算法工程师视角的DeepSeek R1技术报告解读

论文标题：DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

✏️ 在DeepSeek R1的技术报告中，有些内容并未详细介绍，同时在复现R1模型时也存在一些**难点**：

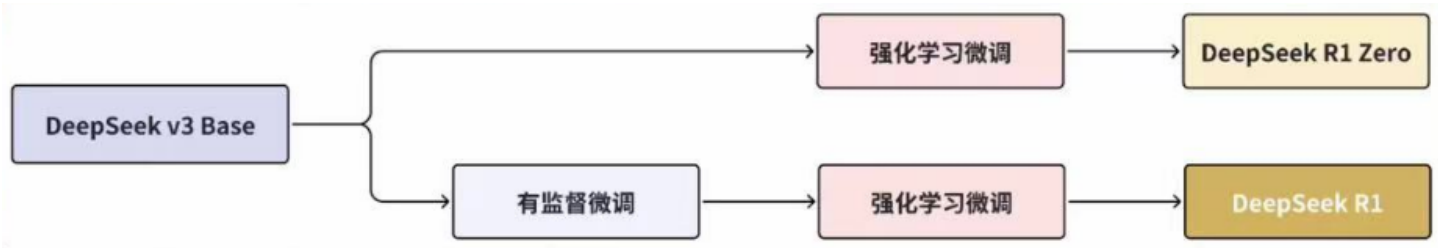
- **训练流程细节**：虽然技术报告中提到了强化学习阶段和知识蒸馏，但省略了一些关键的实现细节。例如，超参数（如学习率、批量大小、奖励缩放因子）的具体设置，生成合成训练数据所用的数据pipeline（如如何安排80W蒸馏样本），以及用于需要与人类偏好对齐的任务的奖励模型架构（如多语言输出中的“语言一致性奖励”）。
- **冷启动数据生成**：报告中提到了创建“高质量冷启动数据”的过程（如人工标准、少样本提示），但没有提供具体的示例或数据集。
- **硬件和基础设施**：文档中没有详细说明所需的计算资源（如GPU集群、训练时间）或软件堆栈优化（如DeepSeek-V3的AMD ROCM集成）。
- **复现难点**：缺少一些关键组件，例如多阶段强化学习的脚本。

🐵 **技术报告概述**

1、DeepSeek-R1-Zero证明了只采用强化学习（RL），而无需SFT作为预备步骤就能展示出显著的推理能力。

2、DeepSeek-R1采用两个RL阶段、两个 SFT 阶段，实现了匹配OpenAI-o1的性能。

3、知识蒸馏让小模型具备强大的推理能力。



DeepSeek-R1-Zero

本节探讨LLM在没有任何监督数据的情况下发展推理能力的潜力，重点关注它们通过纯强化学习过程的自我进化。

Group Relative Policy Optimization（GRPO）

这是一种类似于PPO的强化学习算法（PPO（Proximal Policy Optimization）的核心思想是通过限制策略更新的幅度，确保新策略与旧策略的差异不会过大，从而稳定训练。PPO的详细介绍见之前的强化学习笔记）。

PPO需要三个模型，即policy model，critic model和reward model，这三个模型往往是基于同一个SFT模型修改后得到的。为了节省RL的训练成本，GRPO算法采用group scores取代了critic model。对于用户问题q，GRPO从旧策略 $\pi_{\theta_{old}}$ 采样一系列的输出 $\{o_1, o_2, \cdots, o_G\}$，然后通过最大化以下目标来优化策略：

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$
$$\frac{1}{G}\sum_{i=1}^{G}\left(\min\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}A_i, \text{clip}\left(\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1-\varepsilon, 1+\varepsilon\right)A_i\right) - \beta\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right)\right), \tag{1}$$

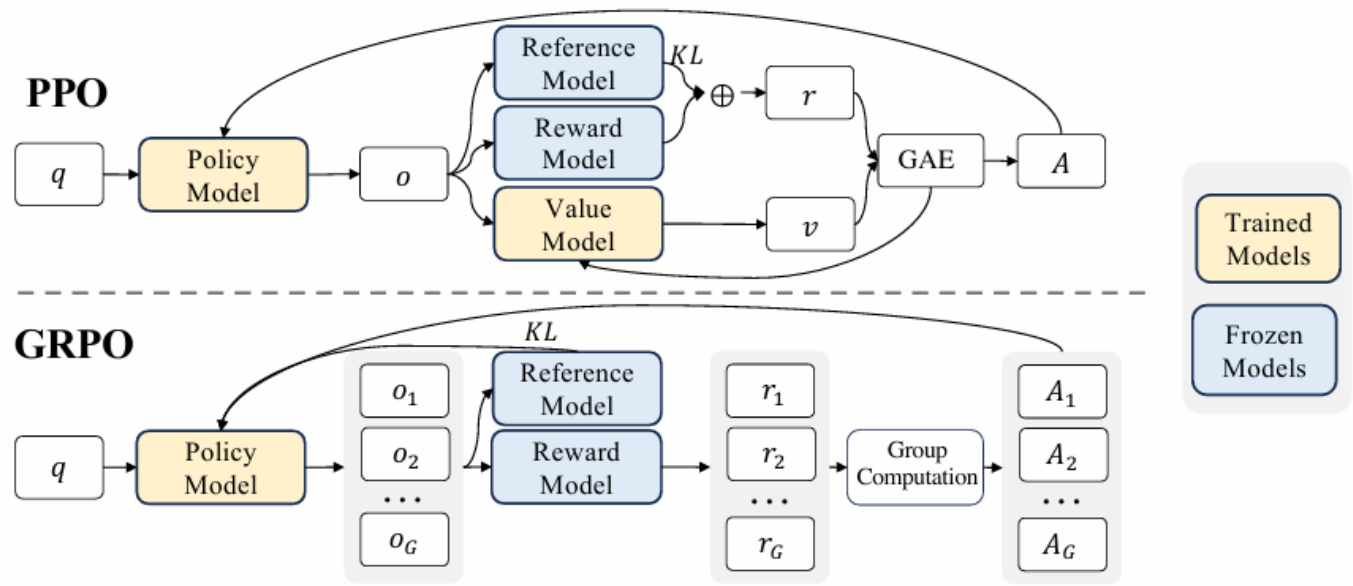
$$\mathbb{D}_{KL}\left(\pi_\theta||\pi_{ref}\right) = \frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{ref}(o_i|q)}{\pi_\theta(o_i|q)} - 1, \tag{2}$$

其中 A_i 是Advantage，它使用对应于每个组内输出的一组奖励$\{r_1, r_2 \ldots\ldots r_G\}$计算：

$$A_i = \frac{r_i - mean(\{r_1, r_2, \cdots, r_G\})}{std(\{r_1, r_2, \cdots, r_G\})}. \tag{3}$$

对目标函数的详解：

- $\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$ 表示新模型和旧模型在同一输入下得到同一输出的概率比，用于限制更新速度，避免原始模型能力的丧失。
- clip函数也是为了将更新限制在一定范围内，即将 $\frac{\pi_\theta(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}$ 限制在 $[1-\epsilon, 1+\epsilon]$ 之间。
- 采用Advantage而不是reward，目的是为了考虑当前回答相比其他回答的优劣。
- GRPO通过使用reward model来简化计算，减少了对critic model的依赖，降低了内存和计算成本。
- KL散度惩罚项，强制新策略与参考策略 π_{ref} 接近，也是为了防止策略偏离特定约束（如安全策略或预训练模型）。



图源：论文DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models

奖励函数设计

采用了基于规则的奖励系统，包括准确性奖励和格式奖励：

- **准确性奖励**：评估模型回复是否正确。
- **格式奖励**：强制模型将思考过程放在'<think>' 和 '</think>'之间。

作者提出没有考虑神经奖励模型，以降低模型训练的复杂性，避免模型迎合奖励模型偏好，而生成错误的答案。

DeepSeek-R1-Zero的训练模板如下图，要求模型先生成推理过程再给出答案。这样能保证模型在强化学习过程中的自由探索和自主学习，而不受内容偏见的影响（如强制反思性推理或推广特定问题解决策略）。

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:

随着训练的深入，模型在自我进化！

- DeepSeek-R1-Zero能在各种推理任务上取得比o1更好的性能，证明了强化学习能获得强大的推理能力。此外通过采用多数投票，性能能进一步提升。
- 随着训练的进行，DeepSeek-R1-Zero自然的学会了通过更多的思考时间，生成更多的token，来更深入的探索和完善其思考过程。在自我进化的过程中，模型自发学会了自我验证、反思和探索其他解决方法的能力，这都是模型与环境交互进行自我优化的结果，而不是通过显示的编程设计。

- **Aha moment**：在某一时刻后，DeepSeek-R1-Zero学会了通过重新评估其初始方法，为问题分配更多的思考时间。这也证明了无需刻意引导模型如何解决问题，只需要强化学习基于适当的激励，就能自我进化出解决问题的高级策略。（如下图所示，模型学习使用拟人化语气重新思考）

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 $\cdots$
$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$
Rearrange to isolate the inner square root term:
$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$
$\cdots$
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 $\cdots$
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: $\cdots$
$\cdots$

- 缺点：生成内容可读性差，在多语言任务表现不佳。

DeepSeek-R1

受DeepSeek-R1-Zero的结果启发，出现了两个自然问题：

1）加入少量高质量数据作为冷启动，能否进一步提高推理性能或加快收敛速度？

2）如何训练一个用户友好的模型，使其不仅能够生成清晰连贯的CoT，而且展现强大的通用能力？

为了解决这些问题，设计了一个训练 DeepSeek-R1 的流程。 该流程包含四个阶段：

1、冷启动

早期的强化学习训练不稳定，因此首先收集少量的长CoT数据来微调模型。这些数据可以通过如下方式生成：

- 使用few shot prompting，其中包含一个长 CoT 作为示例；
- 直接提示模型生成包含反思和验证的详细答案；
- 以可读格式收集 DeepSeek-R1-Zero 的输出，并通过人工标注者的后期处理来细化结果。

作者收集了几千条数据作为冷启动数据，对 DeepSeek-V3-Base进行微调。冷启动数据的引入显著提升了模型的可读性和推理能力，避免了 RL 训练初期的不稳定。

2、面向推理的强化学习

对冷启动微调后的DeepSeek-V3-Base，采用 DeepSeek-R1-Zero 同样的强化学习训练，增强模型在数据、编程等推理任务的性能。

此外，为了缓解在prompt中有多语言时，COT中经常出现的语言混合现象，在强化学习训练时引入语言一致性奖励，奖励为CoT目标语言词汇的比例。引入语言一致性奖励会损害模型性能，但能提升输出的可读性，符合人类偏好。

3、拒绝采样和监督微调

当上一阶段的强化学习收敛时，利用生成的checkpoint来收集SFT数据，这阶段的数据还包含写作、事实问答和自我认知等非推理型任务数据。

- **推理型数据**：从强化学习checkpoint进行**拒绝采样**生成推理轨迹。

> **拒绝采样**是一种在大模型训练中，常用于采集高质量数据的方法。对每个用户的prompt，利用强化学习checkpoint的模型采样K（通常在10到30之间）个输出，利用奖励模型选择最好的一个。以下是deepseek R1对利用拒绝采样收集数据的解释。

步骤1：生成候选样本

- 从当前模型（提议分布 $q(x)$）中采样生成多个候选文本 $x_1, x_2, \ldots, x_N$。
- 例如，对同一输入提示（prompt），生成10个不同输出。

步骤2：定义目标分布 $p(x)$

- 目标分布通常结合任务需求设计，例如：
  - **质量要求**：$p(x) \propto \text{流畅性}(x) \cdot \text{相关性}(x)$
  - **对齐目标**：$p(x) \propto \exp(\text{奖励模型打分}(x)/\beta)$（$\beta$为温度参数）

步骤3：计算接受概率

- 对每个候选样本 x_i，计算其接受概率：

$$\alpha_i = \frac{p(x_i)}{M \cdot q(x_i)}$$

  - $q(x_i)$：当前模型生成 x_i 的概率。
  - M：需满足 $p(x) \leq M \cdot q(x)$，通常通过动态调整或经验设定。

步骤4：接受或拒绝样本

- 从均匀分布 $U(0,1)$ 采样 u，若 $u \leq \alpha_i$，则接受 x_i，否则拒绝。
- **实际优化**：为避免计算全概率 $q(x_i)$（对大模型开销大），常使用以下简化：
  - 用奖励模型的相对分数代替 $p(x)$。
  - 通过Top-k或核采样（Nucleus Sampling）预筛选候选，减少计算量。

- **非推理型数据**：来自 DeepSeek-V3 的 SFT 数据集。

作者收集了约 80 万条训练样本（60 万条推理型数据和 20 万条非推理型数据），并对 DeepSeek-V3-Base 模型进行了两轮微调。

4、面向所有场景的强化学习

为了使模型更加符合人类的偏好，提升其有用性和安全性，我们引入了一个辅助强化学习阶段。

在这个阶段，我们通过结合奖励信号（reward signals）和多样化的提示分布（diverse prompt distributions）来训练模型，以确保其在不同场景下都能表现出色。

对于推理型数据，采用了DeepSeek-R1-Zero中描述的方法，利用基于规则的奖励来引导模型在数学、代码和逻辑推理等领域的学习过程。

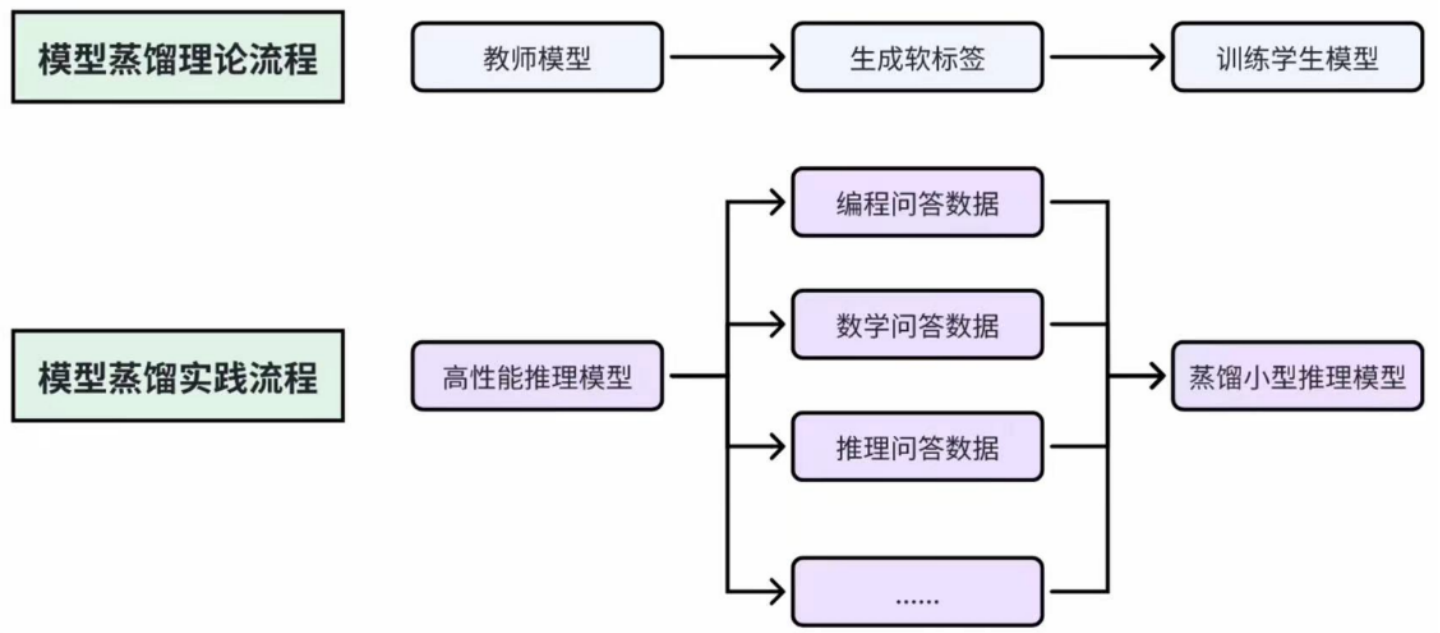
对于非推理型数据，使用奖励模型来捕捉复杂和细微场景中的人类偏好。我们基于DeepSeek-V3的训练流程，采用类似的偏好对和训练提示分布，以确保模型能够理解和适应更广泛的人类偏好。

对于有用性，重点关注最终的摘要部分，确保评估标准强调响应对用户的实用性和相关性，同时尽量减少对底层推理过程的干扰。这样可以确保模型生成的内容对用户来说是直接有用的，而不会因为过多的推理细节而显得冗余。

对于无害性，全面评估模型的整个响应，包括推理过程和最终摘要，以识别和减轻生成过程中可能出现的任何潜在风险、偏差或有害内容。这确保了模型在生成内容时不仅有用，而且安全可靠。

实验结果

- 在通用知识任务上，DeepSeek-R1 的表现显著优于 DeepSeek-V3，尽管略低于 OpenAI-o1-1217，但仍优于其他闭源模型。
- 在创意写作、问答、编辑和摘要等任务上也表现出色，此外，在ArenaHard 和 AlpacaEval 任务上生成的摘要长度简洁，在基于GPT的评估中避免了引入长度偏差，增强了其在这些任务上的鲁棒性。
- 在数学和编程任务上，取得了与 OpenAI-o1-1217 相当的效果
- 利用DeepSeek-R1蒸馏Qwen-32B取得了比o1-mini更好的结果，此外将强化学习应用到蒸馏模型能取得更好的效果。单纯使用强化学习的效果比较差。



知识蒸馏：赋予小模型推理能力

大模型庞大的参数量和计算需求限制了其在很多场景的应用。知识蒸馏是一种将大模型能力迁移到小模型的方法，提升小模型的能力。

如Qwen2.5-7B、Llama-3.1-8B等小模型，其推理能力较弱，利用DeepSeek-R1第三结算生成的80万条训练数据进行两轮微调，让小模型学习DeepSeek-R1的推理模式和生成风格。蒸馏后的模型在推理任务上取得显著的效果提升。

	Benchmark (Metric)	Claude-3.5-Sonnet-1022	GPT-4o 0513	DeepSeek V3	OpenAI o1-mini	OpenAI o1-1217	DeepSeek R1
	Architecture	-	-	MoE	-	-	MoE
	# Activated Params	-	-	37B	-	-	37B
	# Total Params	-	-	671B	-	-	671B
English	MMLU (Pass@1)	88.3	87.2	88.5	85.2	**91.8**	90.8
	MMLU-Redux (EM)	88.9	88.0	89.1	86.7	-	**92.9**
	MMLU-Pro (EM)	78.0	72.6	75.9	80.3	-	**84.0**
	DROP (3-shot F1)	88.3	83.7	91.6	83.9	90.2	**92.2**
	IF-Eval (Prompt Strict)	**86.5**	84.3	86.1	84.8	-	83.3
	GPQA Diamond (Pass@1)	65.0	49.9	59.1	60.0	**75.7**	71.5
	SimpleQA (Correct)	28.4	38.2	24.9	7.0	**47.0**	30.1
	FRAMES (Acc.)	72.5	80.5	73.3	76.9	-	**82.5**
	AlpacaEval2.0 (LC-winrate)	52.0	51.1	70.0	57.8	-	**87.6**
	ArenaHard (GPT-4-1106)	85.2	80.4	85.5	92.0	-	**92.3**
Code	LiveCodeBench (Pass@1-COT)	38.9	32.9	36.2	53.8	63.4	**65.9**
	Codeforces (Percentile)	20.3	23.6	58.7	93.4	**96.6**	96.3
	Codeforces (Rating)	717	759	1134	1820	**2061**	2029
	SWE Verified (Resolved)	**50.8**	38.8	42.0	41.6	48.9	49.2
	Aider-Polyglot (Acc.)	45.3	16.0	49.6	32.9	**61.7**	53.3
Math	AIME 2024 (Pass@1)	16.0	9.3	39.2	63.6	79.2	**79.8**
	MATH-500 (Pass@1)	78.3	74.6	90.2	90.0	96.4	**97.3**
	CNMO 2024 (Pass@1)	13.1	10.8	43.2	67.6	-	**78.8**
Chinese	CLUEWSC (EM)	85.4	87.9	90.9	89.9	-	**92.8**
	C-Eval (EM)	76.7	76.0	86.5	68.9	-	**91.8**
	C-SimpleQA (Correct)	55.4	58.7	**68.0**	40.3	-	63.7

Model	AIME 2024		MATH-500	GPQA Diamond	LiveCode Bench	CodeForces
	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

未来优化方向：

- 通用能力不足：在函数调用、多轮对话、复杂角色扮演和 JSON 输出等任务上，DeepSeek-R1 的表现不如 DeepSeek-V3，未来利用CoT来提升在这些领域的能力。
- 语言混合问题：DeepSeek-R1 主要针对中文和英文优化，处理其他语言时可能出现语言混合问题。
- 提示敏感性：DeepSeek-R1 对提示（prompt）较为敏感，少样本提示（few-shot prompting）会降低其性能，建议使用零样本提示（zero-shot prompting）。
- 软件开发任务：RL在代码编写（尤其是工程级）的大规模应用还受限于评测成本高、过程长等问题。未来将通过拒绝采样软件开发数据或在RL过程中引入异步评估来改善效率，从而解决这一问题。

📚 **一点嗅觉：**

1. **大佬们集体梭哈RL**
   OpenAI偷偷搞的Q-star、Anthropic憋的大招Claude-Next、DeepSeek-r1...这些最拽的实验室全在死磕强化学习。就跟当年全民All in Transformer一样，**现在谁不用RL训练模型，出门都不好意思打招呼**。
2. **RL就是AI的“开窍按钮”**
   以前调教AI像填鸭式教学（SFT微调），现在RL让AI自己琢磨——好比打游戏，菜鸟需要手把手教，但高手都是自己死多了练出来的。**最新模型用RL后，真的会“灵光一闪”自己改代码bug了。**