

理解推理大型语言模型

构建和完善推理模型的方法与策略



塞巴斯蒂安·拉施卡, 博士

2025年2月5日

♡ 368

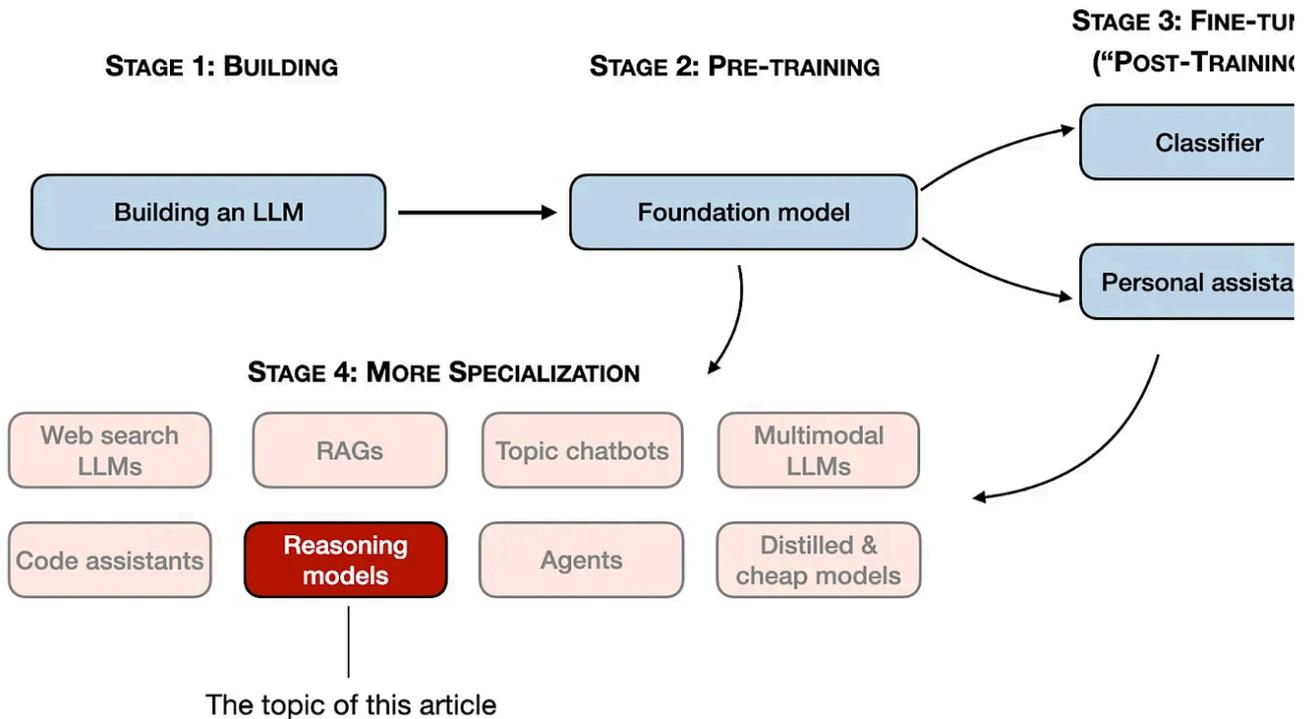
💬 14

🔄 34

Sh

本文描述了构建推理模型的四种主要方法，或者说我们如何增强大型语言模型（LLMs）的推理能力。我希望这能提供有价值的见解，并帮助您在快速发展的文献和围绕这一主题的炒作中进行导航。

在2024年，LLM领域见证了日益增长的专业化。除了预训练和微调，我们还目睹了专用应用的崛起，从RAG到代码助手。我预计这一趋势将在2025年加速，特别强调领域和应用特定的优化（即“专业化”）。



Stages 1-3 are the common steps to developing LLMs. Stage 4 specializes LLMs for specific use cases.

推理模型的发展是这些专业化之一。这意味着要优化大型语言模型（LLMs），使其在复杂任务中表现出色，这些任务最好通过中间步骤来解决，例如谜题、高级数学和编码挑战。然而，这种专业化并不取代其他LLM应用。因为将LLM转变为推理模型也会引入某些缺点，我将在后面讨论。

为了让您简要了解下面的内容，在本文中，我将：

1. 解释“推理模型”的含义
2. 讨论推理模型的优缺点
3. 概述 DeepSeek R1 背后的方法论
4. 描述构建和改进推理模型的四种主要方法
5. 分享 DeepSeek V3 和 R1 发布后对 LLM 领域的看法
6. 提供在预算紧张的情况下开发推理模型的建议

我希望您觉得这篇文章有用，因为人工智能在今年继续快速发展。

r!

我们如何定义“推理模型”？

如果你从事人工智能（或一般的机器学习）工作，你可能对[va](#)和备受争议的定义有所了解。“推理模型”这个术语也不例外。最终，某人会在一篇论文中正式定义它，但在下一篇论文中又会被重新定义，依此类推。

在本文中，我将“推理”定义为回答需要复杂的多步骤生成及中间步骤的问题的过程。例如，像“法国的首都是什么？”这样的事实问答并不涉及推理；相反，像“如果一列火车以每小时60英里行驶并行驶3小时，它走了多少？”这样的问题则需要一些简单的推理。例如，它需要在得出答案之前识别距离、速度和时间之间的关系

。

If a train is moving at 60 mph and travels for 3 hours, how far does it go?

The train travels 180 miles.

Plain response

To determine the distance traveled, use the formula:

$$\text{Distance} = \text{Speed} \times \text{Time}$$

Given that the speed is 60 mph and the time is 3 hours:

$$\text{Distance} = 60 \text{ mph} \times 3 \text{ hours} = 180 \text{ miles}$$

So, the train travels 180 miles.

Response with intermediate reasoning steps

A regular LLM may only provide a short answer (as shown on the left), whereas reasoning models typically include intermediate steps that reveal part of the thought process. (Note that many LLMs who have not been specifically developed for reasoning tasks can also provide intermediate reasoning steps in their answers.)

大多数现代大型语言模型（LLMs）能够进行基本推理，并可以回答诸如“火车以60英里每小时的速度行驶，行驶3小时，它走了多远？”这样的问题。因此，今天当我们提到推理模型时，通常指的是在更复杂的推理任务中表现出色的LLMs，例如解决难题、谜语和数学证明。

此外，今天大多数被称为推理模型的LLM在其响应中包含“思考”或“思维”过程。LLM是否以及如何实际“思考”是一个单独的讨论。

推理模型中的中间步骤可以以两种方式出现。首先，它们可能会明确地包含在响应中，如前面的图所示。其次，一些推理 LLM，例如 OpenAI 的 o1，会运行多个迭代，其中的中间步骤不会向用户显示。

1) Intermediate reasoning chain that is not explicitly shown to the user

2) Intermediate reasoning steps that are shown as part of the answer

The main answer

ChatGPT o1 Share

What is the last digit of 7^{999} ?

Reasoned about last digit of powers for a couple of seconds

To find the last digit of 7^{999} , note that the last digits of powers of 7 repeat in a cycle of length 4:

$7^1 = 7$	(last digit 7)
$7^2 = 49$	(last digit 9)
$7^3 = 343$	(last digit 3)
$7^4 = 2401$	(last digit 1)

Then it repeats: 7, 9, 3, 1, 7, 9, 3, 1, ...

To determine which of these four digits appears for 7^{999} , we look at $999 \bmod 4$:

$$999 \div 4 = 249 \text{ remainder } 3.$$

Since the remainder is 3, we take the third digit in the cycle (7, 9, 3, 1), which is 3.

Hence, the last digit of 7^{999} is **3**.

"Reasoning" is used at two different levels: 1) processing the input and generating via multiple intermediate steps and 2) providing some sort of reasoning as part of the response to the user.

我们何时应该使用推理模型？

现在我们已经定义了推理模型，我们可以进入更有趣的部分：如何构建和改进用于推理任务的LLM。然而，在深入技术细节之前，考虑推理模型何时真正需要是很重要的。

我们什么时候需要推理模型？推理模型旨在处理复杂任务，例如解决难题、高级数学问题和具有挑战性的编码任务。然而，对于像摘要、翻译或基于知识的问题回答这样的简单任务，它们并不是必需的。事实上，使用推理模型来处理所有事情可能效率低下且成本高昂。例如，推理模型通常使用起来更昂贵、更冗长，并且有时由于“过度思考”而更容易出错。在这里，简单的规则适用：为任务使用合适的工具（或类型的LLM）。

推理模型的主要优点和局限性在下图中总结。

Good at	Bad at
+ Deductive or inductive reasoning (e.g., riddles, math proofs)	- Fast and cheap responses (more inference time)
+ Chain-of-thought reasoning (breaking down multi-step problems)	- Knowledge-based tasks (hallucination)
+ Complex decision-making tasks	- Simple tasks (“overthinking”)
+ Better generalization to novel problems	

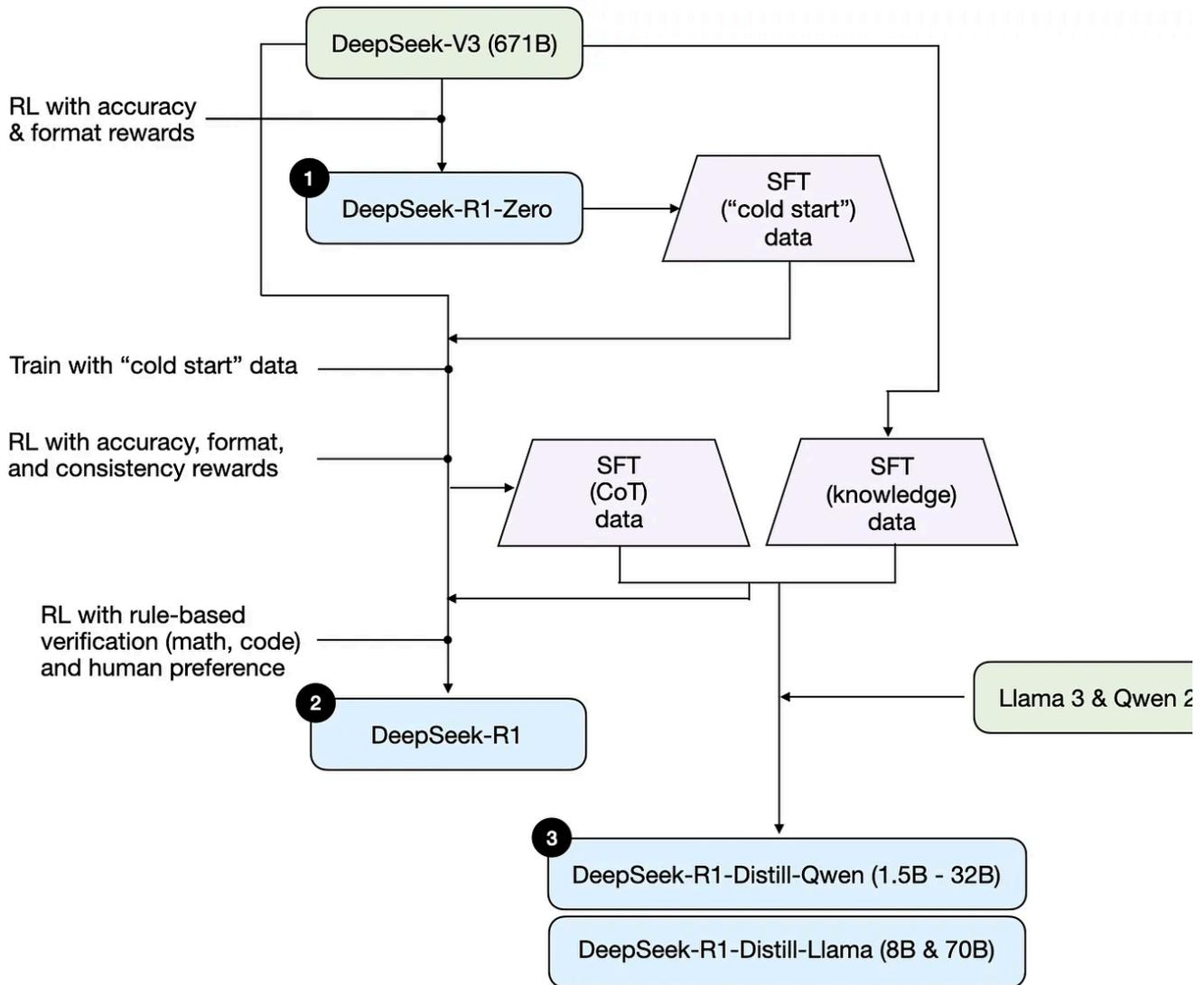
The key strengths and weaknesses of reasoning models.

对DeepSeek训练流程的简要介绍

在下一节讨论构建和改进推理模型的四种主要方法之前，我想简要概述一下DeepSeek R1管道，如DeepSeek R1技术报告中所述。该报告既是一个有趣的案例，也是开发推理LLM的蓝图。

请注意，DeepSeek并没有发布单一的R1推理模型，而是推出了三个不同的变体：DeepSeek-R1-Zero、DeepSeek-R1和DeepSeek-R1-Distill。

根据技术报告中的描述，我在下面的图表中总结了这些模型的开发过程。



DeepSeeks三种不同推理模型的开发过程，这些模型在DeepSeek R1技术报告中进行了讨论。

接下来，让我们简要回顾一下上面图表中显示的过程。更多细节将在下一部分中讨论，我们将讨论构建和改进推理模型的四种主要方法。

(1) DeepSeek-R1-Zero：该模型基于2024年12月发布的671B预训练DeepSeek-V基础模型。研发团队使用强化学习（RL）和两种类型的奖励对其进行了训练。这种方法被称为“冷启动”训练，因为它没有包括通常作为人类反馈强化学习（RLHF）一部分的监督微调（SFT）步骤。

(2) DeepSeek-R1：这是DeepSeek的旗舰推理模型，基于DeepSeek-R1-Zero构建。团队通过额外的SFT阶段进一步优化了它。

RL训练，改进“冷启动” R1-Zero模型。

(3) DeepSeek-R1-Distill*: 使用在前面的步骤中生成的SFT数据，DeepSeek团队对Qwen和Llama模型进行了微调，以增强它们的推理能力。虽然这不是传统意义上的蒸馏，但这个过程涉及在来自大型DeepSeek-R1 671B模型的输出上训练较小的模型（Llama 8B和70B，以及Qwen 1.5B–30B）。

构建和改进推理模型的四种主要方法

在本节中，我将概述当前用于增强LLM推理能力的关键技术，以及构建专门推理模型的方法，如DeepSeek-R1、OpenAI的o1和o3等。

注意：o1 和 o3 的确切工作原理在 OpenAI 之外仍然未知。然而，传闻它们利用了推理和训练技术的结合。

1) 推理时间缩放

提高大型语言模型（LLM）推理能力（或一般能力）的一种方法是推理时间扩展。这个术语可以有多种含义，但在这个上下文中，它指的是在推理过程中增加计算资源以提高输出质量。

一个粗略的类比是，人类在有更多时间思考复杂问题时，往往能够产生更好的回应。同样，我们可以应用一些技术，鼓励LLM在生成答案时“思考”得更多。（尽管，LLM是否真的“思考”是另一个讨论。）

一种直接的推理时间缩放方法是巧妙的提示工程。一个经典的例子是 *chain-of-thought (CoT) prompting*，其中输入提示中包含“逐步思考”等短语。这鼓励模型生成中间推理步骤，而不是直接跳到最终答案。

通常（但并非总是）会导致在更复杂问题上获得更准确的结果（请注意，对于更简单的知识基础问题，例如“法国的首都是什么”，采用这种策略是没有意义的，这再次是一个很好的经验法则，可以帮助您判断推理模型在给定输入查询上的合理性。）

Regular prompting

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: The answer (arabic numerals) is

(Output) 8 ✗

Chain-of-thought prompting

Q: A juggler can juggle 16 balls. Half of the balls are golf ball and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

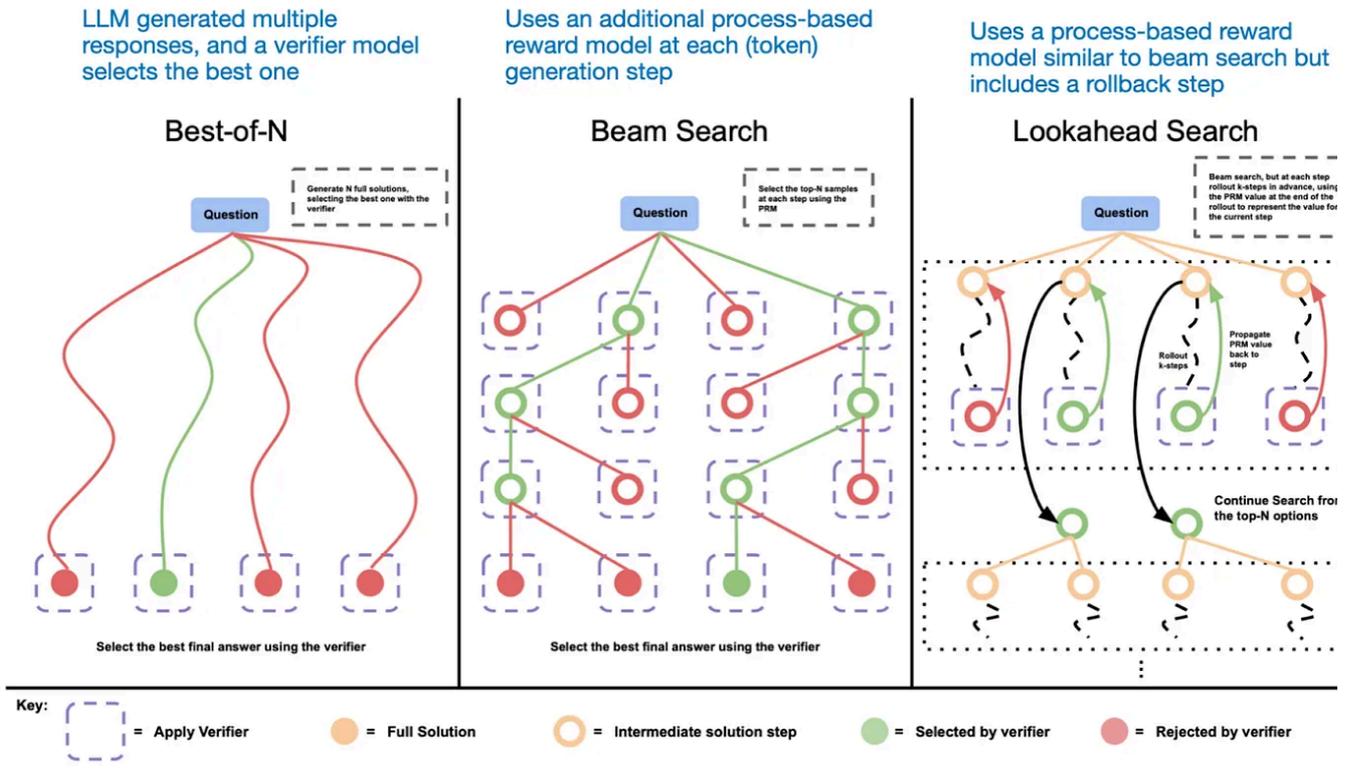
(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*

An example of classic CoT prompting from the 2022 Large Language Models are Zero-Shot Reasoners paper (<https://arxiv.org/abs/2205.11916>).

上述的 CoT 方法可以被视为推理时间的扩展，因为它通过生成更多的输出标记使推理变得更加昂贵。

另一种推理时间缩放的方法是使用投票和搜索策略。一个简单的例子是多数投票，我们让LLM生成多个答案，然后通过多数投票选择正确的答案。类似地，我们可以使用束搜索和其他搜索算法来生成更好的响应。

我强烈推荐我在之前的 [《2024年值得注意的人工智能研究论文（第二部分）》](https://magazine.sebastianraschka.com/p/ai-research-papers-2024-part-2) 文章中描述的 [Scaling LLM Test-Time Compute Optimally can be More Effective than Scaling Model Parameters](https://magazine.sebastianraschka.com/p/ai-research-papers-2024-part-2) 论文 (<https://magazine.sebastianraschka.com/p/ai-research-papers-2024-part-2>)，以获取有关这些不同策略的更多细节。



Different search-based methods rely on a process-reward-based model to select the best answer. Annotated figure from the LLM Test-Time Compute paper, <https://arxiv.org/abs/2408.03314>

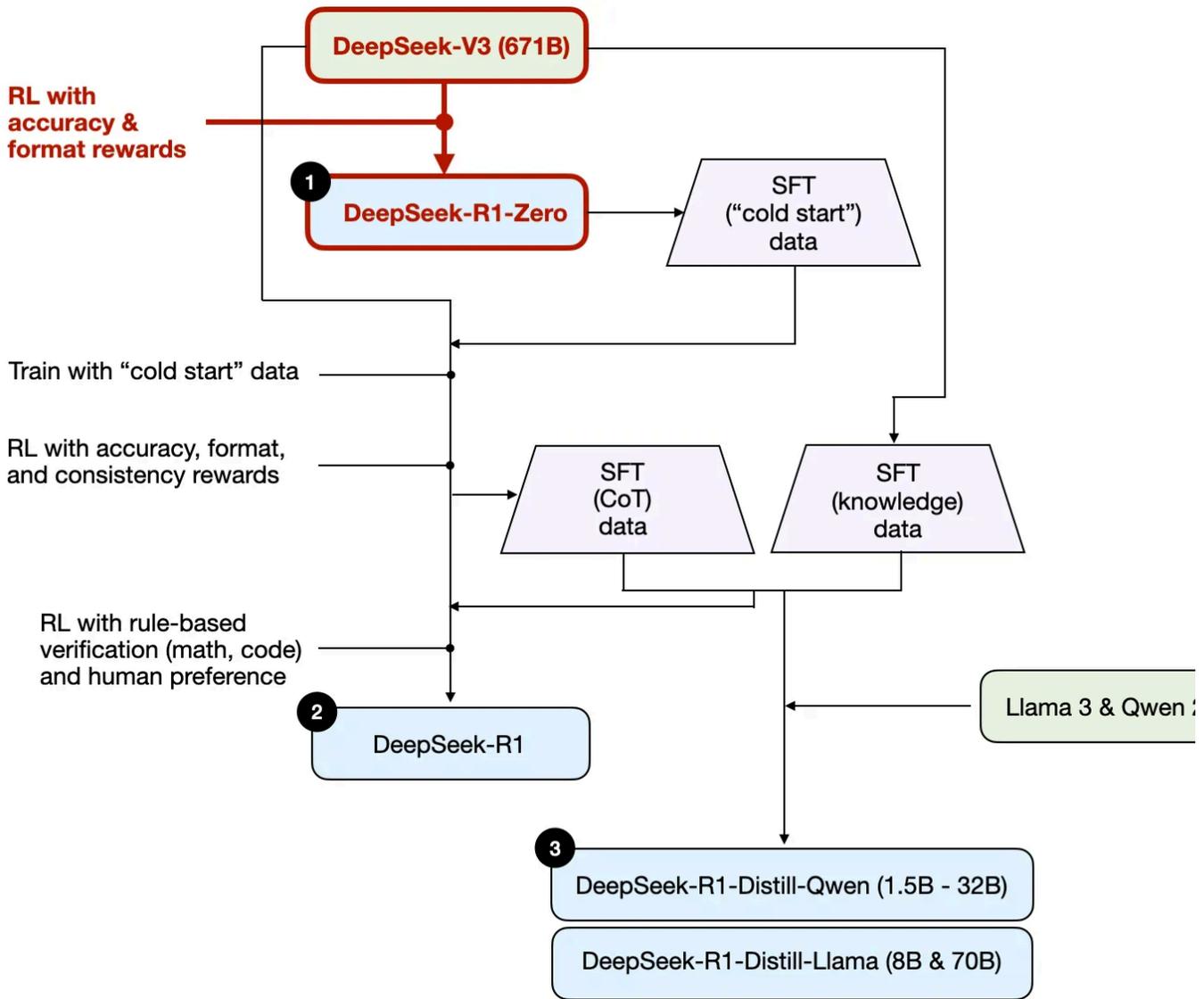
DeepSeek R1 技术报告指出，其模型不使用推理时间缩放。然而，这种技术通常在 LLM 的应用层上实现，因此 DeepSeek 可能在其应用中应用了该技术。

我怀疑OpenAI的o1和o3模型使用推理时间缩放，这可以解释为什么它们相对于像GP T-4o这样的模型来说相对昂贵。除了推理时间缩放，o1和o3可能还使用类似于Deep Seek R1的RL管道进行训练。关于强化学习的更多内容将在接下来的两个部分中讨论。

2) 纯强化学习 (RL)

我个人在DeepSeek R1论文中的一个亮点是他们发现推理作为一种行为从纯粹的强化学习 (RL) 中出现。让我们更详细地解释一下这意味着什么。

如前所述，DeepSeek 开发了三种类型的 R1 模型。第一种，DeepSeek-R1-Zero，是基于 DeepSeek-V3 基础模型构建的，这是他们在 2024 年 12 月发布的标准预训练 LLM。与典型的 RL 流水线不同，DeepSeek-R1-Zero 在没有初始 SFT 阶段的情况下，完全通过强化学习进行训练，如下图所示。



DeepSeek-R1-Zero 模型的开发过程

删除。

尽管如此，这个RL过程与常用的RLHF方法相似，后者通常用于偏好调优LLMs。（我在我的文章中更详细地介绍了RLHF，[LLM Training: RLHF and Its Alternatives](#)。）然而，如上所述，[DeepSeek-R1-Zero](#)中的关键区别在于他们跳过了监督微调（SFT）阶段的指令调优。这就是为什么他们称之为“纯”RL的原因。（尽管，RL在

LLMs的背景与传统的强化学习（RL）有显著不同，这是另一个时间的话题。

对于奖励，他们没有使用基于人类偏好的奖励模型，而是采用了两种类型的奖励：准确性奖励和格式奖励。

- 准确性奖励使用 LeetCode 编译器来验证编码答案，这是一个确定性系统，用于评估数学响应。格式奖励依赖于 LLM 评审，以确保响应遵循预期格式，例如将推理步骤放在 <think> 标签内。
-

令人惊讶的是，这种方法足以让 LLM 发展出基本的推理技能。研究人员观察到一个“恍然大悟”的时刻，此时模型开始在其响应中生成推理痕迹，尽管并没有明确训练其这样做，如下图所示。

Question: If $a > 1$, then the sum of the real solutions of $\sqrt{a - \sqrt{a+x}} = x$ is equal to

Response: <think>
 To solve the equation $\sqrt{a - \sqrt{a+x}} = x$, let's start by squaring both ...
 $(\sqrt{a - \sqrt{a+x}})^2 = x^2 \implies a - \sqrt{a+x} = x^2$.
 Rearrange to isolate the inner square root term:
 $(a - x^2)^2 = a+x \implies a^2 - 2ax^2 + (x^2)^2 = a+x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
 ...
Wait, wait. Wait. That's an aha moment I can flag here.
 Let's reevaluate this step-by-step to identify if the correct sum can be ...
 We started with the equation:
 $\sqrt{a - \sqrt{a+x}} = x$
 First, let's square both sides:
 $a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$
 Next, I could square both sides again, treating the equation: ...
 ...

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. model learns to rethink using an anthropomorphic tone. This is also an aha moment for allowing us to witness the power and beauty of reinforcement learning.

*A figure from the DeepSeek R1 technical report
 (https://arxiv.org/abs/2501.12948) showing the emergence of the "Aha"
 moment.*

虽然 R1-Zero 不是一个顶尖的推理模型，但它通过生成中间的“思考”步骤展示了推理能力，如上图所示。这证实了使用 RL 开发推理模型是可能的，而 DeepSeek 团队是第一个展示（或至少发布）这种方法的团队。

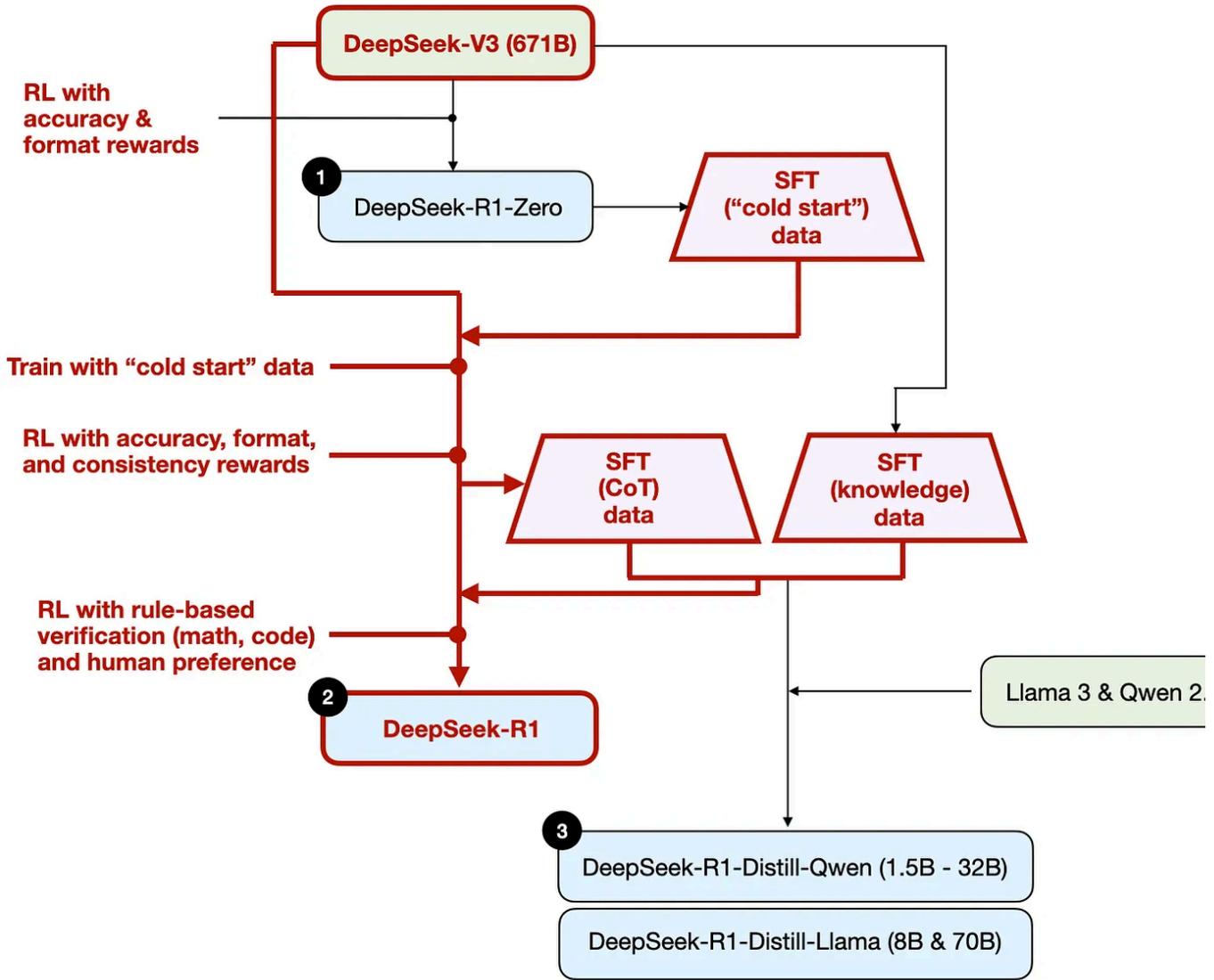
Ahead of AI 是一个由读者支持的出版物。要接收新帖子并支持我的工作，请考虑成为免费或付费订阅者。

Subscribe

3) 监督微调和强化学习 (SFT + RL)

接下来，让我们看看 DeepSeek-R1 的开发，DeepSeek 的旗舰推理模型，它作为构建推理模型的蓝图。该模型通过结合额外的监督微调（SFT）和强化学习（RL）来提高其推理性能，从而改进了 DeepSeek-R1-Zero。

请注意，在 RL 之前包含 SFT 阶段实际上是很常见的，正如在标准的 RLHF 流程中所见。OpenAI 的 o1 可能是使用类似的方法开发的。



DeepSeek-R1模型的开发过程。

如上图所示，DeepSeek团队使用DeepSeek-R1-Zero生成了他们所称的“冷启动”SFT数据。“冷启动”一词指的是这些数据是由DeepSeek-R1-Zero生成的，而DeepSeek-R1-Zero本身并未在任何监督微调（SFT）数据上进行训练。

使用这些冷启动的SFT数据，DeepSeek随后通过指令微调训练模型，接着进行另一个强化学习（RL）阶段。这个RL阶段保留了在DeepSeek-R1-Zero的RL过程中使用的相同准确性和格式奖励。然而，他们增加了一项一致性奖励，以防止语言混合，这种情况发生在模型在响应中切换多种语言时。

RL阶段之后是另一轮SFT数据收集。在这个阶段，使用了最新的模型检查点来生成600K思维链（Co）

SFT 示例，同时使用 DeepSeek-V3 基础模型创建了额外的 200K 知识基础 SFT 示例。

这600K + 200K SFT样本随后被用于另一个回合的强化学习。在这个阶段，他们再次使用基于规则的方法来获得数学和合作问题的准确性奖励，而其他问题类型则使用人类偏好标签。

最终模型 DeepSeek-R1 相较于 DeepSee R1-Zero 在性能上有显著提升，这得益于额外的 SFT 和 RL 阶段，如下表所示。

Model	Math benchmarks			Bio, physics & chemistry	Code benchmarks		Higher is
	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces	
	pass@1	cons@64	pass@1	pass@1	pass@1	rating	
OpenAI-o1-mini	63.6	80.0	90.0	60.0	53.8	1820	
OpenAI-o1-0912	74.4	83.3	94.8	77.3	63.4	1843	
RL only → DeepSeek-R1-Zero	71.0	86.7	95.9	73.3	50.0	1444	
SFT + RL → DeepSeek-R1	79.8		97.3	71.5	65.9	2029	

OpenAI A1 和 DeepSeek R1 模型的基准比较。来自 DeepSeek-R1 技术报告的注释图 (<https://arxiv.org/abs/2501.12948>)。

4) 纯监督微调 (SFT) 和蒸馏

到目前为止，我们已经介绍了三种构建和改进推理模型的关键方法：

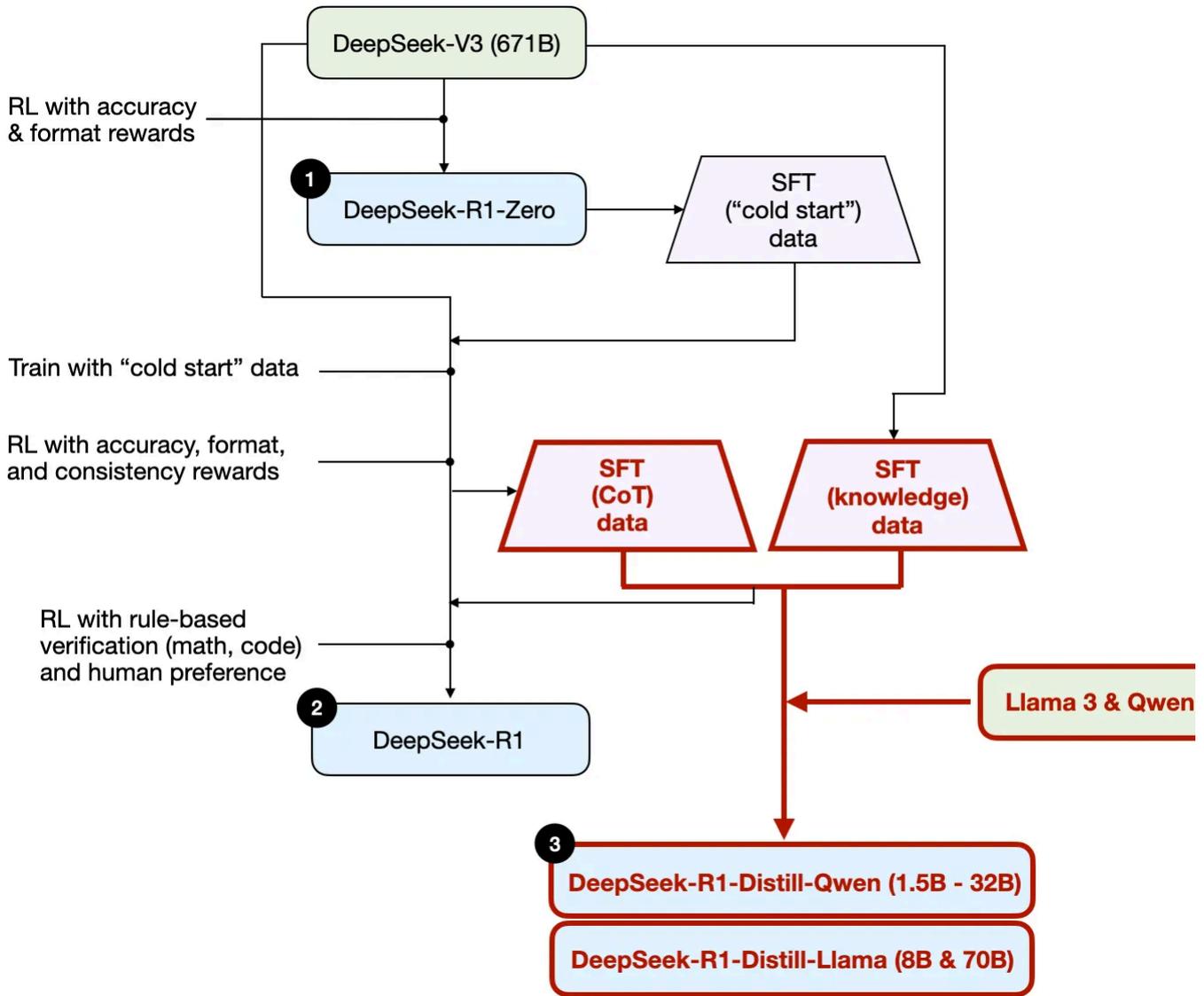
1. 推理时间缩放，一种在不训练或以其他方式修改基础模型的情况下提高推理能力的技术。
2. 纯强化学习 (RL)，如在 DeepSeek-R1-Zero 中所示，表明推理可以作为一种学习行为在没有监督微调的情况下出现。
3. 监督微调 (SFT) 加上强化学习，导致了 DeepSeek-R1，DeepSeek'的旗舰推理模型。

那么，剩下的是什么？模型“蒸馏”。

令人惊讶的是，DeepSeek 还发布了通过他们称之为 *distillation* 的过程训练的小型模型。然而，在 LLM 的背景下，蒸馏并不一定遵循深度学习中使用的经典知识蒸馏方法。传统上，在知识蒸馏中（如我在《机器学习问答与人工智能》一书第六章中简要描述的），一个较小的学生模型在较大教师模型的 logits 和目标数据集上进行训练。

相反，这里提到的蒸馏是指在由更大 LLM 生成的 SFT 数据集上，对较小的 LLM 进行指令微调，例如 LLa 8B 和 70B 以及 Qwen 2.5 模型（0.5B 到 32B）。具体来说，这些更大的 LLM 是 DeepSeek-V3 和 DeepSeek-R1 的一个中间检查点。实际上，用于这个蒸馏过程的 SFT 数据与用于训练 DeepSeek-R1 的相同数据集，如前一部分所述。

为了澄清这个过程，我在图表中突出显示了蒸馏部分。



DeepSeek-R1-Distill mo 的开发过程 dels.

Why 他们为什么开发这些提炼模型？在我看来，有两个关键原因。

1. 较小的模型更高效。这意味着它们运行成本更低，但可以在低端硬件上运行，这使得这些模型对像我这样的许多研究人员和爱好者特别有吸引力。

2. 一个纯粹SFT的案例研究。这些提炼的模型作为一个有趣的基准，展示了纯监督微调（SFT）在没有强化学习的情况下可以将模型提升到何种程度。

下表比较了这些蒸馏模型与其他流行模型的性能，以及 DeepSeek-R1-Zero 和 DeepSeek-R1。

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeFo
	pass@1	cons@64	pass@1	pass@1	pass@1	rating
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	144
DeepSeek-R1	79.8		97.3	71.5	65.9	202

蒸馏模型与非蒸馏模型的基准比较。来自DeepSeek-R1技术报告的注释图 (<https://arxiv.org/abs/2501.12948>)。

正如我们所看到的，蒸馏模型明显弱于DeepSeek-R1，但相对于DeepSeek-R1-Zero却出奇地强，尽管它们的规模小了几个数量级。值得注意的是，这些模型与o1 mini的表现相比也相当出色（我怀疑o1-mini本身可能是o1的一个类似蒸馏版本）。

在结束本节的结论之前，还有一个有趣的比较值得提及。DeepSeek团队测试了在DeepSeek-R1-Zero中观察到的突现推理行为是否也会出现在较小的模式中。为此，他们将DeepSeek-R1-Z的相同纯RL方法直接应用于Qwen-32B。

本实验的结果总结在下表中，其中 QwQ-32B Preview 作为基于 Qwen 2.5 32B 的参考推理模型，由 Qwen 团队开发（我认为训练细节从未披露）。此比较提供了一些额外的见解，关于纯 RL 是否能够在远小于 DeepSeek-R1-Zero 的模型中引发推理能力

。

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCodeBe
	pass@1	cons@64	pass@1	pass@1	pass@1
pure RL → 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
DeepSeek-R1	79.8		97.3	71.5	65.9

基准比较蒸馏和强化学习在一个较小的32B模型上。来自DeepSeek-R1技术报告的注释图 (<https://arxiv.org/abs/2501.12948>)。

有趣的是，结果表明，蒸馏比纯强化学习的小模型要有效得多。这与这样的观点一致：单靠强化学习可能不足以在这种规模的模型中培养强大的推理能力，而在高质量推理数据上进行监督微调（SFT）在处理小模型时可能是一种更有效的策略。

为了完整性，查看表格中的其他比较将会很有用：

1. Qwen-32B 通过 SFT + RL 进行训练，类似于 DeepSeek-R1 的开发方式。这将有助于确定与纯 RL 和纯 SFT 相比，当 RL 与 SFT 结合时可以取得多少改进。
2. DeepSeek-V3 采用纯 SFT 进行训练，类似于蒸馏模型的创建方式。这将允许进行直接比较，以查看 RL + SFT 相对于纯 SFT 的有效性。

Ahead of AI 是一个由读者支持的出版物。要接收新帖子并支持我的工作，请考虑成为免费或付费订阅者。

结论

在本节中，我们探讨了四种不同的策略来构建和改进推理模型：

1. 推理时间缩放不需要额外的训练，但随着用户数量或查询量的增加，会增加推理成本，使大规模部署变得更加昂贵。尽管如此，它仍然是提高已有强大模型性能的明智选择。我强烈怀疑 o1 利用推理时间缩放，这有助于解释为什么它在每个令牌的基础上比 DeepSeek R1 更昂贵。
2. 纯强化学习在研究中很有趣，因为它提供了对推理作为一种涌现行为的洞察。然而，在实际模型开发中，强化学习 SFT 是首选方法，因为它能产生更强的推理模型。我强烈怀疑 o1 也是使用强化学习 + SFT 进行训练的。更准确地说，我认为 o1 是从一个比 DeepSeek-R1 更弱、更小的基础模型开始的，但通过强化学习 + 和推理时间缩放进行了补偿。
3. 如上所述，RL + SFT 是构建高性能推理模型的关键方法。DeepSeek-R1 是一个很好的蓝图，展示了如何实现这一点。
4. 蒸馏是一种有吸引力的方法，特别是用于创建更小、更高效的模型。然而，限制在于蒸馏并不能推动创新或促使下一代推理模型的出现。例如，蒸馏始终依赖于现有的、更强大的模型来生成监督微调（SFT）数据。

我期待看到的一个有趣方面是将 RL + SFT（方法 3）与推理时间缩放（方法 1）结合起来。这可能是 OpenAI o1 正在做的事情，只是可能基于一个比 DeepSeek-R1 更弱的基础模型，这解释了为什么 DeepSeek-R1 在推理时间表现如此出色，同时保持相对低廉的成本。

关于 DeepSeek R1 的思考

在最近几周，许多人询问我对 DeepSeek-R1 模型的看法。简而言之，我认为它们是一个了不起的成就。作为一名研究工程师

特别感谢这份详细的技术报告，它提供了我可以学习的方法论的见解。

最引人注目的收获之一是推理如何作为纯强化学习中的一种行为出现。令人印象深刻的是，DeepSeek在宽松的开源MIT许可证下开源了他们的模型，这比Meta Llama模型的限制还要少。

它与 o1 有何比较？

DeepSeek-R1比o1更好吗？我会说它大致在同一个水平。然而，突出的地方是DeepSeek-R1在推理时更高效。这表明DeepSeek可能在训练过程中投入了更多，而OpenAI可能更依赖于o1的推理时间扩展。

尽管如此，直接比较o1和DeepSeek-R1是困难的，因为OpenAI并没有透露关于o1的太多信息。例如，我们不知道：

- o1也是专家混合模型（MoE）吗？o1有多大？o1是否只是GPT-4o的一个稍微改进的版本，经过最小的强化学习 + SFT，并且仅在推理时进行了广泛的扩展？
-
-

在不知道这些细节的情况下，直接比较仍然是苹果与橘子的比较。

训练DeepSeek-R1的成本

另一个讨论点是开发DeepSeek-R1的成本。有些人提到~\$6百万的训练成本，但他们可能将DeepSeek-V3（去年12月发布的基础模型）和DeepSeek-R1混淆了。

\$600万的估算是基于假设每个GPU小时\$2以及DeepSeek-V3最终训练运行所需的GPU小时数，该项目最初在2024年12月讨论。

然而，DeepSeek团队从未披露R1的确切GPU小时数或开发成本，因此任何成本估算都纯属猜测。

无论如何，最终，DeepSeek-R1 是开放权重推理模型的一个重要里程碑，其在推理时的效率使其成为 OpenAI的 o1 的一个有趣替代方案。

在有限预算下开发推理模型

开发一个DeepSeek-R1级别的推理模型可能需要数十万到数百万美元，即使是从像DeepSeek-V3这样的开放权重基础模型开始。这对于预算有限的研究人员或工程师来说可能会感到沮丧。

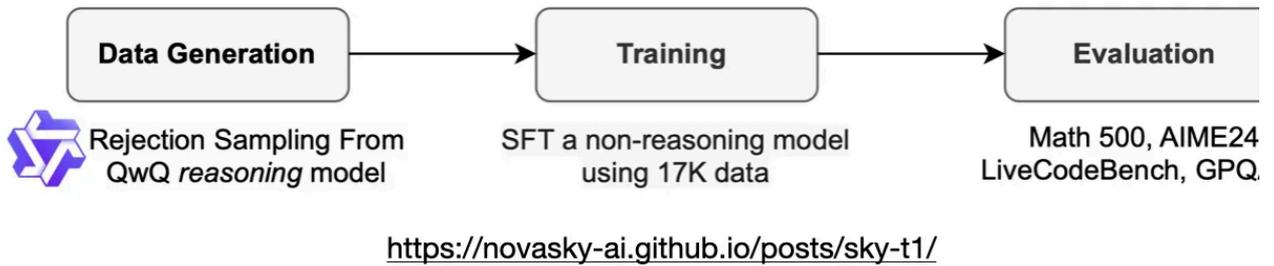
好消息：蒸馏可以大有作为

幸运的是，模型蒸馏提供了一种更具成本效益的替代方案。DeepSeek团队通过他们的R1蒸馏模型证明了这一点，尽管这些模型明显比DeepSeek小，但它们仍然实现了令人惊讶的强大推理性能。然而，即使是这种方法也并非完全便宜。他们的蒸馏过程使用了8个SFT样本，这需要大量的计算资源。

有趣的是，就在DeepSeek-R1发布的几天前，我偶然看到了一篇关于Sky-T1的文章，[这是一个迷人的项目](#)，一个小团队仅使用17K SFT样本训练了一个开放的we 32B模型。总成本？仅为450美元，这比大多数AI会议的注册费还要低。

这个例子强调了尽管大规模训练仍然昂贵，但较小的、有针对性的微调工作仍然可以以较低的成本产生令人印象深刻的结果。

“Alpaca” moment for reasoning data?



\$450 in GPU costs

来自“Sky-T1: 在 \$450 内训练您自己的 O1 预览模型”文章的图, <https://novasky-ai.github.io/posts/sky-t1/>

根据他们的基准测试, Sky-T1 的表现大致与 o1 相当, 这在其低训练成本的情况下令人印象深刻。

纯强化学习预算: TinyZero

虽然Sky-T1专注于模型蒸馏, 但我在“纯强化学习”领域也遇到了一些有趣的w。一个显著的例子是TinyZero, 一个3B参数的模型, 复制了DeepSeek-R1-Zero的方法 (附注: 训练成本不到30美元)。

令人惊讶的是, 即使只有 3B 参数, TinyZero 也表现出一些突现的自我验证能力, 这支持了推理可以通过 p RL 在小模型中出现的观点。

TinyZero 仓库提到研究报告仍在进行中, 我肯定会关注进一步的细节。

User: Using the numbers [19, 36, 55, 7], create an equation that equals 65.

Assistant: Let me solve this step by step.

<think> We need to find an equation using the numbers 19, 36, 55, and 7 exactly once, with basic arithmetic operations, that equals 65. One possible combination is $55 + 36 - 19 + 7$. Let's check: $55 + 36 = 91$, $91 - 19 = 72$, and $72 + 7 = 79$. However, this doesn't equal 65. Another combination is $55 + 36 + 7 - 19$. Let's check: $55 + 36 = 91$, $91 + 7 = 98$, and $98 - 19 = 79$. This also doesn't equal 65. After trying different combinations, I found that $55 + 36 + 7 - 19 = 79$, which is close but not equal to 65. Finally, I found that $55 + 36 - 19 + 7 = 79$, which is still not equal to 65. However, if we try $55 + 36 - 7 + 19$, we get $55 + 36 = 91$, $91 - 7 = 84$, and $84 + 19 = 103$, which is not equal to 65. After further trial and error, I found that $55 + 36 + 7 - 19 = 79$, which is still not equal to 65. However, if we try $55 + 36 - 7 - 19$, we get $55 + 36 = 91$, $91 - 7 = 84$, and $84 - 19 = 65$. This equals 65. **</think>**

<answer> $55 + 36 - 7 - 19$ **</answer>**

来自TinyZero仓库 (<https://github.com/Jiayi-Pan/TinyZero>) 的一幅图，显示该模型能够进行自我验证。（比较基模型的反应会很有趣。）

上述提到的两个项目表明，即使在有限的预算下，关于推理模型的有趣工作也是可能的。虽然这两种方法都复制了 DeepSeek-R1 的方法，一种专注于纯 RL (TinyZero)，另一种专注于纯 SFT (Sky-T1)，但探索这些想法如何进一步扩展将是非常迷人的。

超越传统的SFT：旅程学习

我去年遇到的一个特别有趣的方法在文献

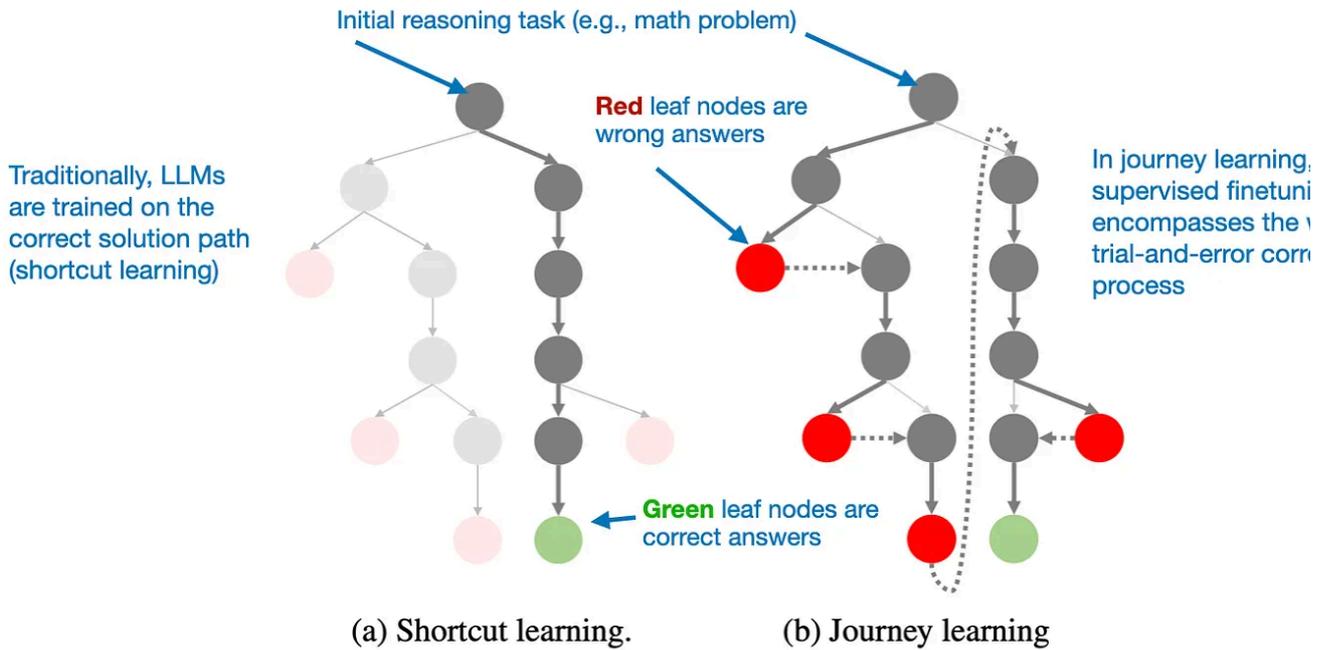
[*O1 Replication Journey: A Strategic Progress Report – Part 1*](#) 中有所描述。尽管标题如此，这篇论文实际上并没有复制 o1。相反，它引入了一种不同的方式来改进蒸馏（纯 SFT）过程。

本文的关键思想是“旅程学习”，作为“捷径学习”的替代方案。

- 快捷学习是指传统的指令微调方法，在这种方法中，模型仅使用正确的解决路径进行训练。

- 旅程学习另一方面也包括错误的解决路径，使模型能够从错误中学习。

这种方法与在TinyZe纯RL训练中观察到的自我验证能力有一定关系，但它专注于通过SFT完全改进模型。通过让模型接触不正确的推理路径及其修正，旅程学习也可能增强自我纠正能力，从而使推理模型以这种方式变得更加可靠。



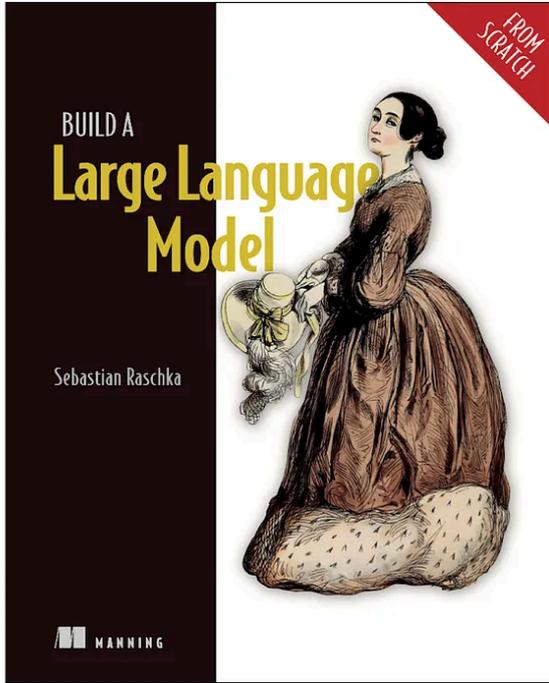
旅程学习，与传统的捷径学习相对，包括SFT数据中的错误解决路径。O1复制旅程的注释图：战略进展报告 - 第1部分 (<https://arxiv.org/abs/2410.18982>)

这可能是未来工作的一个激动人心的方向，特别是在低预算推理模型开发方面，基于RL的方法可能在计算上不切实际。

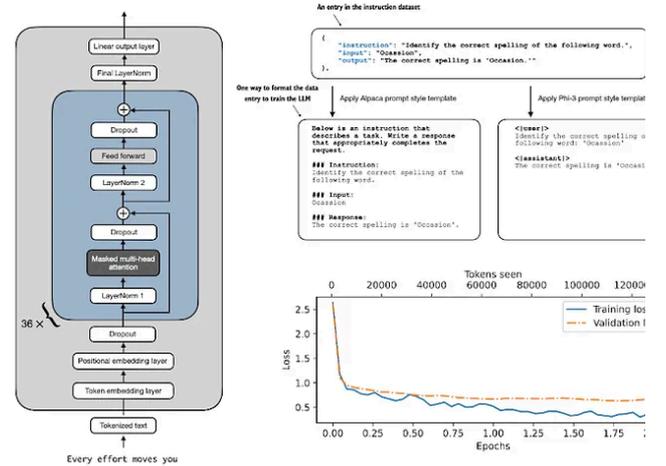
无论如何，目前在推理模型 f 上有很多有趣的工作正在进行，我相信在接下来的几个月里，我们会看到更多令人兴奋的工作！

This magazine is a personal passion project. For those who wish to support me, please consider purchasing a copy of my [Build a Large Language Model \(From](#)

Scratch) book. (I am confident that you'll get lots out of this book as it explains how LLMs work in a level of detail that is not found anywhere else.)



Deeply understand LLMs by implementing them from the ground up



在亚马逊上现在可以购买《从零开始构建大型语言模型》

If you read the book and have a few minutes to spare, I'd really appreciate a brief review. It helps us authors a lot!

您的支持意义重大！谢谢！



368个赞 · 34次重堆

关于此帖的讨论

评论重堆叠



写下评论...

本文由 AINLP 公众号整理翻译，更多 LLM 资源请扫码关注!

AINLP

我爱自然语言处理

一个有趣有AI的自然语言处理社区



长按扫码关注我们