INFO-I590 Fundamentals and Applications of LLMs

DPO and GRPO

Direct Preference Optimization (DPO)
Group Relative Policy Optimization (GRPO)

Jisun An

Direct Preference Optimization (DPO)

RLHF optimization objective

Maximizes the rewards

Prevents the model from changing too drastically

$$\max_{\pi} \mathbb{E}_{x\sim D, y\sim\pi} \left[\boxed{r(x, y)} - \beta \boxed{D_{KL}(\pi(y \mid x) \,\|\, \pi_{\text{ref}}(y \mid x))} \right]$$

Current aligned LLM

SFT (instruction-tuned) LLM

How to get rid of this reward?

(not get rid of it, but turn it into a probability)

Can we avoid rollouts?

DPO (Direct Preference Optimization)

- Idea: Can we fine-tune a model directly using a preference dataset without RL?
- Why Avoid RL?
  - RL is unstable and computationally expensive.
  - It’s hard to pinpoint what to change in long outputs.
  - Generating multiple samples for training (rollouts) increases cost.
  - Training a reliable reward model is challenging.
  - Small reward differences can cause instability.
- Also called as ‘Preference Tuning’

The DPO loss function

Bradley Terry Model

$$P(y_W > y_L \mid x) = \frac{e^{r(x,y_W)}}{e^{r(x,y_W)} + e^{r(x,y_L)}}$$

$$\max_{\pi} \mathbb{E}_{x\sim D, y\sim\pi}\left[\boxed{r(x,y)} - \beta \boxed{D_{KL}(\pi(y \mid x) \parallel \pi_{\text{ref}}(y \mid x))}\right]$$

Mathematical Manipulation

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l)\sim\mathcal{D}}\left[\log \sigma\left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

The DPO loss function

Maximize probability for good response

Minimize probability for bad response

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\boxed{\pi_\theta(y_w \mid x)}}{\boxed{\pi_{\text{ref}}(y_w \mid x)}} - \beta \log \frac{\boxed{\pi_\theta(y_l \mid x)}}{\boxed{\pi_{\text{ref}}(y_l \mid x)}} \right) \right]$$

Aligned model
we are training

Don't change the model too much

- No explicit reward model
- No need for rollouts from the policy

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

(Optional)

Deriving the DPO loss function

$$\begin{aligned}
\max_{\pi} \mathbb{E}_{x\sim\mathcal{D},y\sim\pi}\big[r(x,y)\big] - \beta\mathbb{D}_{\mathrm{KL}}\big[\pi(y|x) \mid\mid \pi_{\mathrm{ref}}(y|x)\big] \\
&= \max_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[r(x,y) - \beta\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)}\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi_{\mathrm{ref}}(y|x)} - \frac{1}{\beta}r(x,y)\right] \\
&= \min_{\pi} \mathbb{E}_{x\sim\mathcal{D}}\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\frac{1}{Z(x)}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right)} - \log Z(x)\right] \qquad (12)
\end{aligned}$$

where we have partition function:

$$Z(x) = \sum_{y}\pi_{\mathrm{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right).$$

* Note that the partition function is a function of only x and the reference policy π_{ref}, but does not depend on the policy π.

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

(Optional)

We can now define a new policy π*,

$$\pi^*(y|x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right),$$

which is a valid probability distribution as $\pi^*(y|x) \geq 0$ for all y and $\sum_y \pi^*(y|x) = 1$. Since $Z(x)$ is not a function of y, we can then re-organize the final objective in Eq 12 as:

$$\min_\pi \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{E}_{y\sim\pi(y|x)}\left[\log\frac{\pi(y|x)}{\pi^*(y|x)}\right] - \log Z(x)\right] = \tag{13}$$

$$\min_\pi \mathbb{E}_{x\sim\mathcal{D}}\left[\mathbb{D}_{\text{KL}}(\pi(y|x)\;||\;\pi^*(y|x)) - \log Z(x)\right] \tag{14}$$

Now, since $Z(x)$ does not depend on π, the minimum is achieved by the policy that minimizes the first KL term. Gibbs' inequality tells us that the KL-divergence is minimized at 0 if and only if the two distributions are identical. Hence we have the optimal solution:

$$\pi(y|x) = \pi^*(y|x) = \frac{1}{Z(x)}\pi_{\text{ref}}(y|x)\exp\left(\frac{1}{\beta}r(x,y)\right) \tag{15}$$

for all $x \in \mathcal{D}$. This completes the derivation.

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

(Optional)

$$\pi_r(y \mid x) = \frac{1}{Z(x)} \pi_{\text{ref}}(y \mid x) \exp\left(\frac{1}{\beta} r(x, y)\right)$$

We can rearrange the above equation to express the reward function in terms of its corresponding optimal policy π_r, the reference policy π_{ref}, and the unknown partition function(x).

$$r(x, y) = \beta \log \frac{\pi_r(y \mid x)}{\pi_{\text{ref}}(y \mid x)} + \beta \log Z(x)$$

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

(Optional)

It is straightforward to derive the DPO objective under the Bradley-Terry preference model as we have

$$p^*(y_1 \succ y_2|x) = \frac{\exp\left(r^*(x, y_1)\right)}{\exp\left(r^*(x, y_1)\right) + \exp\left(r^*(x, y_2)\right)} \tag{16}$$

In Section 4 we showed that we can express the (unavailable) ground-truth reward through its corresponding optimal policy:

$$r^*(x, y) = \beta \log \frac{\pi^*(y|x)}{\pi_{\text{ref}}(y|x)} + \beta \log Z(x) \tag{17}$$

Substituting Eq. 17 into Eq. 16 we obtain:

$$\begin{aligned} p^*(y_1 \succ y_2|x) &= \frac{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right)}{\exp\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} + \beta \log Z(x)\right) + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} + \beta \log Z(x)\right)} \\ &= \frac{1}{1 + \exp\left(\beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)} - \beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)}\right)} \\ &= \sigma\left(\beta \log \frac{\pi^*(y_1|x)}{\pi_{\text{ref}}(y_1|x)} - \beta \log \frac{\pi^*(y_2|x)}{\pi_{\text{ref}}(y_2|x)}\right). \end{aligned}$$

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

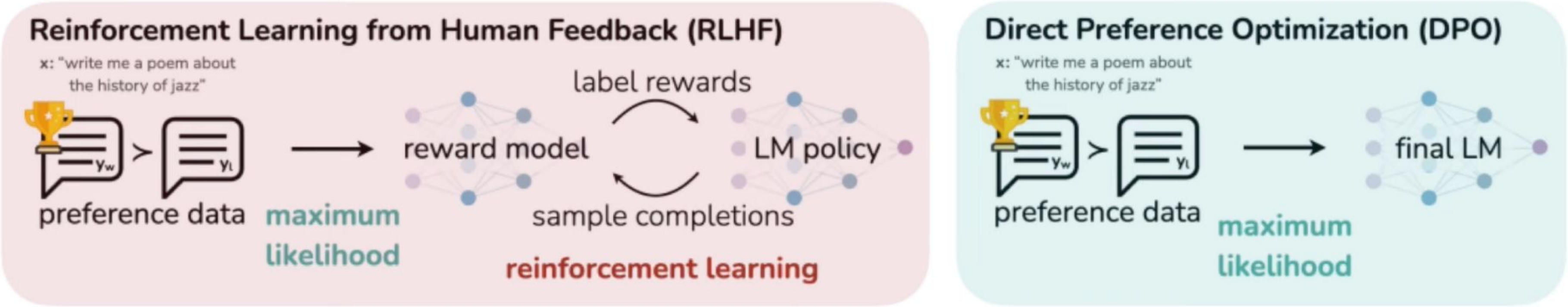
(Optional)

Now that we have the probability of human preference data in terms of the optimal policy rather than the reward model, we can formulate a maximum likelihood objective for a parametrized policy π_θ. Analogous to the reward modeling approach (i.e. Eq. 2), our policy objective becomes:

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} \left[\log \sigma \left(\beta \log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} \right) \right]. \quad (7)$$

Rafailov et al. 2023. Direct preference optimization: Your language model is secretly a reward model. https://arxiv.org/abs/2305.18290

RLHF vs DPO



Img: https://www.youtube.com/watch?v=k2pD3k1485A

Group Relative Policy Optimization (GRPO)

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-AI 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. https://arxiv.org/abs/2501.12948

Group Relative Policy Optimization (GRPO)

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

Zhihong Shao[1,2*†], Peiyi Wang[1,3*†], Qihao Zhu[1,3*†], Runxin Xu[1], Junxiao Song[1]
Xiao Bi[1], Haowei Zhang[1], Mingchuan Zhang[1], Y.K. Li[1], Y. Wu[1], Daya Guo[1*]

[1]DeepSeek-AI, [2]Tsinghua University, [3]Peking University

{zhihongshao,wangpeiyi,zhuqh,guoday}@deepseek.com
https://github.com/deepseek-ai/DeepSeek-Math

Zhihong et al. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. https://arxiv.org/abs/2402.03300

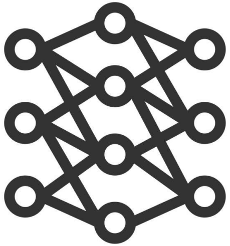
(reminder) RL in language modeling

State S

Policy π

Distribution on
Actions A

Students



a	0.1%
are	3%
can	3%
…	…
…	…
the	0.5%
opened	5%
zebra	0.1%

S_0, A_0 (i.e. selecting ‘opened’),

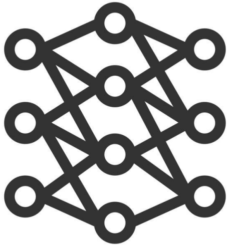
(reminder) RL in language modeling

State S

Policy π

Distribution on
Actions A

Students opened



a	10%
an	10%
books	2%
…	…
…	…
the	10%
their	10%
zebra	0.1%

S_0, A_0, S_1, A_1, (i.e. selecting ‘their’)

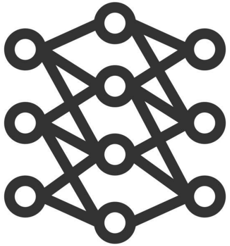
(reminder) RL in language modeling

State S

Policy π

Distribution on
Actions A

Students opened their



apples	1%
ants	1%
books	10%
…	…
…	…
presents	7%
zebra	2%

Update and improve

$S_0, A_0, S_1, A_1, S_2, A_2, S_3, \dots, A_{n\_1}, S_n.$

Reward: R

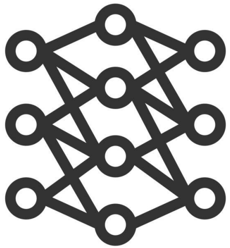
(reminder) RL in language modeling

State S

Policy π

Distribution on
Actions A

Students opened their



apples	1%
ants	1%
books	10%
…	…
…	…
presents	7%
zebra	2%

π(A|S) = P(Choose A when at S)

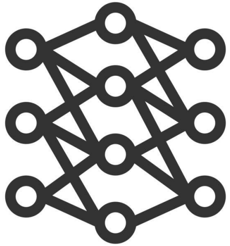
(reminder) RL in language modeling

State S

Policy π

Distribution on
Actions A

Students opened their



Params θ

apples	1%
ants	1%
books	10%
…	…
…	…
presents	7%
zebra	2%

$$\pi(A|S, \theta) = P(\text{Choose } A \text{ when at } S, \text{params} = \theta)$$

The REINFORCE Algorithm

How to improve θ?

- Observe S_0, A_0, S_1, A_1, … , A_{n_1}, S_n and R
- If R is <u>large</u>, then tweek θ to

  Make actions A_t <u>more</u> likely. i.e. ↑ $\pi(A_t|S_t, \theta)$

Large R

Students opened their books

apples	1%	
ants	1%	
books	10%	**+3%**
…	…	
…	…	
presents	7%	
zebra	2%	

The REINFORCE Algorithm

How to improve θ?

- Observe S_0, A_0, S_1, A_1, … , A_{n_1}, S_n and R
- If R is <u>large</u>, then tweek θ to

  Make actions A_t <u>more</u> likely. i.e. ↑ $\pi(A_t|S_t, \theta)$

$$\mathbf{R} \ \nabla_\theta \log \pi(A_t|S_t, \theta)$$

apples	1%	
ants	1%	
books	10%	+3%
…	…	
…	…	
presents	7%	
zebra	2%	

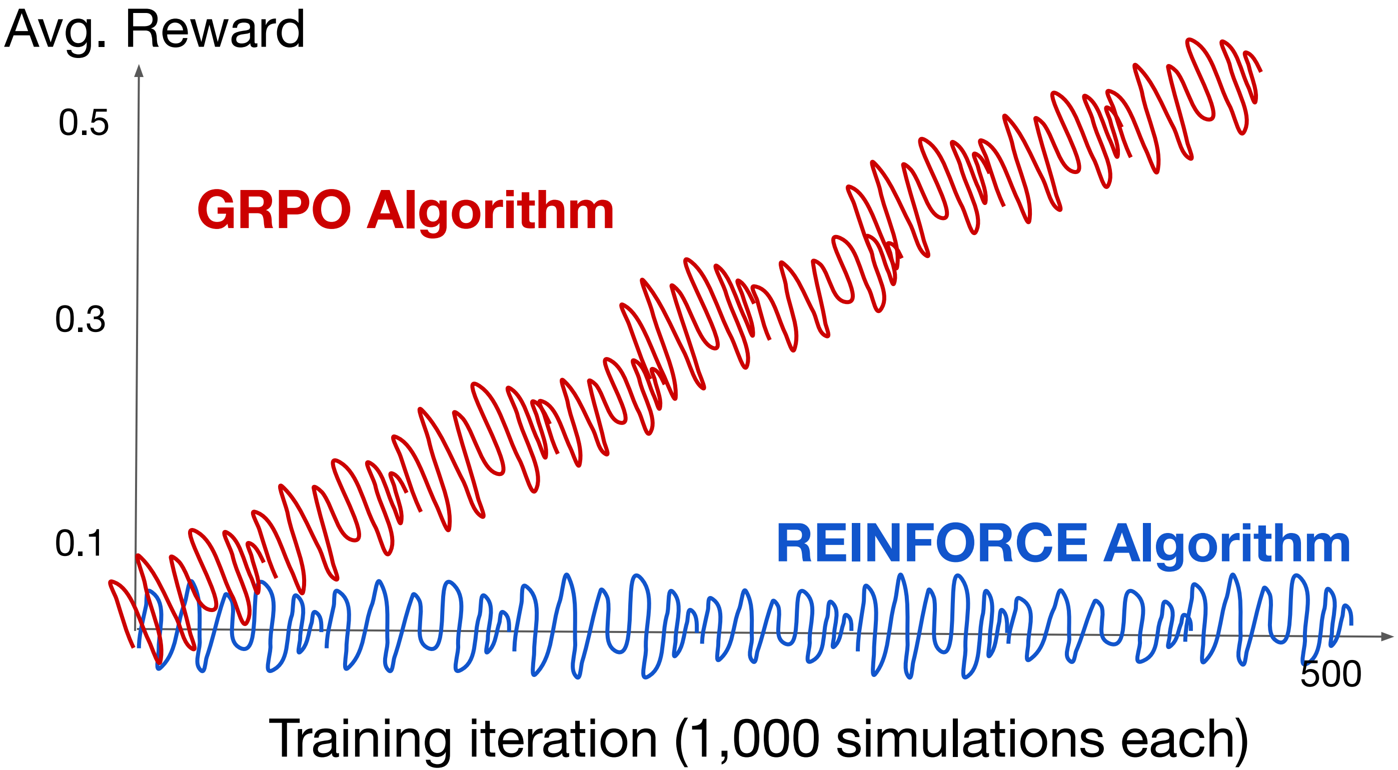
The REINFORCE Algorithm

How to improve θ?

- Observe $S_0, A_0, S_1, A_1, \ldots, A_{n\_1}, S_n$ and R
- If R is <u>large</u>, then tweek θ to
  Make actions A_t <u>more</u> likely. i.e. $\uparrow \pi(A_t|S_t, \theta)$

$$\theta_{new} = \theta + \mathbf{R}\, \nabla_\theta \log \pi(A_t|S_t, \theta)$$

A mockup example



Improvement: The GRPO Algorithm

How to improve θ? Reward size **R** is relative!

- Observe $S_0, A_0^{(1)}, S_1^{(1)}, A_1^{(1)}, \ldots, S_n^{(1)}$ and $R^{(1)}$
- Observe $S_0, A_0^{(2)}, S_1^{(2)}, A_1^{(2)}, \ldots, S_n^{(2)}$ and $R^{(2)}$
- Observe $S_0, A_0^{(3)}, S_1^{(3)}, A_1^{(3)}, \ldots, S_n^{(3)}$ and $R^{(3)}$

1. Students opened their books	R^1: 10
2. Students no playing	R^2: 2
3. Students were quite	R^3: 5

$$\theta_{new} = \theta + \textbf{Advantage}^{(1)} \nabla_\theta \log \pi(A_t|S_t, \theta)$$

$$\textbf{Advantage}^{(1)} = \frac{R^{(1)} - \text{avg}\{R^{(1)}, R^{(2)}, R^{(3)}\}}{\text{std}\{R^{(1)}, R^{(2)}, R^{(3)}\}}$$

DeepSeek's RL objective function

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 π_θ by maximizing the following objective:

Clipping: Prevent excessive jumps

KL divergence

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

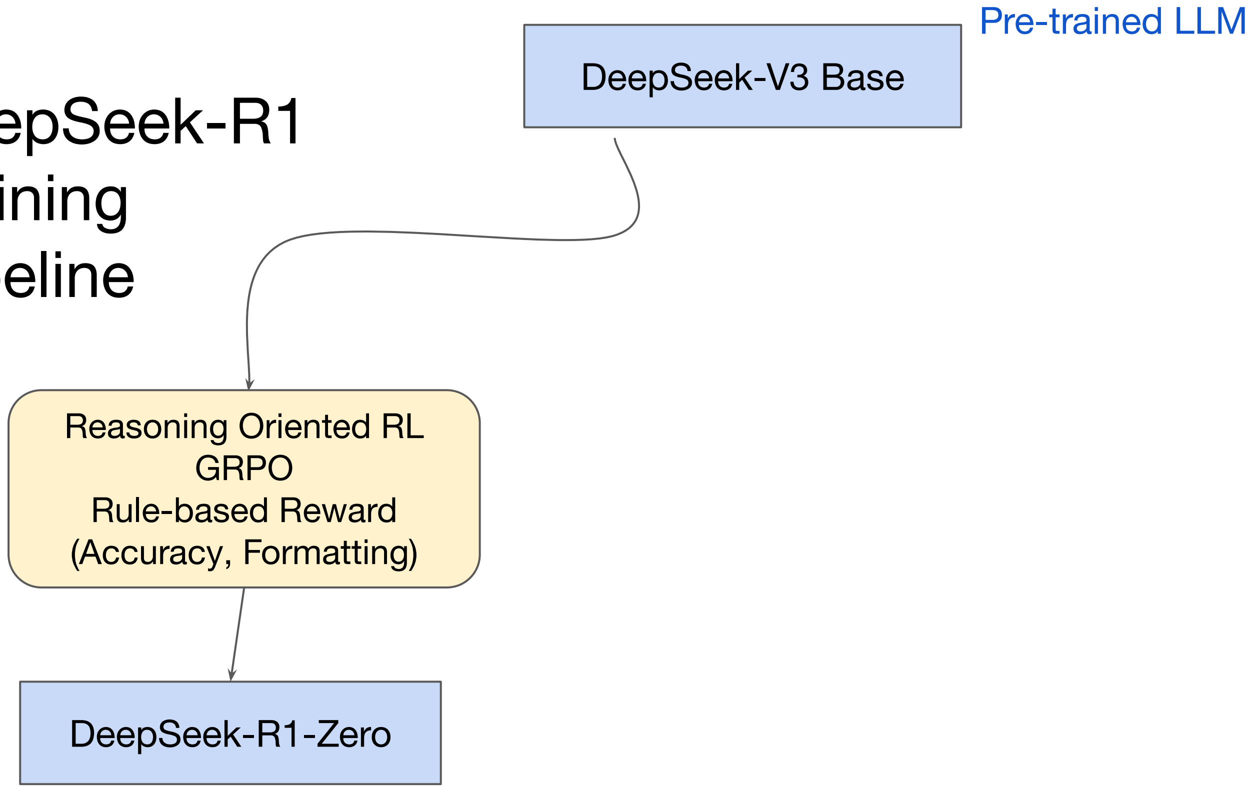
Surrogate objective function
→ Considering momentum

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

where ε and β 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:

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

DeepSeek-R1 Training Pipeline



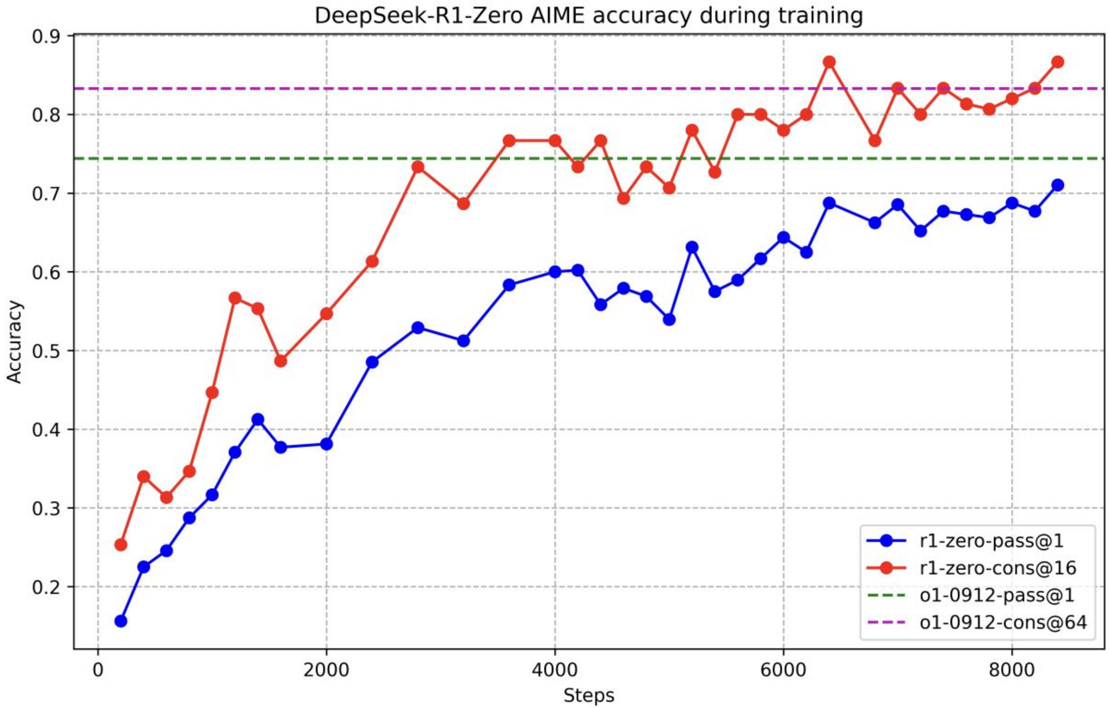
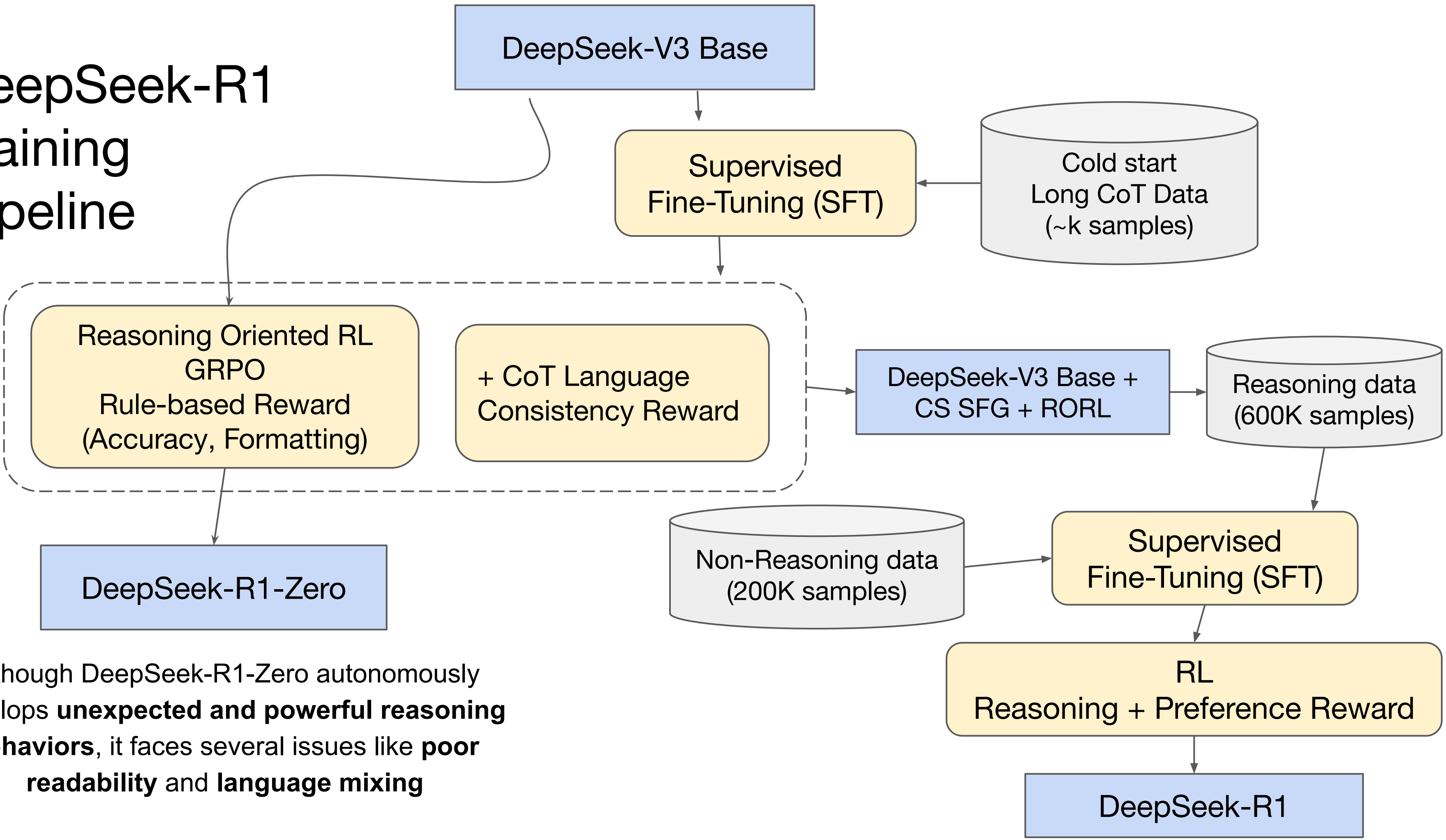


Figure 2 | AIME accuracy of DeepSeek-R1-Zero during training. For each question, we sample 16 responses and calculate the overall average accuracy to ensure a stable evaluation.

DeepSeek-AI 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. https://arxiv.org/abs/2501.12948

DeepSeek-R1 Training Pipeline



Although DeepSeek-R1-Zero autonomously develops **unexpected and powerful reasoning behaviors**, it faces several issues like **poor readability** and **language mixing**

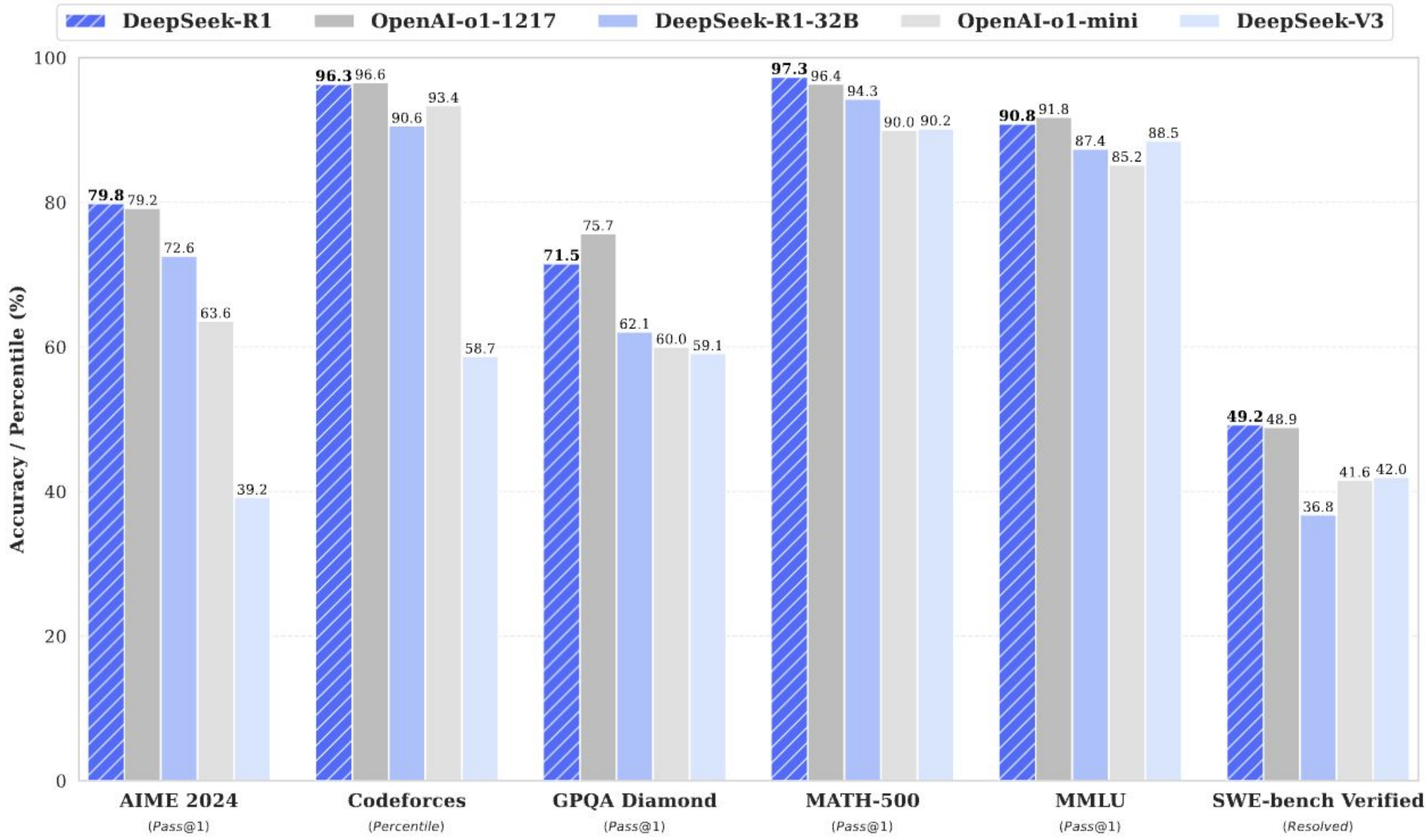
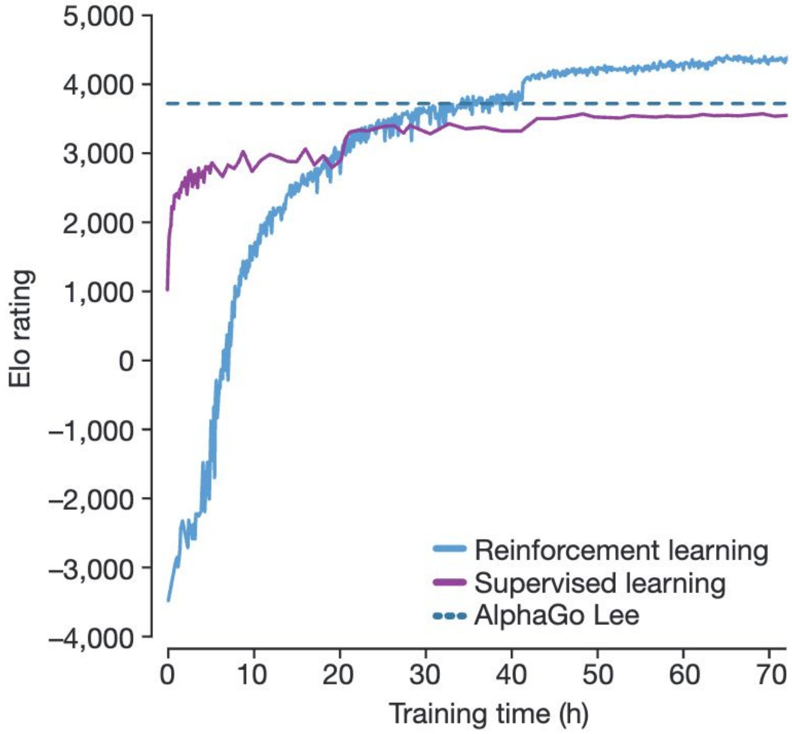


Figure 1 | Benchmark performance of DeepSeek-R1.

DeepSeek-AI 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. https://arxiv.org/abs/2501.12948

Why RL?



Silver et al. 2017. Mastering the game of go without human knowledge. Nature. https://www.nature.com/articles/nature24270

Any Questions?