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

Presented by: Natnael Daba

Feb 18, 2025

Authors

DeepSeek-AI, 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, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J.L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, 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, 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, Wanjia Zhao, 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, Yanhong Xu, 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

DeepSeek-AI
research@deepseek.com

Content

Introduction

- **Goal**: explore the potential of LLMs to develop reasoning capabilities using pure reinforcement learning (RL) process.
- Use DeepSeek-V3-Base[1] as the base model and Group Relative Policy Optimization (GRPO) as RL framework.
    - This results in DeepSeek-R1-Zero.
    - However, DeepSeek-R1-Zero encounters challenges such as poor readability, and language mixing.

[1]Liu, Aixin, et al. "Deepseek-v3 technical report." arXiv preprint arXiv:2412.19437 (2024).

Introduction

- DeepSeek-R1 was introduced to address shortcomings of DeepSeek-R1-Zero.
  - DeepSeek-R1 incorporates a small amount of cold-start data and a multi-stage training pipeline.
- Explored distillation from DeepSeek-R1 to smaller dense models.
  - Demonstrated that the reasoning patterns discovered by larger base models are crucial for improving reasoning capabilities.

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

Contributions

1. Post-Training: Large-Scale Reinforcement Learning on the Base Model
    - Applied RL to a base model without relying on supervised fine-tuning (SFT) as a preliminary step.
    - Validated that reasoning capabilities of LLMs can be incentivized purely through RL, without SFT.
    - Introduced a pipeline for combining multiple RL and SFT stages to discover improved reasoning patterns.

Contributions

2. Distillation: Smaller Models Can Be Powerful Too
    - Demonstrated that the reasoning patterns of larger models can be distilled into smaller models.
    - Validated that this results in better performance compared to the reasoning patterns discovered through RL on small models.

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization (GRPO)
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization (GRPO)
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

Reinforcement Learning Algorithm

- Group Relative Policy Optimization (GRPO)[1]
  - RL algorithm used to train DeepSeek-R1-Zero
  - Foregoes the **critic** model that is typically the same size as the policy model.
  - Estimates the baseline from **group** scores instead.

[1] Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

Reinforcement Learning Algorithm

- **Group Relative Policy Optimization (GRPO)[1]:** optimize the policy model π_θ by **maximizing** the following objective:

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

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

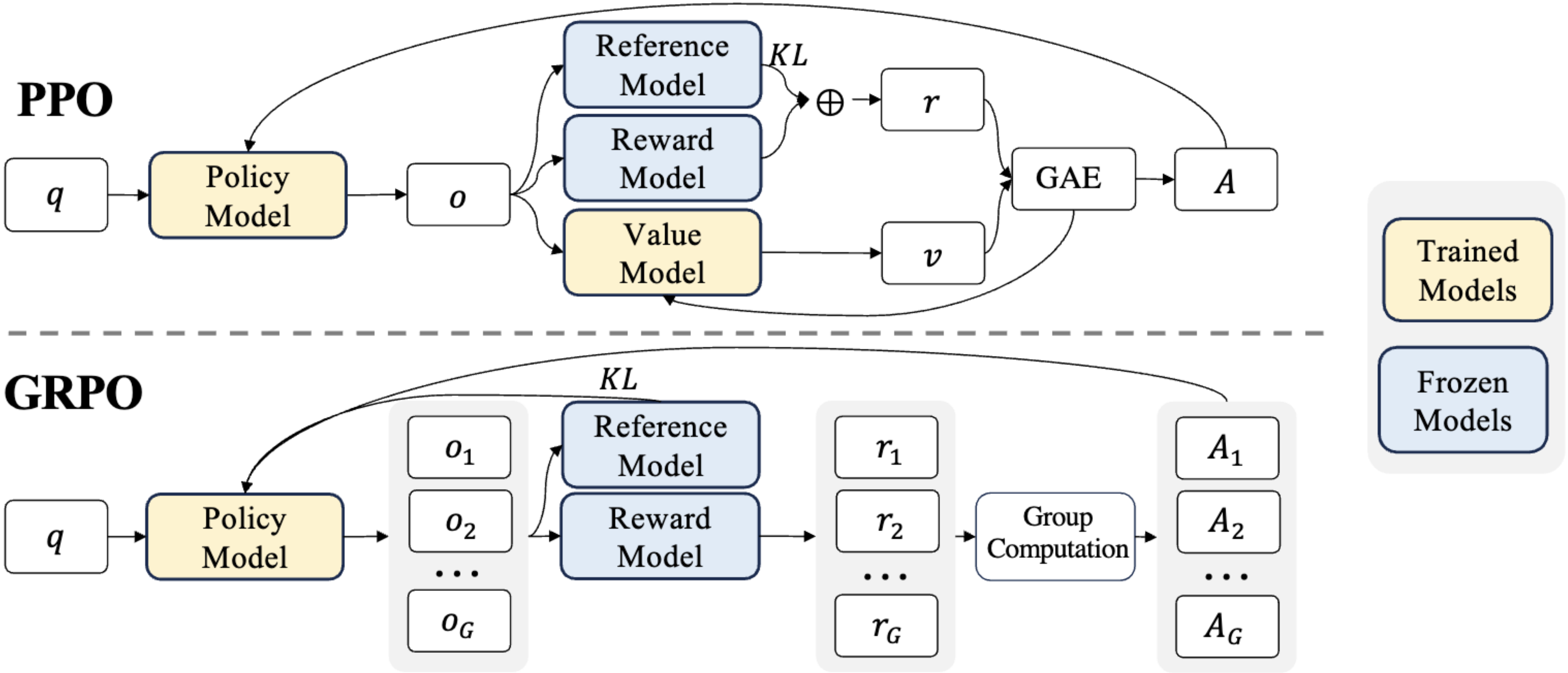
$$D_{KL}(\pi_\theta \| \pi_{\text{ref}}) = \frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - \log\frac{\pi_{\text{ref}}(o_i|q)}{\pi_\theta(o_i|q)} - 1$$

$$A_i = \frac{r_i - \text{mean}(\{r_1, r_2, \ldots, r_G\})}{\text{std}(\{r_1, r_2, \ldots, r_G\})}$$

Advantage computed using a group of rewards $\{r_1, r_2, \ldots, r_G\}$

[1]Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

PPO v.s. GRPO



Demonstration of PPO and our GRPO. GRPO foregoes the value model, instead estimating the baseline from group scores, significantly reducing training resources.

[1]Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

PPO v.s. GRPO

$$\mathcal{J}_{PPO}(\theta) = \mathbb{E}[q \sim P(Q), o \sim \pi_{\theta_{old}}(O|q)] \frac{1}{|o|} \sum_{t=1}^{|o|} \min\left[\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})} A_t, \text{clip}\left(\frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{\theta_{old}}(o_t|q,o_{<t})}, 1-\varepsilon, 1+\varepsilon\right) A_t\right]$$

$$r_t = r_\varphi(q, o_{\le t}) - \beta \log \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{ref}(o_t|q,o_{<t})}$$

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

$$\mathbb{D}_{KL}\left[\pi_\theta || \pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - 1,$$

[1] Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

PPO v.s. GRPO

Algorithm 1 Iterative Group Relative Policy Optimization

Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; hyperparameters ε, β, μ

policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, …, I **do**
    reference model $\pi_{ref} \leftarrow \pi_\theta$
    for step = 1, …, M **do**
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{r_i\}_{i=1}^{G}$ for each sampled output o_i by running r_φ
        Compute $\hat{A}_{i,t}$ for the t-th token of o_i through group relative advantage estimation.
        for GRPO iteration = 1, …, μ **do**
            Update the policy model π_θ by maximizing the GRPO objective (Equation 21)
    Update r_φ through continuous training using a replay mechanism.

Output π_θ

[1]Shao, Zhihong, et al. "Deepseekmath: Pushing the limits of mathematical reasoning in open language models." arXiv preprint arXiv:2402.03300 (2024).

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization (GRPO)
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

Reward Modeling

- **Rule-based** reward system is used to train DeepSeek-R1-Zero.
  - **Accuracy rewards**: evaluates whether the response is correct or not.
    - E.g. For math problems, require the model to provide final answer in a specified format (e.g. within a box[1]) or for LeetCode problems, use compiler generated feedback based on predefined test cases.
  - **Format rewards:** enforces the model to put its thinking process between ‘<think>’ and ‘</think>’ tags.

[1] https://github.com/volcengine/verl/blob/main/verl/utils/reward_score/math.py#L17

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

Training Template

`A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer. The reasoning process and answer are enclosed within <think> </think> and <answer> </answer> tags, respectively, i.e., <think> reasoning process here </think> <answer> answer here </answer>. User: prompt. Assistant:`

Table 1: Template for DeepSeek-R1-Zero. `prompt` will be replaced with the specific reasoning question during training.

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

Performance of DeepSeek-R1-Zero

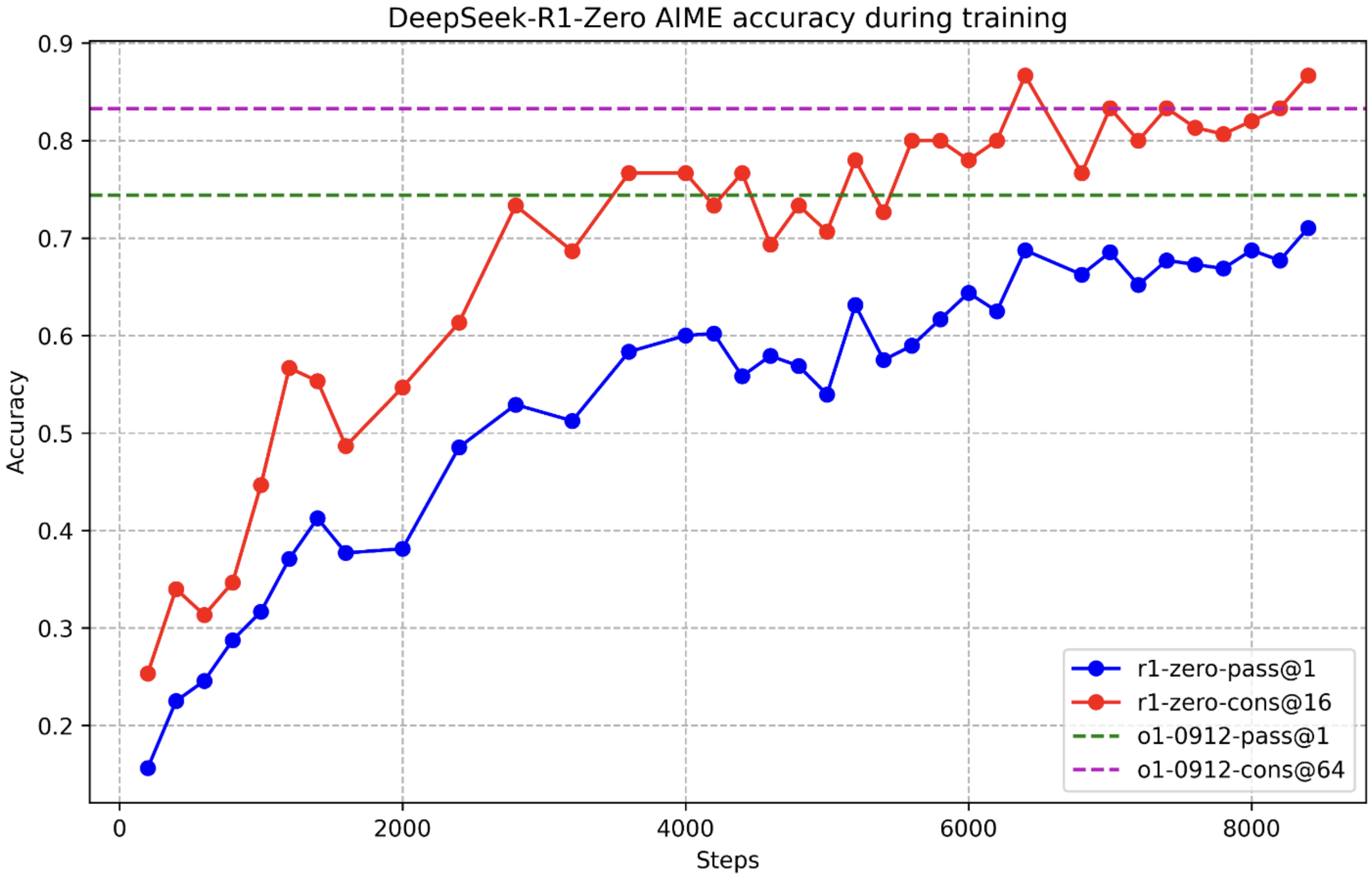


Figure 2: AIME* accuracy of DeepSeek-R1-Zero during training. For each question, accuracy is calculated as average accuracy of 16 sampled responses.

*https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

Performance of DeepSeek-R1-Zero

Model	AIME 2024[1]		MATH-500[2]	GPQA Diamond[3]	LiveCode Bench[4]	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.

[1] https://huggingface.co/datasets/Maxwell-Jia/AIME_2024

[2] https://huggingface.co/datasets/HuggingFaceH4/MATH-500

[3] https://huggingface.co/datasets/Idavidrein/gpqa

[4] https://arxiv.org/abs/2403.07974

Self-evolution Process of DeepSeek-R1-Zero

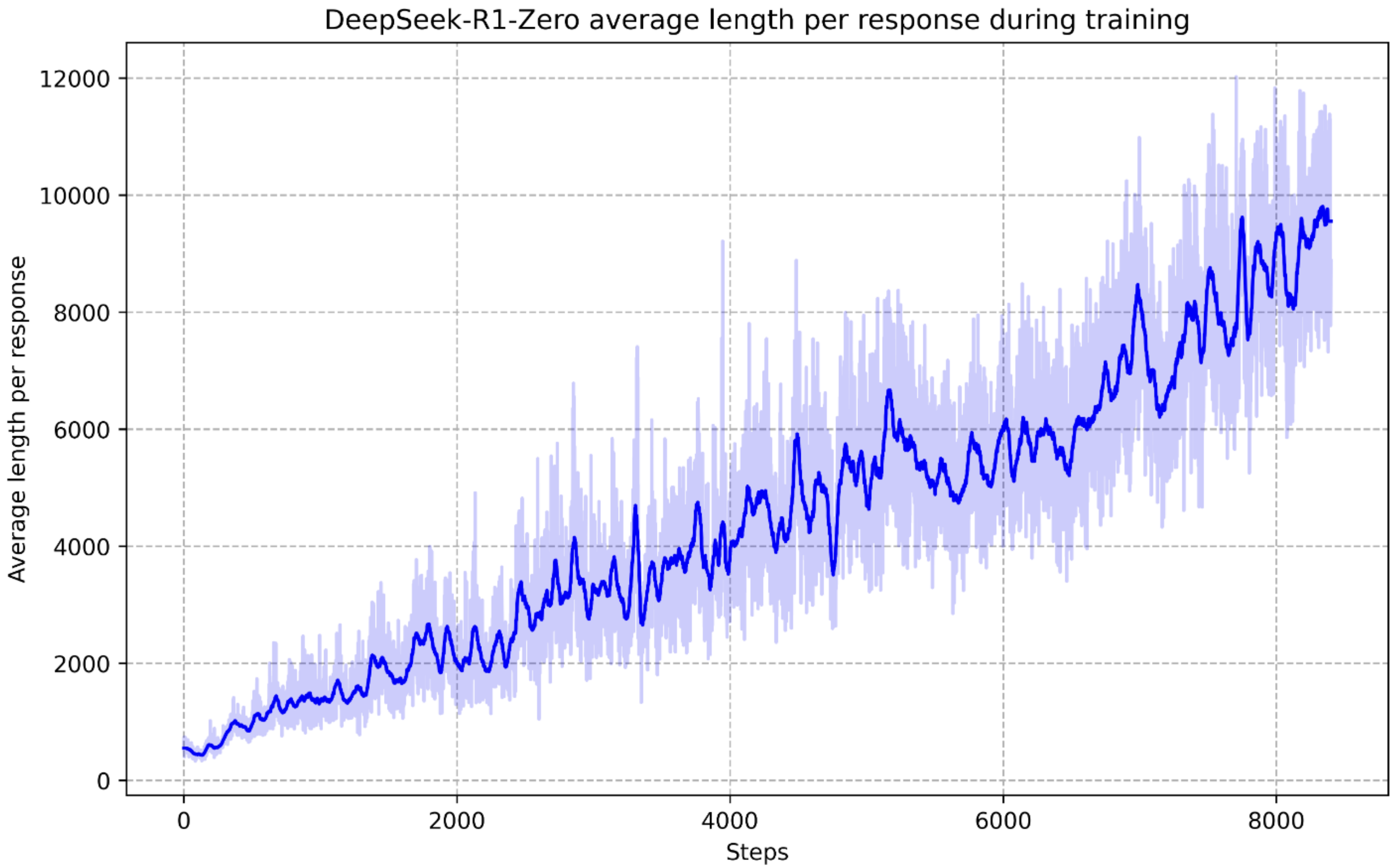
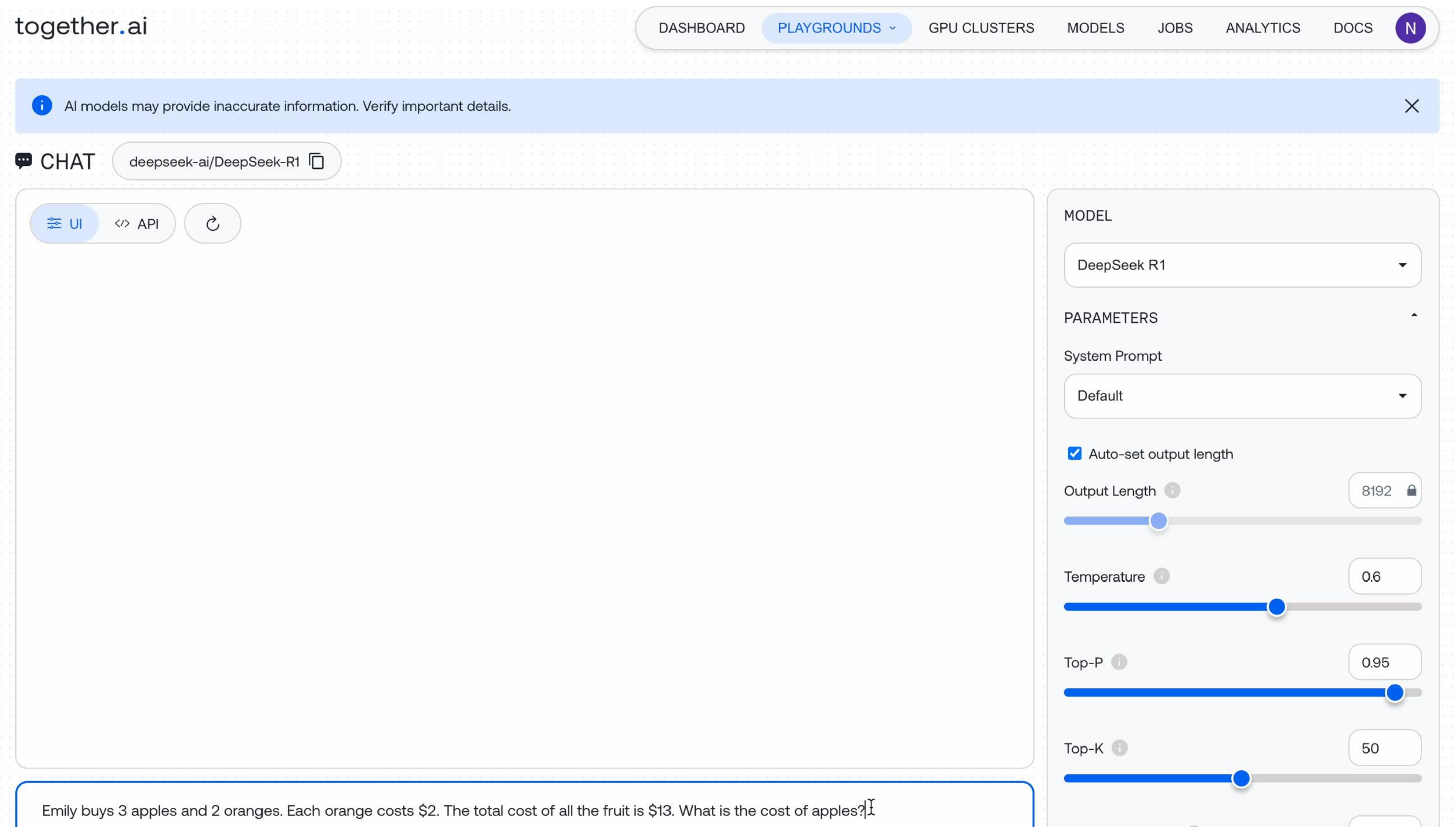


Figure 3: The average response length of DeepSeek-R1-Zero on the training set during the RL process. DeepSeek-R1-Zero naturally learns to solve reasoning tasks with more thinking time.

Aha Moment of DeepSeek-R1-Zero

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

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.



https://www.together.ai/

DeepSeek-R1-Zero: RL on the Base Model

- Reinforcement Learning Algorithm
  - Group Relative Policy Optimization
- Reward Modeling
  - Accuracy rewards
  - Format rewards
- Training Template
- Performance, Self-evolution Process and Aha Moment of DeepSeek-R1-Zero
- Drawback of DeepSeek-R1-Zero

Drawback of DeepSeek-R1-Zero

- Poor readability
- Language mixing

DeepSeek-R1: Reinforcement Learning with Cold Start

- Two questions:
  1. Can reasoning performance be further improved by incorporating a "small" amount of high-quality data as a cold start?
  2. How can we train a user-friendly model that demonstrates strong general capabilities?
- To address these questions, a pipeline to train DeepSeek-R1 is designed.

DeepSeek-R1: Reinforcement Learning with Cold Start

- DeepSeek-R1 training pipeline consists of four stages:
  1. Cold Start
  2. Reasoning-oriented Reinforcement Learning
  3. Rejection Sampling and Supervised Fine-Tuning
  4. Reinforcement Learning for all Scenarios

SFT stages

RL stages

DeepSeek-R1: Reinforcement Learning with Cold Start

- DeepSeek-R1 training pipeline consists of four stages:
  1. Cold Start
  2. Reasoning-oriented Reinforcement Learning
  3. Rejection Sampling and Supervised Fine-Tuning
  4. Reinforcement Learning for all Scenarios

Stage 1: Cold Start

- Approaches explored for collecting cold start data:
  - 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
  - Refining the results through post-processing by human annotators

DeepSeek-R1: Reinforcement Learning with Cold Start

- DeepSeek-R1 training pipeline consists of four stages:
  1. Cold Start
  2. Reasoning-oriented Reinforcement Learning
  3. Rejection Sampling and Supervised Fine-Tuning
  4. Reinforcement Learning for all Scenarios

Stage 2: Reasoning-oriented Reinforcement Learning

- Same RL training process as DeepSeek-R1-Zero i.e., using GRPO
- Language mixing problem is observed again in CoT responses
  - Solution: introduce language consistency reward i.e., the proportion of target language words in the CoT
  - Final reward = accuracy reward + language consistency reward

DeepSeek-R1: Reinforcement Learning with Cold Start

- DeepSeek-R1 training pipeline consists of four stages:
    1. Cold Start
    2. Reasoning-oriented Reinforcement Learning
    3. Rejection Sampling and Supervised Fine-Tuning
    4. Reinforcement Learning for all Scenarios

Stage 3: Rejection Sampling and Supervised Fine-Tuning

- Use RL checkpoint from stage 2 to generate SFT data for stage 3.
- Incorporate data from other domains in addition to reasoning.
  - E.g. writing, role-playing, and other general-purpose tasks

Stage 3: Rejection Sampling and Supervised Fine-Tuning

- Two types of SFT data generated:
  1. **Reasoning data:** curate reasoning prompts and generate reasoning response from RL checkpoint in stage 2 using rejection sampling.
     - **Rejection sampling:** generate multiple responses for each reasoning prompt and retain only the correct responses, as determined by the generative reward model (DeepSeek-V3) and reject the rest.
     - Total collected reasoning data: **600k samples**
  2. **Non-Reasoning data:** writing, factual QS, self-cognition, and translation
     - Portions of non-reasoning SFT training dataset for DeepSeek-V3 + generated responses from DeepSeek-V3 to non-reasoning queries.
     - Total collected non-reasoning data: **200k samples**
- Fine-tune DeepSeek-V3-Base for two epochs using dataset of **800k** samples.

DeepSeek-R1: Reinforcement Learning with Cold Start

- DeepSeek-R1 training pipeline consists of four stages:
  1. Cold Start
  2. Reasoning-oriented Reinforcement Learning
  3. Rejection Sampling and Supervised Fine-Tuning
  4. Reinforcement Learning for all Scenarios

Stage 4: Reinforcement Learning for all Scenarios

- Goal: improve the model's helpfulness and harmlessness.
- Train using reward signals + diverse prompt distributions.
- For reasoning data, use same method as DS-Zero + rule-based rewards.
- For non-reasoning, use reward models to capture human preferences.
- For harmlessness, evaluate the entire response (reasoning + summary).
- For helpfulness, evaluate the final summary.

Distillation: Empower Small Models with Reasoning Capability

- Fine-tune open-source models like Qwen and Llama using the 800k samples curated with DeepSeek-R1.
- This straightforward distillation method significantly enhances the reasoning abilities of smaller models.
- For distilled models, only SFT is applied. I.e., no RL stage.

Pass@1 evaluation

- A simplified adaptation of pass@k metric from Chen et al., 2021.

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

k: number of generated responses ($4 \leq k \leq 64$)

$$p_i = \begin{cases} 1, & \text{if the } i\text{-th response is correct,} \\ 0, & \text{otherwise.} \end{cases}$$

Chen, Mark, et al. "Evaluating large language models trained on code." arXiv preprint arXiv:2107.03374 (2021).

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.

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.

Distillation v.s. Reinforcement Learning

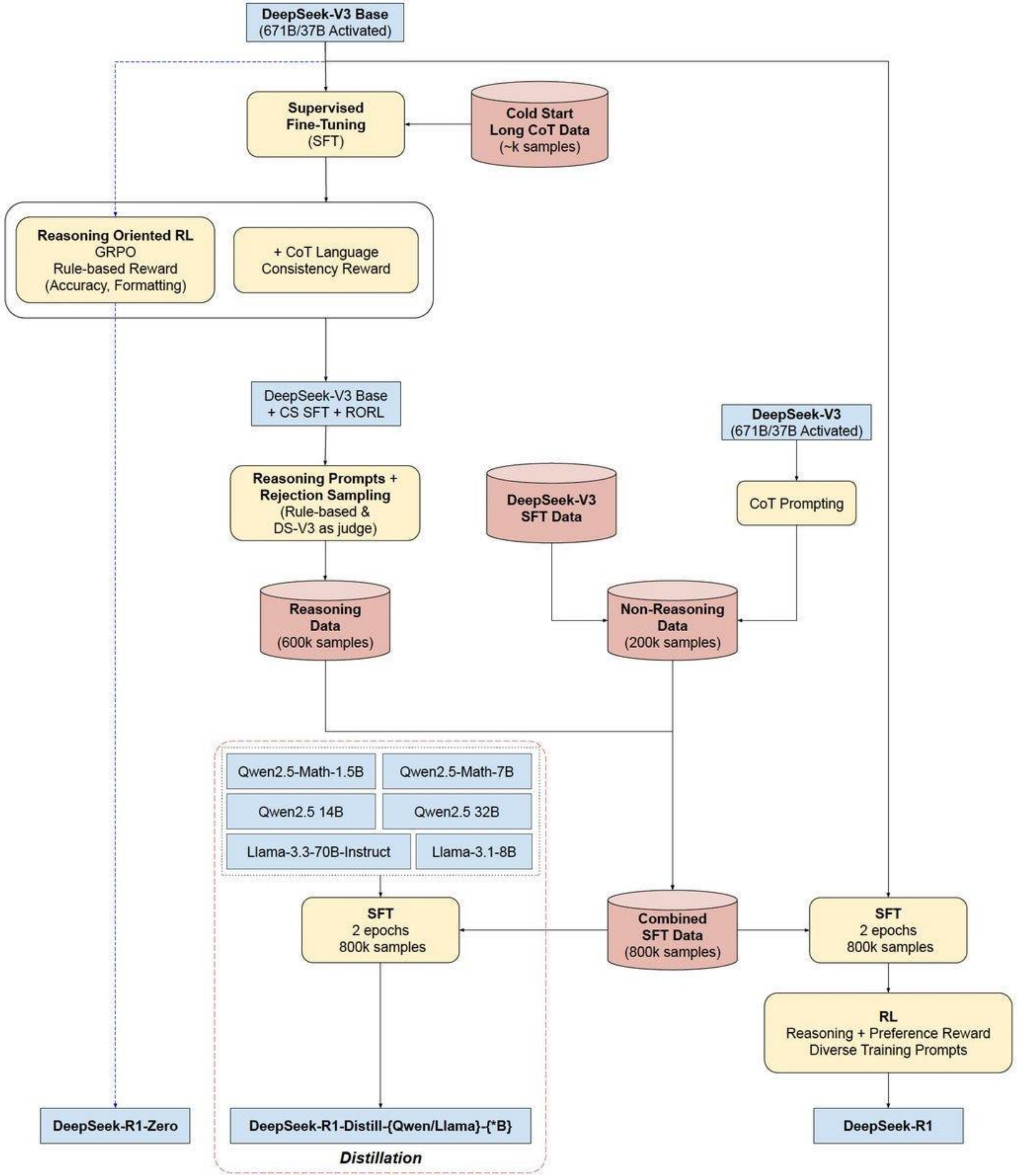
- **Question**: can small models achieve comparable performance through large-scale RL training without distillation?

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.

Distilled from DeepSeek-R1

Qwen-32B-Base + 10K RL steps



https://x.com/SirrahChan/status/1881488738473357753

Resources

- DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning: https://arxiv.org/abs/2501.12948
- DeepSeek-V3 Technical Report: https://arxiv.org/abs/2412.19437
- DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models: https://arxiv.org/abs/2402.03300
- RLHF and PPO
  - https://github.com/hkproj/rlhf-ppo/blob/main/Slides.pdf
  - http://youtube.com/watch?v=qGyFrqc34yc
- RL for LLMs
  - https://phontron.com/class/anlp2024/lectures/#reinforcement-learning-feb-22

Resources

- DeepSeek-R1 paper explanation
  - https://youtu.be/XMnxKGVnEUc?si=UnFmMGe4yZba9PbI
- Deep Dive into LLMs like ChatGPT
  - https://youtu.be/7xTGNNLPyMI?si=h42Tn8sVjfo5ioiH
- HuggingFace implementation of GRPO
  - https://github.com/huggingface/trl/blob/main/trl/trainer/grpo_trainer.py#L108

Thank you