

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

DeepSeek-AI

research@deepseek.com

## Abstract

우리는 첫 번째 세대 추론 모델인 DeepSeek-R1-Zero 및 DeepSeek-R1를 소개한다. DeepSeek-R1-Zero는 대규모 Reinforcement Learning (RL)만을 사용하여, Supervised Fine-Tuning (SFT) 없이 학습된 모델로서 사전 단계가 없었음에도 주목할 만한 추론 능력을 보인다. RL 과정에서 DeepSeek-R1-Zero는 강력하고 흥미로운 여러 추론 행동들을 자연스럽게 습득하였다. 그러나 이 모델은 가독성이 낮고 언어가 혼합되는 등의 문제를 겪는다. 이러한 문제를 해결하고 추론 성능을 더욱 향상시키기 위해, 우리는 DeepSeek-R1를 제안한다. 이는 Multi-Stage Training 및 Cold-Start 데이터를 RL 전에 적용한 모델이다. DeepSeek-R1는 추론 작업에서 OpenAI-o1-1217과 유사한 성능을 달성한다. 연구 커뮤니티를 지원하기 위해, 우리는 DeepSeek-R1-Zero, DeepSeek-R1 그리고 DeepSeek-R1로부터 Qwen과 Llama 기반으로 Distillation된 여섯 가지 Dense 모델(1.5B, 7B, 8B, 14B, 32B, 70B)을 오픈 소스로 공개한다.

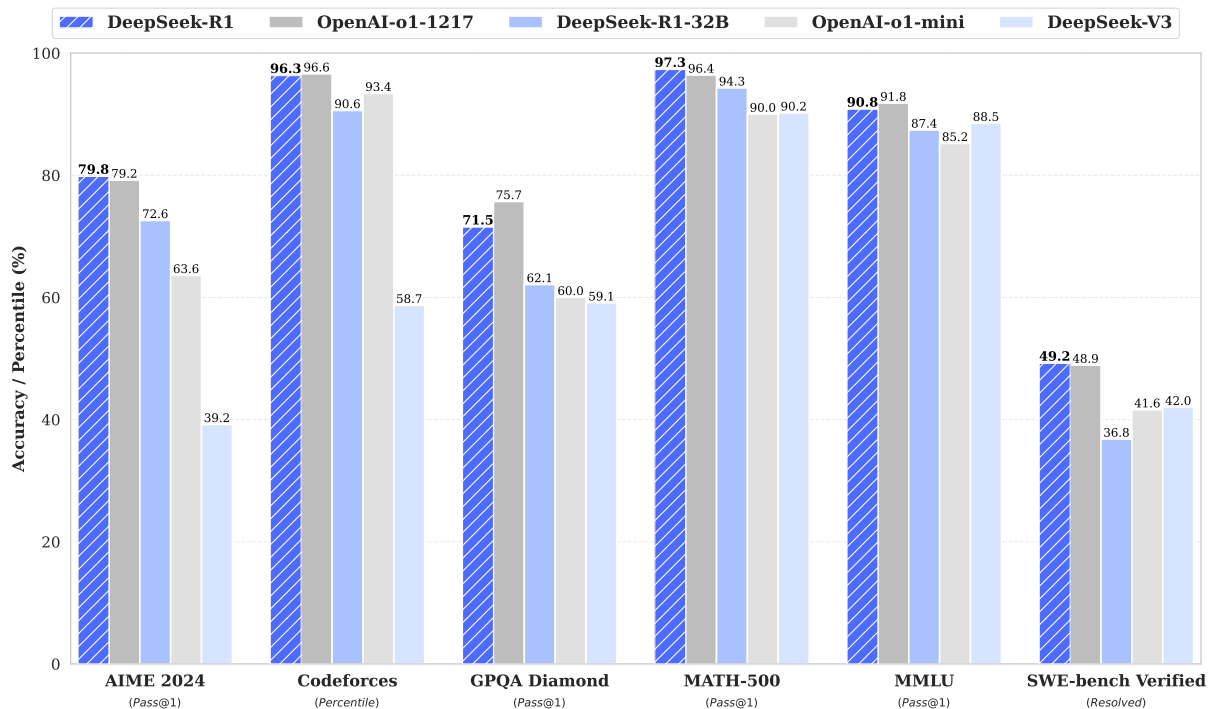


Figure 1 | Benchmark performance of DeepSeek-R1.

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
1.1	Contributions . . . . .	4
1.2	Summary of Evaluation Results . . . . .	4
<b>2</b>	<b>Approach</b>	<b>5</b>
2.1	Overview . . . . .	5
2.2	DeepSeek-R1-Zero: Reinforcement Learning on the Base Model . . . . .	5
2.2.1	Reinforcement Learning Algorithm . . . . .	5
2.2.2	Reward Modeling . . . . .	5
2.2.3	Training Template . . . . .	6
2.2.4	Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero . . . . .	7
2.3	DeepSeek-R1: Reinforcement Learning with Cold Start . . . . .	9
2.3.1	Cold Start . . . . .	9
2.3.2	Reasoning-oriented Reinforcement Learning . . . . .	10
2.3.3	Rejection Sampling and Supervised Fine-Tuning . . . . .	10
2.3.4	Reinforcement Learning for all Scenarios . . . . .	11
2.4	Distillation: Empower Small Models with Reasoning Capability . . . . .	11
<b>3</b>	<b>Experiment</b>	<b>11</b>
3.1	DeepSeek-R1 Evaluation . . . . .	12
3.2	Distilled Model Evaluation . . . . .	14
<b>4</b>	<b>Discussion</b>	<b>14</b>
4.1	Distillation v.s. Reinforcement Learning . . . . .	14
4.2	Unsuccessful Attempts . . . . .	15
<b>5</b>	<b>Conclusion, Limitations, and Future Work</b>	<b>15</b>
<b>A</b>	<b>Contributions and Acknowledgments</b>	<b>20</b>

## 1. Introduction

최근 들어 Large Language Models (LLMs)은 빠른 속도로 반복 및 진화를 거치며 (Anthropic, 2024; Google, 2024; OpenAI, 2024a), Artificial General Intelligence (AGI)에 한 걸음 더 가까워지고 있다.

최근에는 Post-Training이 전체 학습 파이프라인에서 중요한 구성 요소로 주목받고 있다. 이는 Pre-Training에 비해 상대적으로 적은 계산 자원으로도 추론 작업의 정확도를 높이고, 사회적 가치에 부합하며, 사용자 선호도에 적응할 수 있음을 보여주었다. 추론 능력 측면에서 OpenAI의 o1 (OpenAI, 2024b) 시리즈 모델은 Chain-of-Thought 추론 과정을 길게 설정하여 Inference-Time Scaling을 최초로 도입하였다. 이 접근 방식은 수학, 코딩, 과학적 추론 등 다양한 추론 작업에서 큰 성능 향상을 이루었다. 하지만 효과적인 Test-Time Scaling을 달성하는 문제는 여전히 연구 커뮤니티의 미해결 과제로 남아있다. 여러 선행 연구에서는 Process-based Reward Models (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023), Reinforcement Learning (Kumar et al., 2024), Monte Carlo Tree Search, Beam Search 등 탐색 알고리즘 (Feng et al., 2024; Trinh et al., 2024; Xin et al., 2024)을 포함한 다양한 접근법을 시도해왔다. 그러나 이들 방법 중 어느 것도 OpenAI의 o1 시리즈 모델에 필적하는 일반 추론 성능을 달성하지는 못했다.

본 논문에서는 순수 Reinforcement Learning (RL)만을 사용하여 언어 모델의 추론 능력을 향상하는 첫 번째 시도를 수행한다. 우리는 어떠한 Supervised 데이터도 사용하지 않고, 순수 RL 과정을 통해 LLM이 자체적으로 진화하며 추론 능력을 학습할 수 있는 잠재력을 탐구한다. 구체적으로, 우리는 DeepSeek-V3-Base 모델을 기반으로 GRPO (Shao et al., 2024)를 활용하여 추론 성능을 개선하기 위한 RL을 수행한다. 이 과정에서 DeepSeek-R1-Zero는 자연스럽게 강력하고 흥미로운 여러 추론 행동들을 습득하였다. 수천 번의 RL 스텝 후, DeepSeek-R1-Zero는 추론 벤치마크에서 매우 우수한 성능을 보인다. 예를 들어, AIME 2024에서 pass@1 점수가 15.6%에서 71.0%로 상승하고, Majority Voting을 적용하면 86.7%로 추가 상승하여 OpenAI-o1-0912와 동등한 성능을 달성한다.

그러나 DeepSeek-R1-Zero는 가독성이 낮고 언어가 혼합되는 등의 문제를 겪는다. 이러한 문제를 해결하고 추론 성능을 더욱 향상시키기 위해, 우리는 Cold-Start 데이터의 소량 사용과 Multi-Stage Training 파이프라인을 결합한 DeepSeek-R1를 제안한다. 구체적으로, 우리는 먼저 수천 개의 Cold-Start 데이터를 수집하여 DeepSeek-V3-Base 모델을 미세 조정된 뒤, DeepSeek-R1-Zero와 유사한 추론 지향 RL을 수행한다. RL 과정이 수렴에 가까워졌을 때, RL 체크포인트에서 Rejection Sampling을 통해 새로운 SFT 데이터를 생성하고, writing, factual QA, self-cognition 등 여러 도메인에서 DeepSeek-V3의 Supervised 데이터를 결합해 DeepSeek-V3-Base 모델을 재 학습시킨다. 그 후 이 새로운 데이터로 Fine-Tuning된 체크포인트에 대해, 모든 시나리오에 대한 Prompt를 고려하는 추가적인 RL 과정을 거친다. 이 단계들을 통해 얻어진 최종 체크포인트를 DeepSeek-R1라 하며, 이는 OpenAI-o1-1217에 필적하는 성능을 보인다.

또한 우리는 DeepSeek-R1로부터 소규모 Dense 모델로의 Distillation을 추가적으로 탐구하였다. Qwen2.5-32B (Qwen, 2024b)를 기본 모델로 사용하여 DeepSeek-R1로부터 직접 Distillation을 수행한 결과, 동일 모델에 RL을 적용하는 것보다 더 우수한 성능을 얻을 수 있었다. 이는 더 큰 Base 모델이 발견한 추론 패턴이 추론 성능 향상에 핵심적인 역할을 함을 시사한다. 우리는 이러한 Distilled Qwen과 Llama (Dubey et al., 2024) 시리즈를 오픈소스로 공개한다. 특히, Distilled 14B 모델은 최신 오픈소스인 QwQ-32B-Preview (Qwen, 2024a)를 큰 폭으로 능가하며, Distilled 32B와 70B 모델은 Dense 모델 중에서도 추론 벤치마크에서 새로운 기록을 세운다.

## 1.1. Contributions

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

- 우리는 Supervised Fine-Tuning (SFT)을 사전 단계로 사용하지 않고, Base 모델에 직접 RL을 적용한다. 이 접근 방식은 복잡한 문제를 해결하기 위해 Chain-of-Thought (CoT)를 탐색하도록 모델을 유도하며, 그 결과 DeepSeek-R1-Zero가 탄생하였다. DeepSeek-R1-Zero는 Self-Verification, Reflection, 긴 CoT 생성 등의 능력을 보이며, 이는 연구 커뮤니티에 중요한 이정표가 된다. 특히, 이는 SFT 없이 순수 RL만으로 LLM의 추론 능력을 유도할 수 있음을 검증한 최초의 공개 연구 사례로서, 향후 해당 분야의 발전에 큰 가능성을 시사한다.
- 우리는 DeepSeek-R1를 개발하기 위한 파이프라인을 제안한다. 이 파이프라인은 더욱 향상된 추론 패턴을 발견하고 인간의 선호도에 부합하기 위해 두 번의 RL 단계를 포함하며, 모델의 추론과 비추론 능력에 시드를 제공하는 두 번의 SFT 단계도 포함한다. 우리는 이 파이프라인이 산업계에서 더 나은 모델을 개발하는 데 도움이 될 것으로 기대한다.

### Distillation: Smaller Models Can Be Powerful Too

- 우리는 대규모 모델의 추론 패턴이 보다 작은 모델로 Distillation되어, 작은 모델에서 RL을 수행했을 때 발견되는 추론 패턴보다 우수한 성능을 낼 수 있음을 입증한다. 공개되는 DeepSeek-R1와 그 API는 연구 커뮤니티가 향후 더 나은 소형 모델을 Distillation하는 데 도움이 될 것이다.
- DeepSeek-R1가 생성한 추론 데이터를 활용해, 연구 커뮤니티에서 널리 사용되는 여러 Dense 모델을 Fine-Tuning하였다. 평가 결과, Distillation된 소형 Dense 모델이 벤치마크에서 탁월한 성능을 보였다. 예컨대, DeepSeek-R1-Distill-Qwen-7B는 AIME 2024에서 55.5%를 달성하여 QwQ-32B-Preview를 상회하였으며, DeepSeek-R1-Distill-Qwen-32B는 AIME 2024에서 72.6%, MATH-500에서 94.3%, LiveCodeBench에서 57.2%를 기록한다. 이는 이전 오픈소스 모델보다 크게 앞서는 성능이며, o1-mini와 견줄 만한 수준이다. 우리는 Qwen2.5와 Llama3 시리즈를 기반으로 Distillation된 1.5B, 7B, 8B, 14B, 32B, 70B 체크포인트를 커뮤니티에 오픈소스로 공개한다.

## 1.2. Summary of Evaluation Results

- **Reasoning tasks:** (1) DeepSeek-R1는 AIME 2024에서 Pass@1 기준 79.8%를 달성하여 OpenAI-o1-1217을 근소하게 상회한다. MATH-500에서 97.3%의 인상적인 점수를 기록하며, 이는 OpenAI-o1-1217과 대등한 수준이며 다른 모델들을 크게 능가한다. (2) 코딩 관련 과제에서 DeepSeek-R1는 Codeforces에서 2,029 Elo 레이팅을 달성하여, 대회에 참가한 인간 참가자의 96.3%를 상회하는 Expert 수준의 코드 대회 성능을 보인다. 엔지니어링 관련 작업에서도, DeepSeek-R1는 DeepSeek-V3보다 약간 더 우수한 성능을 보이며 실제 개발자 업무에 도움을 줄 수 있을 것으로 기대된다.
- **Knowledge:** MMLU, MMLU-Pro, GPQA Diamond와 같은 벤치마크에서 DeepSeek-R1는 각각 90.8%, 84.0%, 71.5%의 점수를 달성하며, DeepSeek-V3에 비해 현저히 향상된 결과를 보인다. 이 분야에서 OpenAI-o1-1217의 성능에는 약간 미치지 못하지만, DeepSeek-R1는 여타 비공개 모델들을 상회하며 교육 분야 과제에서 경쟁력을 입증한다. Factual 벤치마크 SimpleQA에서도 DeepSeek-R1는 DeepSeek-V3를 뛰어넘어, 사실 기반 질의 처리에 대한 우수한 능력을 보인다. 비슷한 경향으로 OpenAI-o1이 해당 벤치마크에서 4o를 상회한다.
- **Others:** DeepSeek-R1는 창의적 글쓰기, 일반 질의응답, 편집, 요약 등 광범위한 작업에서도 우수한 성능을 발휘한다. AlpacaEval 2.0에서 87.6%의 Length-Controlled Win-Rate를, ArenaHard에서 92.3%의 Win-Rate를 기록하며, 비시험 지향적 질의에서도 높은 지능적 처리를 보여준다. 또한 DeepSeek-R1는 Long-Context 이해가 필요한 작업에서도 DeepSeek-V3보다 훨씬 뛰어난 성능을 발휘하며, Long-Context 벤치마크에서 크게 개선된 결과를

보인다.

## 2. Approach

### 2.1. Overview

이전 연구들은 모델 성능을 향상하기 위해 대량의 Supervised 데이터를 활용해 왔다. 본 연구에서는 Supervised Fine-Tuning (SFT)을 Cold Start로 사용하지 않고도 대규모 Reinforcement Learning (RL)을 통해 추론 능력을 크게 향상시킬 수 있음을 보인다. 더 나아가, 소량의 Cold-Start 데이터를 활용함으로써 성능을 추가로 높일 수 있음을 시연한다. 이후 섹션에서는 (1) SFT 데이터를 전혀 사용하지 않고 Base 모델에 직접 RL을 적용한 DeepSeek-R1-Zero, (2) 수천 개의 긴 Chain-of-Thought (CoT) 예시로 미세 조정된 체크포인트에서 RL을 시작하는 DeepSeek-R1, (3) DeepSeek-R1의 추론 능력을 소형 Dense 모델로 Distillation하는 과정을 소개한다.

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

Reinforcement Learning은 이전 연구 (Shao et al., 2024; Wang et al., 2023)를 통해 추론 과제에서 높은 효과를 보인 바 있다. 그러나 이들 연구는 대부분 Supervised 데이터에 크게 의존하였으며, 이러한 데이터는 수집에 많은 시간이 소요된다. 본 절에서는 **어떠한 Supervised 데이터도 사용하지 않고** 순수한 RL 과정을 통해 LLM이 자체적으로 추론 능력을 학습할 수 있는 잠재력을 탐구한다. 우리는 먼저 RL 알고리즘의 간단한 개요를 살펴본 후, 흥미로운 결과들을 제시하며 연구 커뮤니티에 유용한 통찰을 제공하고자 한다.

#### 2.2.1. Reinforcement Learning Algorithm

**Group Relative Policy Optimization** RL의 학습 비용을 절감하기 위해, 우리는 Group Relative Policy Optimization (GRPO) (Shao et al., 2024)를 채택한다. 이는 일반적으로 Policy 모델과 동일한 크기의 Critic 모델을 사용하지 않고, 대신 그룹 단위로 산출된 점수를 활용하여 Baseline를 추정한다.

구체적으로, 각 질문  $q$ 에 대해, GRPO는 이전 Policy  $\pi_{\theta_{old}}$ 에서 출력  $\{o_1, o_2, \dots, o_G\}$ 를 그룹으로 샘플링한 뒤, 다음 목적함수를 최대화하도록 Policy 모델  $\pi_{\theta}$ 를 최적화한다:

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

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

여기서  $\epsilon$ 과  $\beta$ 는 하이퍼파라미터이며,  $A_i$ 는 각 그룹 내 출력들에 대응하는 보상  $\{r_1, r_2, \dots, r_G\}$ 를 사용해 다음과 같이 계산되는 Advantage이다.

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

#### 2.2.2. Reward Modeling

리워드는 학습 과정에서 트레이닝 시그널의 근원이 되며, 이는 RL의 최적화 방향을 결정짓는다. DeepSeek-R1-Zero를 학습하기 위해, 우리는 주로 두 종류의 리워드로 구성된 Rule-based Reward 시스템을 채택하였다.

- **Accuracy rewards:** 정확도 보상 모델(Accuracy Reward Model)은 응답이 정답인지 여부를 평가한다. 예를 들어, 결정적인 결과가 주어지는 수학(Math) 문제의 경우, 모델은 최종 정답을 특정 형식(예: 박스 안)에 작성해야 하며, 이를 통해 규칙 기반으로 정답 여부를 신뢰도 높게 검증할 수 있다. 마찬가지로, LeetCode 문제에서는 사전에 정의된 테스트 케이스를 활용하여 컴파일러(Compiler)가 피드백을 제공할 수 있다.
- **Format rewards:** 정확도 보상 모델(Accuracy Reward Model) 외에도, 우리는 형식 보상 모델(Format Reward Model)을 적용하여 모델이 자신의 사고 과정을 '`<think>`' 및 '`</think>`' 태그 사이에 작성하도록 강제한다.

DeepSeek-R1-Zero를 개발하는 과정에서 Outcome 또는 Process 기반 Neural Reward Model은 적용하지 않았다. 이는 대규모 RL 과정에서 Neural Reward Model이 Reward Hacking 문제를 일으킬 수 있으며, Reward Model을 재학습하기 위해 추가적인 학습 자원이 필요하고 전체 학습 파이프라인을 복잡하게 만들기 때문이다.

### 2.2.3. Training Template

DeepSeek-R1-Zero를 학습하기 위해, 우리는 먼저 Base 모델이 특정 지침을 따를 수 있도록 유도하는 간단한 템플릿을 설계한다. Table 1에 나타난 바와 같이, 이 템플릿은 DeepSeek-R1-Zero가 먼저 추론 과정을 생성한 후 최종 답안을 도출하도록 요구한다. 우리는 모델이 RL 과정에서 자연스럽게 발전하는 모습을 정확히 관찰할 수 있도록, 반성적(Reflective) 추론을 강제하거나 특정 문제 해결 전략을 유도하는 등의 내용적 편향을 피하고, 구조적 형식에 대한 제한만 두도록 설계하였다.

---

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-Zero를 위한 템플릿. **prompt**는 학습 과정에서 특정 추론 질문으로 대체된다.

## 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.

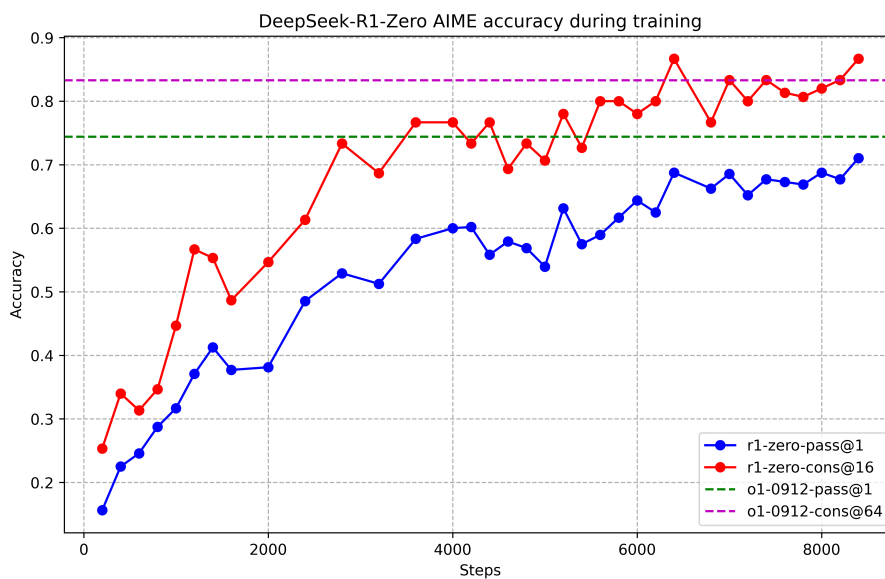


Figure 2 | 학습 과정에서의 DeepSeek-R1-Zero의 AIME 정확도. 각 질문당 16개의 응답을 샘플링하여 전체 평균 정확도를 계산함으로써 안정적인 평가를 보장한다.

**Performance of DeepSeek-R1-Zero** Figure 2는 RL 학습 과정 전반에서 AIME 2024 벤치마크에 대한 DeepSeek-R1-Zero의 성능 변화를 보여준다. 그래프에서 볼 수 있듯이, DeepSeek-R1-Zero는 RL 학습이 진행됨에 따라 안정적이면서도 일관된 성능 향상을 보인다. 특히 AIME 2024에서의 평균 pass@1 점수는 초기 15.6%에서 71.0%로 크게 상승하며, 이는 OpenAI-o1-0912에 상응하는 수준을 달성한다. 이러한 큰 폭의 향상은 시간에 따라 모델 성능을 최적화하는 RL 알고리즘의 효과를 잘 보여준다.

Table 2는 다양한 추론 관련 벤치마크에서 DeepSeek-R1-Zero와 OpenAI의 o1-0912 모델을 비교 분석한 결과이다. 이 연구 결과는 DeepSeek-R1-Zero가 어떠한 Supervised Fine-Tuning 데이터도 사용하지 않고 순수 RL을 통해 견고한 추론 능력을 확보할 수 있음을 시사한다. 이는 모델이 RL만으로도 학습과 일반화가 가능하다는 점에서 주목할 만한 성과이다. 또한 DeepSeek-R1-Zero의 성능은 Majority Voting 기법을 적용함으로써 추가적으로 향상될 수 있다. 예를 들어 AIME 벤치마크에서 Majority Voting을 적용했을 때, DeepSeek-R1-Zero의 성능은 71.0%에서 86.7%로 상승하며, 이는 OpenAI-o1-0912를 상회한다. 이처럼 DeepSeek-R1-Zero는 Majority Voting의 사용 여부와 관계없이 경쟁력 있는 성능을 달성하여, 추론 과제에서의 강력한 기초 역량과 향후 발전 가능성을 보여준다.

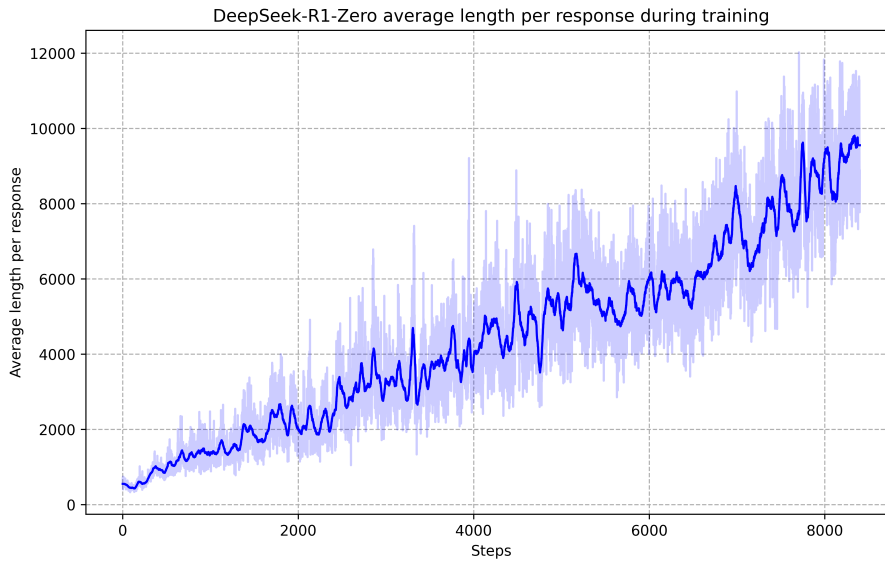


Figure 3 | RL 과정에서의 DeepSeek-R1-Zero의 학습 데이터에 대한 평균 응답 길이. DeepSeek-R1-Zero는 자연스럽게 더욱 많은 Thinking Time을 활용하여 추론 과제를 해결하는 방법을 학습한다.

**Self-evolution Process of DeepSeek-R1-Zero** DeepSeek-R1-Zero의 자기 진화(Self-Evolution) 과정은 RL이 모델의 추론 능력을 자율적으로 향상시킬 수 있음을 보여주는 흥미로운 사례이다. Base 모델로부터 직접 RL을 시작함으로써, Supervised Fine-Tuning 단계의 영향을 배제한 상태에서 모델의 발전 양상을 면밀히 관찰할 수 있다. 이를 통해 모델이 시간 경과에 따라 복잡한 추론 과제를 처리하는 능력이 어떻게 개선되는지를 명확히 파악할 수 있다.

Figure 3에 나타나 있듯이, DeepSeek-R1-Zero의 Thinking Time(즉, 응답 생성 시 소모되는 토큰 수)은 학습이 진행되는 동안 꾸준히 증가하는 추세를 보인다. 이 같은 향상은 외부적인 조정의 결과가 아니라, 모델 내부에서 자연스럽게 이루어지는 발전으로 볼 수 있다. DeepSeek-R1-Zero는 점차 더 긴 Test-Time Computation을 활용함으로써 복잡도가 높은 추론 과제를 해결하는 능력을 획득한다. 이는 수백 개에서 수천 개에 달하는 추론 토큰을 생성하여 보다 깊이 있는 사고 과정을 탐색하고 정제해나가는 과정을 통해 가능해진다.

이와 같은 자기 진화 과정에서 특히 주목할 만한 점은, Test-Time Computation이 늘어남에 따라 반성적(Reflection) 사고나 대안적 문제 해결 방식을 탐색하는 등 고도화된 행동 양상이 자연스럽게 드러난다는 것이다. 이러한 행동들은 명시적으로 프로그래밍된 것이 아니라, RL 환경과의 상호 작용을 통해 자발적으로 나타난다. 이는 DeepSeek-R1-Zero가 더욱 복잡한 과제를 효율적으로 해결할 수 있도록 추론 능력을 크게 강화하며, 향후 더욱 발전된 추론 모델로 나아갈 수 있는 잠재력을 보여준다.

**Aha Moment of DeepSeek-R1-Zero** DeepSeek-R1-Zero의 학습 과정에서 관찰된 한 가지 흥미로운 현상은 이른바 “Aha Moment”이다. Table 3에 나타난 예시와 같이, 이는 모델이 학습 중간 단계에서 스스로의 초기 접근 방식을 재평가하고, 더 많은 Thinking Time을 할당하기 시작하는 순간을 의미한다. 이러한 행동 양상은 모델이 점차 발전된 추론 능력을 갖추고 있음을 보여줄 뿐 아니라, RL을 통해 예상치 못한 정교한 결과물이 도출될 수 있다는 사실을 잘 보여준다.

이는 모델뿐만 아니라, 모델을 관찰하는 연구자들에게도 일종의 “Aha Moment”로 작용한다. 즉, 문제 해결 방법을 명시적으로 가르치는 대신, 적절한 보상을 제공함으로써 모델이 자율적으로 고급 문제 해결 전략을 터득한다는 점은 RL의 강력함과 아름다움을 상징적으로 드러낸다. 이러한



---

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 ...

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

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 ...

We started with the equation:

$$\sqrt{a - \sqrt{a+x}} = x$$

First, let's square both sides:

$$a - \sqrt{a+x} = x^2 \implies \sqrt{a+x} = a - x^2$$

Next, I could square both sides again, treating the equation: ...

...

---

Table 3 | 중간 버전의 DeepSeek-R1-Zero에서 나타난 흥미로운 “aha moment” 예시. 모델이 인격화된(anthropomorphic) 어조로 사고 과정을 재평가하는 방법을 학습한다. 이는 RL의 힘과 아름다움을 직접 경험할 수 있게 해주는 또 하나의 “aha moment”이기도 하다.

“Aha Moment”는 인공지능 시스템에서 RL이 새로운 수준의 지능을 열어갈 수 있음을 시사하며, 미래에는 더욱 자율적이고 적응적인 모델이 등장할 가능성을 시사한다.

**Drawback of DeepSeek-R1-Zero** DeepSeek-R1-Zero는 강력한 추론 능력을 발휘하고, 예기치 못한 강력한 추론 행동들을 자율적으로 발전시키지만, 몇 가지 문제점을 안고 있다. 예컨대, DeepSeek-R1-Zero는 가독성이 낮거나 언어가 혼합되는 등의 문제에 부딪힌다. 우리는 오픈 커뮤니티와 공유 가능한, 보다 가독성이 높은 추론 과정을 생성하고자, Human-Friendly Cold-Start 데이터를 결합한 RL 방식을 사용하는 DeepSeek-R1를 탐색한다.

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

DeepSeek-R1-Zero의 유망한 결과에 영감을 받아, 다음과 같은 두 가지 자연스러운 의문이 제기된다: 1) 소량의 고품질 데이터를 Cold Start로 사용함으로써 추론 성능을 추가로 향상시키거나 학습 수렴 속도를 가속할 수 있을까? 2) 명확하고 일관된 Chain of Thought(CoT)를 생성하는 동시에 높은 범용 역량을 갖춘, 사용자 친화적인 모델을 어떻게 학습시킬 수 있을까? 이러한 질문에 답하기 위해, 우리는 DeepSeek-R1를 학습하기 위한 파이프라인을 고안하였다. 이 파이프라인은 크게 네 단계로 구성되며, 다음과 같은 순서를 따른다.

### 2.3.1. Cold Start

DeepSeek-R1-Zero와 달리, Base 모델에서 RL을 직접 시작할 때 발생하는 불안정한 초기 Cold Start 단계를 방지하기 위해, DeepSeek-R1에서는 길이가 긴 CoT 데이터를 소량 수집하여 모델을 미세 조정된 뒤, 이를 초기 RL Actor로 사용하였다. 이러한 데이터를 수집하기 위해, 우리는 다음과 같은 여러 가지 접근을 탐색하였다: 예시로 긴 CoT를 포함한 Few-Shot Prompting, Reflective하고 Verification을 포함한 상세 답변을 모델에게 직접 생성하도록 하는 방법, DeepSeek-R1-Zero의 출력을 가독성 높은 형식으로 재수집하는 방법, 그리고 인간 주석자(Human Annotator)를 통한 후처리를 거쳐 결과물을 정제하는 작업 등이다.

본 논문에서는 수천 개 규모의 Cold-Start 데이터를 수집하여 DeepSeek-V3-Base 모델을 미세 조정함으로써 RL의 시작점을 마련하였다. DeepSeek-R1-Zero와 비교했을 때, Cold Start 데이터를 활용함으로써 얻을 수 있는 이점은 다음과 같다:

- **Readability:** DeepSeek-R1-Zero의 핵심적인 한계 중 하나는 그 내용이 일반적으로 읽기 적합한 형태가 아니라는 점이다. 예컨대, 여러 언어가 혼합되어 있거나, 사용자 입장에서 답변을 명확히 구분할 수 있는 Markdown 형식 등이 부족한 경우가 발생한다. 이에 반해 DeepSeek-R1의 Cold-Start 데이터를 만들 때는 각 응답의 끝에 요약물을 포함하고, 읽기 편하지 않은 응답을 제거하는 등 가독성을 높이는 패턴을 설계하였다. 구체적으로, 출력 형식을

`|special_token|<reasoning_process>|special_token|<summary>`

로 정의하였으며, 여기서 <reasoning\_process>는 질의에 대한 CoT를 나타내고, <summary>는 추론 결과를 요약하기 위한 용도로 활용된다.

- **Potential:** 인간의 사전 지식을 반영하여 Cold-Start 데이터를 신중하게 설계함으로써, DeepSeek-R1-Zero에 비해 더 나은 성능을 얻을 수 있음을 관찰하였다. 우리는 이러한 반복 학습(iterative training) 방식이 추론 모델에게 보다 적합한 방법이라 판단한다.

### 2.3.2. Reasoning-oriented Reinforcement Learning

Cold-Start 데이터로 DeepSeek-V3-Base 모델을 미세 조정된 뒤, DeepSeek-R1-Zero와 동일한 대규모 Reinforcement Learning 과정을 적용한다. 이 단계는 코딩, 수학, 과학, 논리 추론 등 명확한 해답이 존재하는 문제를 다루는 추론 집약적 과제에서 모델의 추론 능력을 더욱 강화하는데 집중한다. 학습 과정에서 DeepSeek-R1-Zero와 마찬가지로 CoT가 종종 여러 언어가 혼합된 형태로 나타난다. 이를 완화하기 위해, 우리는 RL 과정에서 Language Consistency Reward를 도입하였으며, 이는 CoT에 나타나는 목표 언어 단어의 비율로 계산된다. 모델 성능에 대한 Ablation 실험 결과, 이러한 정렬(Alignment)이 모델의 성능을 약간 떨어뜨릴 수 있음을 보였으나, 인간 선호도 측면에서 더 가독성이 높은 결과를 제공한다. 최종적으로, 추론 과제의 정확도와 언어 일관성에 대한 보상 값을 단순 합산하여 최종 보상을 구성한다. 이후 모델이 추론 과제에서 수렴(Convergence)에 도달할 때까지 미세 조정된 모델에 대해 RL 학습을 진행한다.

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

추론 지향 RL이 수렴하면, 우리는 해당 체크포인트를 활용하여 다음 라운드를 위한 SFT(Supervised Fine-Tuning) 데이터를 수집한다. 초기 Cold-Start 데이터가 주로 추론에 집중되어 있었다면, 이 단계에서는 쓰기(Writer)나 Role-Playing과 같은 범용적 기능 강화를 위해 다른 도메인의 데이터를 추가로 포함한다. 구체적으로, 우리는 데이터를 생성하고 모델을 미세 조정(Fine-Tuning)하는 과정을 아래와 같이 진행한다.

**Reasoning data** 우리는 이전 RL 학습 단계에서 얻은 체크포인트를 이용하여, Rejection Sampling을 수행함으로써 추론 프롬프트와 추론 과정을 구축하였다. 이전 단계에서는 주로 Rule-based Reward를 사용해 평가가 가능한 데이터만을 포함했으나, 이번 단계에서는 DeepSeek-V3를 활용해 정답지(Ground-Truth)와 모델 예측을 함께 입력하여 판별하는 Generative Reward Model을 사용하는 일부 데이터도 추가로 포함하여 데이터셋을 확장한다. 또한 모델 출력이 혼합 언어나 긴 문단, 코드 블록을 포함해 가독성이 떨어지는 경우가 있어, 해당 Chain-of-Thought는 필터링하였다. 각 프롬프트별로 다수의 응답을 샘플링한 뒤, 정답에 해당하는 응답만을 남긴다. 최종적으로 약 60만(600k) 개의 추론 관련 학습 샘플을 수집하였다.

**Non-Reasoning data** Writing, Factual QA, Self-cognition, Translation 등 추론과 직접 연관되지 않은 데이터의 경우, DeepSeek-V3 파이프라인을 채택하고 DeepSeek-V3의 일부 SFT 데이

터셋을 재사용한다. 특정 비추론 과제에 대해서는, 질문에 답하기 전에 DeepSeek-V3를 통해 잠재적 Chain-of-Thought를 생성하도록 Prompting하는 방식도 시도하였다. 반면, 예컨대 “hello” 처럼 간단한 질의에 대해서는 별도의 CoT를 제공하지 않는다. 결국 우리는 추론과 무관한 약 20만(200k) 개의 학습 샘플을 추가로 수집하였다.

이렇게 수집된 약 80만(800k) 개의 샘플로 구성된 데이터셋을 이용해, 우리는 DeepSeek-V3-Base 모델을 두 Epoch에 걸쳐 미세 조정(Fine-Tuning)하였다.

#### 2.3.4. Reinforcement Learning for all Scenarios

모델을 인간 선호도에 더욱 맞추기 위해, 우리는 모델의 추론 능력을 개선함과 동시에, 도움이 되고 해가 없도록(Helpfulness & Harmlessness) 만드는 것을 목표로 하는 2차 RL 단계를 수행하였다. 구체적으로, 우리는 보상(Reinforcement) 시그널과 다양한 Prompt 분포를 결합하여 모델을 학습 시킨다.

추론 데이터의 경우, DeepSeek-R1-Zero와 동일한 방법론을 따라 수학(Math), 코딩(Code), 논리(Logical Reasoning) 영역에서 Rule-based Reward를 사용하여 학습을 유도한다. 반면, 일반 데이터에 대해서는 인간 선호도를 반영하기 위해 Reward 모델을 활용한다. 이는 DeepSeek-V3 파이프라인을 기반으로 하며, 유사한 Preference Pair 분포와 Training Prompt를 채택한다. Helpfulness 측면에서는 최종 요약(Final Summary)에만 초점을 맞추어, 해당 응답이 사용자에게 유익하고 관련성이 높은지를 평가하고, 내부 추론 과정에는 최소한의 간섭만 가하도록 설계하였다. Harmlessness 측면에서는 모델의 전체 응답(추론 과정과 요약 모두)을 평가함으로써, 생성 과정에서 발생할 수 있는 잠재적 위험, 편향, 해로운 콘텐츠 등을 발견하고 완화한다. 결과적으로, 이러한 보상 시그널과 다양한 데이터 분포를 결합함으로써, 추론 능력이 우수하면서도 도움이 되고 안전한 모델을 학습할 수 있게 된다.

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

보다 작은 모델들에게도 DeepSeek-R1과 유사한 추론 능력을 부여하기 위해, 우리는 DeepSeek-R1로부터 큐레이션된 80만(800k) 개의 샘플(\$2.3.3 참고)을 활용하여 Qwen (Qwen, 2024b) 및 Llama (AI@Meta, 2024) 등 오픈소스 모델을 직접 Fine-Tuning하였다. 이러한 간단한 Distillation 방식을 통해서도 소형 모델의 추론 능력이 크게 향상됨을 확인하였다. 본 논문에서 사용한 Base 모델로는 Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, Qwen2.5-14B, Qwen2.5-32B, Llama-3.1-8B, 그리고 Llama-3.3-70B-Instruct를 사용하였으며, Llama-3.3은 Llama-3.1보다 약간 더 우수한 추론 능력을 갖추고 있기에 선택하였다.

Distilled 모델의 경우, RL을 추가로 적용하면 모델 성능을 상당히 높일 수 있음에도 불구하고, 본 연구에서는 오직 SFT만 사용하고 RL 단계는 생략하였다. 이는 Distillation 기법의 효과를 직접적으로 입증하는 데 집중하기 위함이며, RL 단계의 추가 탐색은 더 넓은 연구 커뮤니티에 맡긴다.

### 3. Experiment

**Benchmarks** 우리는 MMLU (Hendrycks et al., 2020), MMLU-Redux (Gema et al., 2024), MMLU-Pro (Wang et al., 2024), C-Eval (Huang et al., 2023), 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<sup>1</sup>, LiveCodeBench (Jain et al., 2024) (2024-08 – 2025-01), Codeforces<sup>2</sup>, Chinese National High School Mathematics Olympiad (CNMO 2024)<sup>3</sup>, American Invitational

<sup>1</sup><https://aider.chat>

<sup>2</sup><https://codeforces.com>

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

Mathematics Examination 2024 (AIME 2024) (MAA, 2024) 등 다양한 벤치마크를 활용하여 모델을 평가한다. 또한 표준 벤치마크 외에도, 우리는 LLM을 평가자로 사용하는 오픈엔디드 생성 과제(Open-ended Generation Tasks)에서 모델 성능을 측정한다. 구체적으로, AlpacaEval 2.0 (Dubois et al., 2024)와 Arena-Hard (Li et al., 2024)의 원래 설정을 따르며, GPT-4-Turbo-1106을 쌍대(Pairwise) 비교의 판정자로 활용한다. 이때, Length Bias를 피하기 위해 최종 요약(Final Summary)만을 입력으로 제공한다. Distilled 모델의 경우, 우리는 AIME 2024, MATH-500, GPQA Diamond, Codeforces, LiveCodeBench에 대한 대표적인 결과를 보고한다.

**Evaluation Prompts** DeepSeek-V3 설정을 따르며, MMLU, DROP, GPQA Diamond, SimpleQA 등 표준 벤치마크는 simple-evals 프레임워크의 프롬프트를 사용하여 평가한다. MMLU-Redux의 경우, Zero-Eval Prompt 형식 (Lin, 2024)을 Zero-Shot 환경에서 적용한다. 원래 Few-Shot 설정인 MMLU-Pro, C-Eval, CLUE-WSC에 대해서는 Prompt를 약간 수정하여 Zero-Shot으로 전환하였다. 이는 Few-Shot CoT가 DeepSeek-R1의 성능에 부정적 영향을 미칠 수 있기 때문이다. 그 외의 데이터셋은 제작자가 제공한 기본 프롬프트와 원래 평가 프로토콜을 따른다. 코드 및 수학 벤치마크 측면에서, HumanEval-Mul 데이터셋은 Python, Java, C++, C#, JavaScript, TypeScript, PHP, Bash 등 8가지 주요 프로그래밍 언어를 포함한다. LiveCodeBench에 대한 모델 성능 평가는 CoT 형식으로 진행하며, 2024년 8월부터 2025년 1월 사이의 데이터를 사용한다. Codeforces 데이터셋은 10개 Div.2 대회의 문제와 전문가가 만든 테스트 케이스를 사용하여 평가한 뒤, 해당 결과로부터 예상 레이팅과 참가자 대비 백분율을 산출한다. SWE-Bench Verified 결과는 agentless 프레임워크 (Xia et al., 2024)를 통해 획득하였으며, AIDER 관련 벤치마크는 “diff” 형식을 사용해 측정한다. DeepSeek-R1의 출력은 각 벤치마크에서 최대 32,768개의 토큰으로 제한된다.

**Baselines** 우리는 DeepSeek-V3, Claude-Sonnet-3.5-1022, GPT-4o-0513, OpenAI-o1-mini, OpenAI-o1-1217 등 강력한 베이스라인들을 대상으로 종합적인 평가를 수행한다. OpenAI-o1-1217 API는 중국 본토에서 접근하기 까다롭기 때문에, 우리는 해당 모델의 성능을 공식 리포트에 기반해 인용한다. Distilled 모델의 경우, 오픈소스 모델 QwQ-32B-Preview (Qwen, 2024a)와도 비교한다.

**Evaluation Setup** 우리는 모델에 대해 최대 생성 길이를 32,768 토큰으로 설정한다. 긴 출력 추론 모델을 Greedy Decoding으로 평가할 경우 반복(Repetition) 경향이 증가하고, 체크포인트 간 편차가 크게 발생함을 관찰하였다. 따라서, 우리는 Pass@k 평가 (Chen et al., 2021)를 기본으로 사용하며, Non-Zero Temperature가 적용된 상태에서 Pass@1을 보고한다. 구체적으로, 샘플링 온도(Temperature)를 0.6, Top-p 값을 0.95로 설정하여 각 질문당  $k$ 개의 응답(통상적으로 4 ~ 64개, 테스트 세트 크기에 따라 다름)을 생성한다. 이후 각 응답의 정답 여부  $p_i$ 를 확인하고, Pass@1을

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

로 정의한다. 이를 통해 더 신뢰도 높은 성능 추정값을 얻을 수 있다. AIME 2024의 경우, 64개의 샘플을 사용하여 Consensus(majority vote) 결과를 보고하며, 이를 cons@64로 표기한다.

### 3.1. DeepSeek-R1 Evaluation

교육용 지식 벤치마크인 MMLU, MMLU-Pro, GPQA Diamond 등에서 DeepSeek-R1은 DeepSeek-V3 대비 우수한 성능을 보인다. 이러한 향상은 주로 STEM 분야 문항에 대한 정확도 향상에 기인하며, 이는 대규모 Reinforcement Learning으로 인해 크게 개선된 것으로 파악된다. 또한 긴 맥락 의존적 QA 과제인 FRAMES에서도 뛰어난 문서 분석 능력을 발휘하여, AI 기반 검색 및 데이터

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

Table 4 | DeepSeek-R1와 다른 대표 모델 간의 비교.

분석 작업에서의 추론 모델 활용 가능성을 시사한다. 사실 기반 벤치마크인 SimpleQA에서도 DeepSeek-R1는 DeepSeek-V3보다 높은 점수를 기록하며, 사실에 근거한 질의를 처리하는 능력을 보여준다. 이와 유사하게 OpenAI-ol은 해당 벤치마크에서 GPT-4o보다 앞서는 결과를 보인다. 반면 Chinese SimpleQA 벤치마크에서는 DeepSeek-V3보다 낮은 성능을 보이는데, 이는 Safety RL 이후 특정 질의에 대한 응답 거부가 주요 원인이다. Safety RL을 적용하지 않을 경우, DeepSeek-R1는 70%가 넘는 정확도를 달성할 수 있었다.

DeepSeek-R1는 포맷 지침 수행 능력을 평가하는 IF-Eval 벤치마크에서도 인상적인 결과를 기록하였다. 이는 Supervised Fine-Tuning(SFT)과 RL의 최종 단계에서 지시사항 준수 데이터를 포함한 학습 전략이 주효했던 것으로 추정된다. 또한 AlpacaEval2.0 및 ArenaHard 평가에서도 뛰어난 성능을 보이는데, 이는 글쓰기 및 Open-Domain QA 영역에서 DeepSeek-R1가 갖는 강점을 나타낸다. 특히 DeepSeek-R1가 DeepSeek-V3보다 현저히 앞선다는 점은, 대규모 RL이 추론 능력뿐 아니라 여러 도메인 전반에서의 성능까지 향상시키는 일반화 효과가 있음을 시사한다. 더불어 DeepSeek-R1가 생성하는 요약은 매우 간결한 편으로, ArenaHard에서 평균 689개의 토큰, AlpacaEval 2.0에서 평균 2,218자 정도를 사용한다. 이는 GPT 기반 평가에서 Length Bias가 개입되는 것을 방지함으로써, 다양한 과업에서 DeepSeek-R1가 안정적인 성능을 보임을 뒷받침한다.

수학 과제에서, DeepSeek-R1는 OpenAI-ol-1217과 견줄 만한 성능을 보이며, 다른 모델들을 큰 격차로 앞선다. 이와 유사한 양상은 LiveCodeBench, Codeforces와 같은 코딩 알고리즘 분야에서도 관찰되는데, 추론 중심 모델이 이러한 벤치마크에서 두드러진 우위를 보인다. 엔지니어링 지향 코딩 과제인 Aider에서는 OpenAI-ol-1217이 DeepSeek-R1를 능가하지만, SWE Verified에서는 양자가 유사한 수준을 유지한다. 현재 RL 학습 데이터가 매우 제한적임을 감안할 때, 추후 버전에서는 DeepSeek-R1의 엔지니어링 관련 성능이 한층 더 향상될 것으로 기대한다.

### 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	<b>1820</b>
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	<b>72.6</b>	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	<b>86.7</b>	<b>94.5</b>	<b>65.2</b>	<b>57.5</b>	1633

Table 5 | DeepSeek-R1로부터 Distillation된 모델과 기타 유사 모델을 추론 관련 벤치마크에서 비교.

Table 5에서 보이듯이, 단순히 DeepSeek-R1의 출력을 Distillation하는 것만으로도 DeepSeek-R1-7B(DeepSeek-R1-Distill-Qwen-7B, 이하 동일 약칭)는 GPT-4o-0513과 같은 비추론 모델보다 전반적으로 우수한 성능을 발휘한다. DeepSeek-R1-14B는 모든 평가 지표에서 QwQ-32B-Preview를 상회하며, DeepSeek-R1-32B와 DeepSeek-R1-70B 역시 대부분의 벤치마크에서 o1-mini를 크게 능가한다. 이는 Distillation이 가진 높은 잠재력을 시사한다. 또한, 우리는 이렇게 Distillation된 모델들에 대해 RL을 추가 적용했을 때 성능이 큰 폭으로 향상됨을 확인하였다. 다만 본 연구에서는 간단한 SFT Distillation을 적용한 모델의 결과만을 제시하며, 추후 연구에서는 RL 적용을 포함한 보다 심층적인 탐색이 가능할 것으로 본다.

## 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	<b>72.6</b>	<b>83.3</b>	<b>94.3</b>	<b>62.1</b>	<b>57.2</b>

Table 6 | Distilled 모델과 RL 모델의 추론 관련 벤치마크 비교.

섹션 3.2에서 확인할 수 있듯이, DeepSeek-R1을 Distillation한 결과, 소형 모델도 뛰어난 성능을 발휘할 수 있음이 드러났다. 그러나 여전히 한 가지 의문이 남는다. Distillation 없이 본 논문에서 언급한 대규모 RL 학습만으로도, 해당 모델이 유사한 성능을 달성할 수 있을까?

이 질문에 답하고자, 우리는 Qwen-32B-Base 모델을 수학(Math), 코딩(Code), STEM 데이터를 활용하여 1만(10K) 스텝 이상 대규모 RL 학습을 수행하였고, 이를 통해 DeepSeek-R1-Zero-Qwen-32B를 얻었다. Table 6에서 제시된 실험 결과에 따르면, 대규모 RL 학습 후 32B Base 모델은 QwQ-32B-Preview와 유사한 수준에 도달한다. 그러나 DeepSeek-R1을 Distillation한 DeepSeek-R1-Distill-Qwen-32B는 모든 벤치마크에서 DeepSeek-R1-Zero-Qwen-32B를 현저히 상회한다.

결과적으로 두 가지 결론을 도출할 수 있다. 첫째, 보다 강력한 모델을 소형 모델로 Distillation하는 전략은 우수한 성능을 이끌어낼 수 있는 반면, 본 논문에서 제시한 대규모 RL만으로 소형 모델을 학습하는 경우는 막대한 계산 자원을 요구함에도 Distillation 성능에 미치지 못할 수 있다. 둘째, Distillation 전략이 경제적이고 효과적이기는 하지만, 지능의 한계를 넘어서는 발전을 위해서는 여전히 더 강력한 Base 모델과 대규모 Reinforcement Learning이 필요할 것으로 보인다.

## 4.2. Unsuccessful Attempts

DeepSeek-R1의 초기 개발 단계에서, 우리는 여러 실패와 시행착오를 겪었다. 여기서는 그 경험을 공유함으로써 통찰을 제공하고자 하며, 이는 해당 접근들이 결코 유효하지 않다는 의미가 아님을 밝힌다.

**Process Reward Model (PRM)** PRM은 추론 과제를 해결하기 위한 더 나은 접근 방식을 모델에 유도하는 합리적인 방법 중 하나로 제안되어 왔다 (Lightman et al., 2023; Uesato et al., 2022; Wang et al., 2023). 그러나 실제 적용 과정에서, 우리는 PRM이 가진 세 가지 주요 제약을 발견하였다. 첫째, 일반적인 추론 문제에서 단계별(Step-by-Step) 과정을 명시적으로 정의하기가 어렵다. 둘째, 현재 진행 중인 중간 단계를 올바른지 판단하는 과정 자체가 쉽지 않다. 모델을 통한 자동화된 주석(Annotation)은 만족스러운 결과를 내기 어렵고, 수작업(Human Annotation)은 확장성 문제를 야기한다. 셋째, 일단 모델 기반 PRM을 도입하면 불가피하게 Reward Hacking (Gao et al., 2022) 문제가 발생하고, 이를 해결하기 위해서는 Reward 모델 재학습에 추가적인 학습 자원이 필요해지며, 전체 학습 파이프라인이 복잡해진다. 결론적으로, PRM은 모델이 생성한 Top-N 응답을 재순위(Reranking)하거나 Guided Search (Snell et al., 2024)를 보조하는 데는 유용하지만, 우리 실험에서처럼 대규모 Reinforcement Learning 과정에서 초래되는 추가 계산 비용과 비교했을 때 그 이점이 제한적임을 확인하였다.

**Monte Carlo Tree Search (MCTS)** AlphaGo (Silver et al., 2017b)와 AlphaZero (Silver et al., 2017a)에서 영감을 받아, 우리는 테스트 시점의 계산량(Compute) 확장을 위해 Monte Carlo Tree Search(MCTS)를 적용하는 방법을 탐색하였다. 이는 답안을 더 작은 단위로 나누어 모델이 해당 공간을 체계적으로 탐색할 수 있도록 하는 방식이다. 이를 구현하기 위해, 우리는 모델에게 문제 해결에 필요한 구체적 추론 단계를 태그(Tag) 형태로 여러 개 생성하도록 Prompting을 유도하였다. 학습 시에는 먼저 사전에 학습된 Value 모델로 MCTS를 수행하여 정답을 찾은 다음, 그 결과로 얻어진 질의-응답 쌍을 이용해 Actor 모델과 Value 모델을 함께 학습하는 과정을 반복하였다.

그러나 이러한 접근은 대규모 학습으로 확장하는 과정에서 여러 문제에 직면하였다. 첫째, 체스 등과 달리 토큰 생성(Token Generation)은 탐색 공간이 기하급수적으로 커서, 노드 확장을 제한적으로 설정해도 모델이 지역 최적해(Local Optima)에 갇히기 쉽다. 둘째, Value 모델이 탐색(서치) 과정의 각 단계를 안내하기 때문에, Value 모델의 정확도가 곧 생성물의 품질을 직접 좌우한다. 이는 정밀한 Value 모델을 학습하는 것이 본질적으로 어렵다는 점을 의미하며, 모델의 성능을 단계적으로 향상시키기가 쉽지 않다. AlphaGo가 Value 모델을 훈련하여 성능을 점진적으로 개선했던 핵심 원리는, 토큰 생성이라는 복잡성이 추가된 우리의 환경에서는 재현하기가 까다롭다.

결국 사전에 학습된 Value 모델을 활용해 추론 중(Inference Time) 성능을 높이는 것은 가능하나, Self-Search를 통해 모델 성능을 반복적으로 향상시키는 것은 여전히 해결 과제로 남아 있음을 확인하였다.

## 5. Conclusion, Limitations, and Future Work

본 연구에서는 Reinforcement Learning을 통한 모델의 추론 능력 향상 과정을 공유하였다. DeepSeek-R1-Zero는 Cold-Start 데이터를 사용하지 않고 순수 RL 방식으로 학습된 모델로, 다

양한 과업에서 우수한 성능을 달성한다. DeepSeek-R1는 Cold-Start 데이터와 반복적 RL Fine-Tuning을 함께 활용하여 더욱 강력한 성능을 확보하였으며, 여러 작업에서 OpenAI-o1-1217에 근접한 성능을 보인다.

또한 우리는 DeepSeek-R1의 추론 능력을 소형 Dense 모델에 Distillation하는 방식을 탐구하였다. DeepSeek-R1를 Teacher 모델로 삼아 약 80만(800K) 개의 학습 샘플을 생성하고, 이를 기반으로 여러 소형 Dense 모델을 Fine-Tuning하였다. 그 결과 DeepSeek-R1-Distill-Qwen-1.5B는 AIME에서 28.9%, MATH에서 83.9%를 달성하며, GPT-4o 및 Claude-3.5-Sonnet을 능가하는 수학 벤치마크 성능을 보인다. 이 외의 Dense 모델들도 동일한 Base Checkpoint를 사용한 다른 Instruction-Tuned 모델에 비해 큰 폭으로 우수한 성능을 기록하였다.

향후 DeepSeek-R1에 대해 다음과 같은 방향에서 추가 연구를 진행하고자 한다.

- **General Capability:** 현재 DeepSeek-R1는 Function Calling, Multi-turn, 복잡한 Role-playing, JSON Output 등의 작업에서 DeepSeek-V3 대비 제한적인 능력을 보인다. 앞으로는 긴 CoT를 활용하여 이러한 영역의 작업을 향상할 수 있는 방법을 모색하고자 한다.
- **Language Mixing:** DeepSeek-R1는 현재 중국어와 영어를 중심으로 최적화되어, 다른 언어 질의를 처리할 때 언어 혼합 문제가 발생할 수 있다. 예를 들어, 영어가 아닌 언어로 질의가 주어지더라도 DeepSeek-R1가 추론 및 응답을 영어로 생성할 가능성이 있다. 이 같은 문제를 해결하기 위해 향후 업데이트에서 다국어 지원을 강화할 예정이다.
- **Prompt Engineering:** DeepSeek-R1를 평가할 때, 모델이 Prompt에 민감하게 반응함을 관찰하였다. Few-Shot Prompting은 일관되게 성능을 저하시킨다. 따라서 최적 성능을 위해서는 문제를 직접 기술하고 결과 형식을 지정하는 Zero-Shot 방식을 권장한다.
- **Software Engineering Tasks:** RL 과정에서 긴 평가 시간이 요구됨에 따라, 대규모 RL을 소프트웨어 엔지니어링 관련 과제에 대규모로 적용하는 것은 아직 이루어지지 않았다. 이로 인해 DeepSeek-R1는 소프트웨어 엔지니어링 벤치마크에서 DeepSeek-V3 대비 큰 폭의 개선을 보이지 않는다. 향후 버전에서는 소프트웨어 엔지니어링 데이터에 대해 Rejection Sampling을 실시하거나, RL 과정에서 비동기 평가(Asynchronous Evaluation)를 도입하여 효율성을 높이고, 더 나은 성능 향상을 목표로 할 것이다.

## 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](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.
- 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/>.
- 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.
- 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.

## 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.