# DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

DeepSeek-AI

research@deepseek.com

## Abstract

We introduce our first-generation reasoning models, DeepSeek-R1-Zero and DeepSeek-R1. DeepSeek-R1-Zero, a model trained via large-scale reinforcement learning (RL) without supervised fine-tuning (SFT) as a preliminary step, demonstrates remarkable reasoning capabilities. Through RL, DeepSeek-R1-Zero naturally emerges with numerous powerful and intriguing reasoning behaviors. However, it encounters challenges such as poor readability, and language mixing. To address these issues and further enhance reasoning performance, we introduce DeepSeek-R1, which incorporates multi-stage training and cold-start data before RL. DeepSeek-R1 achieves performance comparable to OpenAI-o1-1217 on reasoning tasks. To support the research community, we open-source DeepSeek-R1-Zero, DeepSeek-R1, and six dense models (1.5B, 7B, 8B, 14B, 32B, 70B) distilled from DeepSeek-R1 based on Qwen and Llama.

# DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning

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 是一个通过大规模强化学习（RL）训练的模型，在初步步骤中没有使用监督微调（SFT），展示了出色的推理能力。通过 RL，DeepSeek-R1-Zero 自然地涌现出许多强大且引人入胜的推理行为。然而，它遇到了诸如可读性差和语言混用等问题。为了解决这些问题并进一步提升推理性能，我们引入了 DeepSeek-R1，该模型在 RL 之前结合了多阶段训练和冷启动数据。 DeepSeek-R1 在推理任务上的表现与 OpenAI-o1-1217 相当。为了支持研究社区，我们开源了 DeepSeek-R1-Zero、DeepSeek-R1 以及从 DeepSeek-R1 基于 Qwen 和 Llama 蒸馏出的六个密集模型（1.5B、7B、8B、14B、32B、70B）。

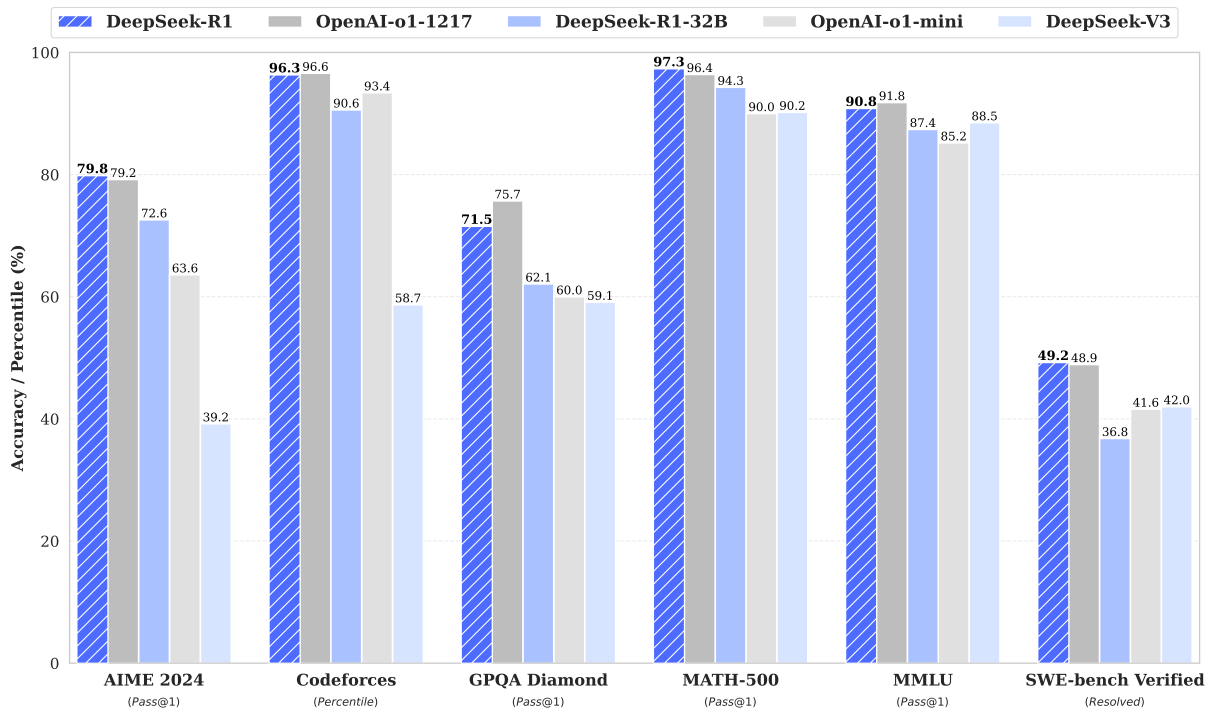


Figure 1 | Benchmark performance of DeepSeek-R1.

## Contents





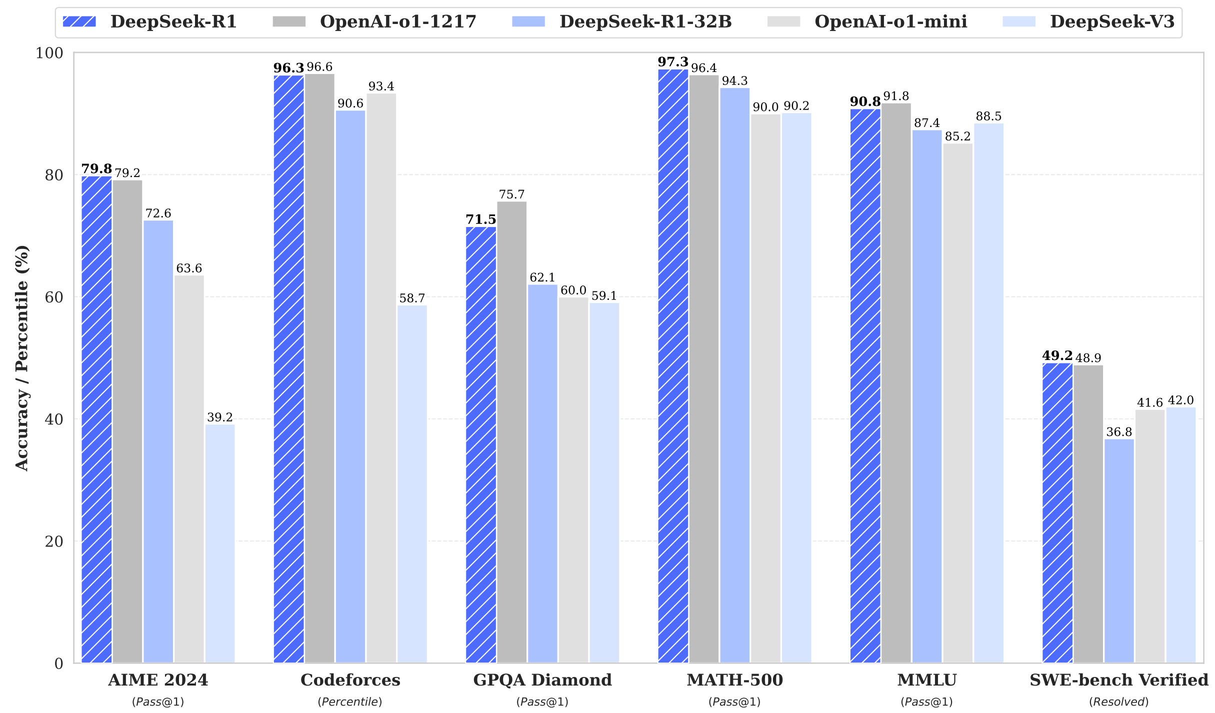


Figure 1 | DeepSeek-R1的基准性能。

## Contents

3

## 1. Introduction

In recent years, Large Language Models (LLMs) have been undergoing rapid iteration and evolution (Anthropic, 2024; Google, 2024; OpenAI, 2024a), progressively diminishing the gap towards Artificial General Intelligence (AGI).

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 (OpenAI, 2024b) 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 (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023), reinforcement learning (Kumar et al., 2024), and search algorithms such as Monte Carlo Tree Search and Beam Search (Feng et al., 2024; Trinh et al., 2024; Xin et al., 2024). However, none of these methods has achieved general reasoning performance comparable to OpenAI's o1 series models.

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 (Shao et al., 2024) 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.

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



## 1. Introduction

近年来，大型语言模型（LLMs）经历了快速的迭代和进化 (Anthropic, 2024; Google, 2024; OpenAI, 2024a)，逐渐缩小了通向通用人工智能（AGI）的差距。

最近，后训练已成为完整训练流程中的一个重要组成部分。研究表明，它可以在推理任务上提高准确性，与社会价值观保持一致，并适应用户偏好，同时相对于预训练所需的计算资源较少。在推理能力方面，OpenAI 的 o1 系列模型 (OpenAI, 2024b) 首次通过增加 Chain-of-Thought 推理过程的长度引入了推理时扩展的方法。这种方法在数学、编程和科学推理等各种推理任务中取得了显著改进。然而，有效的测试时扩展仍然是研究社区的一个开放问题。已有若干先前的工作探索了各种方法，包括基于过程的奖励模型 (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023)、强化学习 (Kumar et al., 2024) 以及如蒙特卡洛树搜索和束搜索等搜索算法 (Feng et al., 2024; Trinh et al., 2024; Xin et al., 2024)。然而，这些方法中没有一种能够实现与 OpenAI 的 o1 系列模型相当的通用推理性能。

在本文中，我们迈出了通过纯强化学习（RL）改善语言模型推理能力的第一步。我们的目标是探索 LLMs 在没有任何监督数据的情况下发展推理能力的潜力，专注于通过纯 RL 过程实现自我进化。具体来说，我们使用 DeepSeek-V3-Base 作为基础模型，并采用 GRPO (Shao et al., 2024) 作为 RL 框架来提高模型在推理方面的性能。在训练过程中，DeepSeek-R1-Zero 自然地表现出许多强大且有趣的推理行为。经过数千次 RL 步骤后，DeepSeek-R1-Zero 在推理基准测试中表现出超凡的性能。例如，AIME 2024 的 pass@1 分数从 15.6% 提高到 71.0%，并且通过多数投票，分数进一步提高到 86.7%，达到了与 OpenAI-o1-0912 相当的性能。

然而，DeepSeek-R1-Zero 遇到了诸如可读性差和语言混杂等问题。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，该模型结合了一小部分冷启动数据和多阶段训练管道。具体来说，我们首先收集了数千条冷启动数据以微调 DeepSeek-V3-Base 模型。随后，我们进行了类似于 DeepSeek-R1-Zero 的推理导向的 RL。当 RL 过程接近收敛时，我们通过对 RL 检查点进行拒绝采样生成新的 SFT 数据，并结合来自 DeepSeek-V3 的写作、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。经过新数据的微调后，检查点会经历额外的 RL 过程，考虑所有场景的提示。经过这些步骤，我们获得了称为 DeepSeek-R1 的检查点，其性能与 OpenAI-o1-1217 相当。

我们进一步探索了从 DeepSeek-R1 到较小密集模型的蒸馏。使用 Qwen2.5-32B (Qwen, 2024b) 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，较大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 (Dubey et al., 2024) 系列。值得注意的是，我们的蒸馏 14B 模型大幅超越了最先进的开源 QwQ-32B-Preview (Qwen, 2024a)，而蒸馏后的 32B 和 70B 模型在密集模型的推理基准测试中创下了新纪录。

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.

We further explore distillation from DeepSeek-R1 to smaller dense models. Using Qwen2.5-32B (Qwen, 2024b) 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 (Dubey et al., 2024) series. Notably, our distilled 14B model outperforms state-of-the-art open-source QwQ-32B-Preview (Qwen, 2024a) by a large margin, and the distilled 32B and 70B models set a new record on the reasoning benchmarks among dense models.

### 1.1. Contributions

**Post-Training: Large-Scale Reinforcement Learning on the Base Model**

- 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.
- 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.

**Distillation: Smaller Models Can Be Powerful Too**

- 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.
- 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,



### 1.1. Contributions

**训练后:在基础模型上进行大规模强化学习**

- 我们直接将强化学习（RL）应用于基础模型，而不需要依赖监督微调（SFT）作为初步步骤。这种方法使模型能够探索解决复杂问题的思维链（CoT），从而开发出DeepSeek-R1-Zero。DeepSeek-R1-Zero展示了诸如自我验证、反思和生成长思维链的能力，标志着研究社区的一个重要里程碑。值得注意的是，这是第一个公开研究验证了大型语言模型（LLM）的推理能力可以通过纯粹的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。
- 我们介绍了开发DeepSeek-R1的管道。该管道包括两个旨在发现改进的推理模式并符合人类偏好的RL阶段，以及两个作为模型推理和非推理能力种子的SFT阶段。我们相信这个管道将通过创建更好的模型来造福行业。

**蒸馏:更小的模型也可以很强大**

- 我们证明了较大模型的推理模式可以被蒸馏到较小的模型中，从而相比通过RL在小模型上发现的推理模式具有更好的性能。开源的DeepSeek-R1及其API将有助于研究社区在未来蒸馏出更好的小型模型。
- 使用由 DeepSeek-R1 生成的推理数据，我们微调了几个在研究社区中广泛使用的密集模型。评估结果显示，这些蒸馏后的较小密集模型在基准测试中表现出色。 DeepSeek-R1-Distill-Qwen-7B 在 AIME 2024 上达到了 55.5%，超过了 QwQ-32B-Preview。此外，DeepSeek-R1-Distill-Qwen-32B 在 AIME 2024 上得分 72.6%，在 MATH-500 上得分 94.3%，在 LiveCodeBench 上得分 57.2%。这些结果显著优于之前的开源模型，并且与 o1-mini 相当。我们向社区开源了基于 Qwen2.5 和 Llama3 系列的 1.5B、7B、8B、14B、32B 和 70B 参数量的蒸馏模型检查点。

### 1.2. Summary of Evaluation Results

- **推理任务**： (1) DeepSeek-R1 在 AIME 2024 上取得了 79.8% 的 Pass@1 成绩，略微超过了 OpenAI-o1-1217。在 MATH-500 上，它获得了令人印象深刻的 97.3% 分数，表现与 OpenAI-o1-1217 持平，并且显著优于其他模型。 (2) 在编程相关任务中，DeepSeek-R1 展现了专家级别的代码竞赛能力，在 Codeforces 上达到了 2,029 的 Elo 评级，超过了 96.3% 的参赛者。对于工程相关任务，DeepSeek-R1 略微优于 DeepSeek-V3，这可以帮助开发人员在实际任务中取得更好的效果。
- **知识**：在MMLU、MMLU-Pro和GPQA Diamond等基准测试中，DeepSeek-R1取得了出色的成绩，显著优于DeepSeek-V3，在MMLU上得分为90.8%，在MMLU-Pro上得分为84.0%，在GPQA Diamond上得分为71.5%。虽然其性能在这些基准测试中略低于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。在事实基准测试SimpleQA上，DeepSeek-R1也超过了DeepSeek-V3，证明了其处理基于事实的查询的能

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.

### 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. (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.
- **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.
- **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.

## 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



力。类似的趋势也在该基准测试中观察到，OpenAI-o1的表现超过了4o。

- **其他**：DeepSeek-R1 在广泛的任务中表现出色，包括创意写作、通用问题回答、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 上的胜率为 92.3%，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中表现出色，显著优于 DeepSeek-V3 在长上下文基准测试中的表现。

## 2. Approach

### 2.1. Overview

先前的研究工作严重依赖大量监督数据来提升模型性能。在本研究中，我们证明了通过大规模强化学习（RL），即使不使用监督微调（SFT）作为冷启动，推理能力也可以显著提高。此外，加入少量冷启动数据可以进一步提升性能。在以下章节中，我们将介绍：(1) DeepSeek-R1-Zero，它直接将RL应用于基础模型，而无需任何SFT数据；(2) DeepSeek-R1，它从使用数千个长链思维（CoT）示例微调的检查点开始应用RL。3) 从DeepSeek-R1蒸馏推理能力到小型密集模型。

### 2.2. DeepSeek-R1-Zero: 基于基础模型的强化学习

强化学习在推理任务中表现出显著的效果，正如我们之前的工作所证明的那样 (Shao et al., 2024; Wang et al., 2023)。然而，这些工作严重依赖于监督数据，而收集这些数据非常耗时。在本节中，我们探讨大型语言模型（LLMs）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粹的强化学习过程实现的自我进化。我们首先简要介绍我们的RL算法，然后展示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*

**组相对策略优化** 为了节省强化学习的训练成本，我们采用了组相对策略优化（GRPO）(Shao et al., 2024)。该方法放弃了通常与策略模型大小相同的评论家模型，并从组得分中估计基线。具体来说，对于每个问题$q$，GRPO从旧策略$\pi_{\theta_{old}}$中采样一组输出$\{o_1, o_2, \cdots, o_G\}$，然后通过最大化以下目标来优化策略模型$\pi_\theta$：

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

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

6

(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.

### 2.2. DeepSeek-R1-Zero: Reinforcement Learning on the Base Model

Reinforcement learning has demonstrated significant effectiveness in reasoning tasks, as evidenced by our previous works (Shao et al., 2024; Wang et al., 2023). 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.

#### 2.2.1. *Reinforcement Learning Algorithm*

**Group Relative Policy Optimization** In order to save the training costs of RL, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), 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 $\{o_1, o_2, \cdots, o_G\}$ from the old policy $\pi_{\theta_{old}}$ and then optimizes the policy model $\pi_\theta$ by maximizing the following objective:

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

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

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

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

#### 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:

- **Accuracy rewards**: The accuracy reward model evaluates whether the response is correct.



| |
|---|
| 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 | DeepSeek-R1-Zero 的模板。
textcolorredprompt 在训练期间将被具体的推理问题替换。

其中 $\varepsilon$ 和 $\beta$ 是超参数，$A_i$ 是优势，使用每组内输出对应的一组奖励 $\{r_1, r_2, \ldots, r_G\}$ 计算得出：

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

#### 2.2.2. *Reward Modeling*

奖励是训练信号的来源，决定了RL的优化方向。为了训练DeepSeek-R1-Zero，我们采用了一个基于规则的奖励系统，该系统主要由两种类型的奖励组成：

- **准确性奖励**：准确性奖励模型评估响应是否正确。例如，在数学问题具有确定性结果的情况下，模型需要以指定格式提供最终答案（例如，在方框内），以便进行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。
- **格式奖励**：除了准确性奖励模型外，我们还采用了一种格式奖励模型，该模型强制将模型的思考过程放在 '<think>' 和 '</think>' 标签之间。

我们不在开发 DeepSeek-R1-Zero 时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遭受奖励劫持问题，并且重新训练奖励模型需要额外的训练资源，并使整个训练管道复杂化。

#### 2.2.3. *Training Template*

为了训练DeepSeek-R1-Zero，我们首先设计了一个简单的模板，引导基础模型遵循我们指定的指令。如表1所示，该模板要求DeepSeek-R1-Zero首先生成一个推理过程，然后是最终答案。我们有意将约束限制在这种结构格式上，避免任何内容特定的偏差——例如，强制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。

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

**Performance of DeepSeek-R1-Zero** 图 2 展示了 DeepSeek-R1-Zero 在 AIME 2024 基准测试中在整个 RL 训练过程中的性能轨迹。如图所示，随着 RL 训练的推进，DeepSeek-R1-Zero 表现出稳定且一致的性能提升。值得注意的是，AIME 2024 的平均 pass@1 分数显著提高，从最初的 15.6% 提升到令人印象深刻的 71.0%，达到了与 OpenAI-o1-0912 相当的性能水平。这一显著改

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.

| 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.

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.

- **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.

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.

#### 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.

#### 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 | DeepSeek-R1-Zero 和 OpenAI o1 模型在推理相关基准测试上的比较。

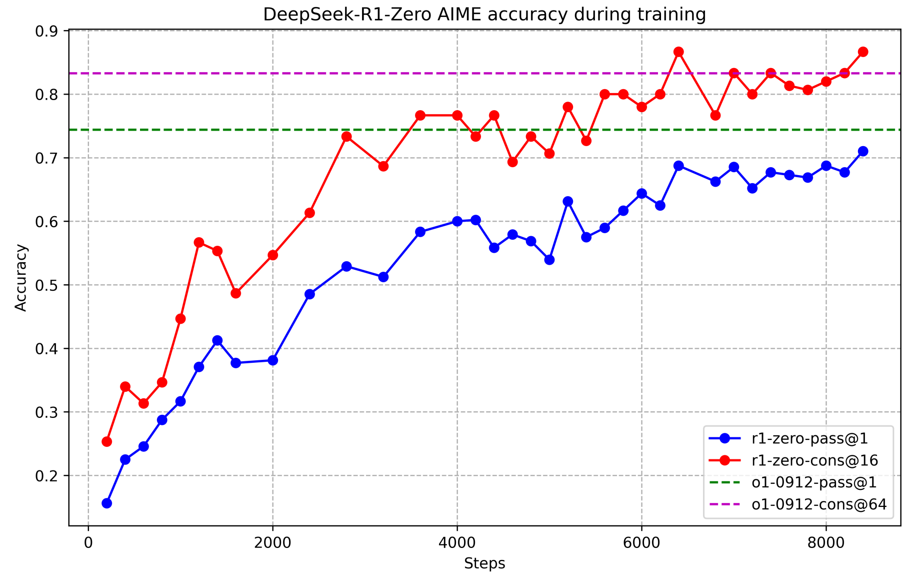


Figure 2 | AIME 在训练期间的 DeepSeek-R1-Zero 准确率。对于每个问题，我们采样 16 个响应并计算总体平均准确率，以确保评估的稳定性。

进突显了我们的 RL 算法在优化模型性能方面的有效性。

表 2 提供了 DeepSeek-R1-Zero 和 OpenAI 的 o1-0912 模型在多种推理相关基准测试上的比较分析。研究结果表明，RL 赋予了 DeepSeek-R1-Zero 强大的推理能力，而无需任何监督微调数据。这是一个值得注意的成就，因为它强调了模型通过 RL 单独学习和泛化的能力。此外，通过应用多数投票，DeepSeek-R1-Zero 的性能可以进一步提升。例如，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%，超过了 OpenAI-o1-0912 的表现。无论是否使用多数投票，DeepSeek-R1-Zero 能够实现如此具有竞争力的性能，突显了其强大的基础能力和在推理任务中进一步发展的潜力。

**DeepSeek-R1-Zero 的自我进化过程** DeepSeek-R1-Zero 的自我进化过程是强化学习 (RL) 如何驱动模型自主提升其推理能力的一个引人入胜的展示。通过直接从基础模型开始进行强化学习，我们可以密切监测模型的进步，而不受监督微调阶段的影响。这种方法提供了清晰的视角，展示了模型随着时间的推移如何演变，特别是在处理复杂推理任务方面的能力。

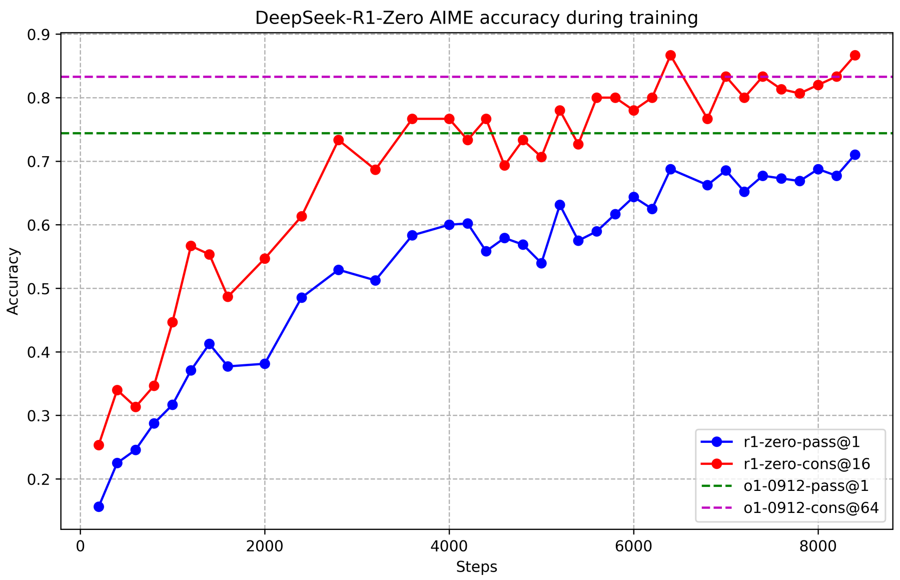


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

**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.

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.

**Self-evolution Process of DeepSeek-R1-Zero** 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



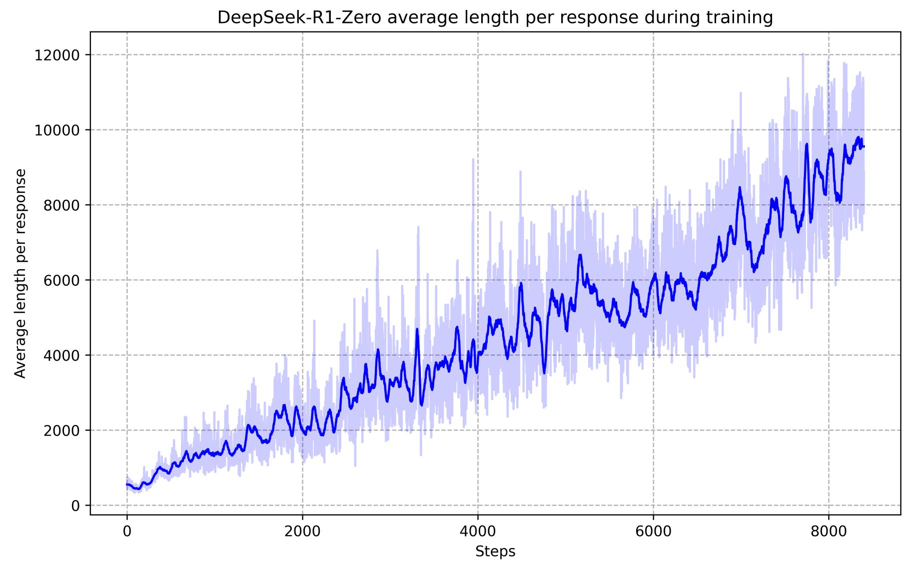


Figure 3 | 在RL过程中，DeepSeek-R1-Zero在训练集上的平均响应长度。DeepSeek-R1-Zero自然学会了用更多的思考时间来解决推理任务。

如图 3 所示，DeepSeek-R1-Zero 的思考时间在整个训练过程中表现出持续的改进。这种改进不是外部调整的结果，而是模型内部的发展。通过利用扩展的测试时间计算，DeepSeek-R1-Zero 自然获得了解决越来越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标记，使模型能够更深入地探索和优化其思维过程。

这一自我进化中最显著的方面之一是随着测试时间计算的增加，出现了复杂的行为。例如反思——模型重新审视和重新评估其先前的步骤——以及探索解决问题的替代方法等行为自发出现。这些行为并不是明确编程的，而是模型与强化学习环境互动的结果。这种自发的发展显著增强了 DeepSeek-R1-Zero 的推理能力，使其能够更高效、准确地应对更具挑战性的任务。

**DeepSeek-R1-Zero 的"恍然大悟"时刻** 在训练 DeepSeek-R1-Zero 期间观察到的一个特别有趣的现象是"恍然大悟"时刻的发生。如表 3 所示，这个时刻发生在模型的中间版本中。在这个阶段，DeepSeek-R1-Zero 学会了通过重新评估其初始方法来为问题分配更多的思考时间。这种行为不仅是模型推理能力增长的证明，也是强化学习可以导致意外和复杂结果的迷人例子。

这一时刻不仅是模型的"恍然大悟"时刻，也是研究人员观察其行为时的"恍然大悟"时刻。它强调了强化学习的力量和美丽：我们不需要明确教模型如何解决问题，只需提供正确的激励，它就能自主发展出高级的问题解决策略。"恍然大悟"时刻有力地提醒了 RL 在解锁人工智能新水平方面的潜力，为未来更加自主和适应性强的模型铺平了道路。

**DeepSeek-R1-Zero 的缺点** 尽管 DeepSeek-R1-Zero 展现了强大的推理能力和自主开发出意外且强大的推理行为，但它仍面临一些问题。例如，DeepSeek-R1-Zero 在可读性和语言混用方面存在挑战。为了使推理过程更易读并将其分享给开源社区，我们探索了 DeepSeek-R1，这是一

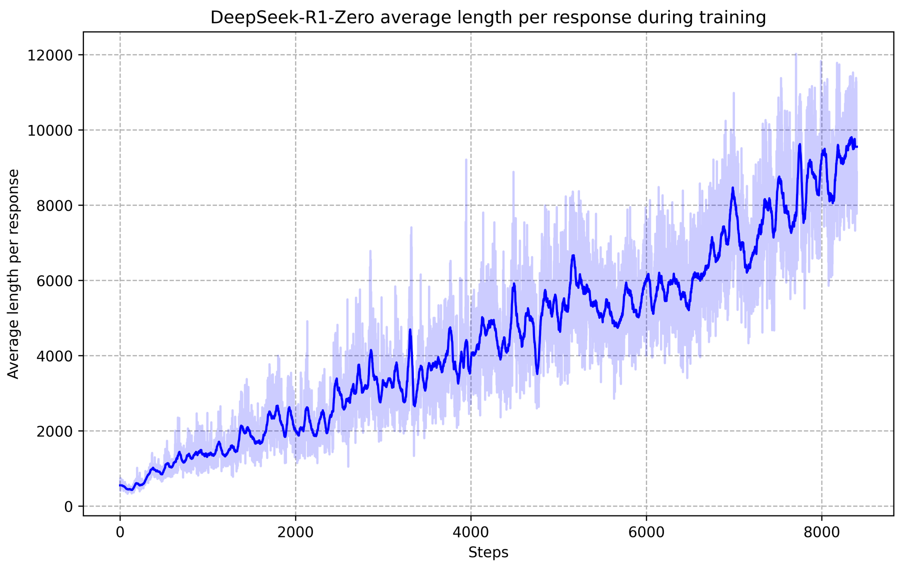


Figure 3 | The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

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.

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.

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.

**Aha Moment of DeepSeek-R1-Zero** 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



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

Table 3 | 一个有趣的 DeepSeek-R1-Zero 中间版本的“啊哈时刻”。该模型学会了用拟人的语气进行反思。这对我们来说也是一个“啊哈时刻”，使我们能够见证强化学习的力量和美丽。

种利用人类友好型冷启动数据的强化学习方法。

### 2.3. DeepSeek-R1: Reinforcement Learning with Cold Start

受到DeepSeek-R1-Zero令人鼓舞的结果的启发，两个自然的问题浮现：1) 通过引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速收敛？2) 如何训练一个用户友好的模型，使其不仅能够产生清晰连贯的思维链（CoT），还能展现出强大的通用能力？为了解决这些问题，我们设计了一个训练DeepSeek-R1的流水线。该流水线包含四个阶段，具体如下。

#### 2.3.1. *Cold Start*

不同于DeepSeek-R1-Zero，为了防止从基础模型开始的RL训练早期出现不稳定的冷启动阶段，对于DeepSeek-R1，我们构建并收集了一小部分长链思考（CoT）数据来微调模型作为初始的RL执行者。为了收集这些数据，我们探索了多种方法：使用长链思考作为示例的少量样本提示，直接提示模型生成带有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处理来优化结果。

在本工作中，我们收集了数千条冷启动数据以微调DeepSeek-V3-Base作为RL的起点。与DeepSeek-R1-Zero, the advantages of cold start data include:

- 可读性：DeepSeek-R1-Zero的一个关键限制是其内容通常不适合阅读。回答可能混合多种语言或缺乏markdown格式来突出显示答案以供用户使用。相比之下，在为DeepSeek-R1创建冷启动数据时，我们设计了一个可读的模式，该模式在每个回答的末尾包含一个摘要，并过滤掉那些对读者不友好的回答。在这里，我们将输出格式定义为|special_token|<推

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

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.

but also a captivating example of how reinforcement learning can lead to unexpected and sophisticated outcomes.

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.

**Drawback of DeepSeek-R1-Zero** 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.

### 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



理过程>|special_token|<摘要>，其中推理过程是查询的CoT，而摘要是用于总结推理结果的。

- 通过精心设计带有先验知识的冷启动数据模式，我们观察到相比 DeepSeek-R1-Zero 有更好的性能表现。我们认为迭代训练是推理模型的更好方法。

#### *2.3.2. Reasoning-oriented Reinforcement Learning*

在使用冷启动数据对DeepSeek-V3-Base进行微调后，我们应用了与DeepSeek-R1-Zero中相同的大型强化学习训练过程。此阶段专注于增强模型的推理能力，特别是在需要强推理的任务中，例如编程、数学、科学和逻辑推理，这些任务涉及具有明确解决方案的明确定义问题。在训练过程中，我们观察到CoT经常表现出语言混用，特别是在RL提示涉及多种语言时。为了解决语言混用的问题，我们在RL训练中引入了语言一致性奖励，该奖励计算为CoT中目标语言单词的比例。尽管消融实验显示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，使其更易读。最后，我们将推理任务的准确性与语言一致性的奖励直接相加，形成最终奖励。然后，我们在微调后的模型上应用RL训练，直到它在推理任务上达到收敛。

#### *2.3.3. Rejection Sampling and Supervised Fine-Tuning*

当面向推理的强化学习（RL）收敛时，我们利用生成的检查点来收集监督微调（SFT）数据以用于下一轮。与主要关注推理的初始冷启动数据不同，此阶段结合了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方面的能力。具体来说，我们按照以下描述生成数据并微调模型。

**推理数据** 我们整理了推理提示，并通过从上述强化学习训练的检查点进行拒绝采样来生成推理轨迹。在前一阶段，我们仅包括可以使用基于规则的奖励进行评估的数据。然而，在此阶段，我们通过纳入额外的数据扩展了数据集，其中一些数据使用生成式奖励模型，将真实值和模型预测输入DeepSeek-V3进行判断。此外，由于模型输出有时是混乱且难以阅读的，我们过滤掉了包含混合语言、长段落和代码块的思维链。对于每个提示，我们采样多个响应并仅保留正确的响应。总共，我们收集了大约60万个与推理相关的训练样本。

**非推理数据** 对于非推理数据，如写作、事实问答、自我认知和翻译，我们采用了DeepSeek-V3的流程，并重用了DeepSeek-V3的SFT数据集的部分内容。对于某些非推理任务，我们在回答问题之前通过提示DeepSeek-V3生成潜在的思维链。然而，对于更简单的查询，如"hello"，我们不会提供思维链作为回应。最终，我们收集了大约20万个与推理无关的训练样本。

我们使用上述整理的数据集（约80万个样本）对DeepSeek-V3-Base进行了两个epoch的微调。

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.

#### 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.

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:

- 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.
- 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.

#### 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



#### 2.3.4. *Reinforcement Learning for all Scenarios*

为了进一步使模型与人类偏好保持一致，我们实施了一个次要的强化学习阶段，旨在提高模型的帮助性和无害性，同时精炼其推理能力。具体来说，我们使用奖励信号和多样化的提示分布组合来训练模型。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。对于通用数据，我们依赖奖励模型来捕捉复杂和微妙场景中的人类偏好。我们基于DeepSeek-V3管道，并采用类似的偏好对和训练提示分布。对于帮助性，我们仅专注于最终摘要，确保评估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。对于无害性，我们评估模型的整个响应，包括推理过程和摘要，以识别和减轻生成过程中可能出现的任何潜在风险、偏见或有害内容。最终，奖励信号和多样化数据分布的集成使我们能够训练出一个在推理方面表现出色，同时优先考虑帮助性和无害性的模型。

### 2.4. Distillation: Empower Small Models with Reasoning Capability

为了使更小的模型具备像DeepSeek-R1这样的推理能力，我们直接微调了开源模型，如Qwen (Qwen, 2024b) 和 Llama (AI@Meta, 2024)，使用由DeepSeek-R1策划的80万样本，具体方法详见§2.3.3。我们的研究结果表明，这种简单的蒸馏方法显著增强了较小模型的推理能力。

我们在这里使用的基模型包括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是因为其推理能力略优于Llama-3.1。

对于蒸馏后的模型，我们仅应用SFT，不包括RL阶段，尽管加入RL可以大幅提升模型性能。我们的主要目标是展示蒸馏技术的有效性，将RL阶段的探索留给更广泛的研究社区。

## 3. Experiment

**基准测试** 我们在MMLU (Hendrycks et al., 2020)、MMLU-Redux (Gema et al., 2024)、MMLU-Pro (Wang et al., 2024)、C-Eval (Huang et al., 2023) 和 CMMLU (Li et al., 2023)、IFEval (Zhou et al., 2023)、FRAMES (Krishna et al., 2024)、GPQA Diamond (Rein et al., 2023)、SimpleQA (OpenAI, 2024c)、C-SimpleQA (He et al., 2024)、SWE-Bench Verified (OpenAI, 2024d)、Aider [1]、LiveCodeBench (Jain et al., 2024)（2024-08 – 2025-01）、Codeforces [2]、中国国家高中数学奥林匹克竞赛（CNMO 2024）[3] 和美国数学邀请赛2024（AIME 2024） (MAA, 2024) 上评估模型。除了标准基准测试外，我们还使用大型语言模型作为评判者来评估模型在开放生成任务上的表现。具体来说，我们遵循AlpacaEval 2.0 (Dubois et al., 2024) 和 Arena-Hard (Li et al., 2024) 的原始配置，这些配置利用GPT-4-Turbo-1106作为评判者进行成对比较。在这里，我们仅将最终摘要提交给评估以避免长度偏差。对于蒸馏模型，我们在AIME 2024、MATH-500、GPQA

[1] https://aider.chat
[2] https://codeforces.com
[3] https://www.cms.org.cn/Home/comp/comp/cid/12.html

as the proportion of target language words in the CoT. Although ablation experiments 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.

#### 2.3.3. *Rejection Sampling and Supervised Fine-Tuning*

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.

**Reasoning data** 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 parapraphs, 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.

**Non-Reasoning data** 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.

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

#### 2.3.4. *Reinforcement Learning for all Scenarios*

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 simultane-



Diamond、Codeforces和LiveCodeBench上报告代表性结果。

**评估提示** 按照DeepSeek-V3的设置，标准基准测试如MMLU、DROP、GPQA Diamond和SimpleQA使用来自simple-evals框架的提示进行评估。对于MMLU-Redux，我们在零样本设置中采用Zero-Eval提示格式 (Lin, 2024)。关于MMLU-Pro、C-Eval和CLUE-WSC，由于原始提示是少样本，我们稍微修改了提示以适应零样本设置。少样本中的链式思维可能会损害 DeepSeek-R1 的性能。其他数据集遵循其创建者提供的默认提示和原始评估协议。对于代码和数学基准测试，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）。LiveCodeBench的模型性能评估使用CoT格式，数据收集时间为2024年8月至2025年1月。Codeforces数据集使用来自10个Div.2比赛的问题以及专家设计的测试用例进行评估，之后计算预期评分和参赛者的百分比。SWE-Bench验证结果通过无代理框架 (Xia et al., 2024) 获得。AIDER相关基准测试使用“diff”格式进行测量。 DeepSeek-R1 的输出在每个基准测试中最多为32,768个标记。

**基线** 我们对多个强大的基线进行了全面评估，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较为困难，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 (Qwen, 2024a)。

**评估设置** 我们将模型的最大生成长度设置为32,768个标记。我们发现，使用贪婪解码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，默认情况下我们使用pass@$k$评估 (Chen et al., 2021) 并报告非零温度下的pass@1。具体来说，我们使用采样温度为0.6和top-$p$值为0.95生成$k$个响应（通常在4到64之间，取决于测试集的大小），然后计算pass@1。

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

其中 $p_i$ 表示第 $i$ 个回答的正确性。此方法提供了更可靠的性能估计。对于 AIME 2024，我们还报告了共识（多数投票）结果 (Wang et al., 2022)，使用 64 个样本，记为 cons@64。

### 3.1. DeepSeek-R1 Evaluation

对于以教育为导向的知识基准测试，如MMLU、MMLU-Pro和GPQA Diamond，DeepSeek-R1相比DeepSeek-V3表现出更优异的性能。这一改进主要归功于STEM相关问题上的准确性提高，通过大规模强化学习实现了显著的进步。此外，DeepSeek-R1在FRAMES上表现出色，这是一个依赖长上下文的问答任务，展示了其强大的文档分析能力。这突显了推理模型在AI驱动的搜索和数据分析任务中的潜力。在事实基准测试SimpleQA上，DeepSeek-R1超越了DeepSeek-V3，证明了其处理基于事实的查询的能力。类似的趋势显示OpenAI-o1在此基准测试中超过了GPT-4o。然而，在中文SimpleQA基准测试中，DeepSeek-R1的表现不如DeepSeek-V3，主要

13

ously 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. 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.

### 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 (Qwen, 2024b) and Llama (AI@Meta, 2024) 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.

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.

## 3. Experiment

**Benchmarks** We evaluate models on MMLU (Hendrycks et al., 2020), MMLU-Redux (Gema et al., 2024), MMLU-Pro (Wang et al., 2024), C-Eval (Huang et al., 2023), and CMMLU (Li et al., 2023), IFEval (Zhou et al., 2023), FRAMES (Krishna et al., 2024), GPQA Diamond (Rein et al., 2023), SimpleQA (OpenAI, 2024c), C-SimpleQA (He et al., 2024), SWE-Bench Verified (OpenAI, 2024d), Aider [1], LiveCodeBench (Jain et al., 2024) (2024-08 – 2025-01), Codeforces [2], Chinese National High School Mathematics Olympiad (CNMO 2024)[3], and American Invitational Mathematics

[1] `https://aider.chat`
[2] `https://codeforces.com`
[3] `https://www.cms.org.cn/Home/comp/comp/cid/12.html`



| | 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 |

Table 4 | DeepSeek-R1与其他代表性模型的比较。

是因为在安全RL后倾向于拒绝回答某些查询。没有安全RL，DeepSeek-R1可以实现超过70%的准确率。

DeepSeek-R1在IF-Eval上也取得了令人印象深刻的结果，该基准测试旨在评估模型遵循格式指令的能力。这些改进可以追溯到在监督微调（SFT）和RL训练的最后阶段加入了指令跟随数据。此外，在AlpacaEval2.0和ArenaHard上也观察到了显著的性能表现，表明DeepSeek-R1在写作任务和开放域问答方面具有优势。其显著优于DeepSeek-V3的表现突显了大规模RL带来的泛化优势，不仅增强了推理能力，还提高了跨多个领域的性能。此外，DeepSeek-R1生成的摘要长度较为简洁，在ArenaHard上的平均长度为689个标记，在AlpacaEval 2.0上的平均长度为2,218个字符。这表明DeepSeek-R1在基于GPT的评估中避免了长度偏差，进一步证明了其在多个任务中的稳健性。

在数学任务上，DeepSeek-R1的表现与OpenAI-o1-1217相当，大幅超越其他模型。在编程算法任务上，如LiveCodeBench和Codeforces，专注于推理的模型主导了这些基准测试。在工程导向的编程任务上，OpenAI-o1-1217在Aider上优于DeepSeek-R1，但在SWE Verified上表现相当。我们相信，随着相关RL训练数据量的增加，DeepSeek-R1的工程性能将在下一个版本中得到改善，目前这类数据仍然非常有限。

Examination 2024 (AIME 2024) (MAA, 2024). 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 (Dubois et al., 2024) and Arena-Hard (Li et al., 2024), 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.

**Evaluation Prompts** 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 (Lin, 2024) 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 (Xia et al., 2024). AIDER-related benchmarks are measured using a "diff" format. DeepSeek-R1 outputs are capped at a maximum of 32,768 tokens for each benchmark.

**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 (Qwen, 2024a).

**Evaluation Setup** 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 (Chen et al., 2021) 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



### 3.2. Distilled Model Evaluation

| 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 | DeepSeek-R1蒸馏模型与其他可比模型在推理相关基准上的比较。

如表5所示，简单蒸馏DeepSeek-R1的输出使得高效的DeepSeek-R1-7B（即DeepSeek-R1-Distill-Qwen-7B，以下类似简称）在所有方面都超过了非推理模型如GPT-4o-0513。DeepSeek-R1-14B在所有评估指标上超过了QwQ-32B-Preview，而DeepSeek-R1-32B和DeepSeek-R1-70B在大多数基准测试中显著超过了o1-mini。这些结果展示了蒸馏的强大潜力。此外，我们发现将强化学习应用于这些蒸馏模型可以带来显著的进一步提升。我们认为这值得进一步探索，因此这里仅呈现简单的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 | 精馏模型和强化学习模型在推理相关基准上的比较。

在第 3.2 节中，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结果。然而，仍有一个问题：该模型是否可以通过论文中讨论的大规模强化学习训练，在没有蒸馏的情况下达到类似的性能？

为了解答这个问题，我们在 Qwen-32B-Base 上使用数学、代码和 STEM 数据进行了大规模的强化学习训练，训练了超过 10K 步，最终得到了 DeepSeek-R1-Zero-Qwen-32B。表 6 中的实验结果显示，经过大规模强化学习训练后的 32B 基础模型的性能与 QwQ-32B-Preview 相当。然而，从 DeepSeek-R1 蒸馏得到的 DeepSeek-R1-Distill-Qwen-32B 在所有基准测试中的表现明显优于 DeepSeek-R1-Zero-Qwen-32B。

is then calculated as

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

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 (Wang et al., 2022) using 64 samples, denoted as cons@64.

### 3.1. DeepSeek-R1 Evaluation

| | 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 |

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

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



因此，我们可以得出两个结论：首先，将更强大的模型蒸馏到较小的模型中可以获得优异的结果，而较小的模型依赖于本文提到的大规模强化学习需要巨大的计算资源，并且可能无法达到蒸馏的效果。其次，虽然蒸馏策略既经济又有效，但要超越智能的边界，仍然可能需要更强大的基础模型和更大规模的强化学习。

### 4.2. Unsuccessful Attempts

在开发 DeepSeek-R1 的早期阶段，我们也遇到了失败和挫折。我们在此分享这些失败经验以提供见解，但这并不意味着这些方法无法开发出有效的推理模型。

**过程奖励模型 (PRM)** PRM 是一种合理的方法，可以引导模型采用更好的方法来解决推理任务 (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023)。然而，在实践中，PRM 存在三个主要限制，可能阻碍其最终成功。首先，明确定义一般推理中的细粒度步骤具有挑战性。其次，确定当前中间步骤是否正确是一项艰巨的任务。使用模型进行自动标注可能无法产生令人满意的结果，而手动标注不利于扩展。第三，一旦引入基于模型的 PRM，不可避免地会导致奖励劫持 (Gao et al., 2022)，重新训练奖励模型需要额外的训练资源，并使整个训练管道复杂化。总之，尽管 PRM 展示了良好的能力，可以在模型生成的前 N 个响应中重新排序或辅助指导搜索 (Snell et al., 2024)，但在我们的实验中，与它在大规模强化学习过程中引入的额外计算开销相比，其优势有限。

**蒙特卡洛树搜索 (MCTS)** 受 AlphaGo (Silver et al., 2017b) 和 AlphaZero (Silver et al., 2017a) 的启发，我们探索了使用蒙特卡洛树搜索 (MCTS) 来增强推理时计算的可扩展性。这种方法涉及将答案分解为更小的部分，以便模型系统地探索解空间。为此，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。在训练过程中，我们首先使用收集到的提示通过由预训练价值模型引导的 MCTS 找到答案。随后，我们使用生成的问题-答案对来训练行为模型和价值模型，迭代改进这一过程。

然而，当扩大训练规模时，这种方法遇到了几个挑战。首先，与国际象棋中相对明确定义的搜索空间不同，令牌生成呈现出指数级更大的搜索空间。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。训练一个细粒度的价值模型本质上是困难的，这使得模型难以迭代改进。虽然 AlphaGo 的核心成功依赖于训练价值模型以逐步提高性能，但在这个设置中复制这一原则由于令牌生成的复杂性而变得困难。

总之，虽然 MCTS 在推理过程中与预训练价值模型结合可以提高性能，但通过自搜索迭代提升模型性能仍然是一个重大挑战。

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%.

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.

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.

### 3.2. Distilled Model Evaluation

| 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.

As shown in Table 5, simply distilling DeepSeek-R1's outputs enables the efficient DeepSeek-



## 5. Conclusion, Limitations, and Future Work

在本工作中，我们分享了通过强化学习增强模型推理能力的历程。$DeepSeek-R1-Zero$ 代表了一种纯粹的 RL 方法，不依赖于冷启动数据，在各种任务中表现出色。$DeepSeek-R1$ 更加强大，利用冷启动数据并结合迭代 RL 微调，最终在一系列任务上实现了与 OpenAI-o1-1217 相当的性能。

我们进一步探索将推理能力蒸馏到小型密集模型中。我们使用 $DeepSeek-R1$ 作为教师模型生成了 80 万个训练样本，并微调了多个小型密集模型。结果非常有希望：DeepSeek-R1-Distill-Qwen-1.5B 在数学基准测试中超过了 GPT-4o 和 Claude-3.5-Sonnet，在 AIME 上达到了 28.9%，在 MATH 上达到了 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微调模型。

在未来，我们计划在以下方向上对 $DeepSeek-R1$ 进行研究投资。

- **通用能力:** 目前，DeepSeek-R1在函数调用、多轮对话、复杂角色扮演和JSON输出等任务上的能力不如DeepSeek-V3。展望未来，我们计划探索如何利用长链思考（CoT）来增强这些领域中的任务表现。
- **语言混合:** DeepSeek-R1 目前针对中文和英文进行了优化，这可能导致在处理其他语言的查询时出现语言混合问题。例如，即使查询不是用英语或中文提出的，DeepSeek-R1 可能仍会使用英语进行推理和响应。我们计划在未来的更新中解决这一限制。
- **提示工程:** 在评估 DeepSeek-R1 时，我们观察到它对提示非常敏感。少样本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。
- **软件工程任务:** 由于长时间的评估影响了RL过程的效率，大规模RL在软件工程任务中尚未得到广泛应用。因此，DeepSeek-R1在软件工程基准测试上相比DeepSeek-V3并没有显示出巨大的改进。未来版本将通过在软件工程数据上实施拒绝采样或在RL过程中引入异步评估来提高效率。

## References

AI@Meta. Llama 3.1 model card, 2024. URL `https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md`.

Anthropic. Claude 3.5 sonnet, 2024. URL `https://www.anthropic.com/news/claude-3-5-sonnet`.

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.

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 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.

## 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?

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.

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.

### 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



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`.

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.

Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. arXiv preprint arXiv:2404.04475, 2024.

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`.

L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization, 2022. URL `https://arxiv.org/abs/2210.10760`.

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`.

Google. Our next-generation model: Gemini 1.5, 2024. URL `https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024`.

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.

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.

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.

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`.

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,

18

these approaches are incapable of developing effective reasoning models.

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

**Monte Carlo Tree Search (MCTS)** Inspired by AlphaGo (Silver et al., 2017b) and AlphaZero (Silver et al., 2017a), 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.

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.

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.



abs/2409.12941, 2024. doi: 10.48550/ARXIV.2409.12941. URL `https://doi.org/10.48550/arXiv.2409.12941`.

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.

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.

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.

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.

B. Y. Lin. ZeroEval: A Unified Framework for Evaluating Language Models, July 2024. URL `https://github.com/WildEval/ZeroEval`.

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`.

OpenAI. Hello GPT-4o, 2024a. URL `https://openai.com/index/hello-gpt-4o/`.

OpenAI. Learning to reason with llms, 2024b. URL `https://openai.com/index/learning-to-reason-with-llms/`.

OpenAI. Introducing SimpleQA, 2024c. URL `https://openai.com/index/introducing-simpleqa/`.

OpenAI. Introducing SWE-bench verified we're releasing a human-validated subset of swe-bench that more, 2024d. URL `https://openai.com/index/introducing-swe-bench-verified/`.

Qwen. Qwq: Reflect deeply on the boundaries of the unknown, 2024a. URL `https://qwenlm.github.io/blog/qwq-32b-preview/`.

Qwen. Qwen2.5: A party of foundation models, 2024b. URL `https://qwenlm.github.io/blog/qwen2.5`.

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.

## 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.

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.

In the future, we plan to invest in research across the following directions for DeepSeek-R1.

- **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.
- **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.
- **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.
- **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.



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.

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, 2017a. URL `http://arxiv.org/abs/1712.01815`.

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, 2017b. doi: 10.1038/NATURE24270. URL `https://doi.org/10.1038/nature24270`.

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`.

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.

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.

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.

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.

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`.

C. S. Xia, Y. Deng, S. Dunn, and L. Zhang. Agentless: Demystifying llm-based software engineering agents. arXiv preprint, 2024.

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

## References


AI@Meta. Llama 3.1 model card, 2024. URL `https://github.com/meta-llama/llama-models/blob/main/models/llama3_1/MODEL_CARD.md`.

Anthropic. Claude 3.5 sonnet, 2024. URL `https://www.anthropic.com/news/claude-3-5-sonnet`.

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`.

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.

Y. Dubois, B. Galambosi, P. Liang, and T. B. Hashimoto. Length-controlled alpacaeval: A simple way to debias automatic evaluators. arXiv preprint arXiv:2404.04475, 2024.

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`.

L. Gao, J. Schulman, and J. Hilton. Scaling laws for reward model overoptimization, 2022. URL `https://arxiv.org/abs/2210.10760`.

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`.

Google. Our next-generation model: Gemini 1.5, 2024. URL `https://blog.google/technology/ai/google-gemini-next-generation-model-february-2024`.

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.





proof assistant feedback for reinforcement learning and monte-carlo tree search, 2024. URL `https://arxiv.org/abs/2408.08152`.

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.

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.

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.

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`.

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`.

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.

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.

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.

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.

B. Y. Lin. ZeroEval: A Unified Framework for Evaluating Language Models, July 2024. URL `https://github.com/WildEval/ZeroEval`.

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`.

OpenAI. Hello GPT-4o, 2024a. URL `https://openai.com/index/hello-gpt-4o/`.

OpenAI. Learning to reason with llms, 2024b. URL `https://openai.com/index/learning-to-reason-with-llms/`.

OpenAI. Introducing SimpleQA, 2024c. URL `https://openai.com/index/introducing-simpleqa/`.




# Appendix

## A. Contributions and Acknowledgments

**核心贡献者**

郭大雅
杨德建
张浩伟
宋俊晓
张若愚
徐若昕
朱启豪
马世荣
王培一
毕潇
张小康
余兴凯
吴宇
吴志锋
勾志斌
邵志宏
李卓舒
高子毅

**贡献者**

刘爱欣
雪冰
王秉轩
吴博超
冯贝
卢成达
赵成刚
邓成琪
阮冲
戴大麦
陈德立
季东杰
李尔航
林芳云
戴福聪
罗富礼*
郝光博
陈冠廷
李国威
张H.
徐汉伟
丁洪辉
高华作
屈慧
李辉
郭建中
李家石
陈景昌
袁景阳
涂金浩
邱俊杰
李军龙
蔡J.L.
倪嘉琪
梁健
陈进
董凯
胡凯*
尤凯超
高凯歌
关康
黄可欣
余快
王磊
张乐聪
赵亮
王立通

22

OpenAI. Introducing SWE-bench verified we're releasing a human-validated subset of swe-bench that more, 2024d. URL `https://openai.com/index/introducing-swe-bench-verified/`.

Qwen. Qwq: Reflect deeply on the boundaries of the unknown, 2024a. URL `https://qwenlm.github.io/blog/qwq-32b-preview/`.

Qwen. Qwen2.5: A party of foundation models, 2024b. URL `https://qwenlm.github.io/blog/qwen2.5`.

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.

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.

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, 2017a. URL `http://arxiv.org/abs/1712.01815`.

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, 2017b. doi: 10.1038/NATURE24270. URL `https://doi.org/10.1038/nature24270`.

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`.

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.

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.

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.

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.




张丽月
徐磊
夏磊毅
张明川
张明华
唐明辉
周明旭
李萌
王苗军
李明明
田宁
黄盼盼
张鹏
王乾城
陈沁宇
杜秋石
葛瑞琪*
张瑞松
潘睿哲
王润基
R.J. 陈
R.L. 金
陈如意
卢上浩
周尚炎
陈山黄
叶胜峰
王世宇
余水萍
周顺风
潘姝婷
S.S. 李
周爽
吴少青
叶胜峰
云涛
柏天
孙天宇
T. 王
曾王鼎
刘文
梁文锋
高文军
余文琴*
张 Wentao
W.L. 肖
安伟
刘晓东
王晓涵
陈小康
聂小涛
程欣
刘欣
谢欣
刘兴超
杨昕宇
李昕远
苏学成
林旭恒
X.Q. 李
金翔跃
沈晓瑾
陈晓莎
孙晓文
王小翔
宋昕楠
周欣仪
王先祖
单鑫霞
Y.K. 李
Y.Q. 王
Y.X. 魏
张阳
徐燕红
李瑶
赵瑶

23

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`.

C. S. Xia, Y. Deng, S. Dunn, and L. Zhang. Agentless: Demystifying llm-based software engineering agents. arXiv preprint, 2024.

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`.

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.



孙耀峰
王瑶辉
于毅
张易超
施亦凡
熊一良
何颖
卞一石
汪一松
谭一轩
马一阳*
刘一元
郭永强
欧渊
王宇端
宫越
邹宇恒
何宇嘉
熊云帆
罗宇翔
尤宇翔
刘宇轩
周宇阳
朱宇兴
黄燕萍
李瑶辉
郑毅
朱煜辰
马云显
唐颖
查宇坤
严玉婷
任振中
任泽辉
沙张立
傅哲
徐哲安
谢振达
张正言
郝哲文
马志成
颜志刚
吴志宇
顾自辉
朱子嘉
刘子君*
李子林
谢子伟
宋子阳
潘子正
黄真
许志鹏
张忠宇
张震

在每个角色中，作者按名字的字母顺序列出。带有 * 标记的名字表示已离开我们团队的个人。

## 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.