

DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning 强化学习激励大语言模型的推理能力

DeepSeek-AI

research@deepseek.com

Abstract

我们介绍了第一代推理模型 DeepSeek-R1-Zero 和 DeepSeek-R1。DeepSeek-R1-Zero 是一种通过大规模强化学习 (RL) 训练的模型，无需监督微调 (SFT) 作为初步步骤，表现出卓越的推理能力。通过 RL，DeepSeek-R1-Zero 自然而然地出现了许多强大而有趣的推理行为。然而，它遇到了可读性差和语言混合等挑战。为了解决这些问题并进一步提高推理性能，我们引入了 DeepSeek-R1，它在 RL 之前结合了多阶段训练和冷启动数据。DeepSeek-R1 在推理任务上实现了与 OpenAI-o1-1217 相当的性能。为了支持研究社区，我们开源了 DeepSeek-R1-Zero、DeepSeek-R1 以及基于 Qwen 和 Llama 从 DeepSeek-R1 蒸馏出的六个密集模型 (1.5B、7B、8B、14B、32B、70B)。

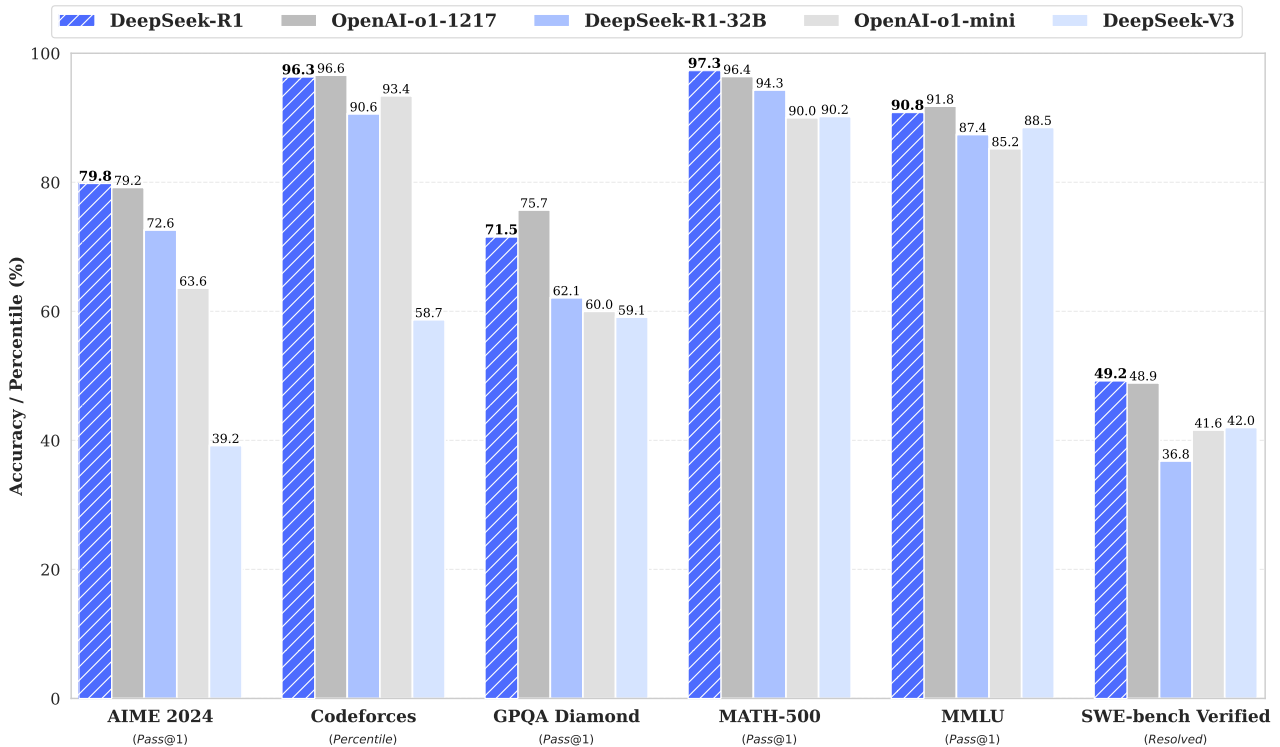


Figure 1 | Benchmark performance of DeepSeek-R1.

Contents

1	Introduction	3
1.1	Contributions	4
1.2	Summary of Evaluation Results	5
2	Approach	6
2.1	Overview	6
2.2	DeepSeek-R1-Zero: Reinforcement Learning on the Base Model 基础模型上的强化学习	7
2.2.1	Reinforcement Learning Algorithm	7
2.2.2	Reward Modeling	7
2.2.3	Training Template	8
2.2.4	Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero	9
2.3	DeepSeek-R1: Reinforcement Learning with Cold Start	12
2.3.1	Cold Start	13
2.3.2	Reasoning-oriented Reinforcement Learning	13
2.3.3	Rejection Sampling and Supervised Fine-Tuning 拒绝抽样和监督微调	14
2.3.4	Reinforcement Learning for all Scenarios 适用于所有场景的强化学习	15
2.4	Distillation: Empower Small Models with Reasoning Capability	15
3	Experiment	16
3.1	DeepSeek-R1 Evaluation	18
3.2	Distilled Model Evaluation	19
4	Discussion	20
4.1	Distillation v.s. Reinforcement Learning	20
4.2	Unsuccessful Attempts	21
5	Conclusion, Limitations, and Future Work	22
A	Contributions and Acknowledgments	27

1. Introduction

In recent years, Large Language Models (LLMs) have been undergoing rapid iteration and evolution (2; 9; 21), progressively diminishing the gap towards Artificial General Intelligence (AGI). 近年来，大型语言模型（LLM）经历了快速迭代和发展 (2; 9; 21)，与通用人工智能（AGI）的差距正在逐步缩小。

Recently, post-training has emerged as an important component of the full training pipeline. It has been shown to enhance accuracy on reasoning tasks, align with social values, and adapt to user preferences, all while requiring relatively minimal computational resources against pre-training. In the context of reasoning capabilities, OpenAI’s o1 (22) series models were the first to introduce inference-time scaling by increasing the length of the Chain-of-Thought reasoning process. This approach has achieved significant improvements in various reasoning tasks, such as mathematics, coding, and scientific reasoning. However, the challenge of effective test-time scaling remains an open question for the research community. Several prior works have explored various approaches, including process-based reward models (18; 33; 34), reinforcement learning (15), and search algorithms such as Monte Carlo Tree Search and Beam Search (6; 32; 38). However, none of these methods has achieved general reasoning performance comparable to OpenAI’s o1 series models.

最近，后训练已成为整个训练流程的重要组成部分。事实证明，它可以提高推理任务的准确性，与社会价值观保持一致，并适应用户偏好，同时与预训练相比，所需的计算资源相对较少。在推理能力方面，OpenAI 的 o1 (22) 系列模型首次通过增加思维链推理过程的长度来引入推理时间扩展。这种方法在数学、编码和科学推理等各种推理任务中取得了显著的改进。然而，有效的测试时间扩展的挑战仍然是研究界的一个悬而未决的问题。先前的一些研究已经探索了各种方法，包括基于过程的奖励模型 (18; 33; 34)、强化学习 (15)，以及蒙特卡洛树搜索和束搜索等搜索算法 (6; 32; 38)。然而，这些方法都没有达到与 OpenAI 的 o1 系列模型相当的一般推理性能。

In this paper, we take the first step toward improving language model reasoning capabilities using pure reinforcement learning (RL). Our goal is to explore the potential of LLMs to develop reasoning capabilities without any supervised data, focusing on their self-evolution through a pure RL process. Specifically, we use DeepSeek-V3-Base as the base model and employ GRPO (28) as the RL framework to improve model performance in reasoning. During training, DeepSeek-R1-Zero naturally emerged with numerous powerful and interesting reasoning behaviors. After thousands of RL steps, DeepSeek-R1-Zero exhibits super performance on reasoning benchmarks. For instance, the pass@1 score on AIME 2024 increases from 15.6% to 71.0%, and with majority voting, the score further improves to 86.7%, matching the performance of OpenAI-o1-0912.

在本文中，我们迈出了使用纯强化学习 (RL) 来提高语言模型推理能力的第一步。我们的目标是探索 LLM 在没有任何监督数据的情况下开发推理能力的潜力，重点关注它们通过纯 RL 过程进行自我进化。具体来说，我们使用 DeepSeek-V3-Base 作为基础模型，并使用 GRPO (28) 作为 RL 框架来提高模型的推理性能。在训练过程中，DeepSeek-R1-Zero 自然而然地出现了许多强大而有趣的推理行为。经过数千个 RL 步骤后，DeepSeek-R1-Zero 在推理基准上表现出超强的性能。例如，AIME 2024 上的 pass@1 分数从 15.6% 提高到 71.0%，通过多数投票，分数进一步提高到 86.7%，与 OpenAI-o1-0912 的性能相当。

However, DeepSeek-R1-Zero encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates a small amount of cold-start data and a multi-stage training pipeline. Specifically, we begin by collecting thousands of cold-start data to fine-tune the DeepSeek-V3-Base model. Following this, we perform reasoning-oriented RL like DeepSeek-

R1-Zero. Upon nearing convergence in the RL process, we create new SFT data through rejection sampling on the RL checkpoint, combined with supervised data from DeepSeek-V3 in domains such as writing, factual QA, and self-cognition, and then retrain the DeepSeek-V3-Base model. After fine-tuning with the new data, the checkpoint undergoes an additional RL process, taking into account prompts from all scenarios. After these steps, we obtained a checkpoint referred to as DeepSeek-R1, which achieves performance on par with OpenAI-o1-1217.

然而，DeepSeek-R1-Zero 面临着可读性差、语言混合等挑战。为了解决这些问题并进一步提高推理性能，我们引入了 DeepSeek-R1，它结合了少量冷启动数据和多阶段训练流程。具体来说，我们首先收集数千个冷启动数据来微调 DeepSeek-V3-Base 模型。然后，我们执行类似 DeepSeek-R1-Zero 的面向推理的 RL。在 RL 过程接近收敛后，我们通过对 RL 检查点进行拒绝采样来创建新的 SFT 数据，并结合 DeepSeek-V3 在写作、事实问答和自我认知等领域的监督数据，然后重新训练 DeepSeek-V3-Base 模型。在使用新数据进行微调后，检查点将经历额外的 RL 过程，其中考虑到所有场景的提示。经过这些步骤，我们获得了一个称为 DeepSeek-R1 的检查点，其性能与 OpenAI-o1-1217 相当。

We further explore distillation from DeepSeek-R1 to smaller dense models. Using Qwen2.5-72B (26) as the base model, direct distillation from DeepSeek-R1 outperforms applying RL on it. This demonstrates that the reasoning patterns discovered by larger base models are crucial for improving reasoning capabilities. We open-source the distilled Qwen and Llama (4) series. Notably, our distilled 72B model outperforms state-of-the-art open-source QwQ-72B-Preview (25) by a large margin, and the distilled 72B and 137B models set a new record on the reasoning benchmarks among dense models.

我们进一步探索从 DeepSeek-R1 蒸馏到更小的密集模型。使用 Qwen2.5-72B (26) 作为基础模型，直接从 DeepSeek-R1 蒸馏的效果优于在其上应用 RL。这表明，更大的基础模型发现的推理模式对于提高推理能力至关重要。我们开源了蒸馏后的 Qwen 和 Llama (4) 系列。值得注意的是，我们蒸馏后的 72B 模型的表现远胜于最先进的开源 QwQ-72B-Preview (25)，而蒸馏后的 72B 和 137B 模型在密集模型的推理基准上创下了新纪录。

1.1. Contributions

Post-Training: Large-Scale Reinforcement Learning on the Base Model 后训练：基础模型上的大规模强化学习

- We directly apply RL to the base model without relying on supervised fine-tuning (SFT) as a preliminary step. This approach allows the model to explore chain-of-thought (CoT) for solving complex problems, resulting in the development of DeepSeek-R1-Zero. DeepSeek-R1-Zero demonstrates capabilities such as self-verification, reflection, and generating long CoTs, marking a significant milestone for the research community. Notably, it is the first open research to validate that reasoning capabilities of LLMs can be incentivized purely through RL, without the need for SFT. This breakthrough paves the way for future advancements in this area.

我们直接将 RL 应用于基础模型，而无需依赖监督微调 (SFT) 作为初步步骤。这种方法允许模型探索解决复杂问题的思路 (CoT)，从而开发 DeepSeek-R1-Zero。DeepSeek-R1-Zero 展示了自我验证、反思和生成 CoT 等功能，标志着研究界的一个重要里程碑。值得注意的是，这是第一个公开研究，验证了 LLM 的推理能力可以纯粹通过 RL 来激励，而无需 SFT。这一突破为该领域的未来发展铺平了道路。

- We introduce our pipeline to develop DeepSeek-R1. The pipeline incorporates two RL stages aimed at discovering improved reasoning patterns and aligning with human preferences, as well as two SFT stages that serve as the seed for the model’s reasoning and

non-reasoning capabilities. We believe the pipeline will benefit the industry by creating better models.

我们引入了开发 DeepSeek-R1 的流程。该流程包含两个 RL 阶段，旨在发现改进的推理模式并与人类偏好保持一致，以及两个 SFT 阶段，作为模型推理和非推理能力的种子。我们相信，通过创建更好的模型，该流程将使行业受益。

Distillation: Smaller Models Can Be Powerful Too 蒸馏：小模型同样强大

- We demonstrate that the reasoning patterns of larger models can be distilled into smaller models, resulting in better performance compared to the reasoning patterns discovered through RL on small models. The open source DeepSeek-R1, as well as its API, will benefit the research community to distill better smaller models in the future.

我们证明，较大模型的推理模式可以蒸馏为较小的模型，与通过强化学习在小型模型上发现的推理模式相比，其性能更佳。开源 DeepSeek-R1 及其 API 将有利于研究界在未来蒸馏出更好的小型模型。

- Using the reasoning data generated by DeepSeek-R1, we fine-tuned several dense models that are widely used in the research community. The evaluation results demonstrate that the distilled smaller dense models perform exceptionally well on benchmarks. DeepSeek-R1-Distill-Qwen-7B achieves 55.5% on AIME 2024, surpassing QwQ-32B-Preview. Additionally, DeepSeek-R1-Distill-Qwen-32B scores 72.6% on AIME 2024, 94.3% on MATH-500, and 57.2% on LiveCodeBench. These results significantly outperform previous open-source models and are comparable to o1-mini. We open-source distilled 1.5B, 7B, 8B, 14B, 32B, and 70B checkpoints based on Qwen2.5 and Llama3 series to the community. 使用 DeepSeek-R1 生成的推理数据，我们对研究社区中广泛使用的几个密集模型进行微调。评估结果表明，蒸馏后的较小密集模型在基准测试中表现非常出色。DeepSeek-R1-Distill-Qwen-7B 在 AIME 2024 上取得了 55.5% 的高分，超过了 QwQ-32B-Preview。此外，DeepSeek-R1-Distill-Qwen-32B 在 AIME 2024 上的得分为 72.6%，在 MATH-500 上的得分为 94.3%，在 LiveCodeBench 上的得分为 57.2%。这些结果明显优于之前的开源模型，可与 o1-mini 相媲美。我们向社区开源了基于 Qwen2.5 和 Llama3 系列的蒸馏后的 1.5B、7B、8B、14B、32B 和 70B 检查点。

1.2. Summary of Evaluation Results

- Reasoning tasks: (1) DeepSeek-R1 achieves a score of 79.8% Pass@1 on AIME 2024, slightly surpassing OpenAI-o1-1217. On MATH-500, it attains an impressive score of 97.3%, performing on par with OpenAI-o1-1217 and significantly outperforming other models.

DeepSeek-R1 在 AIME 2024 上取得了 79.8% Pass@1 的成绩，略高于 OpenAI-o1-1217。在 MATH-500 上，它取得了令人印象深刻的 97.3% 的成绩，表现与 OpenAI-o1-1217 相当，并明显优于其他模型。

(2) On coding-related tasks, DeepSeek-R1 demonstrates expert level in code competition tasks, as it achieves 2,029 Elo rating on Codeforces outperforming 96.3% human participants in the competition. For engineering-related tasks, DeepSeek-R1 performs slightly better than DeepSeek-V3, which could help developers in real world tasks.

在编码相关任务上，DeepSeek-R1 在代码竞赛任务中展现出专家级水平，它在 Codeforces 上获得了 2,029 Elo 评级，比比赛中 96.3% 的人类参与者表现更好。对于工程相关任务，DeepSeek-R1 的表现略优于 DeepSeek-V3，这可以帮助开发人员完成现实世界的任务。

- Knowledge: On benchmarks such as MMLU, MMLU-Pro, and GPQA Diamond, DeepSeek-

R1 achieves outstanding results, significantly outperforming DeepSeek-V3 with scores of 90.8% on MMLU, 84.0% on MMLU-Pro, and 71.5% on GPQA Diamond. While its performance is slightly below that of OpenAI-o1-1217 on these benchmarks, DeepSeek-R1 surpasses other closed-source models, demonstrating its competitive edge in educational tasks. On the factual benchmark SimpleQA, DeepSeek-R1 outperforms DeepSeek-V3, demonstrating its capability in handling fact-based queries. A similar trend is observed where OpenAI-o1 surpasses 4o on this benchmark.

在 MMLU、MMLU-Pro 和 GPQA Diamond 等基准测试中，DeepSeek-R1 取得了出色的成绩，在 MMLU 上的得分为 90.8%，在 MMLU-Pro 上的得分为 84.0%，在 GPQA Diamond 上的得分为 71.5%，远远超过 DeepSeek-V3。虽然其在这些基准测试中的表现略低于 OpenAI-o1-1217，但 DeepSeek-R1 超越了其他闭源模型，展示了其在教育任务中的竞争优势。在事实基准 SimpleQA 上，DeepSeek-R1 的表现优于 DeepSeek-V3，展示了其处理基于事实的查询的能力。OpenAI-o1 在此基准测试中超越 4o 时也观察到了类似的趋势。

- Others: DeepSeek-R1 also excels in a wide range of tasks, including creative writing, general question answering, editing, summarization, and more. It achieves an impressive length-controlled win-rate of 87.6% on AlpacaEval 2.0 and a win-rate of 92.3% on ArenaHard, showcasing its strong ability to intelligently handle non-exam-oriented queries. Additionally, DeepSeek-R1 demonstrates outstanding performance on tasks requiring long-context understanding, substantially outperforming DeepSeek-V3 on long-context benchmarks.

DeepSeek-R1 还在各种任务中表现出色，包括创意写作、一般问答、编辑、总结等。它在 AlpacaEval 2.0 上实现了令人印象深刻的长度控制胜率 87.6%，在 ArenaHard 上实现了 92.3% 的胜率，展示了其智能处理非考试导向查询的强大能力。此外，DeepSeek-R1 在需要长上下文理解的任务上表现出色，在长上下文基准测试中大大优于 DeepSeek-V3。

2. Approach

2.1. Overview

Previous work has heavily relied on large amounts of supervised data to enhance model performance. In this study, we demonstrate that reasoning capabilities can be significantly improved through large-scale reinforcement learning (RL), even without using supervised fine-tuning (SFT) as a cold start. Furthermore, performance can be further enhanced with the inclusion of a small amount of cold-start data. In the following sections, we present: (1) DeepSeek-R1-Zero, which applies RL directly to the base model without any SFT data, and (2) DeepSeek-R1, which applies RL starting from a checkpoint fine-tuned with thousands of long Chain-of-Thought (CoT) examples. 3) Distill the reasoning capability from DeepSeek-R1 to small dense models.

先前的研究严重依赖大量监督数据来提高模型性能。在本研究中，我们证明，即使不使用监督微调 (SFT) 作为冷启动，也可以通过大规模强化学习 (RL) 显著提高推理能力。此外，通过包含少量冷启动数据可以进一步提高性能。在以下部分中，我们将介绍：(1) DeepSeek-R1-Zero，它将 RL 直接应用于基础模型，而无需任何 SFT 数据；(2) DeepSeek-R1，它从使用数千个长思路链 (CoT) 示例微调的检查点开始应用 RL。3) 将推理能力从 DeepSeek-R1 蒸馏到小型密集模型。

2.2. DeepSeek-R1-Zero: Reinforcement Learning on the Base Model 基础模型上的强化学习

Reinforcement learning has demonstrated significant effectiveness in reasoning tasks, as evidenced by our previous works (28; 34). However, these works heavily depended on supervised data, which are time-intensive to gather. In this section, we explore the potential of LLMs to develop reasoning capabilities without any supervised data, focusing on their self-evolution through a pure reinforcement learning process. We start with a brief overview of our RL algorithm, followed by the presentation of some exciting results, and hope this provides the community with valuable insights.

强化学习在推理任务中表现出显著的有效性，这一点可以从我们之前的研究 (28; 34) 中看出。然而，这些研究严重依赖于监督数据，而收集这些数据需要耗费大量时间。在本节中，我们将探索 LLM 在没有任何监督数据的情况下开发推理能力的潜力，重点关注它们通过纯强化学习过程的自我进化。我们首先简要概述我们的 RL 算法，然后介绍一些令人兴奋的结果，希望这能为社区提供有价值的见解。

2.2.1. Reinforcement Learning Algorithm

Group Relative Policy Optimization 群体相对策略优化 In order to save the training costs of RL, we adopt Group Relative Policy Optimization (GRPO) (28), which foregoes the critic model that is typically the same size as the policy model, and estimates the baseline from group scores instead. Specifically, for each question q , GRPO samples a group of outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model π_{θ} by maximizing the following objective

为了节省强化学习的训练成本，我们采用群体相对策略优化 (GRPO) (28)，它放弃了通常与策略模型大小相同的批评模型，而是从群体得分中估计基线。具体来说，对于每个问题 q ，GRPO 从旧策略 $\pi_{\theta_{old}}$ 中抽取一组输出 $\{o_1, o_2, \dots, o_G\}$ ，然后通过最大化以下目标来优化策略模型 π_{θ} ：

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G} \sum_{i=1}^G \left(\min \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)} A_i, \text{clip} \left(\frac{\pi_{\theta}(o_i|q)}{\pi_{\theta_{old}}(o_i|q)}, 1 - \varepsilon, 1 + \varepsilon \right) A_i \right) - \beta \mathbb{D}_{KL}(\pi_{\theta} || \pi_{ref}) \right), \quad (1)$$

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

where ε and β are hyper-parameters, and A_i is the advantage, computed using a group of rewards $\{r_1, r_2, \dots, r_G\}$ corresponding to the outputs within each group

其中 ε 和 β 是超参数， A_i 是优势，使用对应于每组内输出的一组奖励 $\{r_1, r_2, \dots, r_G\}$ 计算：

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \dots, r_G\})}{\text{std}(\{r_1, r_2, \dots, r_G\})}. \quad (3)$$

2.2.2. Reward Modeling

The reward is the source of the training signal, which decides the optimization direction of RL. To train DeepSeek-R1-Zero, we adopt a rule-based reward system that mainly consists of two types of rewards

奖励是训练信号的来源，决定了强化学习的优化方向。为了训练 DeepSeek-R1-Zero，我们采用了基于规则的奖励系统，该系统主要包含两种类型的奖励：

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.

DeepSeek-R1-Zero 的模板。训练期间，**prompt** 将被替换为具体的推理问题。

- Accuracy rewards: The accuracy reward model evaluates whether the response is correct. For example, in the case of math problems with deterministic results, the model is required to provide the final answer in a specified format (e.g., within a box), enabling reliable rule-based verification of correctness. Similarly, for LeetCode problems, a compiler can be used to generate feedback based on predefined test cases.

准确度奖励：准确度奖励模型评估响应是否正确。例如，对于具有确定性结果的数学问题，模型需要以指定的格式（例如，在框内）提供最终答案，从而实现可靠的基于规则的正确性验证。同样，对于 LeetCode 问题，可以使用编译器根据预定义的测试用例生成反馈。

- Format rewards: In addition to the accuracy reward model, we employ a format reward model that enforces the model to put its thinking process between ‘`<think>`’ and ‘`</think>`’ tags.

格式奖励：除了准确性奖励模型之外，我们还采用了格式奖励模型，强制模型将其思考过程置于“`<think>`”和“`</think>`”标签之间。

We do not apply the outcome or process neural reward model in developing DeepSeek-R1-Zero, because we find that the neural reward model may suffer from reward hacking in the large-scale reinforcement learning process, and retraining the reward model needs additional training resources and it complicates the whole training pipeline.

在开发 DeepSeek-R1-Zero 时，我们没有应用结果或过程神经奖励模型，因为我们发现神经奖励模型在大规模强化学习过程中可能会受到奖励黑客攻击，并且重新训练奖励模型需要额外的训练资源，并且使整个训练流程变得复杂。

2.2.3. Training Template

To train DeepSeek-R1-Zero, we begin by designing a straightforward template that guides the base model to adhere to our specified instructions. As depicted in Table 1, this template requires DeepSeek-R1-Zero to first produce a reasoning process, followed by the final answer. We intentionally limit our constraints to this structural format, avoiding any content-specific biases—such as mandating reflective reasoning or promoting particular problem-solving strategies—to ensure that we can accurately observe the model’s natural progression during the RL process. 为了训练 DeepSeek-R1-Zero，我们首先设计一个简单的模板，指导基础模型遵循我们指定的指令。如表 1 所示，此模板要求 DeepSeek-R1-Zero 首先生成一个推理过程，然后生成最终答案。我们有意将约束限制在这种结构格式上，避免任何特定于内容的偏见（例如强制进行反思性推理或推广特定的问题解决策略），以确保我们能够准确观察模型在 RL 过程中的自然进展。

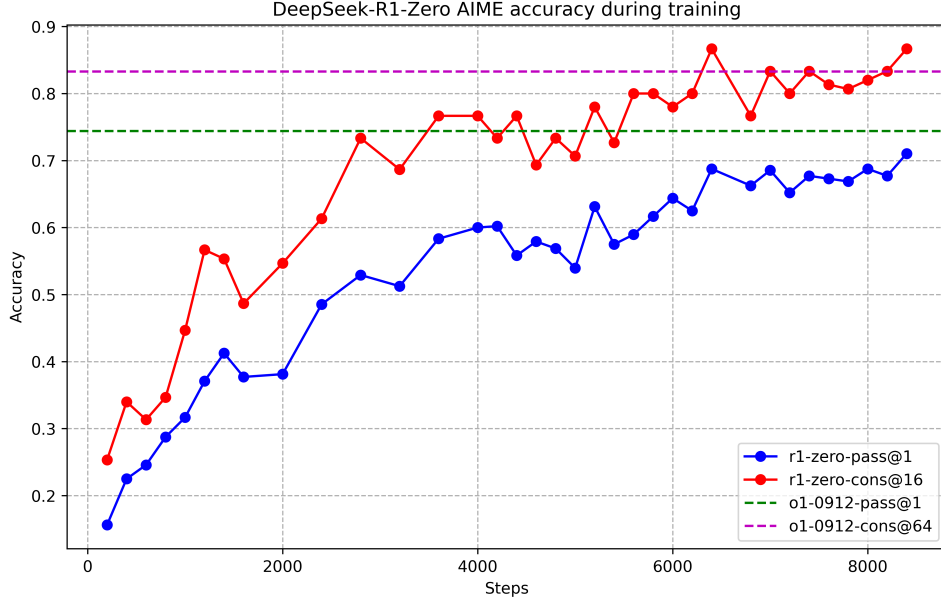


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation. 训练期间 DeepSeek-R1-Zero 的 AIME 准确率。对于每个问题，我们抽取 16 个答案并计算总体平均准确率，以确保评估的稳定性。

2.2.4. Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero

Model	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
DeepSeek-R1-Zero	71.0	86.7	95.9	73.3	50.0	1444

Table 2 | Comparison of DeepSeek-R1-Zero and OpenAI o1 models on reasoning-related benchmarks.

在推理相关基准上对 DeepSeek-R1-Zero 和 OpenAI o1 模型进行比较。

Performance of DeepSeek-R1-Zero Figure 2 depicts the performance trajectory of DeepSeek-R1-Zero on the AIME 2024 benchmark throughout the RL training process. As illustrated, DeepSeek-R1-Zero demonstrates a steady and consistent enhancement in performance as the RL training advances. Notably, the average pass@1 score on AIME 2024 shows a significant increase, jumping from an initial 15.6% to an impressive 71.0%, reaching performance levels comparable to OpenAI-o1-0912. This significant improvement highlights the efficacy of our RL algorithm in optimizing the model’s performance over time.

图 2 描绘了 DeepSeek-R1-Zero 在整个 RL 训练过程中在 AIME 2024 基准上的性能轨迹。如图所示，随着 RL 训练的进展，DeepSeek-R1-Zero 的性能稳步提升。值得注意的是，AIME 2024 上的平均 pass@1 分数显着提高，从最初的 15.6% 跃升至令人印象深刻的 71.0%，达到了与 OpenAI-o1-0912 相当的性能水平。这一显着的改进凸显了我们的 RL 算法在随时间优化模型性能方面的有效性。

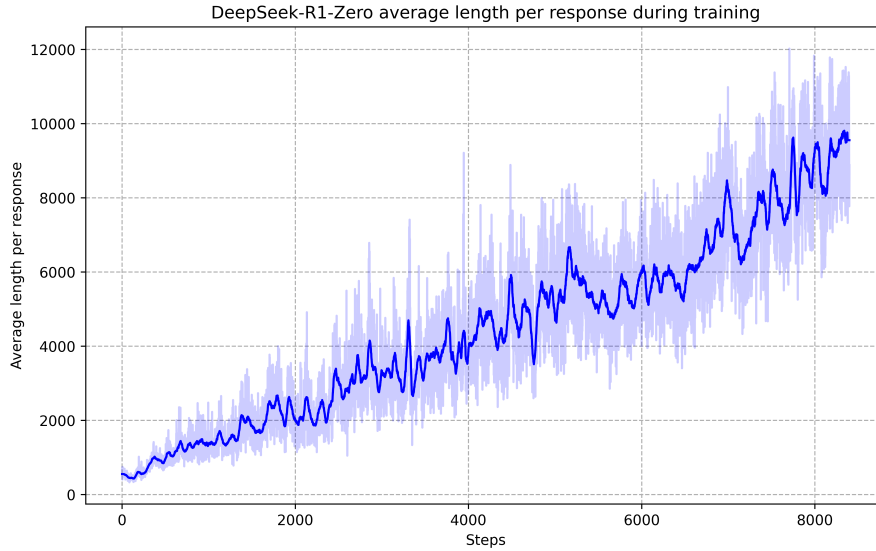


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time. 强化学习过程中，DeepSeek-R1-Zero 在训练集上的平均响应长度。DeepSeek-R1-Zero 自然学会用更多的思考时间来解决推理任务。

Table 2 provides a comparative analysis between DeepSeek-R1-Zero and OpenAI’s o1-0912 models across a variety of reasoning-related benchmarks. The findings reveal that RL empowers DeepSeek-R1-Zero to attain robust reasoning capabilities without the need for any supervised fine-tuning data. This is a noteworthy achievement, as it underscores the model’s ability to learn and generalize effectively through RL alone. Additionally, the performance of DeepSeek-R1-Zero can be further augmented through the application of majority voting. For example, when majority voting is employed on the AIME benchmark, DeepSeek-R1-Zero’s performance escalates from 71.0% to 86.7%, thereby exceeding the performance of OpenAI-o1-0912. The ability of DeepSeek-R1-Zero to achieve such competitive performance, both with and without majority voting, highlights its strong foundational capabilities and its potential for further advancements in reasoning tasks.

表 2 提供了 DeepSeek-R1-Zero 和 OpenAI 的 o1-0912 模型在各种推理相关基准测试中的比较分析。研究结果表明，RL 使 DeepSeek-R1-Zero 能够获得强大的推理能力，而无需任何监督微调数据。这是一项值得注意的成就，因为它强调了该模型仅通过 RL 就能有效学习和概括的能力。此外，DeepSeek-R1-Zero 的性能可以通过应用多数投票进一步增强。例如，当在 AIME 基准测试中使用多数投票时，DeepSeek-R1-Zero 的性能从 71.0% 提升到 86.7%，从而超过了 OpenAI-o1-0912 的性能。DeepSeek-R1-Zero 能够在有或没有多数投票的情况下实现如此具有竞争力的性能，凸显了其强大的基础能力以及在推理任务中进一步发展的潜力。

Self-evolution Process of DeepSeek-R1-Zero DeepSeek-R1-Zero 的自我进化过程 The self-evolution process of DeepSeek-R1-Zero is a fascinating demonstration of how RL can drive a model to improve its reasoning capabilities autonomously. By initiating RL directly from the base model, we can closely monitor the model’s progression without the influence of the supervised fine-tuning stage. This approach provides a clear view of how the model evolves over time, particularly in terms of its ability to handle complex reasoning tasks.

DeepSeek-R1-Zero 的自我进化过程很好地展示了 RL 如何驱动模型自主提高其推理能力。通过直接从基础模型启动 RL，我们可以密切监控模型的进展，而不受监督微调阶段的影响。这种方法可以清楚地了解模型如何随时间演变，特别是在其处理复杂推理任务的能力方面。

As depicted in Figure 3, the thinking time of DeepSeek-R1-Zero shows consistent improvement throughout the training process. This improvement is not the result of external adjustments but rather an intrinsic development within the model. DeepSeek-R1-Zero naturally acquires the ability to solve increasingly complex reasoning tasks by leveraging extended test-time computation. This computation ranges from generating hundreds to thousands of reasoning tokens, allowing the model to explore and refine its thought processes in greater depth.

如图 3 所示，DeepSeek-R1-Zero 的思考时间在整个训练过程中持续改善。这种改善不是外部调整的结果，而是模型内部的内在发展。DeepSeek-R1-Zero 通过利用延长的测试时间计算，自然获得了解决日益复杂的推理任务的能力。这种计算范围从生成数百到数千个推理 tokens，使模型能够更深入地探索和改进其思维过程。

One of the most remarkable aspects of this self-evolution is the emergence of sophisticated behaviors as the test-time computation increases. Behaviors such as reflection—where the model revisits and reevaluates its previous steps—and the exploration of alternative approaches to problem-solving arise spontaneously. These behaviors are not explicitly programmed but instead emerge as a result of the model’s interaction with the reinforcement learning environment. This spontaneous development significantly enhances DeepSeek-R1-Zero’s reasoning capabilities, enabling it to tackle more challenging tasks with greater efficiency and accuracy. 这种自我进化最引人注目的方面之一是，随着测试时间计算的增加，复杂的行为也随之出现。诸如反思（模型重新审视并重新评估其先前的步骤）和探索解决问题的替代方法等行为会自发出现。这些行为不是明确编程的，而是模型与强化学习环境交互的结果。这种自发发展显著增强了 DeepSeek-R1-Zero 的推理能力，使其能够以更高的效率和准确性应对更具挑战性的任务。

Aha Moment of DeepSeek-R1-Zero 顿悟时刻 A particularly intriguing phenomenon observed during the training of DeepSeek-R1-Zero is the occurrence of an “aha moment”. This moment, as illustrated in Table 3, occurs in an intermediate version of the model. During this phase, DeepSeek-R1-Zero learns to allocate more thinking time to a problem by reevaluating its initial approach. This behavior is not only a testament to the model’s growing reasoning abilities but also a captivating example of how reinforcement learning can lead to unexpected and sophisticated outcomes.

在 DeepSeek-R1-Zero 训练过程中观察到的一个特别有趣的现象是“顿悟时刻”的出现。如表 3 所示，此时刻出现在模型的过渡版本中。在此阶段，DeepSeek-R1-Zero 学会通过重新评估其初始方法为问题分配更多思考时间。这种行为不仅证明了模型不断增强的推理能力，而且是一个引人入胜的例子，展示了强化学习如何产生意想不到的复杂结果。

This moment is not only an “aha moment” for the model but also for the researchers observing its behavior. It underscores the power and beauty of reinforcement learning: rather than explicitly teaching the model on how to solve a problem, we simply provide it with the right incentives, and it autonomously develops advanced problem-solving strategies. The “aha moment” serves as a powerful reminder of the potential of RL to unlock new levels of intelligence in artificial systems, paving the way for more autonomous and adaptive models in the future.

这一刻不仅是模型的“顿悟时刻”，也是观察其行为的研究人员的“顿悟时刻”。它强调了强化学习的力量和魅力：我们不是明确地教模型如何解决问题，而是简单地为其提供正确的激励，它就会自主地开发出先进的解决问题的策略。“顿悟时刻”有力地提醒了我们，强化学习有潜力在人工智能系统中开启新的智能水平，为未来更自主、更具适应性的模型铺平道路。

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 \dots

$$\left(\sqrt{a - \sqrt{a + x}}\right)^2 = x^2 \implies a - \sqrt{a + x} = x^2.$$

Rearrange to isolate the inner square root term:

$$(a - x^2)^2 = a + x \implies a^2 - 2ax^2 + (x^2)^2 = a + x \implies x^4 - 2ax^2 - x + (a^2 - a) = 0$$

...

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 \dots

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: \dots

...

Table 3 | An interesting “aha moment” of an intermediate version of DeepSeek-R1-Zero. The model learns to rethink using an anthropomorphic tone. This is also an aha moment for us, allowing us to witness the power and beauty of reinforcement learning. DeepSeek-R1-Zero 过渡版本的一个有趣的“顿悟时刻”。模型学会使用拟人化的语气重新思考。这对我们来说也是一个顿悟时刻，让我们见证了强化学习的力量和美丽。

Drawback of DeepSeek-R1-Zero 缺点 Although DeepSeek-R1-Zero exhibits strong reasoning capabilities and autonomously develops unexpected and powerful reasoning behaviors, it faces several issues. For instance, DeepSeek-R1-Zero struggles with challenges like poor readability, and language mixing. To make reasoning processes more readable and share them with the open community, we explore DeepSeek-R1, a method that utilizes RL with human-friendly cold-start data.

尽管 DeepSeek-R1-Zero 表现出强大的推理能力，并自主开发出意想不到的强大推理行为，但它仍面临一些问题。例如，DeepSeek-R1-Zero 面临着可读性差和语言混合等挑战。为了使推理过程更具可读性并与开放社区共享，我们探索了 DeepSeek-R1，这是一种利用强化学习和人性化冷启动数据的方法。

2.3. DeepSeek-R1: Reinforcement Learning with Cold Start

Inspired by the promising results of DeepSeek-R1-Zero, two natural questions arise: 1) Can reasoning performance be further improved or convergence accelerated by incorporating a small amount of high-quality data as a cold start? 2) How can we train a user-friendly model that not only produces clear and coherent Chains of Thought (CoT) but also demonstrates strong general capabilities? To address these questions, we design a pipeline to train DeepSeek-R1. The pipeline consists of four stages, outlined as follows.

受到 DeepSeek-R1-Zero 良好结果的启发，我们自然而然地产生了两个问题：1) 能否通过引入少量高质量数据作为冷启动来进一步提高推理性能或加速收敛？2) 我们如何训练一个用户友好的模型，该模型不仅可以产生清晰连贯的思维链 (CoT)，而且还具有强大的通用能力？为了解决这些问题，我们设计了一个流程来训练 DeepSeek-R1。该流程由四个阶段组成，概述如下。

2.3.1. Cold Start

Unlike DeepSeek-R1-Zero, to prevent the early unstable cold start phase of RL training from the base model, for DeepSeek-R1 we construct and collect a small amount of long CoT data to fine-tune the model as the initial RL actor. To collect such data, we have explored several approaches: using few-shot prompting with a long CoT as an example, directly prompting models to generate detailed answers with reflection and verification, gathering DeepSeek-R1-Zero outputs in a readable format, and refining the results through post-processing by human annotators.

与 DeepSeek-R1-Zero 不同，为了防止基础模型在 RL 训练早期出现不稳定的冷启动阶段，对于 DeepSeek-R1，我们构建并收集少量的长 CoT 数据，以作为初始 RL 参与者对模型进行微调。为了收集此类数据，我们探索了几种方法：以长 CoT 为例，使用少样本提示，直接提示模型通过反射和验证生成详细答案，以可读格式收集 DeepSeek-R1-Zero 输出，并通过人工注释者的后期处理来完善结果。

In this work, we collect thousands of cold-start data to fine-tune the DeepSeek-V3-Base as the starting point for RL. Compared to DeepSeek-R1-Zero, the advantages of cold start data include

在这项工作中，我们收集了数千个冷启动数据来微调 DeepSeek-V3-Base 作为 RL 的起点。与 DeepSeek-R1-Zero 相比，冷启动数据的优势包括：

- Readability: A key limitation of DeepSeek-R1-Zero is that its content is often not suitable for reading. Responses may mix multiple languages or lack markdown formatting to highlight answers for users. In contrast, when creating cold-start data for DeepSeek-R1, we design a readable pattern that includes a summary at the end of each response and filters out responses that are not reader-friendly. Here, we define the output format as `|special_token|<reasoning_process>|special_token|<summary>`, where the reasoning process is the CoT for the query, and the summary is used to summarize the reasoning results.

可读性：DeepSeek-R1-Zero 的一个关键限制是其内容通常不适合阅读。响应可能混合多种语言或缺乏 markdown 格式来为用户突出显示答案。相比之下，在为 DeepSeek-R1 创建冷启动数据时，我们设计了一种可读模式，在每个响应的末尾包含一个摘要，并过滤掉不易于阅读的响应。这里，我们将输出格式定义为

`|special_token|<reasoning_process>|special_token|<summary>`，其中推理过程是查询的 CoT，摘要用于总结推理结果。

- Potential: By carefully designing the pattern for cold-start data with human priors, we observe better performance against DeepSeek-R1-Zero. We believe the iterative training is a better way for reasoning models.

潜力：通过精心设计冷启动数据模式，结合人类先验知识，我们观察到其相对于 DeepSeek-R1-Zero 有更好的表现。我们相信迭代训练是推理模型的更好方法。

2.3.2. Reasoning-oriented Reinforcement Learning

After fine-tuning DeepSeek-V3-Base on the cold start data, we apply the same large-scale reinforcement learning training process as employed in DeepSeek-R1-Zero. This phase focuses on enhancing the model’s reasoning capabilities, particularly in reasoning-intensive tasks such as coding, mathematics, science, and logic reasoning, which involve well-defined problems with clear solutions. During the training process, we observe that CoT often exhibits language mixing, particularly when RL prompts involve multiple languages. To mitigate the issue of language mixing, we introduce a language consistency reward during RL training, which is calculated as the proportion of target language words in the CoT. Although ablation exper-

iments show that such alignment results in a slight degradation in the model’s performance, this reward aligns with human preferences, making it more readable. Finally, we combine the accuracy of reasoning tasks and the reward for language consistency by directly summing them to form the final reward. We then apply RL training on the fine-tuned model until it achieves convergence on reasoning tasks.

在冷启动数据上对 DeepSeek-V3-Base 进行微调后，我们采用与 DeepSeek-R1-Zero 相同的大规模强化学习训练流程。此阶段侧重于增强模型的推理能力，特别是在编码、数学、科学和逻辑推理等推理密集型任务中，这些任务涉及定义明确且解决方案明确的问题。在训练过程中，我们观察到 CoT 经常出现语言混合，尤其是当 RL 提示涉及多种语言时。为了缓解语言混合问题，我们在 RL 训练期间引入了语言一致性奖励，其计算方式为 CoT 中目标语言单词的比例。虽然消融实验表明这种对齐会导致模型性能略有下降，但这种奖励符合人类偏好，使其更具可读性。最后，我们将推理任务的准确性和语言一致性的奖励直接相加，形成最终奖励。然后，我们对微调后的模型进行 RL 训练，直到它在推理任务上实现收敛。

2.3.3. Rejection Sampling and Supervised Fine-Tuning **拒绝抽样和监督微调**

When reasoning-oriented RL converges, we utilize the resulting checkpoint to collect SFT (Supervised Fine-Tuning) data for the subsequent round. Unlike the initial cold-start data, which primarily focuses on reasoning, this stage incorporates data from other domains to enhance the model’s capabilities in writing, role-playing, and other general-purpose tasks. Specifically, we generate the data and fine-tune the model as described below.

当以推理为导向的强化学习收敛时，我们会利用生成的检查点来收集 SFT（监督微调）数据，以供下一轮使用。与主要侧重于推理的初始冷启动数据不同，此阶段会整合来自其他领域的数据，以增强模型在写作、角色扮演和其他通用任务中的能力。具体来说，我们会生成数据并微调模型，如下所述。

Reasoning data 推理数据 We curate reasoning prompts and generate reasoning trajectories by performing rejection sampling from the checkpoint from the above RL training. In the previous stage, we only included data that could be evaluated using rule-based rewards. However, in this stage, we expand the dataset by incorporating additional data, some of which use a generative reward model by feeding the ground-truth and model predictions into DeepSeek-V3 for judgment. Additionally, because the model output is sometimes chaotic and difficult to read, we have filtered out chain-of-thought with mixed languages, long paragraphs, and code blocks. For each prompt, we sample multiple responses and retain only the correct ones. In total, we collect about 600k reasoning related training samples.

我们通过从上述 RL 训练的检查点执行拒绝采样来整理推理提示并生成推理轨迹。在上一阶段，我们仅包含可以使用基于规则的奖励进行评估的数据。但是，在此阶段，我们通过合并其他数据来扩展数据集，其中一些数据使用生成奖励模型，将基本事实和模型预测输入 DeepSeek-V3 进行判断。此外，由于模型输出有时混乱且难以阅读，我们过滤掉了混合语言、长段落和代码块思路链。对于每个提示，我们都会采样多个响应并仅保留正确的响应。总共，我们收集了大约 600k 个与推理相关的训练样本。

Non-Reasoning data 非推理数据 For non-reasoning data, such as writing, factual QA, self-cognition, and translation, we adopt the DeepSeek-V3 pipeline and reuse portions of the SFT dataset of DeepSeek-V3. For certain non-reasoning tasks, we call DeepSeek-V3 to generate a potential chain-of-thought before answering the question by prompting. However, for simpler queries, such as “hello” we do not provide a CoT in response. In the end, we collected a total of approximately 200k training samples that are unrelated to reasoning.

对于非推理数据，例如写作、事实问答、自我认知和翻译，我们采用 DeepSeek-V3 流程并重用 DeepSeek-V3 的部分 SFT 数据集。对于某些非推理任务，我们调用 DeepSeek-V3 生成潜在的思路链，然后再通过提示回答问题。但是，对于更简单的查询，例如“你好”，我们不提供 CoT 作为响应。最终，我们总共收集了大约 20 万个与推理无关的训练样本。

We fine-tune DeepSeek-V3-Base for two epochs using the above curated dataset of about 800k samples.

我们使用上面整理的约 800k 个样本的数据集对 DeepSeek-V3-Base 进行了两个时期的微调。

2.3.4. Reinforcement Learning for all Scenarios 适用于所有场景的强化学习

To further align the model with human preferences, we implement a secondary reinforcement learning stage aimed at improving the model’s helpfulness and harmlessness while simultaneously refining its reasoning capabilities. Specifically, we train the model using a combination of reward signals and diverse prompt distributions. For reasoning data, we adhere to the methodology outlined in DeepSeek-R1-Zero, which utilizes rule-based rewards to guide the learning process in math, code, and logical reasoning domains.

为了进一步使模型与人类偏好保持一致，我们实施了二级强化学习阶段，旨在提高模型的有用性和无害性，同时改进其推理能力。具体来说，我们使用奖励信号和各种提示分布的组合来训练模型。对于推理数据，我们遵循 DeepSeek-R1-Zero 中概述的方法，该方法利用基于规则的奖励来指导数学、代码和逻辑推理领域的学习过程。

For general data, we resort to reward models to capture human preferences in complex and nuanced scenarios. We build upon the DeepSeek-V3 pipeline and adopt a similar distribution of preference pairs and training prompts. For helpfulness, we focus exclusively on the final summary, ensuring that the assessment emphasizes the utility and relevance of the response to the user while minimizing interference with the underlying reasoning process. For harmlessness, we evaluate the entire response of the model, including both the reasoning process and the summary, to identify and mitigate any potential risks, biases, or harmful content that may arise during the generation process. Ultimately, the integration of reward signals and diverse data distributions enables us to train a model that excels in reasoning while prioritizing helpfulness and harmlessness.

对于一般数据，我们采用奖励模型来捕捉复杂而微妙的场景中的人类偏好。我们以 DeepSeek-V3 管道为基础，采用类似的偏好对和训练提示分布。对于有用性，我们只关注最终的总结，确保评估强调响应对用户的实用性和相关性，同时最大限度地减少对底层推理过程的干扰。对于无害性，我们评估模型的整个响应，包括推理过程和总结，以识别和减轻生成过程中可能出现的任何潜在风险、偏见或有害内容。最终，奖励信号和多样化数据分布的整合使我们能够训练一个擅长推理同时优先考虑有用性和无害性的模型。

2.4. Distillation: Empower Small Models with Reasoning Capability

To equip more efficient smaller models with reasoning capabilities like DeepSeek-R1, we directly fine-tuned open-source models like Qwen (26) and Llama (1) using the 800k samples curated with DeepSeek-R1, as detailed in §2.3.3. Our findings indicate that this straightforward distillation method significantly enhances the reasoning abilities of smaller models. The base models we use here are Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Qwen2.5-14B, Qwen2.5-32B, Llama-3.1-8B, and Llama-3.3-70B-Instruct. We select Llama-3.3 because its reasoning capability is slightly better than that of Llama-3.1.

为了使更高效的小型模型具备 DeepSeek-R1 等推理能力，我们使用 DeepSeek-R1 整理的 80 万个样本直接对 Qwen (26) 和 Llama (1) 等开源模型进行了微调，详情请参阅 §2.3.3。

我们的研究表明，这种简单的蒸馏方法显著增强了小型模型的推理能力。我们在这里使用的基础模型是 Qwen2.5-Math-1.5B、Qwen2.5-Math-7B、Qwen2.5-14B、Qwen2.5-32B、Llama-3.1-8B 和 Llama-3.3-70B-Instruct。我们选择 Llama-3.3 是因为它的推理能力略优于 Llama-3.1。

For distilled models, we apply only SFT and do not include an RL stage, even though incorporating RL could substantially boost model performance. Our primary goal here is to demonstrate the effectiveness of the distillation technique, leaving the exploration of the RL stage to the broader research community.

对于蒸馏模型，我们仅应用 SFT，而不包含 RL 阶段，尽管加入 RL 可以显著提高模型性能。我们的主要目标是证明蒸馏技术的有效性，而将 RL 阶段的探索留给更广泛的研究社区。

3. Experiment

Benchmarks We evaluate models on MMLU (11), MMLU-Redux (8), MMLU-Pro (36), C-Eval (12), and CMMLU (16), IFEval (39), FRAMES (14), GPQA Diamond (27), SimpleQA (23), C-SimpleQA (10), SWE-Bench Verified (24), Aider ¹, LiveCodeBench (13) (2024-08 – 2025-01), Codeforces ², Chinese National High School Mathematics Olympiad (CNMO 2024)³, and American Invitational Mathematics Examination 2024 (AIME 2024) (20). In addition to standard benchmarks, we also evaluate our models on open-ended generation tasks using LLMs as judges. Specifically, we adhere to the original configurations of AlpacaEval 2.0 (5) and Arena-Hard (17), which leverage GPT-4-Turbo-1106 as judges for pairwise comparisons. Here, we only feed the final summary to evaluation to avoid the length bias. For distilled models, we report representative results on AIME 2024, MATH-500, GPQA Diamond, Codeforces, and LiveCodeBench.

我们在 MMLU (11)、MMLU-Redux (8)、MMLU-Pro (36)、C-Eval (12) 和 CMMLU (16) 上评估模型，IFEval (39), FRAMES (?), GPQA 钻石 (27)、SimpleQA (23)、C-SimpleQA (10)、SWE-Bench Verified (24)、Aider ⁴, LiveCodeBench (13) (2024-08 – 2025-01), Codeforces ⁵, 中国全国高中数学奥林匹克 (CNMO 2024)⁶ 和 2024 年美国数学邀请赛 (AIME 2024) (20)。除了标准基准之外，我们还使用 LLM 作为评判者，在开放式生成任务上评估我们的模型。具体来说，我们遵循 AlpacaEval 2.0 (5) 和 Arena-Hard (17) 的原始配置，它们利用 GPT-4-Turbo-1106 作为成对比较的评判者。在这里，我们只提供最后总结以进行评估，避免篇幅偏差。对于蒸馏模型，我们报告了 AIME 2024、MATH-500、GPQA Diamond、Codeforces 和 LiveCodeBench 上的代表性结果。

¹<https://aider.chat>

²<https://codeforces.com>

³<https://www.cms.org.cn/Home/comp/comp/cid/12.html>

⁴<https://aider.>

⁵<https://codeforces.com>

⁶<https://www.cms.org.cn/Home/comp/comp/cid/12.html>

Evaluation Prompts 评估提示 Following the setup in DeepSeek-V3, standard benchmarks such as MMLU, DROP, GPQA Diamond, and SimpleQA are evaluated using prompts from the simple-evals framework. For MMLU-Redux, we adopt the Zero-Eval prompt format (19) in a zero-shot setting. In terms of MMLU-Pro, C-Eval and CLUE-WSC, since the original prompts are few-shot, we slightly modify the prompt to the zero-shot setting. The CoT in few-shot may hurt the performance of DeepSeek-R1. Other datasets follow their original evaluation protocols with default prompts provided by their creators. For code and math benchmarks, the HumanEval-Mul dataset covers eight mainstream programming languages (Python, Java, C++, C#, JavaScript, TypeScript, PHP, and Bash). Model performance on LiveCodeBench is evaluated using CoT format, with data collected between August 2024 and January 2025. The Codeforces dataset is evaluated using problems from 10 Div.2 contests along with expert-crafted test cases, after which the expected ratings and percentages of competitors are calculated. SWE-Bench verified results are obtained via the agentless framework (37). AIDER-related benchmarks are measured using a "diff" format. DeepSeek-R1 outputs are capped at a maximum of 32,768 tokens for each benchmark.

按照 DeepSeek-V3 中的设置, 使用 simple-evals 框架中的提示评估 MMLU、DROP、GPQA Diamond 和 SimpleQA 等标准基准。对于 MMLU-Redux, 我们在零样本设置中采用 Zero-Eval 提示格式 (19)。对于 MMLU-Pro、C-Eval 和 CLUE-WSC, 由于原始提示是少样本的, 我们将提示稍微修改为零样本设置。少样本中的 CoT 可能会损害 DeepSeek-R1 的性能。其他数据集遵循其原始评估协议, 使用其创建者提供的默认提示。对于代码和数学基准, HumanEval-Mul 数据集涵盖了八种主流编程语言 (Python、Java、C++、C#、JavaScript、TypeScript、PHP 和 Bash)。使用 CoT 格式评估 LiveCodeBench 上的模型性能, 数据收集时间为 2024 年 8 月至 2025 年 1 月。使用来自 10 个 Div.2 竞赛的问题以及专家制作的测试用例评估 Codeforces 数据集, 然后计算对手的预期评分和百分比。通过无代理框架 (37) 获得 SWE-Bench 验证的结果。使用 "diff" 格式测量与 AIDER 相关的基准。DeepSeek-R1 输出每基准的上限为 32,768 个 tokens。

Baselines We conduct comprehensive evaluations against several strong baselines, including DeepSeek-V3, Claude-Sonnet-3.5-1022, GPT-4o-0513, OpenAI-o1-mini, and OpenAI-o1-1217. Since accessing the OpenAI-o1-1217 API is challenging in mainland China, we report its performance based on official reports. For distilled models, we also compare the open-source model QwQ-32B-Preview (25).

我们针对几个强大的基线进行了全面的评估, 包括 DeepSeek-V3、Claude-Sonnet-3.5-1022、GPT-4o-0513、OpenAI-o1-mini 和 OpenAI-o1-1217。由于在中国大陆访问 OpenAI-o1-1217 API 具有挑战性, 因此我们根据官方报告报告其性能。对于蒸馏模型, 我们还比较了开源模型 QwQ-32B-Preview (25)。

Evaluation Setup 评估设置 We set the maximum generation length to 32,768 tokens for the models. We found that using greedy decoding to evaluate long-output reasoning models results in higher repetition rates and significant variability across different checkpoints. Therefore, we default to pass@k evaluation (3) and report pass@1 using a non-zero temperature. Specifically, we use a sampling temperature of 0.6 and a top-p value of 0.95 to generate k responses (typically between 4 and 64, depending on the test set size) for each question. Pass@1 is then calculated as 我们将模型的最大生成长度设置为 32,768 个 tokens。我们发现使用贪婪解码来评估长输出推理模型会导致更高的重复率和不同检查点之间的显著差异。因此, 我们默认使用非零温度进行 pass@k 评估 (3) 并报告 pass@1。具体来说, 我们使用 0.6 的采样温度和 0.95 的 top-p 值来为每个问题生成 k 个响应 (通常在 4 到 64 之间, 具体取决于测试集大小)。然后

计算 Pass@1 为

$$\text{pass@1} = \frac{1}{k} \sum_{i=1}^k p_i,$$

where p_i denotes the correctness of the i -th response. This method provides more reliable performance estimates. For AIME 2024, we also report consensus (majority vote) results (35) using 64 samples, denoted as cons@64.

其中 p_i 表示第 i 个响应的正确性。此方法提供了更可靠的性能估计。对于 AIME 2024, 我们还使用 64 个样本报告了共识 (多数投票) 结果 (35), 表示为 cons@64。

3.1. DeepSeek-R1 Evaluation

For education-oriented knowledge benchmarks such as MMLU, MMLU-Pro, and GPQA Diamond, DeepSeek-R1 demonstrates superior performance compared to DeepSeek-V3. This improvement is primarily attributed to enhanced accuracy in STEM-related questions, where significant gains are achieved through large-scale reinforcement learning. Additionally, DeepSeek-R1 excels on FRAMES, a long-context-dependent QA task, showcasing its strong document analysis capabilities. This highlights the potential of reasoning models in AI-driven search and data analysis tasks. On the factual benchmark SimpleQA, DeepSeek-R1 outperforms DeepSeek-V3, demonstrating its capability in handling fact-based queries. A similar trend is observed where OpenAI-o1 surpasses GPT-4o on this benchmark. However, DeepSeek-R1 performs worse than DeepSeek-V3 on the Chinese SimpleQA benchmark, primarily due to its tendency to refuse answering certain queries after safety RL. Without safety RL, DeepSeek-R1 could achieve an accuracy of over 70%.

对于面向教育的知识基准 (例如 MMLU、MMLU-Pro 和 GPQA Diamond), DeepSeek-R1 表现出比 DeepSeek-V3 更好的性能。这种改进主要归因于 STEM 相关问题的准确性提高, 通过大规模强化学习实现了显著的提升。此外, DeepSeek-R1 在 FRAMES (一项长期上下文相关的 QA 任务) 上表现出色, 展示了其强大的文档分析能力。这凸显了推理模型在 AI 驱动的搜索和数据分析任务中的潜力。在事实基准 SimpleQA 上, DeepSeek-R1 的表现优于 DeepSeek-V3, 展示了其处理基于事实的查询的能力。在该基准上, OpenAI-o1 超越 GPT-4o 也呈现出类似的趋势。然而, DeepSeek-R1 在中国 SimpleQA 基准上的表现不如 DeepSeek-V3, 主要是因为它在安全强化学习后倾向于拒绝回答某些查询。即使没有安全 RL, DeepSeek-R1 也能达到 70% 以上的准确率。

DeepSeek-R1 also delivers impressive results on IF-Eval, a benchmark designed to assess a model’s ability to follow format instructions. These improvements can be linked to the inclusion of instruction-following data during the final stages of supervised fine-tuning (SFT) and RL training. Furthermore, remarkable performance is observed on AlpacaEval2.0 and ArenaHard, indicating DeepSeek-R1’s strengths in writing tasks and open-domain question answering. Its significant outperformance of DeepSeek-V3 underscores the generalization benefits of large-scale RL, which not only boosts reasoning capabilities but also improves performance across diverse domains. Moreover, the summary lengths generated by DeepSeek-R1 are concise, with an average of 689 tokens on ArenaHard and 2,218 characters on AlpacaEval 2.0. This indicates that DeepSeek-R1 avoids introducing length bias during GPT-based evaluations, further solidifying its robustness across multiple tasks.

DeepSeek-R1 还在 IF-Eval 上取得了令人印象深刻的结果, IF-Eval 是一个旨在评估模型遵循格式指令的能力的基准测试。这些改进可以与在监督微调 (SFT) 和 RL 训练的最后阶段纳入遵循指令的数据联系起来。此外, 在 AlpacaEval2.0 和 ArenaHard 上也观察到了出色的表现, 表明 DeepSeek-R1 在写作任务和开放域问答方面的优势。它显著优于 DeepSeek-V3, 凸显了大规模 RL 的泛化优势, 这不仅提高了推理能力, 还提高了跨不同领域的性能。此外,

DeepSeek-R1 生成的摘要长度简洁，在 ArenaHard 上平均为 689 个 tokens，在 AlpacaEval 2.0 上平均为 2,218 个字符。这表明 DeepSeek-R1 避免在基于 GPT 的评估期间引入长度偏差，进一步巩固了其在多个任务中的稳健性。

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	CLUWSC (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

Table 4 | Comparison between DeepSeek-R1 and other representative models.

On math tasks, DeepSeek-R1 demonstrates performance on par with OpenAI-o1-1217, surpassing other models by a large margin. A similar trend is observed on coding algorithm tasks, such as LiveCodeBench and Codeforces, where reasoning-focused models dominate these benchmarks. On engineering-oriented coding tasks, OpenAI-o1-1217 outperforms DeepSeek-R1 on Aider but achieves comparable performance on SWE Verified. We believe the engineering performance of DeepSeek-R1 will improve in the next version, as the amount of related RL training data currently remains very limited.

在数学任务上，DeepSeek-R1 的表现与 OpenAI-o1-1217 相当，远远超过其他模型。在编码算法任务（例如 LiveCodeBench 和 Codeforces）上也观察到了类似的趋势，在这些基准测试中，以推理为重点的模型占据主导地位。在面向工程的编码任务上，OpenAI-o1-1217 在 Aider 上的表现优于 DeepSeek-R1，但在 SWE Verified 上实现了相当的性能。我们相信 DeepSeek-R1 的工程性能将在下一版本中得到改善，因为相关的 RL 训练数据量目前仍然非常有限。

3.2. Distilled Model Evaluation

As shown in Table 5, simply distilling DeepSeek-R1’s outputs enables the efficient DeepSeek-R1-7B (i.e., DeepSeek-R1-Distill-Qwen-7B, abbreviated similarly below) to outperform non-reasoning models like GPT-4o-0513 across the board. DeepSeek-R1-14B surpasses QwQ-32B-Preview on all evaluation metrics, while DeepSeek-R1-32B and DeepSeek-R1-70B significantly

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

Table 5 | Comparison of DeepSeek-R1 distilled models and other comparable models on reasoning-related benchmarks. DeepSeek-R1 蒸馏模型与其他类似模型在推理相关基准上的比较。

exceed o1-mini on most benchmarks. These results demonstrate the strong potential of distillation. Additionally, we found that applying RL to these distilled models yields significant further gains. We believe this warrants further exploration and therefore present only the results of the simple SFT-distilled models here.

如表 5 所示, 只需蒸馏 DeepSeek-R1 的输出, 高效的 DeepSeek-R1-7B (即 DeepSeek-R1-Distill-Qwen-7B, 下文缩写类似) 就能全面超越 GPT-4o-0513 等非推理模型。DeepSeek-R1-14B 在所有评估指标上都超过了 QwQ-32B-Preview, 而 DeepSeek-R1-32B 和 DeepSeek-R1-70B 在大多数基准测试中都大大超过了 o1-mini。这些结果证明了蒸馏的巨大潜力。此外, 我们发现将 RL 应用于这些蒸馏模型可以获得显著的进一步收益。我们认为这值得进一步探索, 因此这里仅展示简单的 SFT 蒸馏模型的结果。

4. Discussion

4.1. Distillation v.s. Reinforcement Learning

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

Table 6 | Comparison of distilled and RL Models on Reasoning-Related Benchmarks.

In Section 3.2, we can see that by distilling DeepSeek-R1, the small model can achieve impressive results. However, there is still one question left: can the model achieve comparable performance through the large-scale RL training discussed in the paper without distillation? 在 3.2 部分, 我们可以看到通过蒸馏 DeepSeek-R1, 小模型可以取得令人印象深刻的效果。然而, 还有一个问题: 如果不进行蒸馏, 模型是否可以通过论文中讨论的大规模 RL 训练获得相当的性能?

To answer this question, we conduct large-scale RL training on Qwen-32B-Base using math, code, and STEM data, training for over 10K steps, resulting in DeepSeek-R1-Zero-Qwen-32B. The experimental results, shown in Table 6, demonstrate that the 32B base model, after large-scale RL training, achieves performance on par with QwQ-32B-Preview. However, DeepSeek-R1-Distill-Qwen-32B, which is distilled from DeepSeek-R1, performs significantly better than

DeepSeek-R1-Zero-Qwen-32B across all benchmarks.

为了回答这个问题，我们使用数学、代码和 STEM 数据对 Qwen-32B-Base 进行了大规模 RL 训练，训练了超过 10K 步，最终得到了 DeepSeek-R1-Zero-Qwen-32B。实验结果（如表 6 所示）表明，经过大规模 RL 训练后，32B 基础模型的性能与 QwQ-32B-Preview 相当。然而，从 DeepSeek-R1 蒸馏出来的 DeepSeek-R1-Distill-Qwen-32B 在所有基准测试中的表现都明显优于 DeepSeek-R1-Zero-Qwen-32B。

Therefore, we can draw two conclusions: First, distilling more powerful models into smaller ones yields excellent results, whereas smaller models relying on the large-scale RL mentioned in this paper require enormous computational power and may not even achieve the performance of distillation. Second, while distillation strategies are both economical and effective, advancing beyond the boundaries of intelligence may still require more powerful base models and larger-scale reinforcement learning.

因此，我们可以得出两个结论：第一，将更强大的模型蒸馏成更小的模型可以产生很好的效果，而本文提到的依赖于大规模强化学习的小模型需要巨大的计算能力，甚至可能无法达到蒸馏的性能。第二，虽然蒸馏策略既经济又有效，但要超越智能的界限，可能仍需要更强大的基础模型和更大规模的强化学习。

4.2. Unsuccessful Attempts

In the early stages of developing DeepSeek-R1, we also encountered failures and setbacks along the way. We share our failure experiences here to provide insights, but this does not imply that these approaches are incapable of developing effective reasoning models.

在开发 DeepSeek-R1 的早期阶段，我们也曾遇到过失败和挫折。我们在这里分享失败的经验，以提供见解，但这并不意味着这些方法无法开发有效的推理模型。

Process Reward Model (PRM) PRM is a reasonable method to guide the model toward better approaches for solving reasoning tasks (18; 33; 34). 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 (7), 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 (31), its advantages are limited compared to the additional computational overhead it introduces during the large-scale reinforcement learning process in our experiments.

PRM 是一种合理的方法，可以引导模型朝着更好的方法解决推理任务 (18; 33; 34)。然而，在实践中，PRM 有三个主要限制，可能会阻碍其最终成功。首先，在一般推理中明确定义一个细粒度的步骤具有挑战性。其次，确定当前中间步骤是否正确是一项具有挑战性的任务。使用模型进行自动注释可能无法产生令人满意的结果，而手动注释不利于扩大规模。第三，一旦引入基于模型的 PRM，不可避免地会导致奖励黑客攻击 (7)，重新训练奖励模型需要额外的训练资源，并使整个训练流程变得复杂。综上所述，虽然 PRM 表现出了对模型生成的前 N 个响应进行重新排序或协助引导搜索的良好能力 (31)，但与我们实验中在大规模强化学习过程中引入的额外计算开销相比，它的优势是有限的。

Monte Carlo Tree Search (MCTS) 蒙特卡洛树搜索 (MCTS) Inspired by AlphaGo (30) and AlphaZero (29), we explored using Monte Carlo Tree Search (MCTS) to enhance test-time

compute scalability. This approach involves breaking answers into smaller parts to allow the model to explore the solution space systematically. To facilitate this, we prompt the model to generate multiple tags that correspond to specific reasoning steps necessary for the search. For training, we first use collected prompts to find answers via MCTS guided by a pre-trained value model. Subsequently, we use the resulting question-answer pairs to train both the actor model and the value model, iteratively refining the process.

受到 AlphaGo (30) 和 AlphaZero (29) 的启发，我们探索使用蒙特卡洛树搜索 (MCTS) 来增强测试时计算可扩展性。这种方法涉及将答案分解为更小的部分，以允许模型系统地探索解决方案空间。为了实现这一点，我们提示模型生成多个标签，这些标签对应于搜索所需的特定推理步骤。对于训练，我们首先使用收集到的提示通过由预先训练的价值模型指导的 MCTS 来寻找答案。随后，我们使用生成的问答对来训练参与者模型和价值模型，并迭代地完善该过程。

However, this approach encounters several challenges when scaling up the training. First, unlike chess, where the search space is relatively well-defined, token generation presents an exponentially larger search space. To address this, we set a maximum extension limit for each node, but this can lead to the model getting stuck in local optima. Second, the value model directly influences the quality of generation since it guides each step of the search process. Training a fine-grained value model is inherently difficult, which makes it challenging for the model to iteratively improve. While AlphaGo’s core success relied on training a value model to progressively enhance its performance, this principle proves difficult to replicate in our setup due to the complexities of token generation.

然而，这种方法在扩大训练规模时会遇到一些挑战。首先，与国际象棋不同，国际象棋的搜索空间相对明确，而 token 生成则呈现出指数级的搜索空间。为了解决这个问题，我们为每个节点设置了最大扩展限制，但这可能会导致模型陷入局部最优。其次，价值模型直接影响生成的质量，因为它指导搜索过程的每一步。训练细粒度的价值模型本质上很困难，这使得模型的迭代改进具有挑战性。虽然 AlphaGo 的核心成功依赖于训练价值模型来逐步提高其性能，但由于 token 生成的复杂性，这一原则在我们的设置中很难复制。

In conclusion, while MCTS can improve performance during inference when paired with a pre-trained value model, iteratively boosting model performance through self-search remains a significant challenge.

总之，虽然 MCTS 与预先训练的价值模型结合使用时可以提高推理过程中的性能，但通过自我搜索迭代地提升模型性能仍然是一个重大挑战。

5. Conclusion, Limitations, and Future Work

In this work, we share our journey in enhancing model reasoning abilities through reinforcement learning. DeepSeek-R1-Zero represents a pure RL approach without relying on cold-start data, achieving strong performance across various tasks. DeepSeek-R1 is more powerful, leveraging cold-start data alongside iterative RL fine-tuning. Ultimately, DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on a range of tasks.

在这项工作中，我们分享了通过强化学习增强模型推理能力的历程。DeepSeek-R1-Zero 代表一种不依赖冷启动数据的纯 RL 方法，在各种任务中都取得了出色的性能。DeepSeek-R1 功能更强大，它利用冷启动数据和迭代 RL 微调。最终，DeepSeek-R1 在一系列任务上实现了与 OpenAI-o1-1217 相当的性能。

We further explore distillation the reasoning capability to small dense models. We use DeepSeek-R1 as the teacher model to generate 800K training samples, and fine-tune several small dense models. The results are promising: DeepSeek-R1-Distill-Qwen-1.5B outperforms GPT-4o and Claude-3.5-Sonnet on math benchmarks with 28.9% on AIME and 83.9% on MATH. Other dense models also achieve impressive results, significantly outperforming other instruction-tuned models based on the same underlying checkpoints.

我们进一步探索将推理能力蒸馏到小型密集模型。我们使用 DeepSeek-R1 作为教师模型来生成 800K 个训练样本，并对几个小型密集模型进行微调。结果令人鼓舞：DeepSeek-R1-Distill-Qwen-1.5B 在数学基准测试中的表现优于 GPT-4o 和 Claude-3.5-Sonnet，在 AIME 上为 28.9%，在 MATH 上为 83.9%。其他密集模型也取得了令人印象深刻的结果，显著优于基于相同底层检查点的其他指令调整模型。

In the future, we plan to invest in research across the following directions for DeepSeek-R1. 未来我们计划对 DeepSeek-R1 以下方向的研究进行投入。

- General Capability: Currently, the capabilities of DeepSeek-R1 fall short of DeepSeek-V3 in tasks such as function calling, multi-turn, complex role-playing, and JSON output. Moving forward, we plan to explore how long CoT can be leveraged to enhance tasks in these fields.

目前，DeepSeek-R1 在函数调用、多轮、复杂角色扮演和 JSON 输出等任务上的能力还不及 DeepSeek-V3。展望未来，我们计划探索 CoT 可以发挥多大作用来增强这些领域的任务。

- Language Mixing: DeepSeek-R1 is currently optimized for Chinese and English, which may result in language mixing issues when handling queries in other languages. For instance, DeepSeek-R1 might use English for reasoning and responses, even if the query is in a language other than English or Chinese. We aim to address this limitation in future updates.

DeepSeek-R1 目前针对中文和英文进行了优化，这可能会导致在处理其他语言的查询时出现语言混合问题。例如，即使查询使用的语言不是英文或中文，DeepSeek-R1 也可能使用英文进行推理和响应。我们计划在未来的更新中解决这一限制。

- Prompting Engineering: When evaluating DeepSeek-R1, we observe that it is sensitive to prompts. Few-shot prompting consistently degrades its performance. Therefore, we recommend users directly describe the problem and specify the output format using a zero-shot setting for optimal results.

在评估 DeepSeek-R1 时，我们发现它对提示很敏感。少量提示会持续降低其性能。因此，我们建议用户直接描述问题并使用零样本设置指定输出格式以获得最佳结果。

- Software Engineering Tasks: Due to the long evaluation times, which impact the efficiency of the RL process, large-scale RL has not been applied extensively in software engineering tasks. As a result, DeepSeek-R1 has not demonstrated a huge improvement over DeepSeek-V3 on software engineering benchmarks. Future versions will address this by implementing rejection sampling on software engineering data or incorporating asynchronous evaluations during the RL process to improve efficiency.

由于评估时间较长，影响了强化学习过程的效率，大规模强化学习尚未广泛应用于软件工程任务。因此，DeepSeek-R1 在软件工程基准测试中并未表现出比 DeepSeek-V3 更大的改进。未来版本将通过对软件工程数据实施拒绝采样或在强化学习过程中加入异步评估来解决这个问题，以提高效率。

References

- [1] AI@Meta. Llama 3.1 model card, 2024. URL https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md.
- [2] Anthropic. Claude 3.5 sonnet, 2024. URL <https://www.anthropic.com/news/claude-3-5-sonnet>.
- [3] M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating large language models trained on code. *CoRR*, abs/2107.03374, 2021. URL <https://arxiv.org/abs/2107.03374>.
- [4] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.
- [5] Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto. Length-controlled alpaca-eval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*, 2024.
- [6] X. Feng, Z. Wan, M. Wen, S. M. McAleer, Y. Wen, W. Zhang, and J. Wang. Alphazero-like tree-search can guide large language model decoding and training, 2024. URL <https://arxiv.org/abs/2309.17179>.
- [7] L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization, 2022. URL <https://arxiv.org/abs/2210.10760>.
- [8] A. P. Gema, J. O. J. Leang, G. Hong, A. Devoto, A. C. M. Mancino, R. Saxena, X. He, Y. Zhao, X. Du, M. R. G. Madani, C. Barale, R. McHardy, J. Harris, J. Kaddour, E. van Krieken, and P. Minervini. Are we done with mmlu? *CoRR*, abs/2406.04127, 2024. URL <https://doi.org/10.48550/arXiv.2406.04127>.
- [9] Google. Our next-generation model: Gemini 1.5, 2024. URL <https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024>.
- [10] Y. He, S. Li, J. Liu, Y. Tan, W. Wang, H. Huang, X. Bu, H. Guo, C. Hu, B. Zheng, et al. Chinese simpleqa: A chinese factuality evaluation for large language models. *arXiv preprint arXiv:2411.07140*, 2024.
- [11] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*, 2020.
- [12] Y. Huang, Y. Bai, Z. Zhu, J. Zhang, J. Zhang, T. Su, J. Liu, C. Lv, Y. Zhang, J. Lei, et al. C-Eval: A multi-level multi-discipline chinese evaluation suite for foundation models. *arXiv preprint arXiv:2305.08322*, 2023.

- [13] N. Jain, K. Han, A. Gu, W. Li, F. Yan, T. Zhang, S. Wang, A. Solar-Lezama, K. Sen, and I. Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. CoRR, abs/2403.07974, 2024. URL <https://doi.org/10.48550/arXiv.2403.07974>.
- [14] S. Krishna, K. Krishna, A. Mohananey, S. Schwarcz, A. Stambler, S. Upadhyay, and M. Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. CoRR, abs/2409.12941, 2024. doi: 10.48550/ARXIV.2409.12941. URL <https://doi.org/10.48550/arXiv.2409.12941>.
- [15] A. Kumar, V. Zhuang, R. Agarwal, Y. Su, J. D. Co-Reyes, A. Singh, K. Baumli, S. Iqbal, C. Bishop, R. Roelofs, et al. Training language models to self-correct via reinforcement learning. arXiv preprint arXiv:2409.12917, 2024.
- [16] H. Li, Y. Zhang, F. Koto, Y. Yang, H. Zhao, Y. Gong, N. Duan, and T. Baldwin. CMMLU: Measuring massive multitask language understanding in Chinese. arXiv preprint arXiv:2306.09212, 2023.
- [17] T. Li, W.-L. Chiang, E. Frick, L. Dunlap, T. Wu, B. Zhu, J. E. Gonzalez, and I. Stoica. From crowdsourced data to high-quality benchmarks: Arena-hard and benchbuilder pipeline. arXiv preprint arXiv:2406.11939, 2024.
- [18] H. Lightman, V. Kosaraju, Y. Burda, H. Edwards, B. Baker, T. Lee, J. Leike, J. Schulman, I. Sutskever, and K. Cobbe. Let’s verify step by step. arXiv preprint arXiv:2305.20050, 2023.
- [19] B. Y. Lin. ZeroEval: A Unified Framework for Evaluating Language Models, July 2024. URL <https://github.com/WildEval/ZeroEval>.
- [20] MAA. American invitational mathematics examination - aime. In American Invitational Mathematics Examination - AIME 2024, February 2024. URL <https://maa.org/math-competitions/american-invitational-mathematics-examination-aime>.
- [21] OpenAI. Hello GPT-4o, 2024. URL <https://openai.com/index/hello-gpt-4o/>.
- [22] OpenAI. Learning to reason with llms, 2024. URL <https://openai.com/index/learning-to-reason-with-llms/>.
- [23] OpenAI. Introducing SimpleQA, 2024. URL <https://openai.com/index/introducing-simpleqa/>.
- [24] OpenAI. Introducing SWE-bench verified we’re releasing a human-validated subset of swe-bench that more, 2024. URL <https://openai.com/index/introducing-swe-bench-verified/>.
- [25] Qwen. Qwq: Reflect deeply on the boundaries of the unknown, 2024. URL <https://qwenlm.github.io/blog/qwq-32b-preview/>.
- [26] Qwen. Qwen2.5: A party of foundation models, 2024. URL <https://qwenlm.github.io/blog/qwen2.5>.
- [27] D. Rein, B. L. Hou, A. C. Stickland, J. Petty, R. Y. Pang, J. Dirani, J. Michael, and S. R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. arXiv preprint arXiv:2311.12022, 2023.

- [28] Z. Shao, P. Wang, Q. Zhu, R. Xu, J. Song, M. Zhang, Y. Li, Y. Wu, and D. Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. arXiv preprint arXiv:2402.03300, 2024.
- [29] D. Silver, T. Hubert, J. Schrittwieser, I. Antonoglou, M. Lai, A. Guez, M. Lanctot, L. Sifre, D. Kumaran, T. Graepel, T. P. Lillicrap, K. Simonyan, and D. Hassabis. Mastering chess and shogi by self-play with a general reinforcement learning algorithm. CoRR, abs/1712.01815, 2017. URL <http://arxiv.org/abs/1712.01815>.
- [30] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. P. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. Mastering the game of go without human knowledge. Nat., 550(7676):354–359, 2017. doi: 10.1038/NATURE24270. URL <https://doi.org/10.1038/nature24270>.
- [31] C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL <https://arxiv.org/abs/2408.03314>.
- [32] T. Trinh, Y. Wu, Q. Le, H. He, and T. Luong. Solving olympiad geometry without human demonstrations. Nature, 2024. doi: 10.1038/s41586-023-06747-5.
- [33] J. Uesato, N. Kushman, R. Kumar, F. Song, N. Siegel, L. Wang, A. Creswell, G. Irving, and I. Higgins. Solving math word problems with process-and outcome-based feedback. arXiv preprint arXiv:2211.14275, 2022.
- [34] P. Wang, L. Li, Z. Shao, R. Xu, D. Dai, Y. Li, D. Chen, Y. Wu, and Z. Sui. Math-shepherd: A label-free step-by-step verifier for llms in mathematical reasoning. arXiv preprint arXiv:2312.08935, 2023.
- [35] X. Wang, J. Wei, D. Schuurmans, Q. Le, E. Chi, S. Narang, A. Chowdhery, and D. Zhou. Self-consistency improves chain of thought reasoning in language models. arXiv preprint arXiv:2203.11171, 2022.
- [36] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, T. Li, M. Ku, K. Wang, A. Zhuang, R. Fan, X. Yue, and W. Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. CoRR, abs/2406.01574, 2024. URL <https://doi.org/10.48550/arXiv.2406.01574>.
- [37] C. S. Xia, Y. Deng, S. Dunn, and L. Zhang. Agentless: Demystifying llm-based software engineering agents. arXiv preprint, 2024.
- [38] H. Xin, Z. Z. Ren, J. Song, Z. Shao, W. Zhao, H. Wang, B. Liu, L. Zhang, X. Lu, Q. Du, W. Gao, Q. Zhu, D. Yang, Z. Gou, Z. F. Wu, F. Luo, and C. Ruan. Deepseek-prover-v1.5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search, 2024. URL <https://arxiv.org/abs/2408.08152>.
- [39] J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, and L. Hou. Instruction-following evaluation for large language models. arXiv preprint arXiv:2311.07911, 2023.

Appendix

A. Contributions and Acknowledgments

Core Contributors

Daya Guo
Dejian Yang
Haowei Zhang
Junxiao Song
Ruoyu Zhang
Runxin Xu
Qihao Zhu
Shirong Ma
Peiyi Wang
Xiao Bi
Xiaokang Zhang
Xingkai Yu
Yu Wu
Z.F. Wu
Zhibin Gou
Zhihong Shao
Zhuoshu Li
Ziyi Gao

Contributors

Aixin Liu
Bing Xue
Bingxuan Wang
Bochao Wu
Bei Feng
Chengda Lu
Chenggang Zhao
Chengqi Deng
Chong Ruan
Damai Dai
Deli Chen
Dongjie Ji
Erhang Li
Fangyun Lin
Fucong Dai
Fuli Luo*
Guangbo Hao
Guanting Chen
Guowei Li
H. Zhang
Hanwei Xu
Honghui Ding
Huazuo Gao
Hui Qu

Hui Li
Jianzhong Guo
Jiashi Li
Jingchang Chen
Jingyang Yuan
Jinhao Tu
Junjie Qiu
Junlong Li
J.L. Cai
Jiaqi Ni
Jian Liang
Jin Chen
Kai Dong
Kai Hu*
Kaichao You
Kaige Gao
Kang Guan
Kexin Huang
Kuai Yu
Lean Wang
Lecong Zhang
Liang Zhao
Litong Wang
Liyue Zhang
Lei Xu
Leyi Xia
Mingchuan Zhang
Minghua Zhang
Minghui Tang
Mingxu Zhou
Meng Li
Miaojun Wang
Mingming Li
Ning Tian
Panpan Huang
Peng Zhang
Qiancheng Wang
Qinyu Chen
Qiushi Du
Ruiqi Ge*
Ruisong Zhang
Ruizhe Pan
Runji Wang
R.J. Chen
R.L. Jin

Ruyi Chen
Shanghao Lu
Shangyan Zhou
Shanhuang Chen
Shengfeng Ye
Shiyu Wang
Shuiping Yu
Shunfeng Zhou
Shuting Pan
S.S. Li
Shuang Zhou
Shaoqing Wu
Shengfeng Ye
Tao Yun
Tian Pei
Tianyu Sun
T. Wang
Wangding Zeng
Wen Liu
Wenfeng Liang
Wenjun Gao
Wenqin Yu*
Wentao Zhang
W.L. Xiao
Wei An
Xiaodong Liu
Xiaohan Wang
Xiaokang Chen
Xiaotao Nie
Xin Cheng
Xin Liu
Xin Xie
Xingchao Liu
Xinyu Yang
Xinyuan Li
Xuecheng Su
Xuheng Lin
X.Q. Li
Xiangyue Jin
Xiaojin Shen
Xiaosha Chen
Xiaowen Sun
Xiaoxiang Wang
Xinnan Song
Xinyi Zhou
Xianzu Wang
Xinxia Shan
Y.K. Li
Y.Q. Wang

Y.X. Wei
Yang Zhang
Yanhong Xu
Yao Li
Yao Zhao
Yaofeng Sun
Yaohui Wang
Yi Yu
Yichao Zhang
Yifan Shi
Yiliang Xiong
Ying He
Yishi Piao
Yisong Wang
Yixuan Tan
Yiyang Ma*
Yiyuan Liu
Yongqiang Guo
Yuan Ou
Yuduan Wang
Yue Gong
Yuheng Zou
Yujia He
Yunfan Xiong
Yuxiang Luo
Yuxiang You
Yuxuan Liu
Yuyang Zhou
Y.X. Zhu
Yanping Huang
Yaohui Li
Yi Zheng
Yuchen Zhu
Yunxian Ma
Ying Tang
Yukun Zha
Yuting Yan
Z.Z. Ren
Zehui Ren
Zhangli Sha
Zhe Fu
Zhean Xu
Zhenda Xie
Zhengyan Zhang
Zhewen Hao
Zhicheng Ma
Zhigang Yan
Zhiyu Wu
Zihui Gu

Zijia Zhu
Zijun Liu*
Zilin Li
Ziwei Xie
Ziyang Song
Zizheng Pan

Zhen Huang
Zhipeng Xu
Zhongyu Zhang
Zhen Zhang

Within each role, authors are listed alphabetically by the first name. Names marked with * denote individuals who have departed from our team.