

---

# DAPO : Improving Multi-Step Reasoning Abilities of Large Language Models with Direct Advantage-Based Policy Optimization

---

Jiacai Liu<sup>1,2</sup> Chaojie Wang<sup>1,†</sup> Chris Yuhao Liu<sup>1</sup> Liang Zeng<sup>1</sup> Rui Yan<sup>1</sup>  
Yiwen Sun<sup>2</sup> Yang Liu<sup>1</sup> Yahui Zhou<sup>1</sup>

<sup>1</sup>Skywork AI <sup>2</sup>Fudan University

{jiacai.liu,chaojie.wang}@kunlun-inc.com

## Abstract

The role of reinforcement learning (RL) in enhancing the reasoning of large language models (LLMs) is becoming increasingly significant. Despite the success of RL in many scenarios, there are still many challenges in improving the reasoning of LLMs. One key challenge is the sparse reward, which introduces more training variance in policy optimization and makes it difficult to obtain a good estimation for value function in Actor-Critic (AC) methods. To address these issues, we introduce Direct Advantage-Based Policy Optimization (DAPO), a novel step-level offline RL algorithm with theoretical guarantees for enhancing the reasoning abilities of LLMs. Unlike response-level methods (such as DPO and GRPO) that the update directions of all reasoning steps are governed by the outcome reward uniformly, DAPO employs a critic function to provide step-level dense signals for policy optimization. Additionally, the actor and critic in DAPO are trained independently, ensuring that critic is a good estimation of true state value function and avoiding the co-training instability observed in standard AC methods. We train DAPO on mathematical and code problems and then evaluate its performance on multiple benchmarks. Our results show that DAPO can effectively enhance the mathematical and code capabilities on both SFT models and RL models, demonstrating the effectiveness of DAPO.

## 1 Introduction

In the rapidly evolving landscape of artificial intelligence, large language models (LLMs) have emerged as a cornerstone of natural language processing (NLP) and beyond. These models, trained on vast corpora of text data, have demonstrated an unprecedented ability to understand [16, 3], generate, and reason such as solving mathematical problems [61, 48, 63, 53] and code generations [23, 8]. When the token generation process of a LLM is modeled as a Markov Decision Process (MDP), it can be naturally optimized and aligned with human preference using reinforcement learning (RL) methods, known as Reinforcement Learning from Human Feedback (RLHF). Despite the success of RLHF in various fields [11, 42, 43, 55, 46, 6], it still encounters challenges and difficulties in the field of reasoning especially in long Chain-of-Thought (CoT) models. One of the key challenges is the sparsity of rewards [44, 57]. When using LLMs for mathematical problem-solving and code generation, rewards are only assigned to the terminal tokens. This implies that the intermediate tokens receives no direct reward, and the optimization direction relies solely on the backpropagation of the reward from the terminal token. Consequently, there are two issues in practical RL training:

---

† Corresponding author.

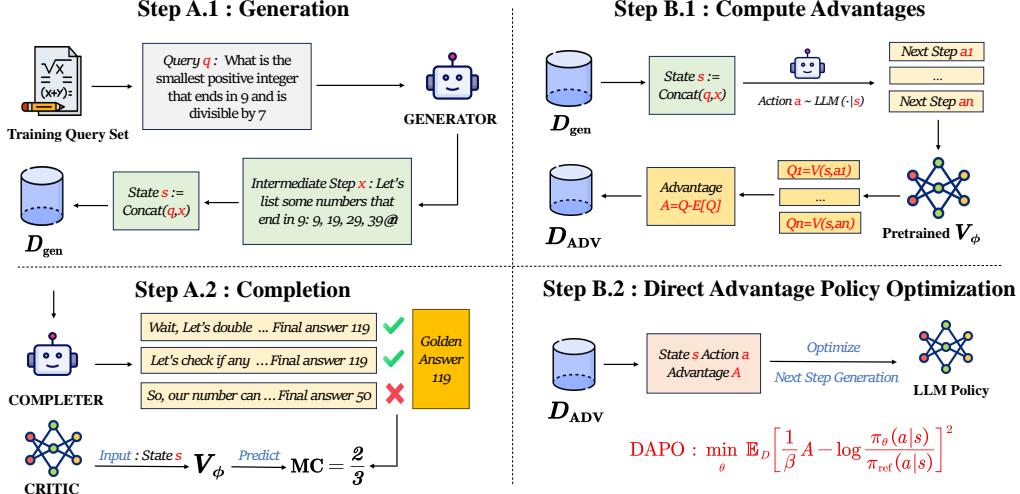


Figure 1: **Direct Advantage-Based Policy Optimization (DAPO)**. The whole training procedure of DAPO consists of two individual stages : 1) **Critic Training (left)**. Given the training query set  $\mathcal{Q}$ , for each query  $q \in \mathcal{Q}$ , DAPO uses the generator  $\pi_{\text{gen}}$  to generate multiple rollouts from  $q$  and derive a dataset of training states denoted as  $\mathcal{D}_{\text{gen}}$ . For each state in  $s \in \mathcal{D}_{\text{gen}}$ , DAPO uses the completer (often  $\pi_{\text{ref}}$ ) to sample multiple sub-trajectory from  $s$  and collects the MC value as the estimation of the true value. Then a critic network  $V_\phi$  is trained to approximate the value function of the completer. 2) **Policy Optimization (right)**. After extracting multiple next steps  $\{a_i\}_{i=1}^n$  of each state  $s$  from the completions in step A.1, DAPO then uses the trained critic to compute the advantages for all actions by  $\forall i \in [n] : A_i = Q_i - \frac{1}{n} \sum_{j=1}^n Q_j$  and  $Q_i$  is the predicted value of  $\text{Concat}(s, a_i)$ . Finally, DAPO fits the policy ratio to the advantage to optimize the generation of reasoning steps.

**High training variance induced by response-level algorithms.** Response-level algorithms (such as DPO [42], GRPO [48], ReMax[30] and etc) treat the complete response as a unified whole and the gradient directions across all reasoning steps within the response are governed by the final reward uniformly. Although response-level algorithms have recently achieved great success, such as Deepseek-R1 [10], the sparsity of the reward indeed introduces more training variance and addressing the training variance introduced by it would lead to a larger performance improvement. See Section D for detailed discussions.

**Bad value estimation in actor-critic methods.** When applying standard actor-critic methods [46, 15, 12] to LLMs, due to the sparsity of the reward and vast generation policy, the critic often fails not only to learn a good semantic representation but also to provide the fine-grained credit assignment for policy optimization, leading to the poor performance of online actor-critic methods [25, 62] such as PPO [46]. Besides, these methods train the actor and critic simultaneously and the regression targets of both actor and critic are non-stationary and change in response to the updates of the other during the training process. This interwoven update mechanism also leads to an unstable training process that is prone to collapse [27, 12] especially when the critic is not a good approximation of the true value function.

In order to overcome these challenges, we introduce Direct Advantage-Based Policy Optimization (DAPO), a new step-level offline actor-critic method designed to enhance LLM’s performance in reasoning tasks. In order to reduce the training variance induced by the sparsity of the reward, we use a critic function to provide dense training signals. To ensure that the critic function is well-trained, DAPO trains the critic individually before policy optimization to ensure it is a good approximation of the true state value function. The procedure of DAPO is visualized in Figure 1. We summarize our contributions as follows:

- We introduce DAPO, a new step-level, offline RL method that learns from all generated samples. DAPO optimizes the generation of the reasoning steps by sampling multiple candidate steps for each intermediate step  $s$  and updating the policy according to the

advantages of each candidate step. Our main theoretical result (see Theorem 3.2) shows that DAPO will produce a better policy until the reference policy is already optimal.

- We conduct extensive experiments to demonstrate the effectiveness of DAPO. DAPO improves the performance on multiple mathematical and coding benchmarks. Our results empirically shows that DAPO consistently outperforms both the response-level baseline GRPO and the actor-critic baseline PPO on multiple base models .

## 2 Preliminaries

In this section, we introduce the mathematical formulation of RL problem we studied and some important related theoretical results.

**RL reasoning.** The objective for RL reasoning can be formulated as

$$\max_{\pi} \mathbb{E}_{x \sim \mu} [\mathbb{E}_{y \sim \pi(\cdot|x)} [r(x, y)] - \beta \text{KL}(\pi_x, \pi_x^{\text{ref}})], \quad (1)$$

where  $\mu$  is the distribution of training prompt  $x$ ,  $y$  is the response sampled from policy  $\pi(\cdot|x)$ ,  $r(x, y) \in \{0, 1\}$  is the binary reward for the correctness of the response,  $\pi_x, \pi_x^{\text{ref}}$  are short for  $\pi(\cdot|x), \pi_{\text{ref}}(\cdot|x)$  respectively,  $\pi_{\text{ref}}$  is the reference policy,  $\beta \geq 0$  is the coefficient of KL regularization.

**Equivalency to multi-step decision problem.** We use the line break \n as the delimiter for reasoning steps. By doing so, a response  $y$  can be divided into multiple reasoning steps, i.e.  $y = (a_0, \dots, a_{T-1})$ , where  $\forall t \geq 0 a_t$  is the reasoning step and  $T$  is the total number of steps. Thus one can easily show that the objective 1 is equivalent to following step-level decision problem:

$$\max_{\pi} \mathbb{E}_{x \sim \mu} \left[ \mathbb{E}_{\forall t: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} r(s_t, a_t) - \beta \text{KL}(\pi_{s_t}, \pi_{s_t}^{\text{ref}}) | s_0 = x \right] \right], \quad (2)$$

where  $s_t := \text{Concat}(x, a_0, \dots, a_{t-1})$  is the state (context prefix), the reasoning step  $a_t$  is sampled from policy  $\pi(\cdot|s_t)$  (here we use  $\pi$  to denote both step generation and response generation policy with a slightly abuse of the notation),  $\pi_{s_t}, \pi_{s_t}^{\text{ref}}$  are short for  $\pi(\cdot|s_t), \pi_{\text{ref}}(\cdot|s_t)$  respectively and the reward function  $r$  satisfies

$$r(s_t, a_t) = \begin{cases} r(x, y) & t = T-1 \\ 0 & t < T-1. \end{cases}$$

For any state  $s \in \mathcal{S}$ , we define as

$$V_{\beta}^{\pi}(s) := \mathbb{E}_{\forall t: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} r(s_t, a_t) - \beta \text{KL}(\pi_{s_t}, \pi_{s_t}^{\text{ref}}) | s_0 = s \right]. \quad (3)$$

Thus the objective (2) is equivalent to finding a policy  $\pi$  that maximizes the state value function, i.e.,

$$\max_{\pi} \mathbb{E}_{x \sim \mu} [V_{\beta}^{\pi}(x)]. \quad (4)$$

We denote  $\pi_{\beta}^*$  as the optimal policy of objective (4) and  $V_{\beta}^* := V_{\beta}^{\pi_{\beta}^*}$  as the state value function of optimal policy. We define the state space  $\mathcal{S}$  as the set of all possible states that can be encountered in the response  $y$  and the action space  $\mathcal{A}$  as the set of all possible singular reasoning steps. For simplicity of analysis, we introduce the following assumptions in this work.

**Assumption 2.1.** We assume there exists a constant  $H_1 < \infty$  such that for any policy  $\pi$  and initial state  $s_0 \in \mathcal{S}$ , the random decision horizon  $T$  satisfies  $T < H_1$  almost surely. We also assume that there exists a constant  $H_2 < \infty$  such that for any singular reasoning step  $a \in \mathcal{A}$ , the token length of  $a$  is no greater than  $H_2$  so that the state space  $\mathcal{S}$  and action space  $\mathcal{A}$  are both finite.

**More value functions.** We define the KL-constrained state-action value function as

$$Q_{\beta}^{\pi}(s, a) := r(s, a) + V_{\beta}^{\pi}(s \circ a), \quad (5)$$

where  $s \circ a := \text{Concat}(s, a)$ . The advantage function  $A_{\beta}^{\pi}$  is defined as

$$A_{\beta}^{\pi}(s, a) := Q_{\beta}^{\pi}(s, a) - V_{\beta}^{\pi}(s) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)}. \quad (6)$$

We denote  $Q_{\beta}^*(s, a) := r(s, a) + V_{\beta}^*(s)$ . For unregularized state value functions, i.e.  $\beta = 0$ , we use  $V^{\pi}, Q^{\pi}, A^{\pi}$  instead of  $V_0^{\pi}, Q_0^{\pi}, A_0^{\pi}$  to be aligned with the standard RL literatures.

**Bellman operators** For arbitrary policy  $\pi \in \Pi$  and any function  $V : \mathcal{S} \rightarrow \mathbb{R}$ , the Bellman operator  $\mathcal{T}_\beta^\pi$  is defined as

$$\forall s \in \mathcal{S} : [\mathcal{T}_\beta^\pi V](s) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[ r(s, a) + V(f(s, a)) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)} \right]. \quad (7)$$

**Performance difference lemma.** Following is the key theoretical analysis tool used in this work.

**Lemma 2.2.** For arbitrary two policies  $\pi, \tilde{\pi} \in \Pi$  and any distribution  $\rho \in \Delta(\mathcal{S})$ ,

$$V_\beta^\pi(\rho) - V_\beta^{\tilde{\pi}}(\rho) = \mathbb{E}_{s \sim d_\rho^\pi} [\mathcal{T}_\beta^\pi V_\beta^{\tilde{\pi}}(s) - V_\beta^{\tilde{\pi}}(s)], \quad (8)$$

where  $V_\beta^\pi(\rho) := \mathbb{E}_{s \sim \rho} [V_\beta^\pi(s)]$  and the visitation measure  $d_\rho^\pi$  is defined as,

$$d_\rho^\pi(s) := \mathbb{E}_{\forall t: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} \mathbf{1}\{s_t = s\} | s_0 \sim \rho \right].$$

The proof of this lemma is deferred to the Appendix C. We also put some additional theoretical results about MDP on Appendix B.

**Remark 2.1.** Note that our definition of visitation measure,  $d_\rho^\pi$  is not standard as the one in [1, 33]. The reason is that the planning horizon in problem (2), i.e.  $T$ , is random while it is the constant  $\frac{1}{1-\gamma}$  in [1, 33]. Thus we do not try to normalize  $d_\rho^\pi$  to a probability distribution by dividing  $T$  directly.

### 3 Direct Advantage-Based Policy Optimization

In this section, we present the details of DAPO. We only consider the case where a binary reward  $r(s, a) \in \{0, 1\}$  is assigned in the terminal state since it is the most common situation in the LLMs setting. We first present the objective of policy optimization with related theoretical results in Section 3.1 assuming that one has access to the true value function.

#### 3.1 Policy objective

We first derive the policy optimization objective in one single iteration provided with the reference policy  $\pi_{\text{ref}}$ . In the perspective of policy optimization, the most important quantity may be the advantage function. Recall Lemma 2.2. The performance difference for any policy  $\pi \in \Pi$  w.r.t  $\pi_{\text{ref}}$  on arbitrary  $\rho \in \Delta(\mathcal{S})$  is measured as

$$V_\beta^\pi(\rho) - V_\beta^{\pi_{\text{ref}}}(\rho) := \mathbb{E}_{s \sim d_\rho^\pi} \left[ \underbrace{\mathcal{T}_\beta^\pi V_\beta^{\pi_{\text{ref}}}(s) - V_\beta^{\pi_{\text{ref}}}(s)}_{:= \mathcal{I}_s(\pi, \pi_{\text{ref}})} \right],$$

where  $\mathcal{I}_s(\pi, \pi_{\text{ref}})$  can be regarded as the policy improvement at state  $s$ . By the definition of Bellman operator  $\mathcal{T}_\beta^\pi$  in (7) and that of function  $V_\beta^\pi$  in (3), one can easily show that

$$\mathcal{I}_s(\pi, \pi_{\text{ref}}) = \mathbb{E}_{a \sim \pi(\cdot|s)} \left[ A^{\pi_{\text{ref}}}(s, a) - \beta \log \frac{\pi(a|s)}{\pi_{\text{ref}}(a|s)} \right].$$

A natural idea that ensures  $V_\beta^\pi(\rho) - V_\beta^{\pi_{\text{ref}}}(\rho) \geq 0$  might be to increase  $\mathcal{I}_s(\pi, \pi_{\text{ref}})$  on as many states  $s$  as possible. For ease of understanding, we first focus on how to increase  $\mathcal{I}_s(\pi, \pi_{\text{ref}})$  at an arbitrary state  $s \in \mathcal{S}$ . Considering parameterized policy  $\pi_\theta$ , the most directly and easily method may be to do multi gradient ascent steps w.r.t  $\mathcal{I}_s(\pi_\theta, \pi_{\text{ref}})$ , i.e.,

$$\forall K \in \mathbb{N}^+, \theta_K = \theta_0 + \eta \sum_{k=0}^{K-1} \nabla_\theta \mathcal{I}_\theta(\pi_{\theta_k}, \pi_{\text{ref}}), \quad (9)$$

where  $\eta$  is the step size. Following lemma gives an equivalent form of  $\nabla_\theta \mathcal{I}_s(\pi_{\theta_k}, \pi_{\text{ref}})$ .

**Lemma 3.1.** *If we define the surrogate function*

$$\mathcal{H}_s^k(\theta) := \frac{1}{2} \mathbb{E}_{a \sim \pi_{\theta_k}(\cdot|s)} \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_\theta(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2,$$

then  $\nabla_\theta \mathcal{I}_s(\pi_{\theta_k}, \pi_{\text{ref}}) = -\beta \nabla_\theta \mathcal{H}_s^k(\theta)$ .

By Lemma 3.1, the update (9) is equivalent to

$$\theta_K = \theta_0 - \eta \beta \sum_{k=0}^{K-1} \nabla_\theta \mathcal{H}_s^k(\theta_k). \quad (10)$$

We now turn to the offline policy optimization problem setting for the sake of implementation simplicity and data efficiency. A natural question is that can we implement (10) approximately in an offline dataset without sampling the on-policy rollouts at each iteration? Our answer is affirmative. We find that (which will be proved latter), just solving the problem

$$\min_{\theta} \frac{1}{2} \mathbb{E}_{a \sim \nu(\cdot|s)} \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_\theta(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2, \quad (11)$$

where  $\nu \in \Delta(\mathcal{A})$  can be any exploratory sampling distribution, can still yield a policy that  $\mathcal{I}_s(\pi_\theta, \pi_{\text{ref}}) > 0$  unless  $\pi_{\text{ref}} = \pi_\beta^*$ . Taking the consideration of optimizing multiple states simultaneously (weighted by a state sampling distribution  $\nu_S$ ), the objective of DAPO is

$$\min_{\theta} \mathcal{L}(\theta) := \frac{1}{2} \mathbb{E}_{s \sim \nu_S, a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_\theta(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2 \right] \quad (12)$$

**Remark 3.1.** *We give a detailed discussion in Section D on how DAPO is specifically designed to improve multi-step reasoning abilities of LLMs, thereby distinguishing it from response-level optimization algorithms such as DPO [42], GRPO [48] and etc.*

Notice that the state space  $\mathcal{S}$  and the action space  $\mathcal{A}$  are both finite under the assumption 2.1. Following theorem establish the validity of Loss (12) and shows that the solution of (12) is a better policy w.r.t  $\pi_{\text{ref}}$  on objective (4).

**Theorem 3.2** (Monotonic improvement). *Suppose  $\forall s \in \mathcal{S}, a \in \mathcal{A}, \nu_S(s) > 0, \nu_{\mathcal{A}}(a|s) > 0, \pi_{\text{ref}}(a|s) > 0$ . Let  $\pi^+$  be the solution in (12). Then for any state  $s \in \mathcal{S}$ , there exists a function  $\lambda_s : \Delta(\mathcal{A}) \rightarrow [0, +\infty)$  such that*

$$V_\beta^{\pi^+}(\mu) - V_\beta^{\pi_{\text{ref}}}(\mu) \geq \mathbb{E}_{s \sim d_{\mathcal{D}}^{\pi^+}} [\lambda_s(\nu_{\mathcal{A}})] \geq 0.$$

The equality holds if and only if  $\pi_{\text{ref}} = \pi_\beta^*$ .

**Remark 3.2.** *The function  $\lambda_s$  serves as a lower bound of the policy improvement  $\mathcal{I}_s(\pi^+, \pi_{\text{ref}})$  in each states  $s$ . Since Theorem 3.2 is established for arbitrary exploratory  $\nu_{\mathcal{A}}$ ,  $\lambda_s$  should be a function of  $\nu_{\mathcal{A}}$ . For the case  $\nu_{\mathcal{A}}(\cdot|s) = \pi_{\text{ref}}(\cdot|s)$ , one can show that*

$$\lambda_s(\pi_{\text{ref}}) = \beta \cdot \text{KL}(\pi_{\text{ref}}(\cdot|s), \pi^+(\cdot|s)),$$

by combining (20) and (21) in the proof of Theorem 3.2. It remains as a great interest for us to derive a more precise expression of  $\lambda_s$  and figure out which  $\nu_{\mathcal{A}}$  should yield biggest lower bound.

Theorem 3.2 implies that DAPO can improve the performance until there is no room for improvement in the trust region of  $\pi_{\text{ref}}$ . Theorem 3.2 also establishes the monotonic improvement property on unregularized state value function. This can be verified by

$$V^{\pi^+}(\mu) = V_\beta^{\pi^+}(\mu) + \mathbb{E}_{s \sim d_\mu^{\pi^+}} [\text{KL}(\pi^+(\cdot|s), \pi_{\text{ref}}(\cdot|s))] \geq V_\beta^{\pi^+}(\mu) \geq V_\beta^{\pi_{\text{ref}}}(\mu) = V^{\pi_{\text{ref}}}(\mu).$$

**Implementation of DAPO.** In practice, DAPO can be implemented by minimizing (12) on an offline training dataset  $\mathcal{D} = \left\{ (s_i, a_i, \hat{A}_i) \right\}_{i=1}^N$ , where  $s_i \sim \nu_S$ ,  $a_i \sim \nu_{\mathcal{A}}(\cdot|s)$  and  $\hat{A}_i$  is an estimation of true advantage  $A^{\pi_{\text{ref}}}(s_i, a_i)$ . The whole procedure can be found in Algorithm 1.

**Advantage estimation.** Recall the definition of Q function in (5) and advantage function in (6). For non-terminal states, one has  $r(s, a) = 0$  and

$$A^{\pi_{\text{ref}}}(s, a) = Q^{\pi_{\text{ref}}}(s, a) - V^{\pi_{\text{ref}}}(s) = V^{\pi_{\text{ref}}}(s \circ a) - \mathbb{E}_{a' \sim \pi_{\text{ref}}(\cdot|s)} [V^{\pi_{\text{ref}}}(s \circ a')].$$

Thus in our implementation, we sample multiple actions  $\{a_i\}_{i=1}^M$  for each state  $s$  from  $\pi_{\text{ref}}(\cdot|s)$  and estimate the advantage of each action by

$$\forall i \in [m] : \hat{A}(s, a_i) := V_\phi(s \circ a_i) - \frac{1}{M} \sum_{j=1}^M V_\phi(s \circ a_j). \quad (13)$$

Here  $V_\phi \approx V^{\pi_{\text{ref}}}$  is a pretrained critic function. Since (13) is fully determined by the critic  $V_\phi$ , thus the training of  $V_\phi$  plays a crucial role for the optimization of training states. In the following, we introduce the optimization method of  $V_\phi$ .

### 3.2 Critic objective

Generally speaking, the critic optimization problem for any target policy  $\pi$  can be formulated as

$$\min_{\phi} \mathbb{E}_{s \sim \mathcal{D}} [\mathcal{L}(V_\phi(s), V^\pi(s))], \quad (14)$$

here  $\mathcal{L}$  is a sample-wise loss function,  $\mathcal{D}$  is the distribution of training states. However, since the target  $V^{\pi_{\text{ref}}}(s)$  can not be accessed to in advance, we need to construct the estimate of  $V^\pi$  as the training samples for the optimization of  $V_\phi$ . One can show that

$$V^\pi(s) = \mathbb{E} \left[ \sum_{t=0}^{T-1} r(s_t, a_t) | s_0 = s \right] = \mathbb{P}^\pi(r(s_{T-1}, a_{T-1}) = 1 | s_0 = s).$$

Thus, one can firstly use a behavior policy  $\pi_{\text{gen}}$  (also called generator) to generate a set of training states  $s$ . Then for each training state  $s$ , one can use  $\pi$  as the completer to sample  $N$  sub-trajectories from  $s$ ,  $\left\{ \tau_i = (s, a_0^{(i)}, a_1^{(i)}, \dots, a_{T_i-1}^{(i)}) \right\}_{i=1}^N$ , and construct the empirical MC mean,

$$\text{MC}_N(s) := \frac{1}{N} \sum_{i=1}^N r(s_{T_i-1}, a_{T_i-1}),$$

as the estimation of  $V^\pi(s)$ , which is guaranteed to converge to  $V^\pi(s)$  as  $n \rightarrow \infty$  by the strong law of large numbers. Notice that the state value  $V^\pi(s)$  is a probability. Thus we use the binary cross-entropy loss in our implementation, i.e.,

$$\mathcal{L}_{\text{BCE}}(y, y_{\text{pred}}) = -(y \log(y_{\text{pred}}) + (1-y) \log(1-y_{\text{pred}}))$$

as the sample-wise loss function  $\mathcal{L}$  in (14) to avoid gradient diminishing if a MSE loss is applied (which is more common in RL literatures). Finally, the critic optimization objective is

$$\min_{\phi} \mathbb{E}_{s \sim \mathcal{D}} [\mathcal{L}_{\text{BCE}}(\text{MC}_N(s), V_\phi(s))]. \quad (15)$$

**Remark 3.3** (Should we treat  $V_\phi$  as a PRM?). *It's worthy to note that the critic training method we presented above is mostly aligned with [56]. However, we argue that the  $V_\phi$  should be treated as a value function other than a process reward model. Here are the reasons : 1) From the optimization perspective,  $V_\phi$  is trained to approximate the true state value of the completer. 2) From the RL perspective, using  $V_\phi$  as a reward function will deviate from the original RLHF optimization objective and easily lead to a serve reward hacking issue on the number of reasoning steps as shown in [13].*

## 4 Experiments

In this section, we conduct DAPO on mathematical and coding datasets individually. The experimental setup is detailed in Section 4.1, and specific conclusions and benchmark results of our experiments are presented in Section 4.2.

Model	In Domain MATH	Out of Domain			
		GSM8K	Minerva MATH	Olympiad Bench	College Math
<b>SFT Models</b>					
Skywork-Math-Llama + DAPO	41.90 46.88 <sup>+4.98</sup>	61.49 67.55 <sup>+6.06</sup>	5.51 7.34 <sup>+1.82</sup>	18.68 22.38 <sup>+3.70</sup>	24.85 25.87 <sup>+1.03</sup>
Llama-3.1-8B-Instruct + DAPO	49.42 53.62 <sup>+4.20</sup>	85.29 86.74 <sup>+1.45</sup>	26.48 23.54 <sup>-2.94</sup>	16.14 20.01 <sup>+3.88</sup>	30.91 30.91 <sup>+1.63</sup>
OpenO1-Llama-8B-v0.1 + DAPO	52.73 60.33 <sup>+7.60</sup>	85.99 88.77 <sup>+2.78</sup>	29.04 29.42 <sup>+0.39</sup>	19.86 24.44 <sup>+4.59</sup>	29.10 32.12 <sup>+3.02</sup>
Qwen2.5-72B-instruct + DAPO	82.90 84.70 <sup>+1.80</sup>	95.40 95.70 <sup>+0.30</sup>	46.30 50.00 <sup>+3.70</sup>	45.45 47.70 <sup>+2.25</sup>	43.00 43.40 <sup>+0.40</sup>
<b>RL Models</b>					
Qwen2-Math-7B-Instruct + DAPO	74.46 75.41 <sup>+0.95</sup>	89.38 89.45 <sup>+0.07</sup>	40.07 37.51 <sup>-2.56</sup>	34.36 37.03 <sup>+2.67</sup>	41.59 42.23 <sup>+0.64</sup>
Skywork-O1-Open-Llama3.1-8B + DAPO	78.10 79.81 <sup>+1.71</sup>	91.64 92.81 <sup>+1.17</sup>	26.10 29.77 <sup>+3.67</sup>	43.11 44.76 <sup>+1.65</sup>	40.40 40.26 <sup>-0.14</sup>
Qwen2.5-Math-7B-Instruct + DAPO	83.42 84.86 <sup>+1.44</sup>	95.78 96.14 <sup>+0.36</sup>	40.06 41.56 <sup>+1.50</sup>	38.96 41.25 <sup>+2.29</sup>	42.65 41.97 <sup>-0.67</sup>

Table 1: Performance of models optimized via DAPO on mathematical benchmarks. We use zero-shot prompting and greedy decoding for all evaluations in the table. Across a wide range of model families and out-of-domain math word problem benchmarks, DAPO achieves steady improvement with only 7.5K training samples from MATH [18].

SFT Model	Out of Domain				
	HumanEval	HumanEval+	MBPP	MBPP+	LiveCodeBench
Llama-3.1-8B-Instruct + DAPO	72.0 75.0 <sup>+3.0</sup>	66.5 68.9 <sup>+2.4</sup>	72.0 77.0 <sup>+5.0</sup>	56.9 66.1 <sup>+9.2</sup>	18.8 20.9 <sup>+2.1</sup>
OpenO1-Llama-8B-v0.1 + DAPO	69.5 72.0 <sup>+2.5</sup>	61.0 64.6 <sup>+3.6</sup>	69.8 75.9 <sup>+6.1</sup>	58.7 63.8 <sup>+5.1</sup>	16.1 13.7 <sup>-2.4</sup>

Table 2: Performance of models optimized via DAPO on code benchmarks. We use zero-shot prompting and greedy decoding for all evaluations in the table. DAPO consistently improves performance when starting with both regular and reasoning LLMs, even when only 4K training samples from TACO [29] is used.

#### 4.1 Experimental setup

**Training dataset.** For all the DAPO experiments on mathematics, we utilize only the 7500 training problems from the dataset MATH [18] to generate the advantage datasets for DAPO training, requiring no additional human annotations. In particular, we only use the questions and the corresponding golden answers in the dataset while the provided solutions are not used for training. For coding experiments, For coding experiments, we subsample the TACO [29] dataset to compile approximately 4,000 competition-level programming questions derived from real-world scenarios. We utilize its provided unit test cases to evaluate whether a solution is accurate.

**Base models.** Considering the reproducibility and of DAPO, our experiments are taken over of several open-source language models including general and math-specific models. For general models, we consider Llama-3.1-8B-Instruct [52], OpenO1-Llama-8B-v0.1 [39], Skywork-O1-Open-Llama3.1-8B [40], Qwen2.5-72B-Instruct [51]. For math-specific models, we consider Skywork-Math-Llama [63], Qwen2-Math-7B-Instruct [59] and Qwen2.5-Math-7B-Instruct [60]. Notice that Skywork-O1-Open-Llama3.1-8B, Qwen2-Math-7B-Instruct and Qwen2.5-Math-7B-Instruct are already trained by RL algorithms [48]. Thus we conduct continue-RL training on three models by DAPO.

**Benchmarks & Metrics.** For mathematical experiments, we evaluate performance on English mathematical benchmarks. In addition to the 5000 test problems from MATH [18], We also add 4 out-of-domain benchmarks, GSM8K [9], Minerva Math [28], Olympiad Bench [17] and College Math [50], to test the performance generalization of DAPO. For coding experiments, we evaluate DAPO on several widely-used benchmarks in code generation, i.e., HumanEval [7], HumanEval+ [22], MBPP [4], MBPP+ [5] and LiveCodeBench [21]. All the evaluations are conducted in a zero-shot greedy

Model	Base	+GRPO	+PPO	+DAPO
Llama-3.1-8B-Instruct	49.42	52.28 <sup>+2.86</sup>	52.41 <sup>+2.99</sup>	<b>53.62<sup>+4.20</sup></b>
OpenO1-Llama-8B-v0.1	52.73	55.63 <sup>+2.90</sup>	54.12 <sup>+1.39</sup>	<b>60.33<sup>+7.60</sup></b>
Qwen2-Math-7B-Instruct	74.46	74.94 <sup>+0.46</sup>	74.93 <sup>+0.47</sup>	<b>75.41<sup>+0.95</sup></b>
Qwen2.5-Math-7B-Instruct	83.42	84.33 <sup>+0.91</sup>	83.76 <sup>+0.34</sup>	<b>84.86<sup>+1.44</sup></b>

Table 3: Comparison with baseline methods on the 5K test samples from MATH [18]. DAPO consistently outperforms both the initial model and model optimized with GRPO and PPO.

sampling (i.e. temperature 0) with a cot prompt template and a maximum amount of 2048 newly generated tokens (except 4096 for OpenO1-Llama-8B-v0.1 and Skywork-O1-Open-Llama3.1-8B).

**DAPO implementations.** For each DAPO experiment, we first use the base model to generate 32 solutions for each training problem and then 6 solutions are selected out of them while making the correct solutions and wrong solutions as balanced as possible. For each selected solution, we use one line break \n to segment the solution steps and use the base model as the completer to do 16 completions for each reasoning step. Then the MC estimations are constructed for the critic training. The training methodology of critic is presented in Section 3.2. In our implementation, the critics are fine-tuned for one epoch on Qwen2.5-Math-7B-Instruct for mathematical experiments and Qwen2.5-coder-7B-Instruct for coding experiments, with a learning rate of 5e-6 and batch size 512. After that, we extract the next steps (action) for each intermediate step (state) in the completion datasets and compute the advantage using the critic for DAPO training. For all experiments, we use the global batch size 2048, learning rate 5e-7 and  $\beta = 0.01$  or  $0.02$ . We also summarize some useful tricks applied in all of the DAPO training in Appendix F.1

**Baseline methods.** We compare the performance of DAPO with PPO as the actor-critic method baseline and GRPO as the response-level method baseline on mathematical tasks using the 7.5K problems from MATH [18] as the training queries. We conduct experiments on four base models : Llama-3.1-8B-Instruct, OpenO1-Llama-8B-v0.1, Qwen2-Math-7B-Instruct and Qwen2.5-Math-7B-Instruct. We train PPO and GRPO until the training accuracy curve converges. The more training details of our baseline methods are in Appendix F.2.

## 4.2 Main results

**Math.** Our results, presented in Table 1, Table 3 and Table 4.2, demonstrate that DAPO consistently enhances the performance of the base model across all tested models on the in-domain benchmark MATH [18], through DAPO training, Skywork-Math-Llama, Llama-3.1-8B-Instruct, OpenO1-Llama-8B-v0.1, and Qwen2.5-7B-instruc achieve 50.54%, 53.62%, 60.27%, 84.55% greedy decoding accuracy respectively. For RL models, Qwen2-Math-7B-Instruct, Skywork-O1-Open-Llama3.1-8B and Qwen2.5-Math-7B-Instruct achieve 76.40%, 79.81%, and 84.86% greedy decoding accuracy respectively, indicating that DAPO can further enhance the performance even the base model are already trained by RL methods previously. Table 3 shows that DAPO has a greater improvement than PPO and GRPO on various base models in our experiments. In Figure 3, we present the accuracy curve on the MATH TEST during the training process of DAPO. As can be clearly observed, DAPO steadily enhances the accuracy throughout the training until it stabilizes.

**Code generation.** Our results for the code generation task are shown in Table 2. DAPO improves the coding performance of both Llama-3.1-8B-Instruct and OpenO1-Llama-8B-v0.1 on multiple wildly used benchmarks. For Llama-3.1-8B-Instruct, DAPO has increased by 3.0% and 2.4% respectively on HumanEval [7] and HumanEval+ [22], by 5.0% and 9.2% respectively on MBPP [4] and MBPP+ [5], and by 2.1% on LiveCodeBench [21]. For OpenO1-Llama-8B-v0.1, DAPO has increased by 2.5% and 3.6% respectively on HumanEval [7] and HumanEval+ [22], by 6.1% and 5.0% respectively on MBPP [4] and MBPP+ [5], and decreased by 2.4% on LiveCodeBench [21]. Overall, these results strongly demonstrate the effectiveness of DAPO.

**On the iterative DAPO.** As suggested by our theoretical results (Theorem 3.2), there still remains the room for performance improvement if  $\pi_{\text{ref}}$  is not the optimal policy. Thus we conduct two iterative DAPO experiments on Skywork-Math-Llama and Qwen2-Math-7B-Instruct. At iteration 2, we use the model optimized by DAPO at iteration 1 as  $\pi_{\text{ref}}$ . Our experiment results are summarized in Table

4.2 and Table 10. We can conclude from both tables that DAPO can further improves the performance of Skywork-Math-Llama and Qwen2-Math-7B-Instruct.

Model	Base	+DAPO iter1	+DAPO iter2
Skywork-Math-Llama	41.90	46.88 <sup>+4.98</sup>	<b>50.54</b> <sup>+8.64</sup>
Qwen2-Math-7B-Instruct	74.46	75.41 <sup>+0.95</sup>	<b>76.40</b> <sup>+1.94</sup>

Table 4: Performance of iterative DAPO on 5K test samples from MATH [18]. Both Skywork-Math-Llama and Qwen2-Math-7B-Instruct show score improvement with additional iterations.

**Ablation studies of hyperparameters.** We conduct some preliminary ablation studies on the different components of DAPO. For the choice of KL coefficient  $\beta$ , we test different  $\beta$  in the experiments on Skywork-Math-Llama and the results are reported in Table 4.2. It can be seen that a suitable choice of  $\beta$  yields best improvement. We find  $\beta = 0.01$  generally performs well on various models. Regarding the number of completions  $n$  used for MC value estimation, we test  $n = 8$  and  $n = 16$  in

Model \ $\beta$	Base	$\beta = 0.002$	$\beta = 0.01$	$\beta = 0.02$	$\beta = 0.05$	$\beta = 0.1$
Skywork-Math-Llama	41.90	44.52	<b>46.88</b>	46.70	45.56	44.50

Table 5: Performance of DAPO on 5K test samples from MATH [18] with different  $\beta$ .

the experiments on Qwen2-Math-7B-Instruct and the results are reported in Table 4.2. We find that the difference is not significant between  $n = 8$  and  $n = 16$  in our experiment. However, to make sure a larger performance gain as possible, we use  $n = 16$  in all our experiments.

Model \ $n$	Base	$n = 8$	$n = 16$
Qwen2-Math-7B-Instruct	74.46	75.30	<b>75.41</b>

Table 6: Performance of DAPO on 5K test samples from MATH [18] with different number of completions for MC value estimation.

**On the computation cost of DAPO.** One major limitation that may hinder the efficient training-time scaling of DAPO is the high computation cost, particularly during the critic pre-training stage. We leave it as a future work for us to find a more efficient and scalable method for critic training. Please refer Section E for more discussions. That being said, given similar computation resources budget to the response-level baseline method, i.e. GRPO, we find that DAPO still outperforms GRPO in our experiment. We run GRPO experiment on Meta-Llama-3.1-8B-Instruct to more than 1000 training steps. The training and test curves can be found in Figure 4. It can be observed that the model exhibits serve overfitting after around 300 steps and the test performance begins to drop quickly. In contrast, as shown in Figure 3, DAPO improves the test performance steadily without exhibiting overfitting, ultimately achieving better performance gains compared to GRPO. This validates the effectiveness of granular policy optimization enabled by estimating step-level advantages.

## 5 Conclusion

In this work, we propose an offline step-level RLHF method called Direct Advantage-Based Policy Optimization (DAPO), which aims to optimize the generation of reasoning steps. DAPO significantly improves performance on both mathematical and coding benchmarks, demonstrating its effectiveness. Compared with response-level algorithms, DAPO leverages the critic function for more fine-grained policy optimization. Compared with standard actor-critic methods, DAPO separates the training of the actor and critic into two distinct stages, stabilizing the RL training process while obtaining a good value function estimation. As we discussed before, the main limitation of DAPO is the high computation cost, indicating a need to find a more efficient implementation method. Besides, it also remains a great interest to see whether using a larger set of high-quality training queries would yield greater performance improvements.

## References

- [1] Alekh Agarwal, Sham M. Kakade, Jason D. Lee, and Gaurav Mahajan. On the theory of policy gradient methods: Optimality, approximation, and distribution shift. *Journal of Machine Learning Research*, 22(98):1–76, 2021.
- [2] Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. *arXiv preprint arXiv:2402.14740*, 2024.
- [3] Hessa Abdulrahman Alawwad, Areej Alhothali, Usman Naseem, Ali Alkhathlan, and Amani Jamal. Enhancing textbook question answering task with large language models and retrieval augmented. *arXiv:2402.05128*, 2024.
- [4] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.
- [5] Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and et al. Program synthesis with large language models. *arXiv:2108.07732*, 2021.
- [6] Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, Scott Johnston, Shauna Kravec, Liane Lovitt, Neel Nanda, Catherine Olsson, Dario Amodei, Tom Brown, Jack Clark, Sam McCandlish, Chris Olah, Ben Mann, and Jared Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv:2204.05862*, 2022.
- [7] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and et al. Evaluating large language models trained on code. *arXiv:2107.03374*, 2021.
- [8] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. *ArXiv*, abs/2304.05128, 2023.
- [9] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv:2110.14168*, 2021.
- [10] DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shiron 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, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv:2501.12948*, 2025.
- [11] Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *arXiv preprint arXiv:2304.06767*, 2023.
  - [12] Scott Fujimoto, Herke van Hoof, and David Meger. Addressing function approximation error in actor-critic methods. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1582–1591. PMLR, 2018.
  - [13] Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. On designing effective rl reward at training time for llm reasoning. *arXiv:2410.15115*, 2024.
  - [14] Chengxi Li Guoxin Chen, Minpeng Liao and Kai Fan. Step-level value preference optimization for mathematical reasoning. *arXiv:2406.10858*, 2024.
  - [15] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 1856–1865. PMLR, 2018.
  - [16] Ji-Eun Han, Jun-Seok Koh, Hyeon-Tae Seo, Du-Seong Chang, , and Kyung-Ah Sohn. Psydial: Personality-based synthetic dialogue generation using large language models. *arXiv:2404.00930*, 2024.
  - [17] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.
  - [18] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021.
  - [19] Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models. *arXiv:2501.03262v1*, 2025.
  - [20] Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv:2503.24290*, 2025.
  - [21] Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, , and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv:2403.07974*, 2024.
  - [22] Yuyao Wang Jiawei Liu, Chunqiu Steven Xia and Lingming Zhang. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. *arXiv:2305.01210*, 2023.

- [23] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. Swe-bench: Can language models resolve real-world github issues? *arXiv:2310.06770*, 2023.
- [24] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv:2503.01491*, 2025.
- [25] Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment. *arXiv preprint arXiv:2410.01679*, 2024.
- [26] Xin Lai, Zhuotao Tian, Yukang Chen, Senqiao Yang, Xiangru Peng, and Jiaya Jia. Step-dpo: Step-wise preference optimization for long-chain reasoning of llms. *arXiv:2406.18629*, 2024.
- [27] Deep Reinforcement Learning and the Deadly Triad. Hado van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. *arXiv:1812.02648*, 2018.
- [28] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.
- [29] Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv:2312.14852*, 2023.
- [30] Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models. In *Forty-first International Conference on Machine Learning*, 2023.
- [31] Weibin Liao, Xu Chu, and Yasha Wang. Tpo: Aligning large language models with multi-branch multi-step preference trees. *arXiv:2410.12854*, 2024.
- [32] Guanlin Liu, Kaixuan Ji, Renjie Zheng, Zheng Wu, Chen Dun, Quanquan Gu, and Lin Yan. Enhancing multi-step reasoning abilities of language models through direct q-function optimization. *arXiv preprint arXiv:2410.09302*, 2024.
- [33] Jiacai Liu, Wenye Li, Dachao Lin, Ke Wei, and Zhihua Zhang. On the convergence of projected policy gradient for any constant step sizes. *arXiv:2311.01104*, 2024.
- [34] Jiacai Liu, Wenye Li, and Ke Wei. Elementary analysis of policy gradient methods. *arxiv:2404.03372*, 2024.
- [35] Yuliang Liu, Junjie Lu, Zhaoling Chen, Chaofeng Qu, Jason Klein Liu, Chonghan Liu, Zefan Cai, Yunhui Xia, Li Zhao, Jiang Bian, Chuheng Zhang, Wei Shen, and Zhouhan Lin. Adaptivestep: Automatically dividing reasoning step through model confidence. *arXiv:2406.06592*, 2025.
- [36] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv:2503.20783v1*, 2025.
- [37] Zimu Lu, Aojun Zhou, Ke Wang, Houxing Ren, Weikang Shi, Junting Pan, Mingjie Zhan, and Hongsheng Li. Sstep-controlled dpo: Leveraging stepwise error for enhanced mathematical reasoning. *arXiv:2407.00782*, 2024.
- [38] Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. Improve mathematical reasoning in language models by automated process supervision. *arXiv:2406.06592*, 2025.
- [39] O1-OPEN. Open-source o1. <https://huggingface.co/01-OPEN>, 2025. Accessed: 2025-01-30.
- [40] Skywork o1 Team. Skywork-o1 open series. <https://huggingface.co/Skywork>, November 2024.

- [41] Zehan Qi, Xiao Liu, Iat Long Iong, Hanyu Lai, Xueqiao Sun, Wenyi Zhao, Yu Yang, Xinyue Yang, Jiadai Sun, Shuntian Yao, Tianjie Zhang, Wei Xu, Jie Tang, and Yuxiao Dong. Webbrl: Training LLM web agents via self-evolving online curriculum reinforcement learning. *arXiv:2411.02337*, 2024.
- [42] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.
- [43] Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, Aliaksei Severyn, Jonathan Mallinson, Lior Shani, Gil Shamir, Rishabh Joshi, Tianqi Liu, Remi Munos, and Bilal Piot. Offline regularised reinforcement learning for large language models alignment. *arXiv:2405.19107*, 2024.
- [44] Martin A. Riedmiller, Roland Hafner, Thomas Lampe, Michael Neunert, Jonas Degraeve, Tom Van de Wiele, Vlad Mnih, Nicolas Heess, and Jost Tobias Springenberg. Learning by playing solving sparse reward tasks from scratch. In Jennifer G. Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, pages 4341–4350. PMLR, 2018.
- [45] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arxiv:1506.02438*, 2020.
- [46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv:1707.06347*, 2017.
- [47] Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv:2410.08146*, 2024.
- [48] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Haowei Zhang Xiao Bi, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.
- [49] Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*, pages 1057–1063, 1999.
- [50] Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. Mathscale: Scaling instruction tuning for mathematical reasoning. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.
- [51] Qwen Team. Qwen2.5: A party of foundation models, September 2024.
- [52] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, and et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv:2302.13971*, 2023.
- [53] Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q\*: Improving multi-step reasoning for llms with deliberative planning. *arXiv:2406.14283*, 2024.
- [54] Huajie Wang, Shibo Hao, Hanze Dong, Shenao Zhang, Yilin Bao, Ziran Yang, and Yi Wu. Offline reinforcement learning for llm multi-step reasoning. *arXiv:2412.16145*, 2024.
- [55] Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. Mathcoder: Seamless code integration in llms for enhanced mathematical reasoning. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

- [56] Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, 2024.
- [57] Albert Wilcox, Ashwin Balakrishna, Jules Dedieu, Wyame Benslimane, Daniel Brown, and Ken Goldberg. Monte carlo augmented actor-critic for sparse reward deep reinforcement learning from suboptimal demonstrations. *Advances in neural information processing systems*, 35:2254–2267, 2022.
- [58] Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P. Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv:2405.00451*, 2024.
- [59] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.
- [60] An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and et al. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv:2409.12122*, 2024.
- [61] Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*, 2023.
- [62] Yufeng Yuan, Yu Yue, Ruofei Zhu, Tiantian Fan, and Lin Yan. What’s behind ppo’s collapse in long-cot? value optimization holds the secret. *arXiv:2503.01491*, 2025.
- [63] Liang Zeng, Liangjun Zhong, Liang Zhao, Tianwen Wei, Liu Yang, Jujie He, Cheng Cheng, Rui Hu, Yang Liu, Shuicheng Yan, Han Fang, and Yahui Zhou. Skywork-math: Data scaling laws for mathematical reasoning in large language models – the story goes on. *arXiv:2407.08348*, 2024.

## NeurIPS Paper Checklist

The checklist is designed to encourage best practices for responsible machine learning research, addressing issues of reproducibility, transparency, research ethics, and societal impact. Do not remove the checklist: **The papers not including the checklist will be desk rejected.** The checklist should follow the references and follow the (optional) supplemental material. The checklist does NOT count towards the page limit.

Please read the checklist guidelines carefully for information on how to answer these questions. For each question in the checklist:

- You should answer [Yes] , [No] , or [NA] .
- [NA] means either that the question is Not Applicable for that particular paper or the relevant information is Not Available.
- Please provide a short (1–2 sentence) justification right after your answer (even for NA).

**The checklist answers are an integral part of your paper submission.** They are visible to the reviewers, area chairs, senior area chairs, and ethics reviewers. You will be asked to also include it (after eventual revisions) with the final version of your paper, and its final version will be published with the paper.

The reviewers of your paper will be asked to use the checklist as one of the factors in their evaluation. While "[Yes]" is generally preferable to "[No]", it is perfectly acceptable to answer "[No]" provided a proper justification is given (e.g., "error bars are not reported because it would be too computationally expensive" or "we were unable to find the license for the dataset we used"). In general, answering "[No]" or "[NA]" is not grounds for rejection. While the questions are phrased in a binary way, we acknowledge that the true answer is often more nuanced, so please just use your best judgment and write a justification to elaborate. All supporting evidence can appear either in the main paper or the supplemental material, provided in appendix. If you answer [Yes] to a question, in the justification please point to the section(s) where related material for the question can be found.

IMPORTANT, please:

- **Delete this instruction block, but keep the section heading “NeurIPS Paper Checklist”.**
- **Keep the checklist subsection headings, questions/answers and guidelines below.**
- **Do not modify the questions and only use the provided macros for your answers.**

### 1. Claims

Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

Answer: [Yes]

Justification: The claims match our theoretical results (see Theorem 3.2) in Section 3, the experimental results in Section 4 and the discussion in Section D.

Guidelines:

- The answer NA means that the abstract and introduction do not include the claims made in the paper.
- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

### 2. Limitations

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: We discuss the main limitation, i.e. high computational cost, in Section E where we also discuss the potential approaches to address this issue.

Guidelines:

- The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

### 3. Theory assumptions and proofs

Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: The assumptions are stated in Section 2 and all proof are included in the Appendix.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

### 4. Experimental result reproducibility

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: we provide detailed setting of our experiments in Section 4 and Section F. The main code of our methods is included in our supplementary material.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general, releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  - (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  - (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  - (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  - (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

## 5. Open access to data and code

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We use the open-source training dataset and the main code of our methods is included in our supplementary material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so “No” is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (<https://nips.cc/public/guides/CodeSubmissionPolicy>) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.

- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

## 6. Experimental setting/details

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: We provide detailed experimental setting in Section 4 and Section F.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

## 7. Experiment statistical significance

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Our evaluate our experimental results with greedy decoding, i.e. using temperature 0. Thus there is no statistic random error in our experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

## 8. Experiments compute resources

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: We provide sufficient information on the computation cost in Section E.

Guidelines:

- The answer NA means that the paper does not include experiments.

- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

## 9. Code of ethics

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics <https://neurips.cc/public/EthicsGuidelines>?

Answer: [Yes]

Justification: Our research confirms with the NeurIPS Code of Ethics.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

## 10. Broader impacts

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [No]

Justification: It is difficult to discuss the potential societal impacts of our paper.

Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

## 11. Safeguards

Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

## 12. Licenses for existing assets

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We have properly cited all used datasets in this paper.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, [paperswithcode.com/datasets](https://paperswithcode.com/datasets) has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

## 13. New assets

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [NA]

Justification: Our paper does not release new assets.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

## 14. Crowdsourcing and research with human subjects

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: Our paper does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

**15. Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: Our paper does not involve human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

**16. Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorosity, or originality of the research, declaration is not required.

Answer: [NA]

Justification: Our method only involves standard usage of LLMs.

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (<https://neurips.cc/Conferences/2025/LLM>) for what should or should not be described.

## A Related work

By Theorem B.2, the optimal policy  $\pi_\beta^*$  has the analytic solution given by

$$\forall x, y : \log \frac{\pi_\beta^*(y|x)}{\pi_{\text{ref}}(y|x)} = \frac{1}{\beta} (r(x, y) - \hat{V}_{\text{ref}}(x)), \quad (16)$$

where

$$\hat{V}_{\text{ref}}(x) := \beta \log \left\{ \mathbb{E}_{y' \sim \pi_{\text{ref}}(\cdot|x)} \left[ \exp \left\{ \frac{1}{\beta} r(x, y') \right\} \right] \right\}.$$

In order to solve  $\pi_\beta^*$ , DPO [42] use the Bradley-Terry model to parametrize the reward model  $r$  and infers the optimal policy directly from pair-wise data without accessing to  $\hat{V}_{\text{ref}}$ . DRO [43] takes a straight way and solves the optimal policy directly by

$$\min_{\theta} \mathbb{E}_{x,y} \left[ \left( \log \frac{\pi_\theta(y|x)}{\pi_{\text{ref}}(y|x)} - \frac{1}{\beta} (r(x, y) - \hat{V}_{\text{ref}}(x)) \right)^2 \right].$$

one-step However, when it comes to the multi-step RLHF problem (4), the decision horizon  $T$  is generally larger than one. In this case, by Theorem B.2, the optimal policy  $\pi_\beta^*$  satisfies

$$\log \frac{\pi_\beta^*(a|s)}{\pi_{\text{ref}}(a|s)} = \frac{1}{\beta} (r(s, a) + V^*(s') - V_\beta^*(s)), \quad (17)$$

where  $s$  is the current state,  $a$  is the action,  $s' = f(s, a)$  is the successive state and  $V_\beta^*$  is the optimal value function of  $\pi_\beta^*$  defined in (3) that can be accessed in advance. Several policy gradient ascent [49] based methods are applied to solve the optimal policy. PPO use GAE [45] to estimate the policy gradient and a clip trick to keep the training process stable. Considering the computation efficiency, GRPO [48], RLOO [2], Remax [30] remove the critic component and use the sampling rewards to estimate the gradient. [26, 37, 14] adapt the DPO algorithm to the step-level setting, developing a series of methods known as step-DPO. These methods also uses the BT model to parametrize the reward model and infer the optimal policy from (17) without the knowledge of  $V_\beta^*$ . However, these methods still requires the collection of massive pairwise trajectory data. DQO [32] takes the standard actor-critic route like PPO and solves the optimal policy using the Soft Actor-Critic [15](SAC) Method. WebRL [41] directly solves equation (17) by using a MSE loss and replacing  $V^*$  with the value function of some known behavior policy  $\mu$ . OREO [54] also proposes an offline actor-critic methods for LLM multi-step reasoning. [58] and TPO [31] uses MCTS and implicit reward respectively to provide dense signals for policy optimization. [47] use the advantage function of an additional prover policy  $\mu$  as a PRM in the online policy gradient. In the below, we give a holistic comparison between DAPO and other methods in Table 7.

Method	DPO	GRPO	PPO	DQO	OREO	[58]	DAPO
Step-wise optimization	×	×	✓	✓	✓	✓	✓
Step signal provider	NA	NA	critic	critic	critic	MCTS	critic
Learn from listwise samples	×	✓	✓	✓	✓	✗	✓
interwoven update of actor-critic	NA	NA	✗	✗	✗	NA	✓

Table 7: Comparison between DAPO and other methods

A significant difference between DAPO and other actor-critic methods is that DAPO decouples the training of the actor and critic into two distinct stages. This separation offers the following advantages during the actor training stage: **Enhanced training stability**. The non-stationary critic along with the interwoven update between actor and critic bring more instability to the training process. DAPO address this issue by removing the critic in the actor training stage. **Substantially reduced memory usage**. During the actor training stage, DAPO only loads the actor model and the advantage dataset, leading to significantly lower GPU memory consumption compared to PPO, DQO, and OREO.

## B More on MDP

**Bellman Optimality Operators** We define the Bellman optimality operator  $\mathcal{T}_\beta$  as

$$\forall s \in \mathcal{S} : [\mathcal{T}_\beta V](s) := \max_{\pi \in \Pi} [\mathcal{T}_\beta^\pi V](s)$$

**Lemma B.1.** *Given any function  $V : \mathcal{S} \rightarrow \mathbb{R}$ ,*

$$\forall s \in \mathcal{S} : [\mathcal{T}_\beta V](s) = \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} (r(s, a) + V(f(s, a))) \right\} \right] \quad (18)$$

**Optimal policy** In the following we give the optimality conditions of the multi-step RL.

**Theorem B.2** ([1, 34]). *if a policy  $\hat{\pi} \in \Pi$  satisfies*

$$\forall s \in \mathcal{S} : V_\beta^{\hat{\pi}}(s) = [\mathcal{T}_\beta V_\beta^{\hat{\pi}}](s),$$

*then  $\hat{\pi} = \pi_\beta^*$ . Besides, the optimal policy  $\pi_\beta^*$  has the unique analytic form of*

$$\forall s \in \mathcal{S}, a \in \mathcal{A} : \log \frac{\pi_\beta^*(a|s)}{\pi_{\text{ref}}(a|s)} = \frac{1}{\beta} (Q_\beta^*(s, a) - V_\beta^*(s)).$$

## C Proofs

### C.1 Proof of Lemma B.1

By the definition of  $\mathcal{T}_\beta$ , for any state  $s \in \mathcal{S}$ , we define

$$\pi_\beta^V(\cdot|s) \in \arg \max_{p \in \Delta(\mathcal{A})} \mathbb{E}_{a \sim p} \left[ r(s, a) + V(f(s, a)) - \beta \log \frac{p(a)}{\pi_{\text{ref}}(a|s)} \right].$$

One can easily find that

$$\forall s \in \mathcal{S}, a \in \mathcal{A} : \pi_\beta^V(a|s) := \frac{\pi_{\text{ref}}(a|s) \cdot \exp \left\{ \frac{1}{\beta} (r(s, a) + V(f(s, a))) \right\}}{\mathbb{E}_{a' \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} (r(s, a') + V(f(s, a'))) \right\} \right]}$$

and we omit the proof of it here. Then

$$\begin{aligned} \mathcal{T}_\beta V(s) &= \mathcal{T}_\beta^{\pi_\beta^V} V(s) \\ &= \mathbb{E}_{a \sim \pi_\beta^V(\cdot|s)} \left[ r(s, a) + V(f(s, a)) - \beta \log \frac{\pi_\beta^V(a|s)}{\pi_{\text{ref}}(a|s)} \right] \\ &= \mathbb{E}_{a \sim \pi_\beta^V(\cdot|s)} \left[ r(s, a) + V(f(s, a)) - \beta \log \left( \frac{\exp \left\{ \frac{1}{\beta} (r(s, a) + V(f(s, a))) \right\}}{\mathbb{E}_{a' \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} (r(s, a') + V(f(s, a'))) \right\} \right]} \right) \right] \\ &= \beta \log \left( \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} (r(s, a) + V(f(s, a))) \right\} \right] \right) \end{aligned}$$

### C.2 Proof of Lemma 2.2

*Proof.* Recall the definition of state value functions in (3). Then one has

$$V_\beta^\pi(\rho) - V_\beta^{\tilde{\pi}}(\rho) = \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (r(s_t, a_t) - \beta \text{KL}(\pi(\cdot|s_t), \pi_{\text{ref}}(\cdot|s_t))) | s_0 \sim \rho \right] - V_\beta^{\tilde{\pi}}(\rho)$$

$$\begin{aligned}
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (r(s_t, a_t) - \beta \text{KL}(\pi(\cdot|s_t), \pi_{\text{ref}}(\cdot|s_t))) | s_0 \sim \rho \right] \\
&\quad - \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (V_{\beta}^{\tilde{\pi}}(s_t) - V_{\beta}^{\tilde{\pi}}(s_{t+1})) | s_0 \sim \rho \right] \\
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (r(s_t, a_t) - \beta \text{KL}(\pi(\cdot|s_t), \pi_{\text{ref}}(\cdot|s_t)) + V_{\beta}^{\tilde{\pi}}(s_{t+1}) - V_{\beta}^{\tilde{\pi}}(s_t)) | s_0 \sim \rho \right] \\
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s_t) - V_{\beta}^{\tilde{\pi}}(s_t)) | s_0 \sim \rho \right].
\end{aligned}$$

Note that under assumption 2.1, the state space  $\mathcal{S}$  is finite. Thus

$$\begin{aligned}
V_{\beta}^{\pi}(\rho) - V_{\beta}^{\tilde{\pi}}(\rho) &= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s_t) - V_{\beta}^{\tilde{\pi}}(s_t)) \left( \sum_{s \in \mathcal{S}} \mathbb{I}\{s_t = s\} \right) | s_0 \sim \rho \right] \\
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{s \in \mathcal{S}} \sum_{t=0}^{T-1} (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s_t) - V_{\beta}^{\tilde{\pi}}(s_t)) \mathbb{I}\{s_t = s\} | s_0 \sim \rho \right] \\
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{s \in \mathcal{S}} \sum_{t=0}^{T-1} (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s) - V_{\beta}^{\tilde{\pi}}(s)) \mathbb{I}\{s_t = s\} | s_0 \sim \rho \right] \\
&= \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{s \in \mathcal{S}} \mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s) - V_{\beta}^{\tilde{\pi}}(s) \sum_{t=0}^{T-1} \mathbb{I}\{s_t = s\} | s_0 \sim \rho \right] \\
&= \sum_{s \in \mathcal{S}} (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s) - V_{\beta}^{\tilde{\pi}}(s)) \cdot \mathbb{E}_{\forall t \in \mathbb{N}: a_t \sim \pi(\cdot|s_t)} \left[ \sum_{t=0}^{T-1} \mathbb{I}\{s_t = s\} | s_0 \sim \rho \right] \\
&\stackrel{(a)}{=} \sum_{s \in \mathcal{S}} d_{\rho}^{\pi}(s) \cdot (\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s) - V_{\beta}^{\tilde{\pi}}(s)) \\
&= \mathbb{E}_{s \sim d_{\rho}^{\pi}} [\mathcal{T}_{\beta}^{\pi} V_{\beta}^{\tilde{\pi}}(s) - V_{\beta}^{\tilde{\pi}}(s)],
\end{aligned}$$

where (a) leverages the definition of visitation measure  $d_{\rho}^{\pi}$ .  $\square$

### C.3 Proof of Lemma 3.1

*Proof.* Recall that

$$\mathcal{H}_s^k(\pi_{\theta}, \pi_{\text{ref}}) = \frac{1}{2} \mathbb{E}_{a \sim \pi_{\theta_k}(\cdot|s)} \left[ \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2 \right]$$

A direct computation yields that

$$\begin{aligned}
\nabla_{\theta} \mathcal{H}_s^k(\pi_{\theta}, \pi_{\text{ref}}) &= \frac{1}{2} \mathbb{E}_{a \sim \pi_{\theta_k}(\cdot|s)} \left[ \nabla_{\theta} \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2 \right] \\
&= \mathbb{E}_{a \sim \pi_{\theta_k}(\cdot|s)} \left[ \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \nabla_{\theta} \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \right] \\
&= -\mathbb{E}_{a \sim \pi_{\theta_k}(\cdot|s)} \left[ \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \nabla_{\theta} \log \pi_{\theta}(a|s) \right].
\end{aligned}$$

Note that

$$\mathcal{I}_s(\pi_{\theta}, \pi_{\text{ref}}) = \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ A^{\pi_{\text{ref}}}(s, a) - \beta \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right].$$

Then

$$\begin{aligned}
\nabla_{\theta} \mathcal{I}_s (\pi_{\theta}, \pi_{\text{ref}}) &= \nabla_{\theta} \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ A^{\pi_{\text{ref}}} (s, a) - \beta \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right] \\
&= \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ \nabla_{\theta} \log \pi_{\theta}(a|s) \left( A^{\pi_{\text{ref}}}(s, a) - \beta \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \right] \\
&\quad + \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ \nabla_{\theta} \left( A^{\pi_{\text{ref}}}(s, a) - \beta \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \right] \\
&= \beta \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ \nabla_{\theta} \log \pi_{\theta}(a|s) \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \right] \\
&\quad - \beta \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} [\nabla_{\theta} \log \pi_{\theta}(a|s)] \\
&\stackrel{(a)}{=} \beta \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} \left[ \nabla_{\theta} \log \pi_{\theta}(a|s) \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right) \right],
\end{aligned}$$

where (a) is due to

$$\mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} [\nabla_{\theta} \log \pi_{\theta}(a|s)] = \nabla_{\theta} \mathbb{E}_{a \sim \pi_{\theta}(\cdot|s)} [1] = \nabla_{\theta} (1) = 0.$$

Thus it's easy to verify that  $\nabla_{\theta} \mathcal{I}_s (\pi_{\theta_k}, \pi_{\text{ref}}) = -\beta \nabla_{\theta} \mathcal{H}_s^k (\pi_{\theta}, \pi_{\text{ref}})$  and the proof is completed.  $\square$

#### C.4 Proof of Theorem 3.2

*Proof.* Notice that  $\forall s \in \mathcal{S}, \nu_{\mathcal{S}}(s) > 0$ . Then it's obviously that for any state  $s \in \mathcal{S}$ ,

$$\pi^+(\cdot|s) \in \arg \min_{p \in \Delta(\mathcal{A})} \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \left( \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) - \log \frac{p(a)}{\pi_{\text{ref}}(a|s)} \right)^2 \right].$$

Since  $\forall s \in \mathcal{S}, a \in \mathcal{A}, \nu_{\mathcal{A}}(a|s) > 0, \pi_{\text{ref}}(a|s) > 0$ , one has  $\forall s \in \mathcal{S}, a \in \mathcal{A}, \pi^+(a|s) > 0$  otherwise the loss becomes infinity. For ease of notation, we define

$$u_{s,a} := \log \frac{p(a)}{\pi_{\text{ref}}(a|s)} \quad \text{and} \quad A_{s,a} := \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a).$$

Then  $\left\{ u_{s,a}^+ := \log \frac{\pi^+(a|s)}{\pi_{\text{ref}}(a|s)} \mid a \in \mathcal{A} \right\}$  is the solution of the following optimization problem:

$$\begin{aligned}
\min_{u_s := (u_{s,a})_{a \in \mathcal{A}} \in \mathbb{R}^{|\mathcal{A}|}} : \quad \mathcal{L}_s(u_s) &:= \frac{1}{2} \cdot \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} [(A_{s,a} - u_{s,a})^2] \\
\text{s.t.} : \quad \sum_{a \in \mathcal{A}} p(a) &= \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} [\exp \{u_{s,a}\}] = 1.
\end{aligned}$$

Denote the Lagrangian function as

$$\hat{\mathcal{L}}_s(u_s, \lambda) := \mathcal{L}_s(u_s) + \lambda \cdot (\mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} [\exp \{u_{s,a}\}] - 1).$$

Since it's a convex problem, by KKT conditions, there exists a  $\lambda_s^* \in \mathbb{R}$  such that  $(u_s^+, \lambda_s^*)$  is the solution of the equations:

$$\begin{cases} \nabla_{u_s} \hat{\mathcal{L}}_s(u_s^+, \lambda_s^*) = 0 \\ \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} [\exp \{u_{s,a}^+\}] = 1 \end{cases} \tag{19}$$

Notice that

$$\begin{aligned}
\nabla_{u_s} \hat{\mathcal{L}}_s(u_s, \lambda) &= \nabla_{u_s} \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \frac{1}{2} (A_{s,a} - u_{s,a})^2 + \lambda \cdot \frac{\pi_{\text{ref}}(a|s)}{\nu(a|s)} \cdot \exp \{u_{s,a}\} \right] \\
&= \nu_{\mathcal{A}}(\cdot|s) \odot \left( u_s - A_s + \lambda \cdot \frac{\pi_{\text{ref}}(\cdot|s)}{\nu_{\mathcal{A}}(\cdot|s)} \cdot \exp \{u_s\} \right).
\end{aligned}$$

Plugging it into (19) yields that

$$\forall a \in \mathcal{A} : A_{s,a} - u_{s,a}^+ = \lambda_s^* \cdot \frac{\pi_{\text{ref}}(a|s)}{\nu_{\mathcal{A}}(a|s)} \cdot \exp \{u_{s,a}^+\}. \quad (20)$$

A direct computation yields that

$$\begin{aligned} \lambda_s^* &= \lambda_s^* \cdot \mathbb{E}_{a \sim \pi_{\text{ref}}} [\exp \{u_{s,a}^+\}] \\ &\stackrel{(a)}{=} \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \lambda_s^* \cdot \frac{\pi_{\text{ref}}(a|s)}{\nu_{\mathcal{A}}(a|s)} \cdot \exp \{u_{s,a}^+\} \right] \\ &= \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} [A_{s,a} - u_{s,a}^+], \end{aligned}$$

where (a) is due to (20). We now to show that  $\lambda_s^* \geq 0$ . Notice that

$$\begin{aligned} \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} [A_{s,a} - u_{s,a}^+] &= \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \frac{1}{\beta} A^{\pi_{\text{ref}}}(a|s) - \log \frac{\pi^+(a|s)}{\pi_{\text{ref}}(a|s)} \right] \\ &= \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \log \frac{\pi_{\text{ref}}(a|s)}{\pi^+(a|s)} \right] \\ &= \text{KL}(\pi_{\text{ref}}(\cdot|s), \pi^+(\cdot|s)). \end{aligned} \quad (21)$$

Thus if  $\lambda_s^* < 0$ , one has  $\forall a \in \mathcal{A} : A_{s,a} - u_{s,a}^+ < 0$  due to (20) and

$$\mathbb{E}_{a \sim \pi_{\text{ref}}} [A_{s,a} - u_{s,a}^+] < 0,$$

which contradicts to (21). Recall the definition of  $\mathcal{I}_s(\pi, \pi_{\text{ref}})$ . Thus

$$\begin{aligned} \mathcal{I}_s(\pi^+, \pi_{\text{ref}}) &= \mathbb{E}_{a \sim \pi^+(\cdot|s)} \left[ A^{\pi_{\text{ref}}}(s, a) - \beta \log \frac{\pi^+(a|s)}{\pi_{\text{ref}}(a|s)} \right] \\ &= \beta \cdot \mathbb{E}_{a \sim \pi^+(\cdot|s)} [A_{s,a} - u_{s,a}^+] \\ &\stackrel{(a)}{=} \beta \cdot \mathbb{E}_{a \sim \pi^+(\cdot|s)} \left[ \lambda_s^* \cdot \frac{\pi_{\text{ref}}(a|s)}{\nu_{\mathcal{A}}(a|s)} \cdot \exp \{u_{s,a}^+\} \right] \\ &= \beta \cdot \lambda_s^* \cdot \mathbb{E}_{a \sim \pi^+(\cdot|s)} \left[ \frac{\pi^+(a|s)}{\nu_{\mathcal{A}}(a|s)} \right] \\ &= \beta \cdot \lambda_s^* \cdot \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \left( \frac{\pi^+(a|s)}{\nu_{\mathcal{A}}(a|s)} \right)^2 \right] \\ &= \beta \cdot \lambda_s^* \cdot \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \left( \frac{\pi^+(a|s)}{\nu_{\mathcal{A}}(a|s)} \right)^2 \right] \\ &\geq \beta \cdot \lambda_s^* \\ &:= \lambda_s(\nu_{\mathcal{A}}) \end{aligned} \quad (22)$$

where the last inequality is due to

$$\mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \left( \frac{\pi^+(a|s)}{\nu_{\mathcal{A}}(a|s)} \right)^2 \right] \geq \left( \mathbb{E}_{a \sim \nu_{\mathcal{A}}(\cdot|s)} \left[ \frac{\pi^+(a|s)}{\nu_{\mathcal{A}}(a|s)} \right] \right)^2 = \left( \sum_{a \in \mathcal{A}} \pi^+(a|s) \right)^2 = 1.$$

Plugging (22) into Lemma 2.2 directly yields that

$$V_{\beta}^{\pi^+}(\mu) - V_{\beta}^{\pi_{\text{ref}}}(\mu) = \mathbb{E}_{d_{\mu}^{\pi^+}} [\mathcal{I}_s(\pi, \pi_{\text{ref}})] \geq \mathbb{E}_{d_{\mu}^{\pi^+}} [\lambda_s(\nu_{\mathcal{A}})] \geq 0.$$

We now to show that  $V_{\beta}^{\pi^+}(\mu) = V_{\beta}^{\pi_{\text{ref}}}(\mu)$  holds if and only if  $\pi_{\text{ref}}$  is already the optimal policy. Notice that

$$V_{\beta}^{\pi^+}(\mu) = V_{\beta}^{\pi_{\text{ref}}}(\mu) \Leftrightarrow \forall s \in \mathcal{S} : \lambda_s(\nu_{\mathcal{A}}) = 0 \Leftrightarrow \forall s \in \mathcal{S}, a \in \mathcal{A} : u_{s,a}^+ - A_{s,a} = 0.$$

Thus plugging it into (21) yields that

$$\text{KL}(\pi_{\text{ref}}(\cdot|s), \pi^+(\cdot|s)) = 0$$

and  $\forall s, a \in \mathcal{S} \times \mathcal{A} : A^{\pi_{\text{ref}}}(s, a) = \beta \cdot \log \frac{\pi^+(a|s)}{\pi_{\text{ref}}(a|s)} = 0$ . Hence for any state  $s \in \mathcal{S}$ ,

$$\begin{aligned}
\mathcal{T}_\beta V_\beta^{\pi_{\text{ref}}}(s) &= \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} (r(s, a) + V_\beta^{\pi_{\text{ref}}}(f(s, a))) \right\} \right] \\
&= \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} Q_\beta^{\pi_{\text{ref}}}(s, a) \right\} \right] \\
&= \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} Q^{\pi_{\text{ref}}}(s, a) \right\} \right] \\
&= \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} A^{\pi_{\text{ref}}}(s, a) + \frac{1}{\beta} V^{\pi_{\text{ref}}}(s) \right\} \right] \\
&= \beta \log \mathbb{E}_{a \sim \pi_{\text{ref}}(\cdot|s)} \left[ \exp \left\{ \frac{1}{\beta} V^{\pi_{\text{ref}}}(s) \right\} \right] \\
&= V^{\pi_{\text{ref}}}(s) \\
&= V_\beta^{\pi_{\text{ref}}}(s)
\end{aligned}$$

which means  $V_\beta^{\pi_{\text{ref}}}$  is the fixed point of the Bellman optimality operator thus  $\pi_{\text{ref}}$  is the optimal policy by Theorem B.2  $\square$

## D Discussion I : connection between DAPO and multi-step reasoning.

In this section, we give a detailed discussion on the connections between DAPO and multi-step reasoning and show that DAPO indeed achieves step-level fine-grained policy optimization and reduces the training variance.

**Response-level policy gradient methods suffer from high training variance.** Policy gradient (PG) methods updates the parameter by applying gradient ascent w.r.t the objective (4) directly and generally takes the form of

$$\theta \leftarrow \theta + \eta \nabla_\theta V_\beta^\pi(\mu).$$

For simplicity of analysis, we assume  $\beta = 0$ . By policy gradient theorem [1],

$$\nabla_\theta V^{\pi_\theta}(\mu) = \mathbb{E}_{x \sim \mathcal{D}} \mathbb{E}_{\forall t: a_t \sim \pi_\theta(\cdot|x)} \left[ \sum_{t=0}^{T-1} \nabla_\theta \log \pi_\theta(a_t|s_t) A^{\pi_\theta}(s_t, a_t) \right]. \quad (23)$$

Current response-level PG methods mainly focus on the estimation of advantage in (23). Suppose  $n$  responses  $y_1, \dots, y_n$  are sampled from  $\pi_\theta(\cdot|x)$  for each prompt  $x$ . We denote  $y_i := \text{Concat}(x, a_0^{(i)}, \dots, a_{T_i-1}^{(i)})$ ,  $s_t^{(i)} := \text{Concat}(x, a_0^{(i)}, \dots, a_{t-1}^{(i)})$ , where  $T_i$  is the number of reasoning steps in response  $y_i$ , and  $A_t^{(i)}$  as the estimation of advantage  $A^{\pi_\theta}(s_t^{(i)}, a_t^{(i)})$ . We summarize the construction of  $A_t^{(i)}$  of different methods in Table 8. It can be observed that all methods use  $r(x, y_i)$

Methods	Advantage estimation $A_t^{(i)}$	Sign of $A_t^{(i)}$	Does $A_t^{(i)}$ vary with $t$ ?
REINFORCE	$r(x, y)$	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No
REINFORCE++ [19]	$r(x, y)$ with global batch normalization	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No
ReMax [30]	$r(x, y_i) - r(x, \hat{y})$ where $\hat{y}$ is sampled from the greedy policy	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No
GRPO [48]	$\frac{r(x, y_i) - \hat{p}}{\sqrt{\hat{p}(1-\hat{p})}}$ , where $\hat{p} := \frac{1}{n} \sum_{j=1}^n r(x, y_j)$	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No
ORZ [20]	$r(x, y_i) - V_\phi(s_t^{(i)})$	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No
DR.GRPO [36]	$r(x, y_i) - \hat{p}$ , where $\hat{p} := \frac{1}{n} \sum_{j=1}^n r(x, y_j)$	non-negative when $r(x, y_i) = 1$ and non-positive when $r(x, y_i) = 0$	No

Table 8: Summary of advantage estimation of current response-level PG methods.

as the MC estimation of state-value function . Thus these methods generally try to increase the probability of **all** the reasoning steps if the response  $y$  is correct and decrease the probability of **all** the reasoning steps if  $y$  is incorrect. However, this update pattern indeed introduces high training variance since the sign of the true advantage value, i.e.  $A^{\pi_\theta}(s_t, a_t)$ , may mismatch the final reward of the trajectory. Following is a concrete example. The trajectory shown in Figure 2 contains 14 reasoning steps and is a correct trajectory sampled from Llama-3.1-8B-instruct. The values and advantages of each reasoning step are also reported on the right-hand side of the figure. There is a noticeable drop in value (accuracy) in step 11, decreasing from 0.94 to 0.41, resulting in an advantage of -0.53. This means the model is more likely to make a mistake starting from step 11, rather than increasing it, even if the final answer is correct. **Thus, there exists responses that yield correct answers while containing reasoning steps from which the model is likely to make mistakes, and vice versa.** Thus, using the same advantage estimation for all reasoning steps in the entire response introduces high training variance.

Question	Reasoning Steps	Value	Advantage
What is the volume, in cubic inches, of a rectangular box, whose faces have areas of 24 square inches, 16 square inches and 6 square inches?	Step 1 : To find the volume of a rectangular box, we need to multiply its length, width, and height.	0.41	0
	Step 2 : Let's call the length, width, and height of the box $l$ , $w$ , and $h$ respectively. We are given the areas of the faces as follows:	0.53	0.12
	Step 3 : - The face with area 24 has dimensions $l * w$ .	0.53	0
	Step 4 : - The face with area 16 has dimensions $w * h$ .	0.81	0.28
	Step 5 : - The face with area 6 has dimensions $l * h$ .	0.56	-0.25
	Step 6 : So, we have the following equations from the given areas:	0.53	-0.03
	Step 7 : $lw = 24$ , $wh = 16$ , $lh = 6$ .	0.84	0.31
	Step 8 : To find the volume, we can multiply all three equations together to get:	0.84	0
	Step 9 : $(lw)(wh)(lh) = 24 * 16 * 6$ .	0.87	0.03
	Step 10 : This simplifies to $(lwh)^2 = 2304$ .	0.94	0.07
	Step 11 : Taking the square root of both sides gives: $lwh = \sqrt{2304}$ .	0.41	-0.53
	Step 12 : $lwh = \sqrt{144 * 16}$ .	1	0.59
	Step 13 : $lwh = 12 * 4 = 48$ .	1	0
	Step 14 : The volume is $\boxed{48}$ cubic inches.	1	0

Figure 2: A reasoning trajectory sampled from Llama-3.1-8B-instruct with the true state value and advantage reported in the right hand side.

**Step-level fine-grained policy optimization via DAPO.** Even though the objective (1) optimized by response-level algorithms can be reformulated to a multi-step decision problem (2) equivalently, it by no means implies that these methods can achieve step-level optimization as illustrated above. For DAPO, it can be directly observed from (12) that, given the previous reasoning steps (state  $s$ ), DAPO optimizes the distribution of the next reasoning step generation (action  $a$ ) depends on the step-level advantage estimation other than the final reward. Thus DAPO indeed enables fine-grained step-wise learning by optimizing the actor using the advantage datasets. Our main theoretical result (Theorem 3.2) further shows that updating the actor’s probability in the direction of the advantage can indeed yield policy improvement if the advantages are well-estimated.

## E Discussion II : Computational cost of DAPO

In this section, we provide a detailed analysis of the computational cost of DAPO and discuss potential approaches for efficient training-time scaling. Consider an actor model  $M_{\text{actor}}$  with  $N_{\text{model}}^{\text{actor}}$

non-embedding parameters is optimized by DAPO on an offline advantage dataset  $\mathcal{D}$  with batch size  $B_{\text{actor}}$  over  $K_{\text{actor}}$  training steps. For the critic pre-training, we first generate  $n$  initial responses from which  $n_1$  responses are selected. For each reasoning step in these filtered responses, we then produce  $m$  completions for MC value estimation. Then a critic model  $M_{\text{critic}}$  with  $N_{\text{model}}^{\text{critic}}$  non-embedding parameters is optimized on the MC value dataset with batch size  $B_{\text{critic}}$  over  $K_{\text{critic}}$  training steps. To simplify the analysis, we assume both the response length and the number of reasoning steps per response remain constant at  $L$  and  $T$  respectively (each reasoning step is of length  $L/T$ ). Following [24], we estimate inference computational cost as  $2N_{\text{model}}N_{\text{infer}}$  and the training computational cost can be estimated by  $6N_{\text{model}}N_{\text{train}}$ , where  $N_{\text{infer}}$  and  $N_{\text{train}}$  are inference and training tokens respectively. We report  $N_{\text{infer}}$  and  $N_{\text{train}}$  for our DAPO implementation in Table E.

Step	$N_{\text{infer}}$	$N_{\text{train}}$	Computational cost $N_{\text{model}}(2N_{\text{infer}} + 6N_{\text{train}})$
Generation	$n_1 L$	0	$2n_1 L \cdot N_{\text{model}}^{\text{actor}}$
Completion	$\frac{nmL}{2}(T-1)$	0	$nmL(T-1) \cdot N_{\text{model}}^{\text{actor}}$
Critic Optimization	0	$nT \cdot B_{\text{critic}}K_{\text{critic}}$	$6nT \cdot B_{\text{critic}}K_{\text{critic}} \cdot N_{\text{model}}^{\text{critic}}$
Making Advantage Dataset	$nmL$	0	$2nmL \cdot N_{\text{model}}^{\text{critic}}$
Actor Optimization	0	$L/T \cdot B_{\text{actor}}K_{\text{actor}}$	$6L/T \cdot B_{\text{actor}}K_{\text{actor}} \cdot N_{\text{model}}^{\text{actor}}$

Table 9: Computational cost estimation of DAPO.

Suppose  $N_{\text{model}}^{\text{actor}} = N_{\text{model}}^{\text{critic}} = N_{\text{model}}$ . By Table E, the total computational cost  $c$  of DAPO is

$$\begin{aligned} c &= N_{\text{model}} \left( L(2n_1 + nm(T-1) + 2nm) + 6 \left( nTB_{\text{critic}}K_{\text{critic}} + \frac{L}{T}B_{\text{actor}}K_{\text{actor}} \right) \right) \\ &= N_{\text{model}} \cdot \mathcal{O} \left( LTnm + \left( \frac{L}{T} + nT \right) BK \right), \end{aligned}$$

where  $B := \max\{B_{\text{actor}}, B_{\text{critic}}\}$  and  $K := \max\{K_{\text{actor}}, K_{\text{critic}}\}$ . In our implementation  $n_1 = 32, n = 6, m = 16, T \approx 20, L \approx 2048$ .

**Possible approaches to efficient training-time scaling.** Suppose the length of reasoning step  $L_{\text{step}} := \frac{L}{T}$  is a constant. Then the computation cost  $c = N_{\text{model}} \cdot L_{\text{step}} \cdot \mathcal{O}(T^2nm + nTBK)$ . The computational cost  $c$  may explode due to square term  $T^2$  as the response length  $L$  increased. A potential approach to mitigate this issue is **not** to do MC value estimation for **all** reasoning steps as in [38]. Another possible approach is to find an efficient and effective method for dividing reasoning steps, ensuring that the total number of reasoning steps  $T$  does not increase too rapidly as in [35].

## F More details of experiments

### E.1 DAPO

Model	In Domain MATH	Out of Domain			
		GSM8K	Minerva MATH	Olympiad Bench	College Math
Skywork-Math-LLama + DAPO iter 1	41.90	61.49	5.51	18.68	24.85
	46.88 <small>+4.98</small>	67.55 <small>+6.06</small>	7.34 <small>+1.82</small>	22.38 <small>+3.70</small>	25.87 <small>+1.03</small>
	50.54 <small>+8.64</small>	69.04 <small>+7.55</small>	8.09 <small>+2.58</small>	23.86 <small>+5.18</small>	27.20 <small>+2.35</small>
Qwen2-Math-7B-Instruct + DAPO iter 1	74.46	89.38	40.07	34.36	41.59
	75.41 <small>+0.95</small>	89.45 <small>+0.07</small>	37.51 <small>-2.56</small>	37.03 <small>+2.67</small>	42.23 <small>+0.64</small>
	76.40 <small>+1.94</small>	89.30 <small>-0.08</small>	38.25 <small>-1.82</small>	37.64 <small>+3.27</small>	41.16 <small>-0.43</small>

Table 10: **Full Mathematical benchmark results of iterative DAPO** using zero-shot greedy inference.

**Useful training tricks** Following are some useful tricks applied in all of the DAPO experiments.

- **State-wise learning.** Notice that for each training state (intermediate step)  $s$ , there are multiple actions (next step) for DAPO training. Rather than shuffling all state-action pairs, we try to place different actions from the same state within the same batch for DAPO training in order to provide contrastive gradients.
- **Learning uniformly on approximate action space.** Suppose we have  $n$  actions  $\{a_i\}_{i=1}^n$  for each training state  $s$ . Although the true action space  $\mathcal{A}$  is extremely large for LLM, we consider the unique actions of  $\{a_i\}_{i=1}^n$  as the approximate action space denoted as  $\tilde{\mathcal{A}}_s$ . After computing the advantage for each action, we only add the unique actions to the advantage dataset for DAPO training, which means the training action distribution is  $\nu_{\mathcal{A}}(\cdot|s) = \text{Unif}(\tilde{\mathcal{A}}_s)$  in our implementation.
- **Learning on the states with large advantage gap.** Considering that the approximation error between the prediction of critic network and the true state value, we only learn those states that have a large advantage gap to make sure the actions with higher advantage are actually better than those actions with a lower advantage. We define the advantage gap as

$$\Delta_s := \max_{a_i \in \tilde{\mathcal{A}}_s} A(s, a_i) - \min_{a_j \in \tilde{\mathcal{A}}_s} A(s, a_j).$$

In our implementation, we only learn those states with  $\Delta_s >= 0.1$

## F.2 Baseline methods

Hyperparameter	Value
actor learning rate	5e-7
critic learning rate	9e-6
KL coefficient $\beta$	0.01
discount factor $\gamma$	1
GAE factor $\lambda$	0.95
actor training epoch	1
critic training epoch	1
rollout max new tokens	2048 (4096 for OpenO1-Llama-8B-v0.1)
rollout temperature	1
rollout batch size	1024
global train batch size	1024

Table 11: The hyperparameters we employ to train PPO

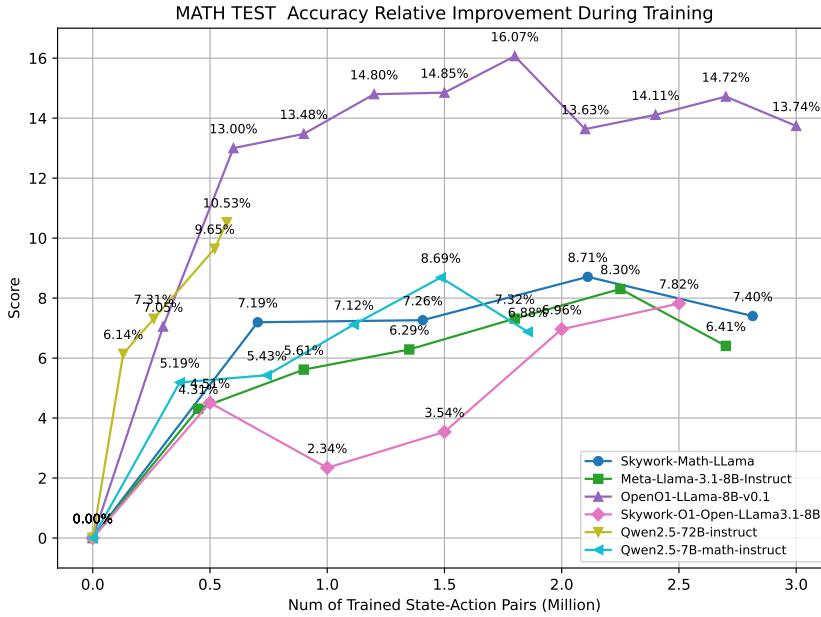
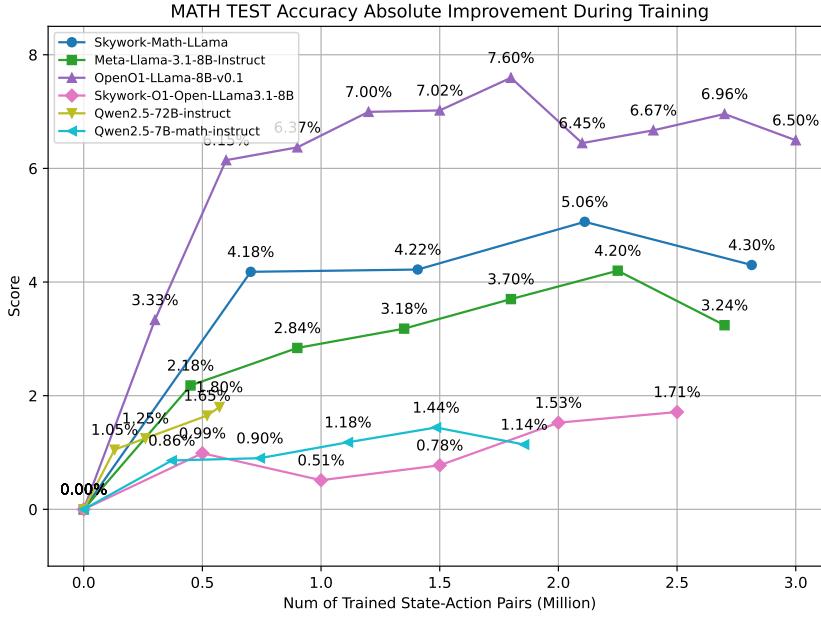


Figure 3: **Accuracy curve on MATH TEST during training process.** Let  $x$  be the accuracy of base model on MATH TEST and  $y$  be the accuracy after DAPO training. Absolute improvement refers to  $y - x$  and relative improvement refers to  $\frac{y-x}{1-x}$ .

Hyperparameter	Value
query batch size	64
group size	16
KL coefficient $\beta$	0.01
rollout max new tokens	2048 (4096 for OpenO1-Llama-8B-v0.1)
rollout temperature	1
global train batch size	1024
actor learning rate	1e-6

Table 12: The hyperparameters we employ to train GRPO

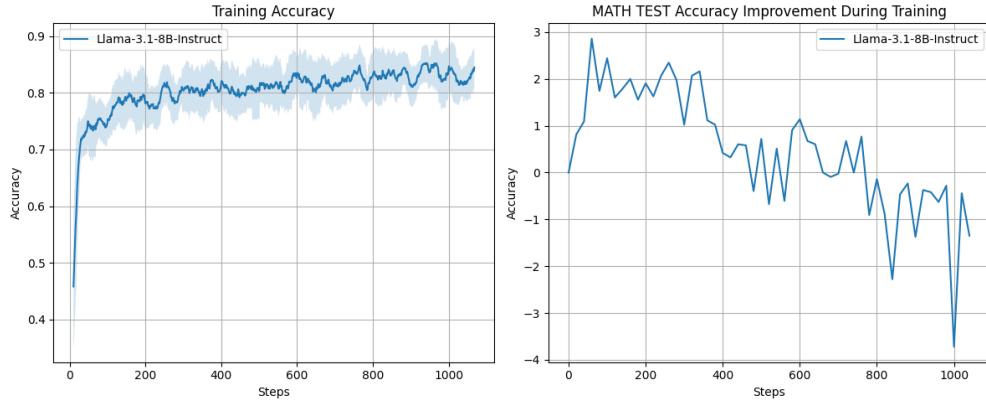


Figure 4: Performance of Meta-Llama-3.1-8B-instruct optimized via GRPO. The model exhibits serve overfitting after around 300 steps. **Left:** Accuracy on training batch during RL process. **Right:** Performance improvement (in percentage points) on MATH 5K test problems.

## G Pseudocode of DAPO

---

**Algorithm 1** Direct Advantage-Based Policy Optimization

---

1: **Input:** Training problems  $\mathcal{D}$ , initial actor  $\pi_{\theta_0}$ , initial critic  $V_{\phi_0}$ , maximum iteration number  $T$ , KL coefficient  $\beta$ .

2: **for** iteration  $i = 0, 1, \dots, T - 1$  **do**

3:   **[Step A.1 Generation]**

4:      $\mathcal{D}_{\text{gen}} \leftarrow \emptyset$ ,  $\pi_{\text{gen}} \leftarrow \pi_{\theta_i}$ ,  $\pi_{\text{ref}} \leftarrow \pi_{\theta_i}$ .

5:     **for** problem  $x \in \mathcal{D}$  **do**

6:       Sample multiple responses from the generator, i.e.  $G(x) \leftarrow \{y\} \stackrel{\text{iid}}{\sim} \pi_{\text{gen}}(\cdot|x)$ .

7:       For each response  $y \in G(x)$ , split it into multiple reasoning steps (e.g. by using the line break  $\backslash n$ ) so that  $y = (a_0, \dots, a_{T-1})$ , and construct the states

$$\mathcal{S}(y) \leftarrow \{\text{Concat}(x, a_0, \dots, a_t)\}_{t=0}^{T-1}.$$

8:       Add the states derived from all responses into  $\mathcal{D}_{\text{gen}}$ , i.e.,  $\mathcal{D}_{\text{gen}} \leftarrow \mathcal{D}_{\text{gen}} + \bigcup_{y \in G(x)} \mathcal{S}(y)$ .

9:       Stop generation when certain criteria are met (e.g., when the computation budget is reached).

10:   **end for**

11:   **[Step A.2 Completion]**

12:      $\mathcal{D}_{\text{com}} \leftarrow \emptyset$ ,  $\mathcal{D}_{\text{MC}} \leftarrow \emptyset$ .

13:     **for** state  $s \in \mathcal{D}_{\text{gen}}$  **do**

14:       Sample multiple completions  $C(s) \leftarrow \{\tilde{y}\} \stackrel{\text{iid}}{\sim} \pi_{\text{ref}}(\cdot|s)$  using  $s$  as the context prefix.

15:       Reward each completion and compute the MC value as

$$\text{MC}(s) \leftarrow \frac{1}{|C(s)|} \sum_{\tilde{y} \in C(s)} r(s, \tilde{y}).$$

16:        $\mathcal{D}_{\text{MC}}.\text{add}((s, \text{MC}(s)))$ ,  $\mathcal{D}_{\text{com}}.\text{add}((s, C(s)))$ .

17:   **end for**

18:   Update the critic (starting from  $\phi_i$ ) by optimizing (15) on MC value dataset  $\mathcal{D}_{\text{MC}}$ , i.e.

$$\phi_{i+1} \leftarrow \arg \min_{\phi} \mathbb{E}_{s \sim \mathcal{D}_{\text{MC}}} [\mathcal{L}_{\text{BCE}}(\text{MC}(s), V_{\phi}(s))]$$

19:   **[Step B.1 Compute Advantages]**

20:      $\mathcal{D}_{\text{adv}} \leftarrow \emptyset$

21:     **for**  $(s, C(s)) \in \mathcal{D}_{\text{com}}$  **do**

22:       Set action set  $\mathcal{A}(s) \leftarrow \emptyset$ .

23:       For each completion  $\tilde{y} \in C(s)$ , extract the first reasoning step  $a$  from  $\tilde{y}$  and add it into  $\mathcal{A}(s)$ .

24:       Compute advantage for each action  $a$  in  $\mathcal{A}(s)$  using optimized critic, i.e.,

$$\forall a \in \mathcal{A}(s) : \hat{A}(s, a) \leftarrow V_{\phi_{i+1}}(s \circ a) - \frac{1}{|\mathcal{A}(s)|} \sum_{\tilde{a} \in \mathcal{A}(s)} V_{\phi_{i+1}}(s \circ \tilde{a})$$

25:       Add the estimated advantages into  $\mathcal{D}_{\text{adv}}$ , i.e.,

$$\mathcal{D}_{\text{adv}} \leftarrow \mathcal{D}_{\text{adv}} + \left\{ (s, a, \hat{A}(s, a)) : a \in \mathcal{A}(s) \right\}$$

26:   **end for**

27:   **[Step B.2 Policy Optimization]**

28:   Update the actor (starting from  $\theta_i$ ) by optimizing (12) on the advantage dataset  $\mathcal{D}_{\text{adv}}$ , i.e.,

$$\theta_{i+1} \leftarrow \arg \min_{\theta} \frac{1}{2} \mathbb{E}_{(s, a, \hat{A}) \sim \mathcal{D}_{\text{adv}}} \left[ \left( \frac{1}{\beta} \hat{A}(s, a) - \log \frac{\pi_{\theta}(a|s)}{\pi_{\text{ref}}(a|s)} \right)^2 \right]$$

29: **end for**

30: **Output:** Last iterate actor  $\pi_{\theta_T}$ , last iterate critic  $V_{\phi_T}$ .

---