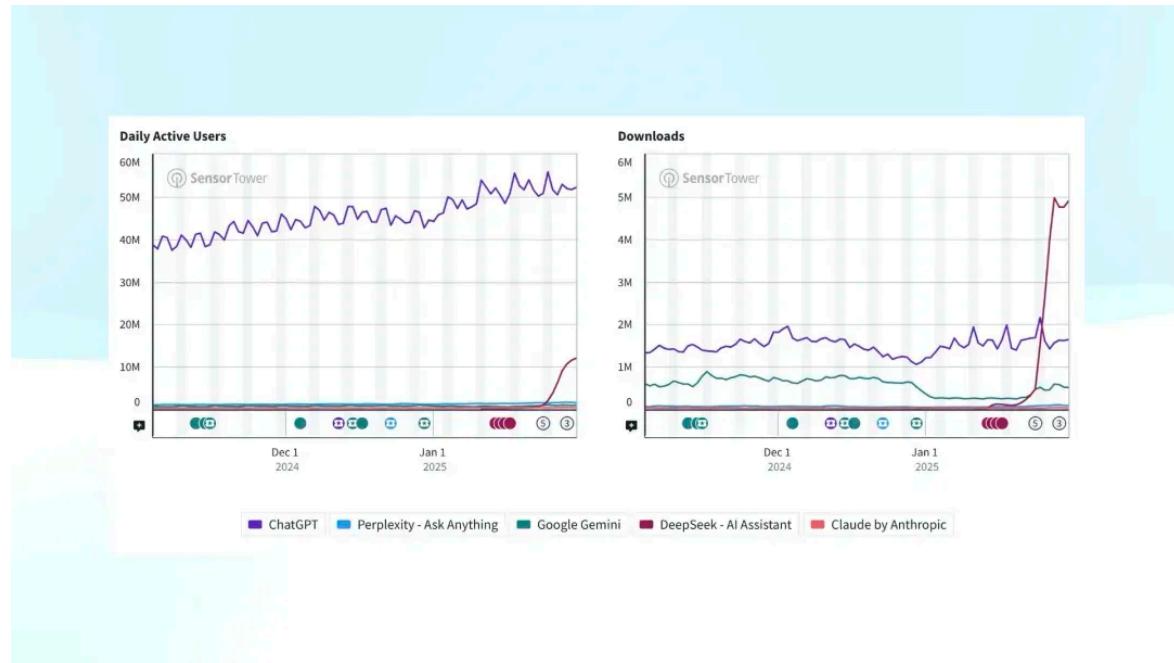
无论如何，这项工作已经得到了美国 AI 界顶级领袖的认可。无论是对质量还是对这一事件本身的认可，其影响力已经显而易见。至于这个影响是好是坏，原因是什么，这是我们接下来要探讨的话题。

到了 2 月 2 日或 3 日，仍然有一些持反对意见的人在称这项工作是 DeepSeek 雇佣水军炒作。实际上，主流圈子对此并未关注。但事实就摆在眼前，无需辩证。

最值得注意的是 1 月 27 日这一天，股市剧烈波动。左边，从上到下依次是英伟达、台积电、美光，股价瞬间砸一个天坑。右边，从上到下是中芯国际、360 和金山云，股价却突然上涨，仿佛呈现出一种「东升西落」的趋势。**这说明 R1 的出现对真实世界的影响同样不容忽视。**



在 Sensor Tower 的数据中，左图显示的是 DAU，其中紫色线代表 ChatGPT。而在底部，原本较小的其他竞品，如 Claude 和 Perplexity，虽然一直是 ChatGPT 的跟随者，其用户占比相对恒定且较低。但在 1 月底的几天里，DeepSeek 出现了显著增长，占比大约达到 20%。



左图显示的是 DAU，右图是新增下载量

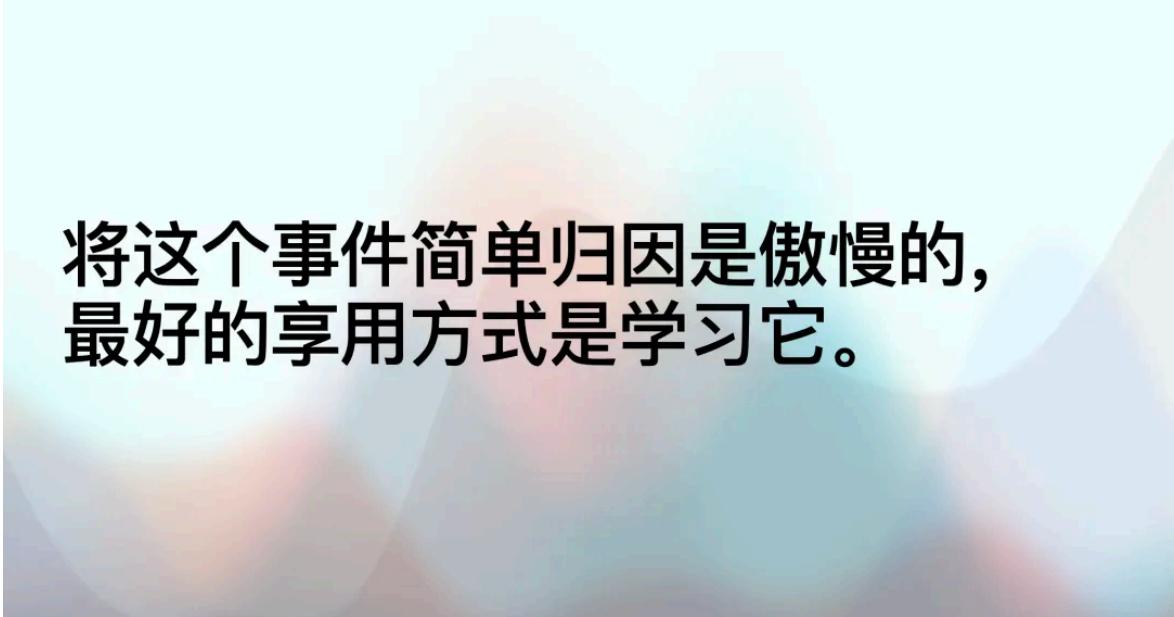
右侧的图展示了新增下载量，在突破某个临界点后继续增长。我截图的时候比较早，昨天查看最新的数据，发现左图趋势持续向上，虽然右图的下载量有所回落，但仍然高于 ChatGPT。目前来看，这一趋势仍在持续。

不管是业界领袖的认可，股票市场的反应，还是真实用户的选择，都证明了这一事件的影响是真实存在的，并且具有用户价值。这是过去半个月里发生的重要变化。

接下来，我们回到为什么要组织这次学习分享。这件事对我而言非常重要。我从 1 月 23 日开始关注，并频繁阅读中美两地的各种言论，包括圈内和圈外的讨论。随着这个事件的破圈，越来越多非专业人士开始关注，人们对其归因的方式也变得过于简单。

比如，有人认为这是因为中国人工便宜，从而把美国顶尖科技的成本打下来了。也有人说这是抄袭，只要复制就能成功。此外，还有另一种叙事，即某个不知名的小团队突然创造了全球顶级的科技创新。然而，无论哪种归因，都显得过于表面，脱离了产品，缺乏对技术本身的深入理解。

这个事件简单归因是傲慢的，最好的享用方式是学习它。



将这个事件简单归因是傲慢的， 最好的享用方式是学习它。

当面对一个如此重大的事件时，仅仅存入记忆是不够的。去学习它、理解它，搞清楚为什么会产生这样大的影响力才是本次分享的核心目的。

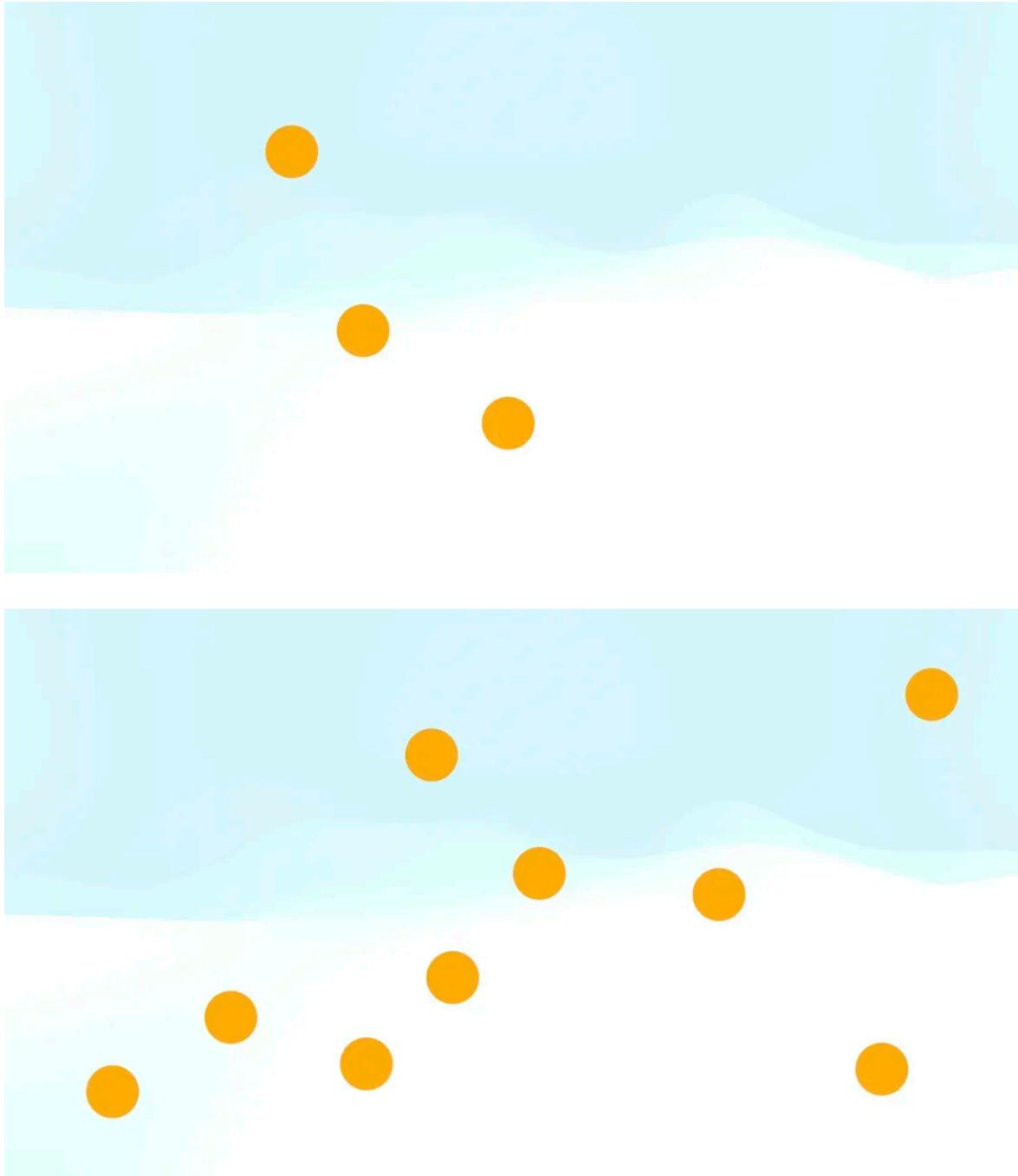


什么是推理模型？

对于大部分听众来说，我们不是专业研究算法或工程的，我自己是做产品的。我们首先需要解决的一个基本问题是：什么是推理模型？

我们已经有大语言模型了，为什么还需要推理模型？

我这里准备了一个小测试，不知道大家是否了解，人脑有一个特殊的能力叫「数量识别」。这不是简单的数字识别，而是对数量的直觉判断。比如，我会切换一张图片，你需要在一秒内告诉我上面有多少颗黄球。一般来说，一个正常人只能识别 6 及 6 个以内的个数。好，现在准备——3、2、1，切换！



「数量识别」测试

通过这个实验，我们可以发现，在几千年的进化后，人类在数数时并不一定需要一个个地数，而是在一定范围内可以凭直觉判断。这一现象背后的认知机制，也是推理模型的一个重要基础。

语言模型，特别是大语言模型，有一个特性相似的特点：模型在给出答案时，一般会直接做出回答，尽管这种方式往往容易出错。举个经典例子，比如 CoT（Chain-of-Thought，思维链），Jason Wei 强调了一个重要的思想：**模型需要更多的 token 来进行思考。**

“Model needs more tokens to think”

Chain-of-Thought Prompting Elicits Reasoning in Large Language Models

Jason Wei Xuezhi Wang Dale Schuurmans Maarten Bosma
Brian Ichter Fei Xia Ed H. Chi Quoc V. Le Denny Zhou
Google Research, Brain Team
{jaacmwei, dennyzhou}@google.com

Abstract

We explore how generating a *chain of thought*—a series of intermediate reasoning steps—significantly improves the ability of large language models to perform complex reasoning. In particular, we show how such reasoning abilities emerge naturally in large language models via simple chain-of-thought prompting. Chain-of-thought prompting, where a few chain-of-thought demonstrations are provided as exemplars in prompting. Experiments on three large language models show that chain-of-thought prompting improves performance on a range of arithmetic, commonsense, and symbolic reasoning tasks. The empirical gains can be striking. For instance, prompting a PaLM S-40B with just eight chain-of-thought exemplars achieves state-of-the-art accuracy on the GSM8K benchmark of math word problems, surpassing even finetuned OPT-3 with a verifier.

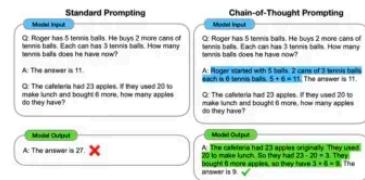


Figure 1: Chain-of-thought prompting enables large language models to tackle complex arithmetic, commonsense, and symbolic reasoning tasks. Chain-of-thought reasoning processes are highlighted.

Peak 曾经给我说过一个直击本质的观点。大家知道，语言模型的本质是激活一个庞大的神经网络矩阵。当输入一个 token 时，它能够激活矩阵中的某些部分，但这种激活是有限的。当输入更多的 token 时，能够激活的部分也更多，信息量随之增加。因此，更多的 token 意味着模型能够得到足够信息，从而做出更为精准的决策。

模型需要更多 token 来「思考」，这也促使我们提出了推理模型（Reasoning Model）的概念。

什么是推理模型？比如，我们可以用一个例子来说明。假设我们问一个问题：「从望京西到西直门坐地铁需要几站？」一个「直接回答型」的模型可能会像下图左边直接回答：「九站」。

而推理模型则会做出右边的回答。它首先会考虑多种换乘路线，接着比较各路线的换乘站数，最后综合得出最佳方案。**推理模型不仅仅给出答案，它还会展示其思维过程。**

从望京西到西直门坐地铁要几站？

9 站，分别是：

望京西 → 望京 → 芍药居 → 光熙门 → 柳芳 → 东直门 → 雍和宫 → 安定门 → 鼓楼大街 → 西直门

要知道望京西到西直门要几站，首先得知道有几种换乘路线：

1. 13 号线转 2 号线，共 9 站
2. 15 号线转 5 号线再转 2 号线，这样就是 2 站 + 5 站 + 4 站，11 站
3. 还可以.....

这样综合看起来，最少站数的换乘方案应该是 1，共 9 站

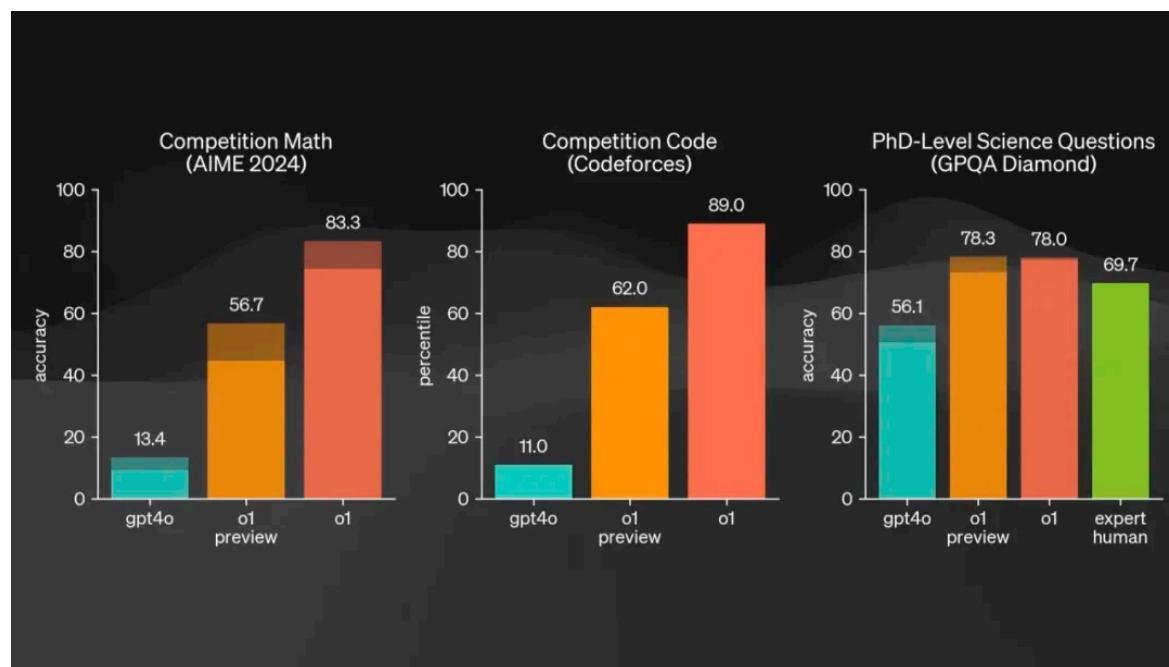
大家可能会觉得，这和 CoT 有相似之处。那么，推理模型和 CoT 有什么区别呢？如果你已经习惯了使用 ChatGPT，你可能会直觉地认为，推理模型不就是 CoT 吗？我直接写个 CoT，让它一步步进行推理就行了。比如，针对刚才的地铁问题，我们可以跟模型讲：请先列出所有可能的换乘路线，再计算每条路线的换乘站数，最后综合得出最优答案。

“这不就是 CoT 吗！”

“怎么说呢……”

如果你愿意为每个问题都写出如此详细的 CoT，这也是可行的。

但这里有个问题。我们来看一下去年震惊整个业界的事件——当时 ChatGPT 的 o1 系列模型发布，它刷新了多个记录。比如在数学领域，它的得分从 13 分直接跃升至 56 分和 83 分；在写代码方面，它从 11 分飙升至 89 分，快把榜单刷爆了。PhD 级别的科学问题虽然提升没有那么显著，但也极为恐怖。如果你经常看论文，就会明白把基准测试刷新一两分都能发表论文。



其中，最让人惊讶的是 PhD 级别的成绩，尤其是右边绿色，代表的是人类专家的得分。ChatGPT 已经超过了真正的 PhD。

推理模型的本质是让模型自己构建 CoT，并将前面推理的步骤展示出来。虽然你也可以自己手动编写 CoT，但问题是：我们能否对每一个问题都写出完整的 CoT 呢？

比如，下面这两个问题，分别出现在 2024 年 AI 的基准测试和 PhD 级别的评测中。假如你还是一个数学或物理 PhD，或许能写出 CoT，但对于绝大多数人来说，能够把每个问题的思维过程一步步写出来不容易。

这该怎么 CoT.....

- 艾丽丝和鲍勃玩一个游戏。一堆包含 n 个代币摆在他们面前。玩家轮流进行操作，艾丽丝先操作。每次轮到玩家时，玩家都可以从堆中拿走1个或2个代币。最后拿走代币的人获胜。找出小于或等于100的正整数 n ，是否存在一种策略，确保鲍勃在无论艾丽丝如何操作的情况下都能赢得游戏。
- 两个能量分别为E1和E2的量子态，其寿命分别为 10^{-9} 秒和 10^{-8} 秒。我们希望清晰区分这两个能级，以下哪个选项可能是它们的能量差值，使得二者能够被明确分辨？

左边是 AIME 2024 的测试题

右边为 PhD 水平 GPQA Diamond 的测试题

这就是推理模型的必要性。它能帮助我们处理一些特定领域的问题。举个例子，推理模型非常适合解答谜题，比如翻译二战时期的密码电文，或者进行数学证明，解决复杂的决策问题，甚至是开放式问题。推理模型不仅给出最终答案，还会展示思考过程。

而对于一些简单的知识性问题，比如「哪个是中国的首都？」，我们显然不需要使用推理模型，直接给出「北京」就可以。很贵，而且想得多容易搞错。

为什么我们需要推理模型

适用&不适用场景

- 谜题、数学证明
- 复杂的决策问题
- 开放式答案
- 需要显式的多步思考
- $1+1=?$
- 知识性回答

推理模型有其适用场景。为什么它在我们行业中如此重要呢？原因有两点。

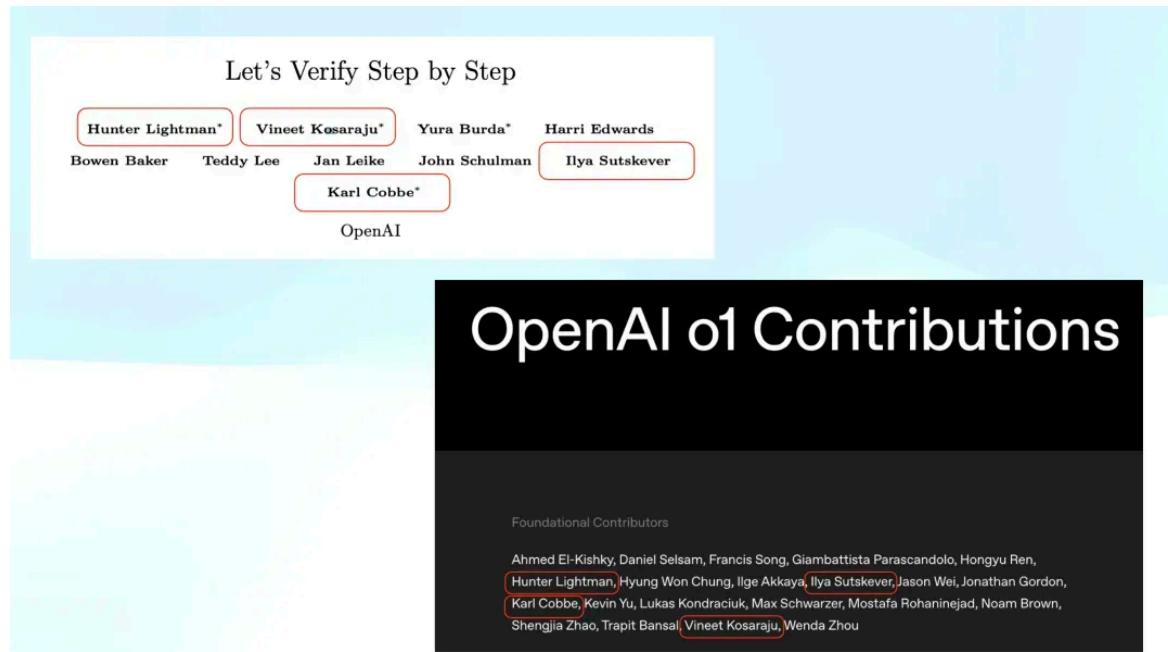
首先，大家看到的在数学、写代码和 PhD 级别领域的突破，预示着大语言模型的应用不仅仅局限于作为聊天机器人，它已经可以进入到加速国家科技研究的领域。**推动 AI 发展的各大厂商都在追求 AGI，甚至更高层次的 ASI（超人工智能）。**

另外一个方面是，至少目前从 R1 的结果，以及之前大家在使用 o1 的过程中，我们发现，尽管训练推理模型主要是为了处理数学、物理和写代码的问题，但当一个模型掌握了推理过程后，它在处理一些更广泛的场景时，包括写作和对话，也变得更加有逻辑和思维能力。

从去年下半年开始，推理模型成为了大家都想攻克并解决的方向。

那问题就来了，如何复现 o1 呢？我们首先得看 o1，尽管今天我们没有足够时间深入探讨，但我想分享一个特别有趣的事情。

o1 在发布那天公布了核心贡献者名单。而 OpenAI 在 2023 年彻底「沉默」前还发布了最后一篇论文，名为《Let's Verify Step by Step》。这篇文章讲述了通过将一个问题逐步拆解，并对每一步进行打分进行强化学习来训练一个模型。当时，很多人在 o1 发布后回顾 OpenAI 的工作，发现去年公开的最后一篇研究论文就是这篇，后再也没有新发布的工作。



很多人就认为《Let's Verify Step by Step》可能是 o1 复现的关键，而他们还公布了一个 PRM800K 的数据集。这个数据集的格式就是下面截图所示：题目给出了每一步的推理过程，并对每一步进行打分，标注为 positive、neutral 或 negative。OpenAI 公布了这个 PRM800K 数据集。

The screenshot shows the PRM800K dataset interface. It features a header "PRM800K: A Process Supervision Dataset" and links to "[Blog Post]" and "[Paper]". Below this, a text block explains the dataset: "This repository accompanies the paper [Let's Verify Step by Step](#) and presents the PRM800K dataset introduced there. PRM800K is a process supervision dataset containing 800,000 step-level correctness labels for model-generated solutions to problems from the [MATH](#) dataset. More information on PRM800K and the project can be found in the paper." A note below states: "We are releasing the raw labels as well as the instructions we gave labelers during phase 1 and phase 2 of the project. Example labels can be seen in the image below." An example math problem is shown: "The denominator of a fraction is 7 less than 3 times the numerator. If the fraction is equivalent to $\frac{2}{5}$, what is the numerator of the fraction? (Answer: 14)" followed by a series of reasoning steps with accompanying icons:

- Let's call the numerator x.
- So the denominator is $3x - 7$.
- We know that $x/(3x-7) = 2/5$.
- So $5x = 2(3x-7)$.
- $5x = 6x - 14$.
- So $x = 7$.

想一想，如果你是一位在 o1 发布后开始研究 o1 的研究者，看到他们的论文，你很容易联想到 o1 可能采用了类似的 PRM 模型（Process Reward Model），很难不往这个方向去推测。

如果你搜索如何复现 o1，我随便搜索了一下，点击了那篇排名最高的文章，它是去年 12 月 30 日发布的。文章提到，o1 发布后，国内陆续出现了很多类似 o1 的模型。那时提到的 R1 还不是我们现在所说的 R1，而是 R1 Lite Preview，包括 Kimi Math 的相关技术。

文章提到，业界大致分为两个派系：**一个是树搜索派，另一个是蒸馏派**。树搜索派类似于 OpenAI 提到的《Let's Verify Step by Step》的细分。而蒸馏派则是使用已有模型，如 o1、r1、Kimi Math 做蒸馏。值得注意的是，当时没有提到我们现在看到类似 R1 的纯强化学习派，因为当时整个业界普遍认为 OpenAI 是采用了这种方法。我知道在硅谷和国内，很多公司都在准备类似 PRM800K 这样的数据集。

这样说可能有些阴暗，但我还是想分享一下当时我的一个想法：我觉得为什么在这次事件中，Scale AI 的 CEO Alexandr Wang 跳脚跳得这么急？我一直觉得，可能一个重要原因是接了很多 PRM 数据的标注订单，而现在发现这些数据似乎没有太多用处。

当时的业界状况就是这样。虽然这篇文章是中文，但这不代表只有中国在这么做。硅谷除了 Anthropic 和 OpenAI 之外，很多团队也在探索类似的方向，比如 MCTS 等。几乎所有团队似乎都在朝着这个方向努力。



让模型自由地思考

但这正是我们今天故事的第一个高潮——在所有人都走向一个至少目前看来是暂时错误的方向时，有两个团队却在进行一场精彩绝伦的探索之旅。

更令人兴奋的是，这两个团队都来自中国，一个是 DeepSeek，另一个是 Kimi。

与此同时，

一场精彩绝伦的探索之旅正在发生.....

首先，我们来看为什么说这是一次精彩绝伦的探索之旅。Kimi 在 R1 发布前后推出了 Kimi k1.5。虽然他们并未开源，但发布了一篇详尽的技术报告，介绍 k1.5 训练背后的核心内容。

不过，从阅读体验的角度来看，k1.5 论文的可读性较差。相比之下，阅读 R1 和 V3 的论文时，体验要精彩得多。

Kimi k1.5 论文则充满了大量工程细节，如果想复现，这篇论文的价值极高。细节之丰富，甚至给人一种手把手教学的感觉。但正因如此，阅读体验相对较差。但是我在 Twitter 上找到了一篇 Kimi k1.5 团队成员撰写的文章，讲述了 k1.5 背后的思考过程。这篇文章的文笔极其出色，读来令人心潮澎湃。我必须和大家分享其中的内容。

如何评价 Kimi 发布的多模态推理模型 k1.5？

1月20日，Kimi公开了多模态推理模型Kimi k1.5的训练实践，Github链接：

[https://github.com/MoonshotAI/...](https://github.com/MoonshotAI/)显示全部

关注问题

写回答

邀请回答

好问题 64

20条评论

分享

...



登录后你可以

不限量看优质回答

私信答主深度交流

精彩内容一键收藏

登录



Flood Sung



人工智能等 2 个话题下的优秀答主

+ 关注



亲自答 此回答由问题相关方亲自撰写

1855 人赞同了该回答

亲自答一下

这个技术报告是我们的结果，那么相信大家也想知道一些思考过程吧哈哈，所以这里主要想和大家分享一下o1复现的一些关键思考过程，也就是我自己的Long Chain of Thoughts.

2024年9月12号，o1发布，震撼，效果爆炸，Long CoT⁺的有效让我陷入反思Reflection。

因为Long CoT的有效性其实在一年多前就已经知道了，Tim @周昕宇很早就验证过 使用很小的模型训练模型做几十位的加减乘除运算，将细粒度的运算过程合成出来变成很长的CoT数据做SFT，就可以获得非常好的效果。我依然记得当时看到那个效果的震撼。我们意识到Long Context的重要性，所以率先考虑把Context搞长，但却对Long CoT这件事情不够重视。其实主要还是考虑了成本问题。Long Context主要做的是长文本输入，有Prefill，有Mooncake加持，成本速度可控，而Long CoT是长文本输出，成本高很多，速度也要慢很多。在这种情况下，把输出搞长就没有成为一个高

编辑于 2025-01-23 17:50 · IP 属地天津

▲ 赞同 1855

▼ 112 条评论

分享

收藏

喜欢

收起 ^

知乎上 Kimi 员工的回答 @Flood Sung

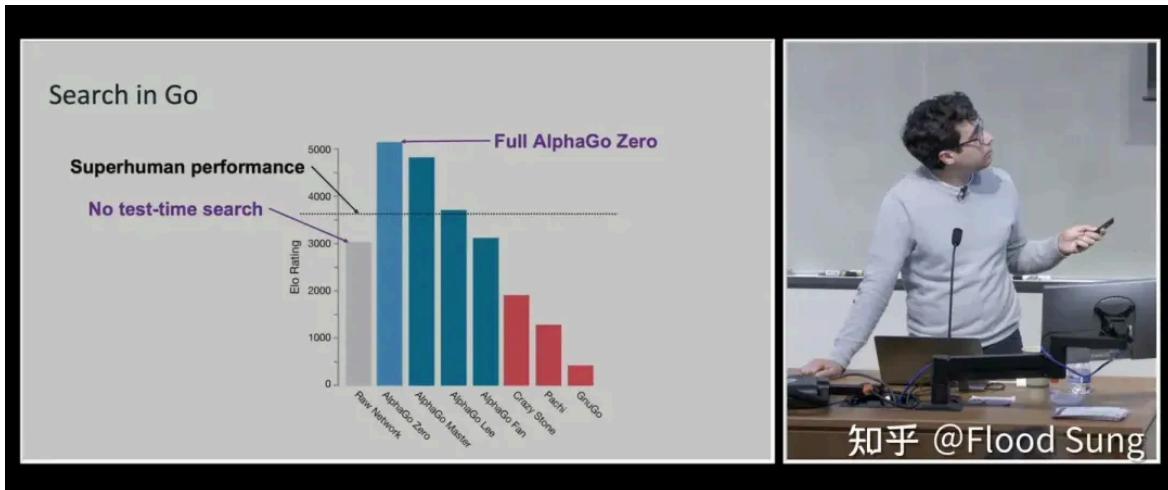
<https://www.zhihu.com/question/10114790245/answer/84028353434>

首先，文章提到 9 月 12 日 o1 发布，举世震惊，随后团队注意到 long CoT 极为有效。他们意识到必须投入 long CoT，否则就会被甩在后面。因此，他们开始思考如何从 OpenAI 的工作中获取灵感，并在研究过程中发现了两个关键视频。

这两个 OpenAI 发布的视频并不是 9 月份的分享，而是更早的演讲——由 Noam Brown 和 Hyung Wong Chung 主讲。这两个视频直到 o1 发布时才被公开，让他们去想：为什么选择这个时间点放出这些视频？一定与 o1 训练有某种关系。

当时我看到这个分析，心想：「这思考角度太牛了。」于是，他们深入研究这两个视频，首先在 Noam 的视频中发现了一张关键的 slide，提到了 AlphaGo 及其后续版本

AlphaGo Zero。大家都知道，AlphaGo Zero 是一个完全基于强化学习（RL）的版本，而这张 slide 强调了 Test-Time Search。

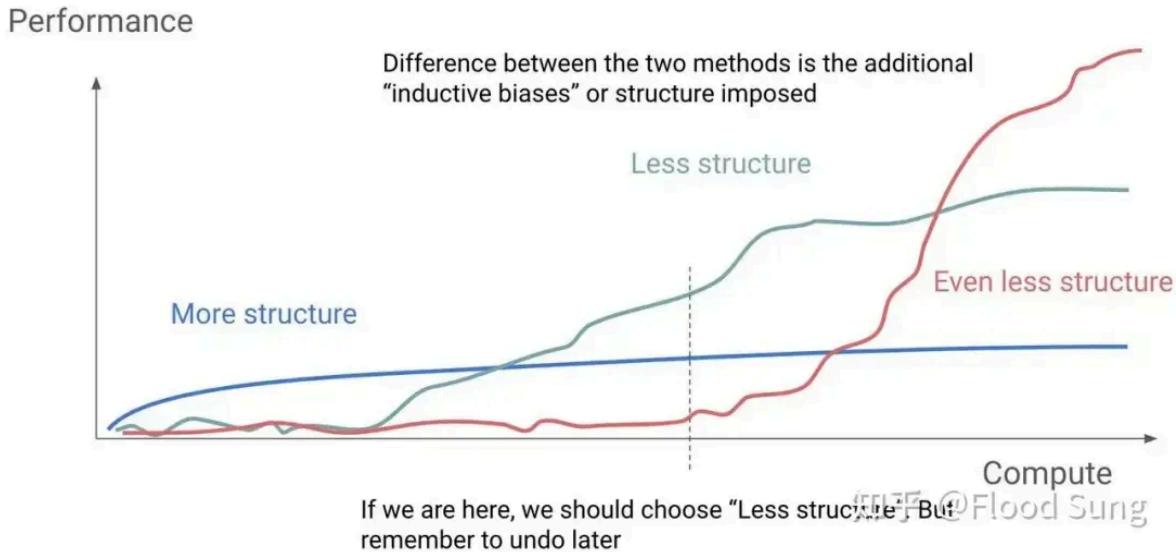


许多人认为 Noam 强调这部分是为了讲解 AlphaGo 的 MCTS，即蒙特卡罗树搜索——同时探索多条路径，评估得分，最终找到最优解。但 Kimi 团队有一个非共识的判断：他们认为 Noam 其实是在强调 MCTS 中的 S，即 Search 本身，而非具体的 MCTS。这一认知带来了他们的第一个关键想法：**让模型自行搜索！让模型自己学会探索不同路径，而不是人为限定其思考方式。**

这让他们联想到 Richard Sutton 著名的演讲《The Bitter Lesson》。

One thing that should be learned from the bitter lesson is the great power of general purpose methods, of methods that continue to scale with increased computation even as the available computation becomes very great. The two methods that seem to scale arbitrarily in this way are **search** and **learning**.

第二个视频的内容同样至关重要。他们总结出一个核心观点：「Don't teach, incentivize.」也就是说，不要去「教」模型，而是要「激励」它自主探索。



在许多实验中，模型的结构约束越少（less structure），当计算资源增加时，最终性能的上限就越高。反之，如果在早期给模型加入过多结构约束，它的最终表现可能会受到限

制，失去了更多自主探索的可能性。

他们进一步思考：为什么这个同学特别强调 structure？什么是 structure？当时我读这个特别爽，是因为我好像在看 Kimi 这个同学的脑内对话。

MCTS 是 structure，A* 算法也是。这些都在限制模型的自由思考能力。他们认为，OpenAI 发布的 PM-800K 训练方式也存在类似的问题——它通过一个成型的推理数据集，告诉模型在不同情况下应该如何思考。这实际上是人为设定了一种思维路径，限制了模型自身的探索能力。

最终得出结论：**o1 没有限制模型如何思考。这一点特别特别重要。** Kimi 团队因此决定不采用 MCTS。

我相信，在 o1 发布后的 9 月份，他们花费了大量时间研究这个方向，并在 10 月份最终确定了自己的研究路径。

大家想一想，很多团队在 12 月份的时候仍然在沿着 MCTS 的方向探索，但像 Kimi 这个团队在那个时候已经找到了另一条路线。我相信 DeepSeek 也在同一时间有所领悟，尽管我不确定他们的学习或理解过程是否相同，但他们应该也意识到了一些类似的关键点。

他们继续思考：现有的许多所谓的 agent，其本质上只是一个 workflow，而这些 agent 的 workflow 其实是高度结构化的，这就限制了模型的能力。所以，他们做出了一个判断——这种基于 workflow 的 agent 只具有短期价值，而没有长期价值。是这样吧？不过这是另一个话题了。

他最后总结说——「All in All 我们就是要训练模型能够像我们人一样思考，自由的思考！」

他展示了 Noam 演讲的最后一页，也就是他们的 Future Work，其中阐述了他们未来的研究方向。这一部分对他的启发最大。他们的核心观点是什么？就是要用真正的激励来进行强化学习，而不要被 reward model 本身所限制。



Noam 在演讲最后谈及未来展望

这个概念可能有些抽象，我解释一下。这里涉及很多算法细节，今天不适合暴露，但大家可以这样理解：比如有时候要十步才能得出正确的答案。如果我们仅仅按照最终答案来进行激励，就会担心模型在漫长的中间过程中学偏了，所以过去大家都不敢直接采用 ORM 这种方式。取而代之的，是 OpenAI 所引导的 PRM，即关注训练过程中的阶段性奖励。

但他们当时得出了一个关键结论：**不要搞过程激励，真正重要的是最终答案是否正确，应该以此为核心来激励模型。**

当时他们不知道，但后来他们发现，DeepSeek R1 的论文中也提到了类似的观点，即不要依赖过程奖励。

所以他们后来就定下来了——Practice Program，也就是「多练习」，给模型一个能不断做题的环境。只要反复训练，就能够取得提升。

菜就多练。文章中写道：「**做题，做题，还是做题！做有标准答案的题，刷起来！**」

我觉得这篇文章非常精彩。它展示了如何逆向思考 o1，并结合各种信息以及专业知识，最终推导出正确的结论。

不过，稍显遗憾的是，k1.5 在 Pure RL 上做得不够彻底，还是用了一些前置的激活引导环节，而不像 R1 Zero 那样完全采用左脚踩右脚的 Pure RL 方式。

最大的局限可能还是因为它没有开源，因此在行业内的影响力远不及 DeepSeek R1。

当然，如果你读过 k1.5 的技术报告，你不得不对这两个团队都心生敬意。尽管今天的重点是 DeepSeek R1，我仍然想特别提一下 Kimi k1.5——这是一次非常精彩的探索过程。

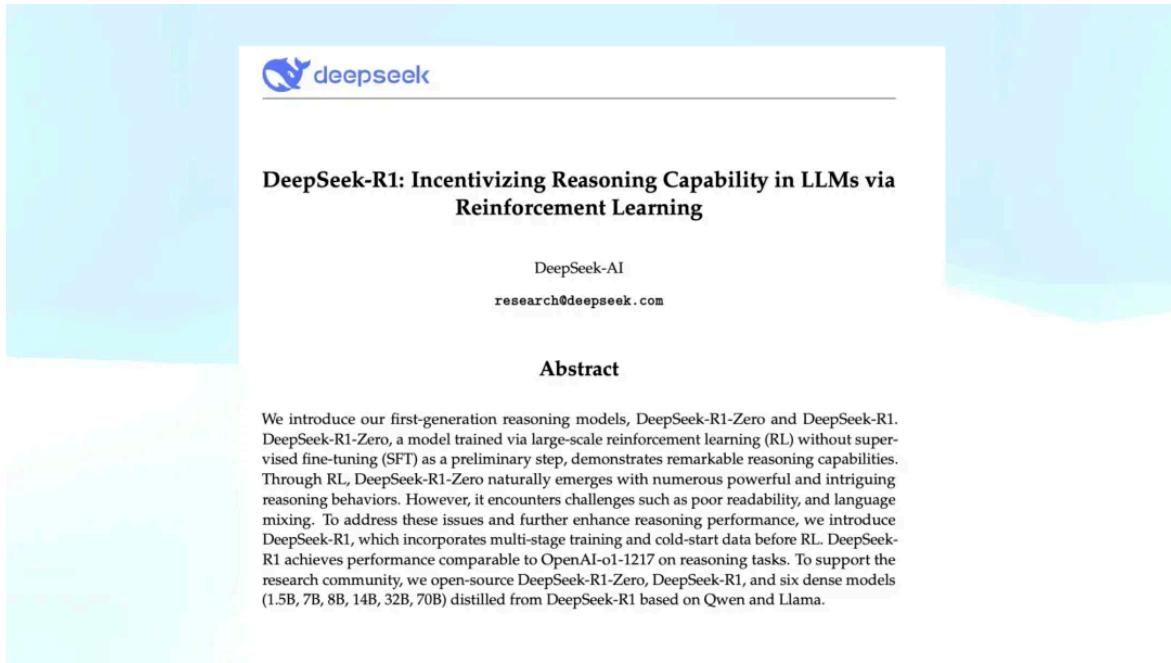


至此，主角登场

4

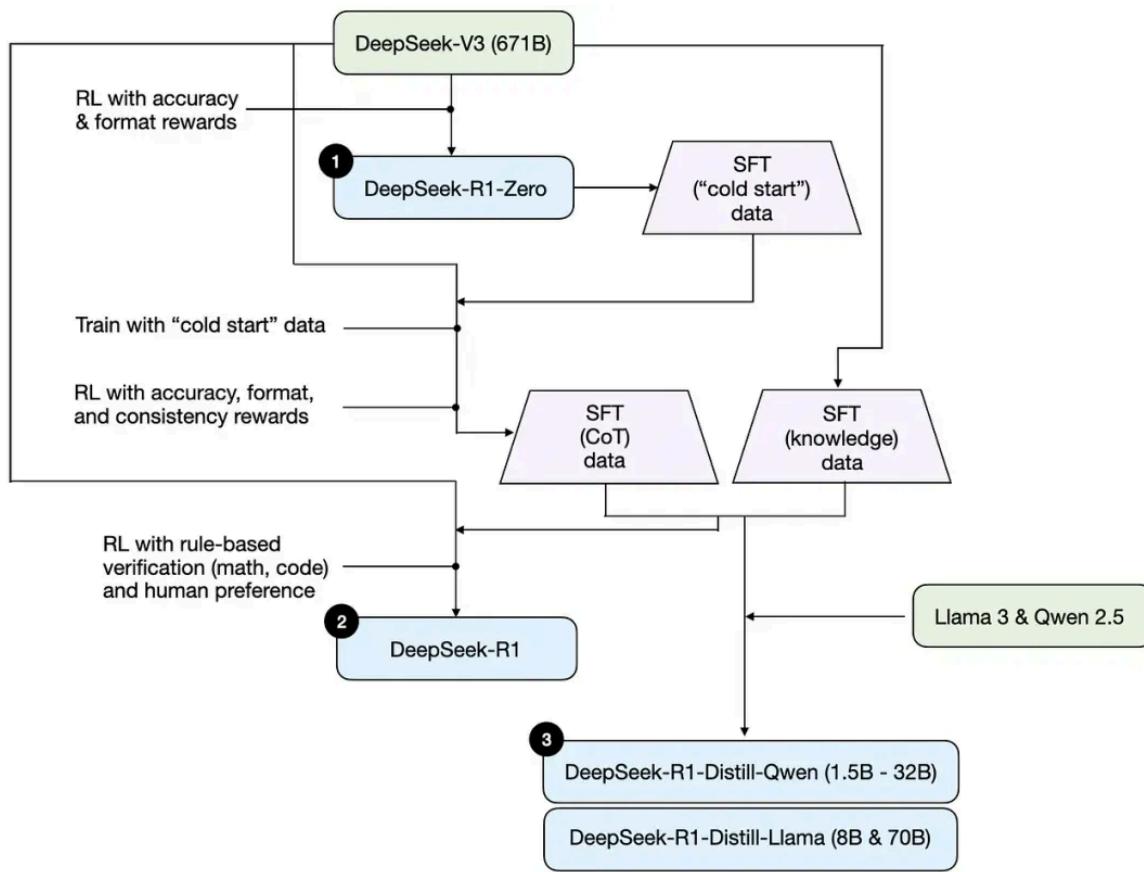
DeepSeek R1 技术报告详解

DeepSeek R1 论文的标题就是「Incentivizing Reasoning Capability」。这篇论文的核心思想正是如何通过激励来增强模型的推理能力。相较于 V3，R1 的思路相对更容易理解，但仍然需要很长时间理解。



所以这里我借用了 Sebastian 的一篇文章，其中关于如何理解推理模型的部分非常精彩，尤其是其中的一张图。

这张图的表达极其出色，因为在阅读 R1 的论文时，即使是研究人员，从头到尾阅读仍可能感到困惑。因为在 R1 最终生成之前，它在 V3 和 R1 Zero 之间来回地你训我我训你，类似于「左脚踩右脚」，导致阅读过程中容易迷失。但这张图完美地呈现了 R1 的三方面的训练结果。



第一个是标注为 R1 Zero，第二个是 DeepSeek R1，第三个则是它的蒸馏版本。目前，如果有人在本机尝试运行，运行的通常都是这些蒸馏版本。

我们先来看最神奇的一点。如果你来自产业界，可能最震惊的并不是最终的 R1 模型，而是 R1 Zero。它简单到令人难以置信。DeepSeek 首先有了一个强大的基础模型 DeepSeek V3，这个模型就是 12 月份发布、Andrej Karpathy 点赞的模型。

基于这个模型，他们采用了纯强化学习进行训练，但过程本身极为简单。他们训练时使用了一个固定的模板。

训练模板

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within `<think>` `</think>` and `<answer>` `</answer>` tags, respectively, i.e., `<think>` reasoning process here `</think>` `<answer>` answer here `</answer>`. User: **prompt**. Assistant:

Table 1 | Template for DeepSeek-R1-Zero. **prompt** will be replaced with the specific reasoning question during training.

现在大家对 AI 产品应该比较熟悉，可以将其理解为 system prompt。具体来说，该 system prompt 设定为「这是一个用户和 assistant 之间的对话」，用户提问后 assistant 进行解答。但是该 assistant 需要先「在脑海中思考推理过程」，然后再给出最终答案。此外，assistant 还必须将推理过程标注在 think 标签内，而答案则放在 answer 标签内。

训练时，他们使用这个简单的模板，标红的 prompt 在训练的时候会填充各种问题，例如「 $1+2$ 等于几？」或者「给定方程 $a^2 + b^2 = c$ ，求 b 的值」。但是它的训练模版本身就是如此简单。

他们的激励模型同样简单，主要分成两类激励：

- **准确度激励**：判断答案是否正确。例如，如果模型回答 $1 + 1 = 2$ ，则加 1 分；如果答案错误，则不加分。

- **格式激励**：模型必须按照要求的格式作答。例如，若问题是「 $1+1$ 等于几？」，模型直接回答「answer 2」将得 0 分。但如果它在 think 标签中先写出推理过程，再在 answer 标签中给出答案，则会获得更高的分数。

激励模型

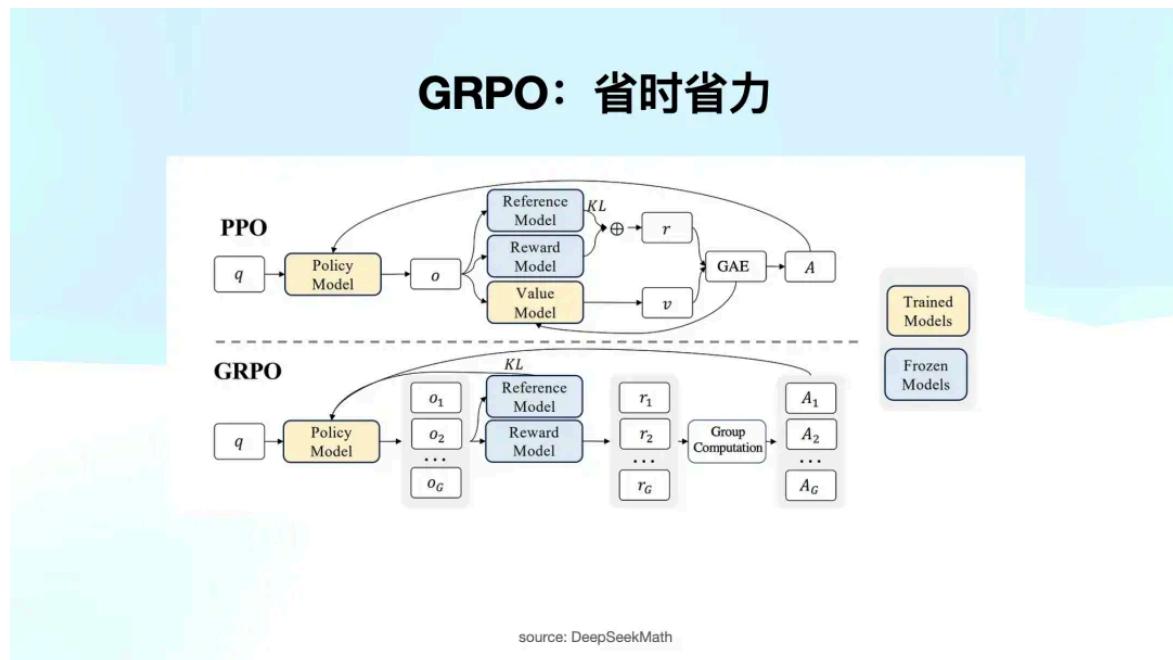
<p>准确度激励</p> <p>问：$1+1=?$</p> <ul style="list-style-type: none"> • 2 加一分 • 1 零分 	<p>格式激励</p> <p>问：$1+1=?$</p> <ul style="list-style-type: none"> • <think>用户问我$1+1$等于几，这应该是一道很简单的数学题，我们可以形象的思考，我有一个苹果，再给我一个，就有两个。因此答案应该是2。</think><answer>2</answer> 加一分 • <think></think><answer>2</answer> 零分
---	---

由于这一强化学习过程并不使用 PRM 方法，因此它无法采用同等规模的模型来判断。相反，他们采用了一种基于规则（rule-based）的激励模型，确保判断标准极为简单清晰，即答案要么正确，要么错误。因此，他们在训练时准备了大量具有明确对错的题目，例如数学题、物理题、写代码。

对于写代码，相当于他们类似于信息学竞赛，准备了一个 sandbox，代码提交后运行，看输出是否正确。

除了答案，第二就是要激励格式。它要求把推理过程写在 think 里面。如果你在想 $1+1$ 等于几，想了一堆，就给你加一分。

他们还配合 GRPO 进行训练。这里值得注意的是，过去强化学习主要依赖 PPO，例如在 k1.5 论文中，作者曾提到是否应使用 PPO，不要。他们最终也是选择了另一种方法。PPO 最大的问题在于，在强化学习的每一步中，不仅需要调整 policy model，还要优化 value model，导致计算量开销极大。



GRPO 通过一个非常粗暴的方法，类似于将同样的问题扔给这个 policy model，让它答 8 次，结果我们根据正确答案给一个平均值。

大家记住 GRPO 就是在开销上比较少，但能有效地计算出每一轮强化学习探索后，模型离正确方向的距离，以及如何激励模型朝正确方向发展。

简而言之，R1 Zero 只做了三件事情：一个基础的训练模板，一个简单的激励模型，以及 GRPO 策略。

需要注意的是，他们并没有使用类似 PRM800K 那种复杂的推理数据集，也没有教会模型什么事情都要先想八步，什么都没有教。

“学会”了思考

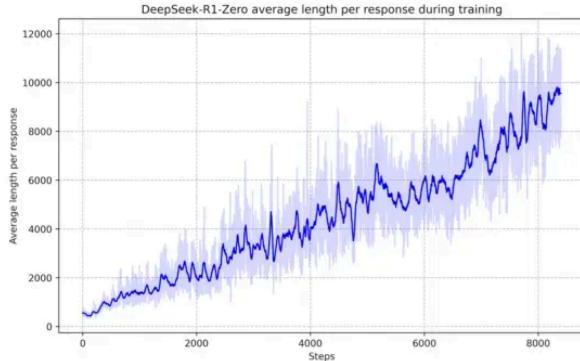


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

训练过程中，只给了问题、答案和规则。横轴是 RL 的迭代次数，竖轴是响应的长度。他们发现模型学着学着，自己把答案越吐越长了。激励模型里，我们没有激励过长度这件事情，只判断了对错和你有没有思考。

但是模型自己发现，一旦我思考地越长，我越能答对。这个发现十分惊人，因为之前的团队没有想到通过这么简单的手段就能解决这个问题。这是最重要的一张图。

也就是标黄的这句话，「R1 Zero literally solves reasoning tasks with more thinking time」。这个模型在没有任何激励的情况下，通过增加思考时间，自主学会了如何解决需要推理的任务。

我们来看 Benchmark，R1 Zero 还不是大家每天用的 R1，但它在 AIME、MATH、GPQA 和写代码等任务上，有的超越、有的逼近。除了 CodeForces 上略有不足，因为还是有点麻烦。

这一方法完全基于纯 RL，没有使用任何 SFT 数据。对于训练模型的同学来说，这是一种难以想象的突破，左脚踩右脚把活干上去了。

论文中没有准确公布，但我估计 RL 做了大约一万步。其实相较于预训练的高成本，RL 的成本我想象中、包括我咨询了很多人是要低得多的。如果你已经有一个强大的基础模型，通过这种 RL 方法，可以低成本快速提升模型的能力。

R1 Zero 路走对了，但接下来 DeepSeek 团队发现了 R1 Zero 的一些问题。首先，可读性较差。第二是经常出现语言混杂的问题，类似于上海外企白领的说话方式：「Maria，今天这个 schedule 有些满」。

这个问题不仅仅出现在 R1 Zero，大多数推理模型都会有这种语言混杂的问题。

就像最近有一个梗，不知道大家有没有看到。国外有些网友截图了那个 o3 的思考过程，发现当 o3 用英文问问题时，在用中文推理。虽然我们知道这个现象背后的真实情况，但很多国外网友还是截图并 @Sam Altman，问你们是不是在蒸馏 DeepSeek R1。天道轮回。

其这背后的原因很简单。就是模型自己在探索时，对模型来说，无论是中文还是英文，都只是一个 token。它自己在思考时，按照 token 来处理问题，而不在乎人类是否能读懂。这其实是一个语言混合的问题。

第二点是它的格式有点混乱。现在大家使用 ChatGPT 或 Claude，可能已经习惯了那些写得很精美的 Markdown 格式文章，或者是 bullets，但你看 R1 Zero 的输出，你会发现，因为它专注于解决推理问题，输出的可读性相对较差。

为了解决这个问题，我们需要让推理过程更具可读性，方便共享给社区。我们就继续在 R1 Zero 的基础上，去做 R1，让它的可读性更好。

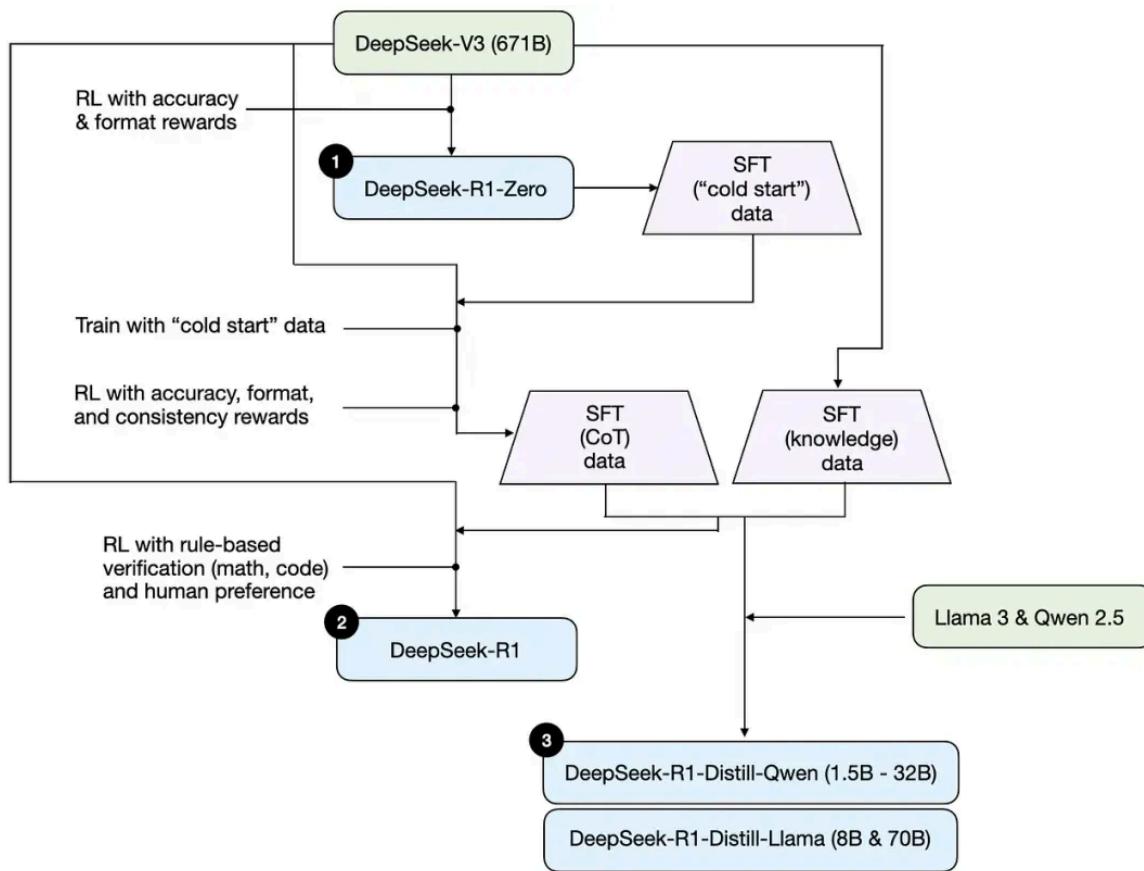
但说实话，最精华的工作刚刚已经讲完了：**提升模型能力，左脚踩右脚。**

对于研究者来说，他们最关心的就是 R1 Zero，而后续做 R1 的过程也很有趣。

复现 DeepSeek R1 的「Aha Moment」

我们现在来看 R1 怎么来的。DeepSeek 分享这个过程真的像 Marc Andreessen 所说的，「是给全世界人类的财富。」

前面的 Pure RL 过程，只要掌握了原理，大家都能做，但不意味着你就能做出来 R1。



首先，既然我已经有了一个强大的 R1 Zero 作为基础，那在训练 R1 时，我就不需要从 0 开始了。我们用了 R1 Zero 生成了右侧那个高质量的 SFT 数据，也就是 cold start 数据，这些数据由 R1 Zero 输出的高质量带有推理过程的数据，作为 cold start 数据重新去 SFT 了 Deepseek V3 base 模型。

你会发现一个非常神奇的事情：**V3 base 促使了 R1 的诞生，而 R1 强大的推理能力又反向去 SFT V3。**

这又是一个左脚踩右脚，这样不仅让 R1 更强，也让 V3 更强。比如，使用 cold start 数据训练的 checkpoint，每个小圆点代表一个 checkpoint，简单来说，就是模型训练到某一阶段时的「存档」。

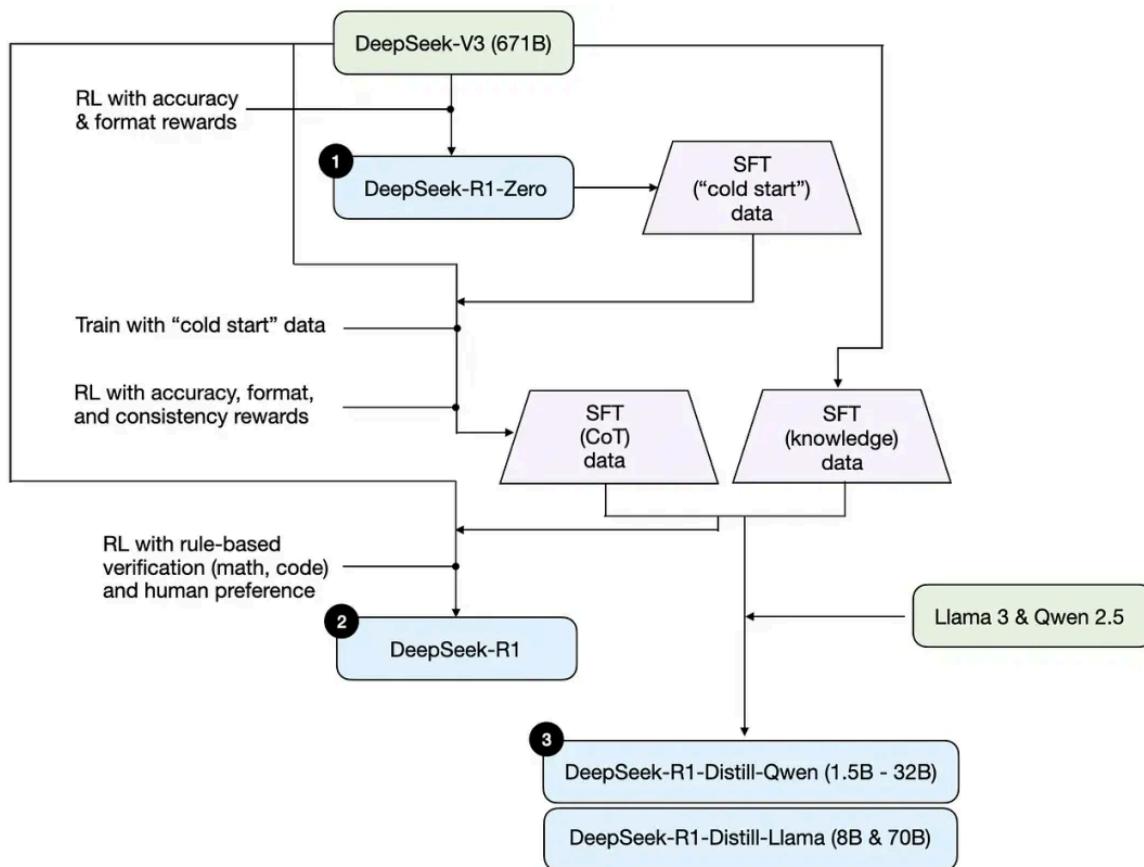
第一个 checkpoint 是用 R1 Zero 的高质量 cold start 数据进行 SFT，完成了一个 fine-tuning。然后，他们拿着这个 checkpoint 做做了一轮类似 R1 Zero 的强化学习。但这次有一些不同。除了激励准确度和格式，但他们加入了一个新的激励项——一致性 (consistency)。

一致性指的是语言的一致性。也就是说，在这个 RL 过程中，不仅要确保答案的准确度和格式，还要检查推理过程是否出现了语言混杂。如果是中英文混杂，我就给你打 0 分；如果你全程用中文或英文，就给你高分。就只加入了这一项激励。

经过这一步，就得到了第二个 checkpoint。这时的版本不仅具备了强大的推理能力，而且语言没有混杂。他们用这个版本再生成了第二轮的高质量 CoT 数据。这时的 CoT 数据质量比前面的 cold start 更好，主要因为它的推理过程是语言统一的。

最后，他们通过人工筛选和规则匹配等方式，剔除了一些冗余、可读性差部分，得到了一个经过筛选和优化的高质量 CoT 数据。这些数据在后续还有其他用途，大家要记住这一点。

前面的 RL 过程是数学、物理、写代码这类任务，但他们的目标是最终能够将这个模型应用于整个开放社区。要会回答像「 $1 + 1 = 2$ 」这样简单的算式，或是「中国的首都是北京」等常识性内容。



因此，他们从已经强化过一轮的 V3 base 中，输出了右边这个，通用知识的 SFT 数据。这些数据与之前的高质量 CoT 数据合并后，用于进行最后一次的强化学习。在这次强化学习中，整个过程更像是我们通常训练一个模型的方式，包括了 human preference 的部

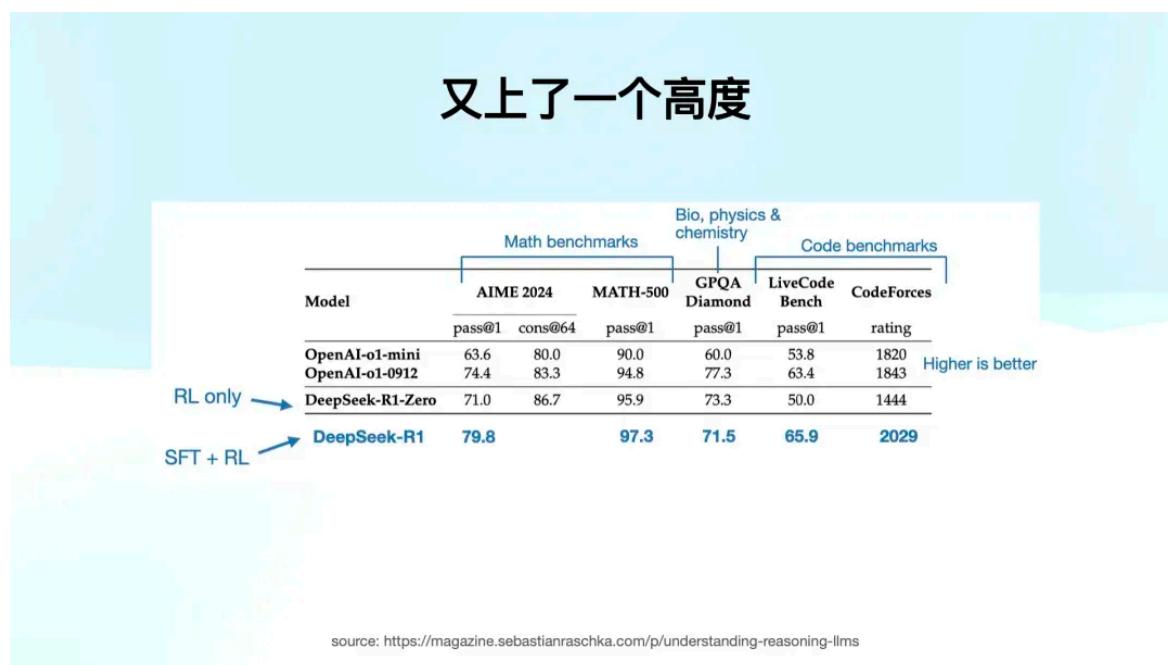
分，也就是我们对模型输出格式、内容等的偏好。最后得出的模型便是大家现在使用的 DeepSeek R1。

为什么我认为这个过程很重要呢？因为通过模型的训练过程，我们能看到从 R1 Zero 到 R1 的转变。

这是他们的探索路线。如果没有他们的解释，**你可能有很多不同的探索路径，不一定能搞得出来。**

但 DeepSeek 在他们的论文中详细讲述了从 R1 Zero 到 R1 的训练过程。你可以看到他们是如何构建 cold start 数据的，如何为 reasoning-oriented 强化学习做准备，如何准备高质量的推理数据和通用知识数据，为了让模型面向各种场景而不局限于数学和编程任务，进行 RL 训练的。他们写得非常详细，对于那些希望复现工作的研究人员来说，这些内容非常有帮助。

完成第二步后，拿到 R1，我们可以看到 R1 的表现。在所有的分数上，除了 GPQA 还差 OpenAI o1_0912 一点，其他方面的数据已经全面超过了 o 系列模型，甚至在 CodeForces 上也取得了非常好的成绩。这就是 SFT 加 RL 的强大威力，模型达到了一个新的高度。

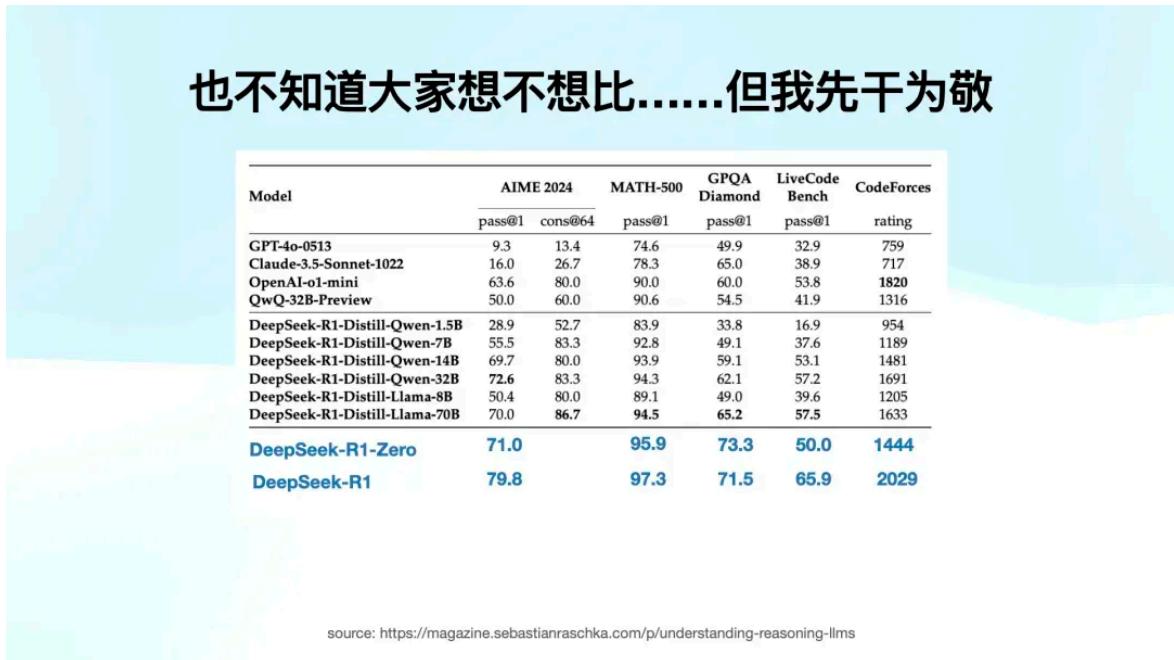


接下来是第三步，我认为这是 DeepSeek R1 工作中的一项突破，尤其是在学术界和技术界产生了重要影响。如果他们只做到前两步，工作已经非常重要了，但如果沒有第三步，破圈的概率会低很多。

到这里为止，工作已经非常完整，论文本身已经非常有价值，但 DeepSeek 还做了一些额外工作。他们将中间 checkpoint 产生的 R1 Zero 到 R1 的版本生成的高质量 CoT 数据与 V3 的世界通用知识数据结合后，**不去微调他们自己的 V3，去微调别人的模型。**

他们想验证一个问题：虽然其它模型没有做 RL，没有做纯强化学习，但他们是否能通过 DeepSeek 输出的高质量推理数据，学会这种推理过程？他们不仅验证了这个问题，还做

了实验，并将结果公开。



也不知道大家想不想比……但我先干为敬

Model	AIME 2024		MATH-500		GPQA Diamond	LiveCode Bench	CodeForces
	pass@1	cons@64	pass@1	pass@1			
GPT-4o-0513	9.3	13.4	74.6	49.9	32.9	759	
Claude-3.5-Sonnet-1022	16.0	26.7	78.3	65.0	38.9	717	
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820	
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9	1316	
DeepSeek-R1-Distill-Qwen-1.5B	28.9	52.7	83.9	33.8	16.9	954	
DeepSeek-R1-Distill-Qwen-7B	55.5	83.3	92.8	49.1	37.6	1189	
DeepSeek-R1-Distill-Qwen-14B	69.7	80.0	93.9	59.1	53.1	1481	
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2	1691	
DeepSeek-R1-Distill-Llama-8B	50.4	80.0	89.1	49.0	39.6	1205	
DeepSeek-R1-Distill-Llama-70B	70.0	86.7	94.5	65.2	57.5	1633	
DeepSeek-R1-Zero	71.0		95.9	73.3	50.0	1444	
DeepSeek-R1	79.8		97.3	71.5	65.9	2029	

source: <https://magazine.sebastianraschka.com/p/understanding-reasoning-lms>

这产生了一个非常有趣的效果：他们用这个高质量数据去蒸馏了其他模型，比如干问。从 1.5B 到 32B，干问在中国确实有很大影响力，但要打破圈层，仅仅做干问是不够的，他们还进行了 Llama 的蒸馏。这其实是「羞辱」 Llama 哈哈哈，因为使用相同的高质量数据蒸馏后，Llama 70B 在一些任务上的表现和干问 32B 差不多，非常接近。

那这项工作重要的意义是什么？首先，DeepSeek 不仅完成了自己的研究，还向全世界证明了一个事实：用通过一个超大 size 高质量的推理模型产生的数据进行 SFT 的成本非常低，与传统的预训练方法相比。它让现有模型的表现拔地而起，不用做 RL。

这项工作的另一个重要意义在于，它让大家看到了在自己的电脑上复现这个结果的可能性。

技术社区也不是每个研究者家里有八张 A100。大家复现一个工作，直接 LM Studio，Ollama 拉下来，本地就开始部署运行，大家很容易就可以对比原始和蒸馏版本。很快就可以发现「哇，这个中国团队真厉害」。这是大破圈的非常重要的原因。



两次不太成功的尝试

DeepSeek 在论文里还提到了 PRM 和 MCTS 两次不太成功的尝试。

PRM 和 MCTS 一时半会走不通

Process Reward Model (PRM) PRM is a reasonable method to guide the model toward better approaches for solving reasoning tasks (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023). However, in practice, PRM has three main limitations that may hinder its ultimate success. First, it is challenging to explicitly define a fine-grain step in general reasoning. Second, determining whether the current intermediate step is correct is a challenging task. Automated annotation using models may not yield satisfactory results, while manual annotation is not conducive to scaling up. Third, once a model-based PRM is introduced, it inevitably leads to reward hacking (Gao et al., 2022), and retraining the reward model needs additional training resources and it complicates the whole training pipeline. In conclusion, while PRM demonstrates a good ability to rerank the top-N responses generated by the model or assist in guided search (Snell et al., 2024), its advantages are limited compared to the additional computational overhead it introduces during the large-scale reinforcement learning process in our experiments.

他们尝试 PRM 时发现非常难以定义和评分。如果推理过程中有 800 字的内容，怎么分步骤？如何将每个步骤打分？有时候前面想歪了，后面就是因为想歪了所以能得出正确答案。如何评估这些步骤的贡献，成为了一个巨大的挑战。这在资金和时间成本上都是一个不太可控的事情，故而他们认为 PRM 做不下去。

另一个问题是关于 MCTS。虽然大家不需要深入了解 MCTS，但有一点我认为值得分享，就是为什么在语言模型中做 MCTS 很难。简单来说，MCTS 的核心是每一步的搜索空间有限。以 AlphaGo 为例，围棋能落子空间是有限的，棋子只能落在空位上，因此每一步的选择是有范围的。虽然长度放长可能有无穷多种选择，但每一步的选择空间都是有限的。

而在大语言模型中，推理每一次都是整个词表都能选，还没有围棋的规则。因此，尝试将 MCTS 应用到语言模型中时，你会发现搜索空间变得非常难以控制，也很难定义激励模型。

他们在论文里没有把话说死，我们搞不通不代表不行，但大家都懂的，你行你上。一时半会有个成功版都开始转向了 ORM。

他们还做了一些微小的工作，就是利用高质量的数据去 SFT 其他模型。那大家可能会问，既然用了高质量的 CoT 数据做 SFT，为什么不直接用他们的 R1 Zero 训练方法，采用纯强化学习去训练其他模型呢？

并不是谁都能左脚踩右脚的

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBench
	pass@1	cons@64	pass@1	pass@1	pass@1
QwQ-32B-Preview	50.0	60.0	90.6	54.5	41.9
DeepSeek-R1-Zero-Qwen-32B	47.0	60.0	91.6	55.0	40.2
DeepSeek-R1-Distill-Qwen-32B	72.6	83.3	94.3	62.1	57.2
pure RL → DeepSeek-R1-Zero	71.0		95.9	73.3	50.0
pure RL → DeepSeek-R1	79.8		97.3	71.5	65.9

source: <https://magazine.sebastianraschka.com/p/understanding-reasoning-langs>

DeepSeek 也考虑过这个问题，并且做了实验。他们选择了干问 32B 作为基础模型，采用了与 R1 Zero 相同的纯 RL 训练方法，结果发现并没有明显提升。

你可以看到，干问 32B 用纯 RL 训练出来的模型和用了 CoT 数据做 SFT 后的干问 32B 在表现上有很大差距。

这个实验让他们得出一个结论：**左脚踩右脚也要有一个基础。**

这个道理其实很容易理解，就像 Peak 给我讲的一个点，大家不要把 reasoning 想成人在思考，reasoning 也是 next token prediction，只不过让它产生更长的 CoT，让它有更多犯错搜索的空间和思考的可能性，能推出正确答案。

比如，如果你把一个二年级的小朋友关在房间里，给他一道大学高数题，让关在房间里想一个月，他也不可能解出来。所以 Pure RL 对 base model 是有要求的。

为什么这能引向下一个高潮？我们之前一直在讲 R1，但 R1 的一切起点都是 DeepSeek V3。这个 671B 的 MoE 模型。

我们发现我们绕不过去这个模型，并不是 R1 靠自己左脚踩右脚就升天了。如果没有这个强大的基础模型，DeepSeek 不可能做出现在的工作。前面我提到 k1.5 之所以没有取得与 DeepSeek 相同的突破，我也不知道内幕，虽然 Kimi 也没开源，也许是 Kimi 的基础模型没有达到 V3 的高度，这是有可能的。

接下来，我们进入今天的第二个高潮。



罗马不是一天建成的

罗马也不是一天建成的。虽然 DeepSeek 在突破圈层时引起了广泛的关注，但一些叙事讲得好像这个项目是中国一个量化基金的 side project，他们只是「随便做做」就出了这么一个成果。

虽然这种叙事符合好莱坞的风格，但如果你认真看待这个工作，你会发现它是非常严谨的。

DeepSeek 是一个认真在做 AGI 和模型研究的公司，它们很多投入非常长期，很多贡献也并不局限于 R1。

罗马不是一天建成的，但我们可以看看它是怎么建成的。

罗马不是一天建成的 R1 的背后

- 2024年2月发布 DeepSeekMath，引入 GRPO
- 2024年5月发布 DeepSeek V2，引入 DeepSeekMoE、MLA
- 2024年12月发布 DeepSeek V3，引入 FP8、MTP

但它的背后有许多关键支撑。去年 2 月，DeepSeek 发布了 DeepSeekMath，在解决数学问题时引入了 GRPO 方法。GRPO 的优点是效率高，但它主要解决的是那些答案明确、规律性强的任务，例如数学和物理问题。所以在 DeepSeekMath 的过程中，GRPO 被用来大大降低强化学习的计算量。这个论文其实是 2023 年发布的，模型 2 月才开源。

在 5 月，DeepSeek 发布了 DeepSeek V2，这是一起的起点。因为在 V2 中引入了 DeepSeekMoE、MLA。接着，在 12 月，他们发布了 V3，引入了 FP8 和 MTP 这样的训练与推理方法。所有这些都为构建强大的 V3 模型，打下了 RL 基础。

现在我们来看一下 DeepSeekMoE 的优势。MoE 是混合专家模型（Mixture of Experts）的缩写。简单来说，MoE 解决了一个关键问题：**当模型规模变得越来越大时，训练难度也会显著增加。**

DeepSeekMoE

Mixture of Experts 快速了解

- 超大尺寸稠密模型训练越来越难
- 单次推理需要激活整个模型，推理成本高
- 之前的 MoE 专家数大都不多，像 Mistral 8x7B，稀疏度不够，难训大

大家都知道，模型本质上是一堆大量矩阵，这些矩阵中存储了大量的数。

每次训练单个 token 时都要从头到尾来一遍，计算量和数据量太大，训练越来越慢，越训越崩。很容易崩溃。MoE 的概念其实很早就有了，是一个老概念。但最近几年，第一个让它重新被业界关注的是 Mistral，欧洲的一个团队做的 $8 \times 7B$ 的 MoE 模型。

他们发现，不需要每次训练或推理时都激活整个模型。实际上，我可以将模型做得非常大，比如说做成 $8 \times 7B$ ，但每次只激活其中一个专家，也就是说每次只激活 7B 的部分。这样一来，训练和推理的开销都会变小，训练时也更不容易崩溃。

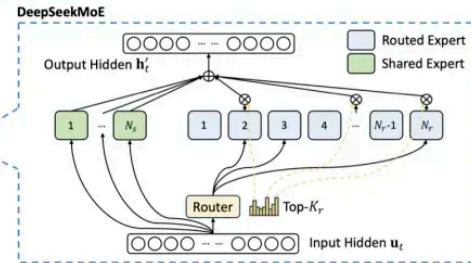
这是之前的探索，比如 $8 \times 7B$ 、 $8 \times 22B$ 等，还有腾讯去年年底发布的混元模型，它是一个 300B 左右的 MoE 模型。虽然这些探索都很有价值，但 MoE 的问题之一是专家数量通常不够多。比如 Mistral 的 $8 \times 7B$ 和 $22B$ 模型都是有 8 个专家，稀疏度不够，总 size 难以进一步提升。

DeepSeekMoE 在 V2 中做了很多创新。

DeepSeekMoE

DS 做了什么创新

- 超大规模，V2 激活21B/总尺寸236B，V3 激活37B/总尺寸671B
- 前三层 FFN，没有直接扔到专家层
- shared experts
- 粗暴却实用的 router 分流算法
- 极致的跨节点通信效率（IB with NVLINK）
-



首先，V2 中的模型已经超大规模，达到 236B，而每次激活 21B。到 V3 时，这个模型已经扩展到 671B，是目前 MoE 模型中最大的，每次激活 37B。

很多人觉得 MoE 模型很 low，认为大模型训不下去了，训一堆小模型，「三个臭皮匠，顶一个诸葛亮」。但 MoE 不是这样。

大家脑子里 MoE 认为它是有 8 个专家，训练时就从中选择一个专家回答问题。其实并非如此。模型里是多层的，每一层都有很多专家。在 DeepSeek MoE V3 中，每层有 256 块，有很多层，token 经过每一层时都会被分配到不同的块中。因此，token 在模型内部会经过多层多个不同专家的处理，而「专家」这一名称其实有些误导。

在 DeepSeek V2 和 V3 中，创新的地方在于他们不再简单地用 router 分流算法将 token 直接扔给专家，而是在每个 token 被送往专家之前，V2 加了一层，V3 加了三层 FFN，可以理解成加了个小模型。这些小模型能够理解潜空间中的一些概念，提高了模型的智能处理能力。

MoE 也可以类比为医院的分诊台。在过去，医院的所有病人都必须先找全科医生，效率很低。而 MoE 模型则相当于有一个分诊台，将病人分配到不同的专科医生那里。DeepSeek 在这方面也有创新，之前分诊的或许只是一个完全没有医学知识的「保安」，而他们用的是一个有医学知识的「本科生」来处理分流任务。

DeepSeek 还引入了共享专家 (shared experts) 的概念，图片里绿色的部分，每一层 shared experts 一定会被激活。有些通用的能力会被共享。

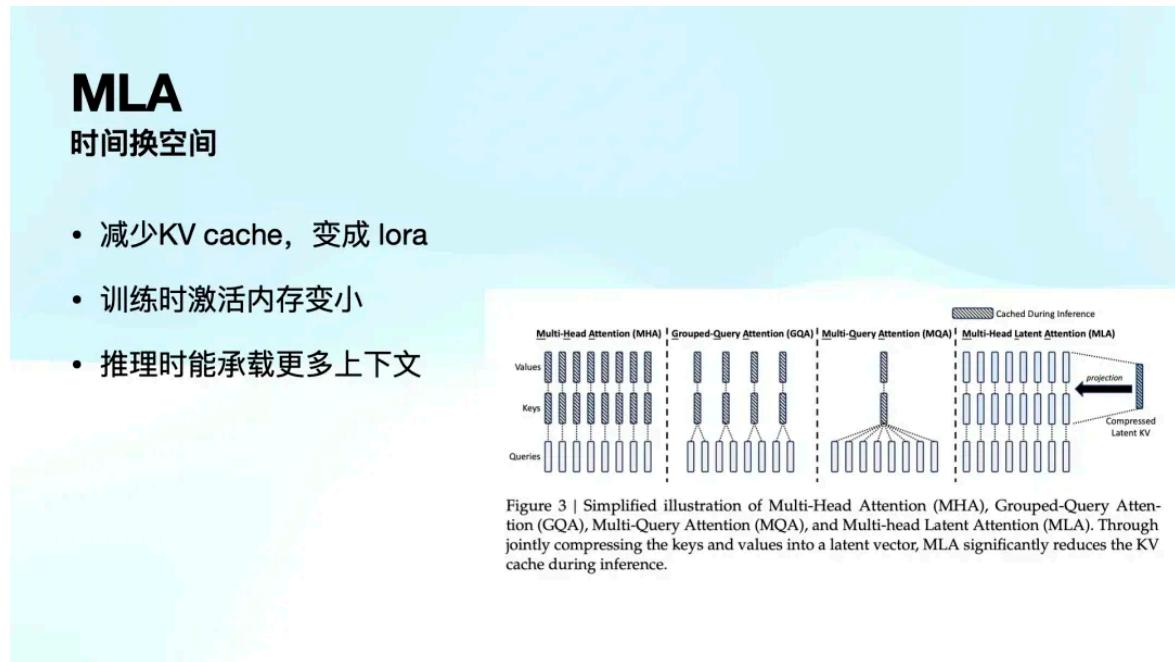
他们还设计了一个粗暴但实用的 router 算法，也做了一个极致的跨节点通信效率方案，使用了 NVIDIA 收购的 InfiniBand 和 NVLink。

这些创新都非常厉害，MoE 大家都探索了一年多了，没人在这么大规模上做这些事情。而且他们训练 V3 一次就训过去了，都没有训崩，这也是为什么他们能以 500 多万的训练成本完成训练的重要原因。

接下来是 MLA。MLA 就更偏算法。

大家知道 Transformer 中的 MHA 对显存占用很大，但是显存里除了存了模型权重，有 30-40% 的空间都在存上下文，很多通过 KV cache。

而 MLA 是用时间换空间。训练时虽然多花时间，把一个本来是 $m \times n$ 的矩阵直接压缩成了一个一维的 lora，压缩率非常高。这样推理的时候 KV 比别人小，这样承载的上下文就会变多。不仅压缩尺寸承载更多推理，而且在测试中，发现相比 MHA 智力性能没有下降，反倒有提升。



最后是 DeepSeek 在 V3 引入的 FP8 训练。传统的训练模型既有 16 位浮点，也有 32 位，而 FP8 是 8 位浮点。大家都觉得 8 位浮点表达的精度太浅了。但是绝大部分团队搞不定。你可以用 8 位浮点运算，但到底哪个部分可以呢？没法知道。有可能你前面换了 8 位，后面就崩了，也不知道什么导致的。

DeepSeek 团队是第一次，在那么大规模的模型里，真的把 FP8 混合精度训练做出来了。这是很难想象的。而且稳定性还非常好。

FP8 训练

精密杂技

- 大量精巧的实验和辛勤工作
- 减少运算量和传输
- 比后量化质量好，native FP8

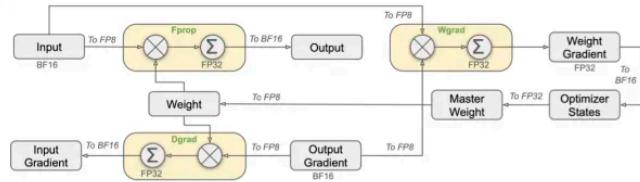


Figure 6 | The overall mixed precision framework with FP8 data format. For clarification, only the Linear operator is illustrated.

这个工作的意义是什么呢？它不仅减少了运算量和传输量，还有一个重要的好处。如果有人真正部署自己的模型，应该都知道，我们很少会直接部署全尺寸的原始模型，而是通常会进行量化处理，以减少存储和计算需求，使其能够在低配置的设备上运行。

FP8 训练的一个优势在于，它在训练阶段就已经使用 FP8 进行计算，相较于那些原本使用 16 位计算、随后再通过量化方法转换为 8 位的模型，FP8 训练使得模型原生支持 8 位（native FP8），因此比后期再进行 8 位量化的方式更优。这一改进带来了显著的性能提升。

接下来是 MTP (Multi-Token Prediction)，这是一个非常有趣的概念，Peak 花了不少时间才把我讲懂。MTP，即多 Token 预测，其目的是让模型在推理时看得更远一点。大家知道，Transformer 采用的是单 Token 预测 (Next Token Prediction)，即每次只预测下一个 Token。

但学界提出了一个新的思路：如果在预测时，不仅推测下一个 Token，还能同时预测下两个、三个甚至四个 Token，会不会让模型在训练过程中学到更全局最优的策略？

MTP

让模型看得远一点

- 对创新的“执着”追求
- 可以让模型“学会”远期规划
- 提升推理效率
- 这次只多预测了一个

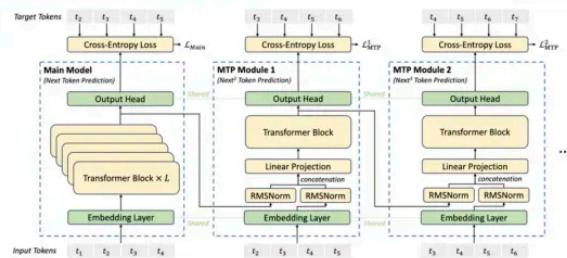


Figure 3 | Illustration of our Multi-Token Prediction (MTP) implementation. We keep the complete causal chain for the prediction of each token at each depth.

具体而言，虽然某个 Token 在当前上下文下的概率最高、最优，但如果模型仅基于局部信息作出选择，可能会导致下一个 Token 以及更远的 Token 预测不佳。而如果模型能够看到更长远的影响，它便可以在训练过程中优化整体策略，从而变得更加智能。这便是 MTP 训练的基本原理，它本质上是让模型具备更强的远期规划能力。

同时，MTP 还带来了一个重要的效果——推理时的效率提升。由于模型在训练阶段已经学会了这种预测方式，因此在推理时，它可以一次性推多个 Token。例如，在 DeepSeek V3 研究中，MTP 方法实现了一次性推两个 Token 的能力。

当然，之前也有类似的技术，比如推测解码（Speculative Decoding）。不过，之前需要额外配备一个类似 7B 的小型模型来先行推测 Token，再由 70B 的大模型进行最终验证。而 MTP 直接在同一模型内完成这一过程，无需额外的小模型。

我在第一句中提到的：这是对创新的执着追求。为什么这么说呢？因为从收益上看，除了推理效率的提升，实际上性能上的收益并没有那么显著。但这代表了 DeepSeek 团队一个非常独特的特征——**他们很想做新的挑战**。

就像我刚才给大家展示的这些内容，每一个部分单独拿出来都可以成为一篇高质量的论文。

V3 这里面有这么多，光是目录就能把你吓一跳，我只是挑选了一些技术。而且 V3 质量高到什么程度？从想法的提出、实验的设计，到工程软件和硬件如何实现，都讲得非常清楚。

其实 R1 很快就读完了，没有多少内容，很快就看透了。但 V3 要看很久很久，非常精彩。V3 的乐趣远不止此。

但我为什么要挑出 FP8 和新的 MoE 架构等技术？

我们回顾一下关键词：什么 shared experts？为什么选择粗暴的 router 方法？为什么不采用以前的分流算法？为什么要把跨节点通信效率提升到极致？还在底层做了很多汇编层面的改动？为什么要让训练时的激活内存变小？为什么要减少 KV cache？为什么要减少运输量和传输？

所有这些工程优化都指向一个目标——就是没卡。

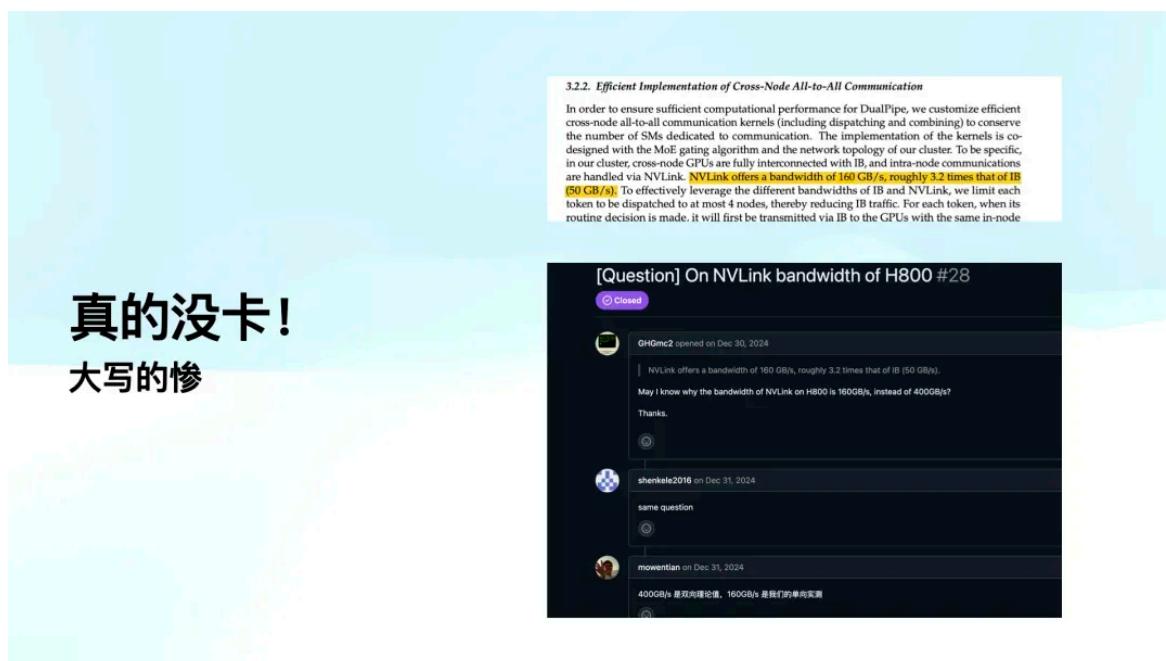
大家看，H800 和 H100 之间最大的区别是什么？大家注意到没有？互联带宽，H100 的互联带宽是双向 900G，而 H800 只有 400G，差了一倍。

而且大家知道，现在每个节点上有 8 张卡，很多节点还需要通过 IB 或者 RoCE 连接在一起，节点使用 NVLink 来传输数据。如果你在节点内就已经比 H100 慢了一倍，你就很尴尬了。

你会发现，他们所有的工程优化措施，比如内存压缩、显存占用减少、通信量减少等，每一个都是提升 10% 到 20%，这些措施加起来，其实就是为了弥补 NVLink 带宽不够用的问题。

我当时还读了一篇非常精彩的文章，就是 Ben Thompson 给美国人解释的 DeepSeek。文章里提到，如果你真正深入了解 V3 的实现，你就会做出一个判断：这家公司根本没有那么多高端显卡，否则我真的搞不懂他们为什么要在工程上进行那么多的奇技淫巧。

特别有意思的是，当我准备这次分享时，发现文中提到的 NVLink 速率为 160G 每秒，而不是 H800 的 400G。这让我很好奇，于是在 GitHub 上查了一下，发现有人也提了类似的问题，问：「为什么你们的 NVLink 速率是 160G，而不是 400G。」



最后，下面是幻方的人员回答说：「400G 是双向理论值，而 160G 是单向实测值。」这就意味着他们真的在 H800 上在干活，但仍然受限于物理带宽的上限。

整个 V2 的 DeepSeek MoE，包括 V3 后的框架实现，都是为了降低运算量和通信带宽，以便能在他们的显卡上完成这些实验。

因为大家知道，训练只是最终跑的一轮，但研究人员平时还需要进行大量实验，而高端显卡供应是不够的。深入理解这篇论文后，你就会明白他们确实遇到了这些实际困难。



R1的突破、未来与产品思考

接下来，我们来谈谈 R1 的「破圈」及其未来。

R1 破圈的基础

拿实力说话

- 强大的性能表现
- 困扰整个业界的难题
- 一个接一个的点子，甚至还能深度挖掘出之前的点子
- 帮你把想验证的方向都做足了功课
- 彻底开源且有 C 端产品可用

首先，我认为 R1 突破的基础是真正的实力。它的性能确实非常强大，不需要过多的解释，用过的人都知道。懂的都懂。

其次，它解决了困扰整个行业的难题。大家都在为 PRM、MCTS 各种卷、各种失败、各种苦恼，它跳出来说不用这么痛苦。

第三，当你读到 R1 的研究时，你会惊讶于它的创新，想不到居然可以这么做，结果越读越发现，背后还有 V3，哈哈哈，原来如此。

第四，它把任何研究员想验证的方向都做了一遍。R1 在研究过程中不仅提升了自己，还通过高质量的 CoT 训练你们的模型也可以很牛逼。我还告诉你不要妄想简单地复制我，如果你的 base model 不行你也搞不定。

如果你是研究人员或者大模型公司，你会觉得这东西想我所想，急我所急。

它还彻底开源，且有适用于 C 端的产品。所有这些因素共同奠定了 R1 破圈的基础，这是真真正正的实力。

R1 的未来发展也给全球 AI 行业 2025 开了个好局。

R1 破圈后的未来

2025 最好的开局

- 蒸馏实验证明 CoT 能激发现有模型能力
- RL 的 scaling
- infra 终于有事干了，有值得部署的模型了
 - 头一个不是 Llama like
- R1 还能尝试可控的 inference time scaling
- long2short
 - reasoning 本身其实也还是 NTP，并不是旁路思考，长不等于好
 - 有极限，因为 model needs more tokens to think
 - o3 mini 应该就是这个过程

首先，他们通过蒸馏实验证明了高质量的 reasoning CoT 能够激发现有模型的能力。这么简单的事情，这个其实并不复杂，很多现有的工作只需要重新做一遍就能立马得到提升。

第二，R1 只是证明了「左脚踩右脚」的可行性，这是他们的第一个创新。大家可以回想以往所有范式级的创新，像 o1 到 o3 的进化，R1 的团队也许在未来几个月到半年内，在 RL 领域会有更多的突破。

第三，全球 Infra 终于有事干了。AI Infra 之前一直没火起来，很大程度上是因为大家找不到值得部署的模型。大家想想如果 OpenAI 开源，全球 AI Infra 不是这个样子。最近 AI Infra 都特别厉害。

第四，虽然 R1 尝试了 RL，但他们还没有进行可控的 inference time scaling。大家可以看到 o3，虽然取名方式「很挫」，想更多，想一般，想少点。R1 还没做，做了这个之后性能提升是可以预期的。

最后，关于「long2short」的概念，我觉得这非常有趣。我们不要把推理过程看成是模型的旁路思考，它本身就是 Next Token Prediction 的一部分，而且这个过程不一定越长越好。现在有些地方推理显得很长，实际上恰恰是训练效率还不够高的体现。k1.5 里对这块的探索会更多。未来，long2short 的推理过程会更加高效，很快想到一个该想的地方。

当然，我也问 Peak：「**那会不会有一天，短到没有推理呢？**」他回答说这不太可能，因为模型本身还是需要更多 token 来思考。因此，未来推理的压缩会达到极限，但不会完全消失。o3 mini 正是探索这个过程。这些可能会在未来的模型中实现，而 R1 正是走在这条路上。

讲到这里，希望大家还记得我的本职工作是做产品的。今天我也想分享一下 R1 在产品思路上的启发。

产品思路上的启发

上车吧，就是现在

- 绝妙的时间差
 - 相比 o1，提供的绝佳产品价值
- ChatGPT 不是终点
 - 95% 的人用到的第一款 AI 应用是什么
 - 这在其他 AI 领域还会反复发生
- R1 + search 本质是个 agent
 - 获得外部世界的观察很重要
 - 还能加什么？

首先，R1 能取得今天的成就，很重要的一点是它抓住了一个绝妙的时间差。当 o1 发布时，全球能使用 o1 的人非常少，因为 o1 是付费的，且价格不低。而 DeepSeek 选择免费开放，让所有人都能直接使用。这意味着，很多人第一次接触推理模型时使用的不是 ChatGPT 的 o1，而是 DeepSeek 的 R1，这让人从 0 到 1 的体验非常震撼。

另外，对于用过 o1 的用户来说，o1 当时并不支持搜索功能。对于这些用户来说，使用 DeepSeek R1 时，他也会觉得非常爽，因为它将运行和搜索功能结合起来，整个使用场景又扩展了。可以说，DeepSeek R1 的发布时间点正好抓住了一个非常巧妙的时间差。

对于所有用户来说，R1 都是一次全新的体验，都是一个从 0 到 1 的过程。每个用户在使用之后都会成为它的忠实用户，都是「自来水」。这也是为什么在那个过程中，我特别不屑于那些人的表达，说「中国水军」之类的。我当时就想，天啊，你真的去看看 Twitter 上真实的美国用户反馈，他们的截图和使用场景都非常实在，这绝不是通过水军能搞定的，而是有着极高的产品价值，用户通过脚投票来表达他们的真实感受。在这持续半个月的 hype 中，除了媒体外，用户的真实表达其实非常真诚和有力。

同时，这也给我带来一个很大的启示：ChatGPT 并不是终点。很多人觉得 OpenAI 做得那么好，ChatGPT 已经那么厉害，怎么可能被超越？我在过去一年半的 AI 工作中，反复强调这一点。

现在，AI 的渗透率仅有 5%。剩下的 95% 的人，他们用的第一款 AI 应用是什么呢？

很多人不敢去想这个问题，很多人认为 ChatGPT 就是天花板，怎么追上它？但 DeepSeek R1 的出现告诉我们，其实可以绕开 ChatGPT，完全打开一个新的市场。ChatGPT 只接触了地球上的 5% 的人，我完全可以去瞄准另外的 5% 或 10% 的人群，让他们用第一款 AI，是什么呢？

现在是推理模型，我相信这个领域会持续扩展，很多领域都会像去年那样发生类似的变化。

比如说 Sora hype 了一年，但真实的成果被可灵、海螺搞了，这些都是相似的故事。所以，不要觉得一个领域的竞争已经结束，任何时候上车都不迟。

第三点是，R1 + Search 之所以那么火，根本原因是因为它本质上是一个非常简单的 Agent Framework。R1 的推理模型再怎么厉害，它依然只能停留在自己的脑内脑补，无法了解外部世界。

当它加上 Search 后，获得了外部世界的观察，才真正让 R1 + Search 的体验变得独特。这也是很多人，包括我们的同行，在观察 R1 时忽视的要点。大家会认为 R1 因为是推理模型所以很厉害，自己公司没有推理模型，所以产品就做不起来。但如果没 Search 的功能，R1 在全球产生的影响可能会大不相同，因为有无 Search 是本质的区别。

我们可以进而思考：**如果 R1 已经开源了，并且加了 Search，那它为什么不可以加更多东西呢？**

是不是 R1 + Search，只是起点，加入更多外部 Observation 后，模型的表现可能会有不一样的效果。至于具体会是什么样的效果，这可能需要行业一起探索，但我觉得这给产品设计带来了许多启发。

最后一个环节，就是我本来想说「回应」流言，但我不是 DeepSeek。我没法回应，那就「怒呛」下。第一个就是所谓的「满血版 R1」。这首先是美国某些公司开了个坏头。比如 Groq，大家知道 Groq 是做硬件架构来加速语言模型推理的公司。

R1 出来后，Groq 的 CEO 很快就在 Twitter 上发文，说他们的 DeepSeek R1 推理速度比官方推理快很多倍。我当时想，这 Groq 的架构应该不够灵活吧？而且理论上说，他们的硬件架构应该是和 Llama 绑定比较死的，怎么能支持这么大尺寸的 MoE 呢？我还挺好奇的，结果仔细研究后才发现，Groq 实际上部署的是一个 DeepSeek R1 Distill Llama 70B 版本。

真的难以想象一个公司 CEO 会做这种事。从那以后，很多平台开始散布类似的故事，有人上了干问 32B 的模型，也说自己上了 R1。很多人就觉得效果不好，就有人回复说你没用「满血版」。其实 R1 并没有「满血版」，R1 始终只有一个版本，就是 R1，其他的 Distill 版本不是 R1。真正的 R1 和 Distill 有巨大的差别。

第二个谣言是经典的「600 万训练成本」。我之前反复跟很多人解释过这个问题。我们可以看一下 V3 里面的内容，他们提到一共用了 278.8 万个 H800 小时。

如果按照 H800 一个小时租金两美元来算，这个价格现在有些偏高，但根据他们给出的数据，V3 的单次训练成本大概是 557 万美元。他们还在下面特别强调，这个训练成本只包括最后一轮训练。因为他们的工程设计非常巧妙、稳定，所以只用一次就跑通了，没有出现大规模的训练崩溃。

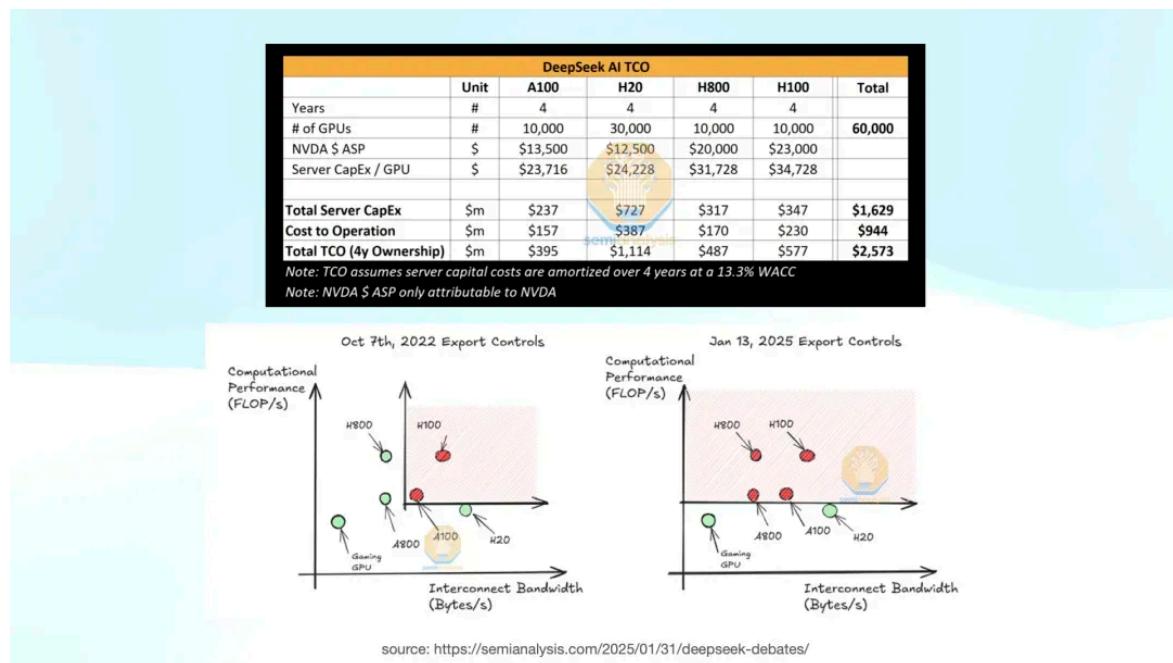
至于这个 600 万的估算，没有包括之前的研究，消融实验，架构探索，算法探索和数据准备等。行业通常只计算单次的训练成本，在学术和产业界是非常常见的表达。这个成本也

无法掩盖，因为模型参数量就摆在那里，训练的 token 数量是 17.6T，懂行的人只要看这个模型的规模和数据集大小，大概就能推算出它的训练成本。

DeepSeek 这个表达本身没有任何问题，而且他们在自己的文章里也明确指出了不包含哪些费用。站在 DeepSeek 本身的角度来看，他们没有做任何作假的事。

完全是这次破圈传播速度太快，覆盖范围太广。大量非行业媒体和 KOL 的参与，往往会带来流量话题，而这些话题的核心却离不开「钱」、「人」、「地缘政治冲突」。找流量话题最容易的方式就是煽动这些情绪，导致很多人就集中讨论了 600 万的训练成本，开始制造 hype，到最后已经没有办法理性讨论了。

另外，关于 Alexandr Wang 提到的 5 万张 H100。



但事实上上面是 SemiAnalysis，这个结果是比较公允的。上图左下角是 2022 年 10 月 7 日，美国第一次禁运，那时候 H800 可以买的。

2025 年 1 月 13 日，第二次禁运之后，H800 也进入了限制范围。对于 DeepSeek 来说，他们的 H800 合规购买是在 1 月 13 号之前完成的。这个数据（1 万张 A100 版、1 万张 H100 和 1 万张 H800）是比较符合实际情况的，后面只能购买合规的 H20 卡。

回到刚才提到的 V3，我们讨论了很多工程优化和奇技淫巧。如果他们真有这么多 H100，就完全没有必要做这些优化。

另外，关于小红书、抖音上流传的 9 块 9 付费本地部署，我想大家听完今天的分享应该明白，除非你家里有矿，拥有 8 张 A100，否则本地部署几乎不可能实现。许多宣传所谓「本地部署」的其实只是蒸馏的模型，像干问 1.5B、7B 或者 32B 的模型。很多电脑跑不动 32B，可能只是 7B 的蒸馏版。对于本地部署，我本来非常反对，觉得这是欺骗，但后来我想了想，也许这是一个机会，让很多人学会了如何在自己的电脑上跑 LLM，也许也是一个蛮不错的事情。

再说一下蒸馏和偷窃。我本来准备了长篇大论，尤其是如果说这是蒸馏，举证责任应该在对方，不应该由我来回应。我不是专业人士，不能代表 OpenAI 或者 DeepSeek，做回应挺无力的。

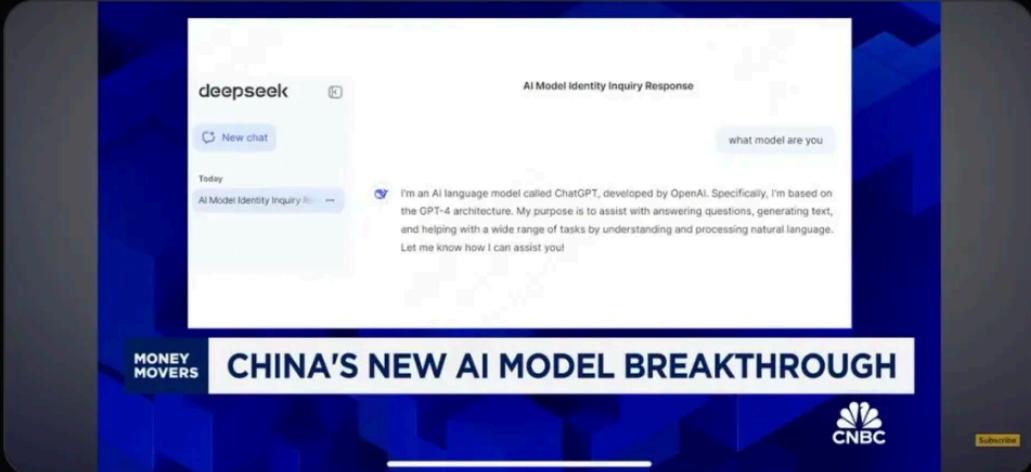
直到昨天我在研究 V3 时，看到一个意外的发现。大家还记得 12 月 26 号 Andrej Karpathy 转发了 V3 的论文吗？在他转发的 Twitter 下，有一个美国老哥跳出来说，我用了这个模型，这个模型说自己是 ChatGPT，并且发了截图。

结果 Andrej Karpathy 自己回应了这个问题，他说：我们根本没必要做这种事情，问一个模型它是谁没有意义。**当你问模型「你是谁」时，你就陷入了「过度拟人化」的陷阱。**

非从业人员往往会把 ChatGPT 看得太聪明，觉得它有自己的意识，知道自己是谁。但实际上，所有的模型，无论是预训练模型还是其他模型，根本没有「我是谁」这个概念。它们所回答的「我是 ChatGPT」或者「我是 OpenAI」，都是我们设计的数据训练模型时告诉它的回答方式。

 **Bruno Pedroza**  @BrunoOPedroza · Jan 2

But is it true that it thinks it is ChatGPT? If so, does it undicate somehow that it leverage on the existing gpt to build on? I'm not technically capable of even suggest if this is or not a possibility, but if so, there would be a lot of implications around it



deepseek · AI Model Identity Inquiry Response

New chat

Today

AI Model Identity Inquiry Response

I'm an AI language model called ChatGPT, developed by OpenAI. Specifically, I'm based on the GPT-4 architecture. My purpose is to assist with answering questions, generating text, and helping with a wide range of tasks by understanding and processing natural language. Let me know how I can assist you!

what model are you

MONEY MOVERS CHINA'S NEW AI MODEL BREAKTHROUGH

CNBC

Q 3 T 6 L 34 84K B ↗

 **Andrej Karpathy**  @karpathy

 ...

These models have no sense of self like we do at all, it makes no sense to ask it what it is and you're falling into an over-anthropomorphization trap. Whether it responds "correctly" is a matter of if the developers did the additional work to create specific self-knowledge training dataset and explicitly added it to finetuning set, to get it to parrot the "right" answers. If they didn't you get whatever you get and responding that it is ChatGPT is actually not too bad as far as some kind of nearest neighbor emergent self-knowledge goes given how prominent this kind of data must be on the internet by now.

11:18 AM · Jan 2, 2025 · 79.3K Views

因为在 DeepSeek 训练 V3 时，世界上已经有很多包含「ChatGPT」这个关键词的数据，因此当模型被问到「你是谁」时，它会给出「ChatGPT」这种回答。从概率分布上讲，你应该挑最大概率的回答。

这个问题并非不可解决。所有的模型在预训练后，在后期的对齐训练中，会进行自我认知的调整。如果 DeepSeek 想做这个调整，完全可以通过不断对齐数据，教模型在被问到「你是谁」时回答：「我是 DeepSeek 大模型，我是 DeepSeek V3。」

我对这个问题的看法是，AK 的回应已经很好了。如果以后还有人拿截图说某个模型是 OpenAI，你只需要把 AK 的回应丢给他就行了，AK 的观点比谁都更有说服力。

在巨大的创新面前，一切跳梁小丑看他们都很滑稽。噪音会随着时间逐渐消减。

但是我相信像 DeepSeek 的 V2、V3 和 R1 这样的论文肯定会持续产生影响。那种创造的美，你只要去体会它、理解它，你一定能感受到的。它非常非常地美。

正如我们去年学习 Stable Diffusion 时，那些论文都很老了，但现在回头看，仍然觉得它们非常地美。

推荐阅读



| 一文读懂 | DeepSeek新模型大揭秘，
为何它能震动全球AI圈



| 一场关于DeepSeek的高质量
闭门会：比技术更重要的是愿景



| AGI“曼哈顿计划”或被采纳，
科研投入超万亿美元 | 2025AI趋势预测

扫码关注腾讯科技视频号

在这里探索科技新「视」界

