AAPO: Enhancing the Reasoning Capabilities of LLMs with Advantage Margin

Jian Xiong[1,2*], Jingbo Zhou[2†], Jingyong Ye[1], Qiang Huang[1], Dejing Dou[1,3†]
[1]College of Computer Science and Artificial Intelligence, Fudan University,
[2]Frontier Research Department, Baidu Inc., [3]BEDI Cloud
{jxiong24, jyye21, huangq25}@m.fudan.edu.cn, zhoujingbo@baidu.com, doudejing@fudan.edu.cn

Abstract

Reinforcement learning (RL) has emerged as an effective approach for enhancing the reasoning capabilities of large language models (LLMs), especially in scenarios where supervised fine-tuning (SFT) falls short due to limited chain-of-thought (CoT) data. Among RL-based post-training methods, group relative advantage estimation, as exemplified by Group Relative Policy Optimization (GRPO), has attracted considerable attention for eliminating the dependency on the value model, thereby simplifying training compared to traditional approaches like Proximal Policy Optimization (PPO). However, existing group relative advantage estimation method still suffers from training inefficiencies, particularly when the estimated advantage approaches zero. To address this limitation, we propose Advantage-Augmented Policy Optimization (AAPO), a novel RL algorithm that optimizes the cross-entropy (CE) loss using advantages enhanced through a margin-based estimation scheme. This approach effectively mitigates the inefficiencies associated with group relative advantage estimation. Experimental results on multiple mathematical reasoning benchmarks and model series demonstrate the superior performance of AAPO. Code is available at https://github.com/JianxXiong/AAPO.

1 Introduction

Reinforcement learning (RL) has emerged as a powerful approach for enhancing the reasoning and decision-making capabilities of large language models (LLMs). While LLMs have demonstrated strong performance in both language understanding and generation tasks (Brown et al., 2020; Chowdhery et al., 2023; Touvron et al., 2023; Zhao et al., 2026), traditional training strategies such as pre-training and supervised fine-tuning (SFT) (Radford et al., 2018; Bommasani et al., 2022; Liu et al., 2023) often fall short in enabling effective chain-of-thought (CoT) (Wei et al., 2022) reasoning for complex decision-making tasks. To address this limitation, recent research has explored RL-based training paradigms, which have shown considerable empirical success in specialized domains such as mathematical reasoning. Models including GPT-o1 (OpenAI, 2024), DeepSeek-R1 (Guo et al., 2025), and QwQ (Qwen Team, 2024) exemplify this promising direction, demonstrating the potential of RL to substantially improve the reasoning capabilities of LLMs.

A recent advancement in LLM post-training with RL is the introduction of novel methods for advantage estimation, a technique for quantifying how favorable a specific action is in a given state. In this context, the term advantage measures the relative benefit of an action compared to the average in that state, thereby providing an informative learning signal during training. Traditionally, approaches such as Proximal Policy Optimization (PPO) (Schulman et al., 2017), exemplified by InstructGPT (Ouyang et al., 2022), rely on a value model to estimate the advantage. Although PPO offers stable and reliable performance, maintaining a separate value model leads to substantial consumption of GPU resources.

In contrast to conventional approaches, the group relative advantage estimation method was originally proposed in Group Relative Policy Optimization (GRPO) (Shao et al., 2024) and has since been widely adopted for enhancing reasoning capabilities in LLMs. Group relative advantage estimation method removes the need for value models entirely by evaluating responses relative to the average within a group of sampled responses. This approach significantly reduces GPU memory usage and computational costs while maintaining competitive performance in downstream reasoning tasks.

[*]This work was done when the first author was an intern in Frontier Research Department, Baidu Inc., under the supervision of Jingbo Zhou.
[†]Corresponding authors.

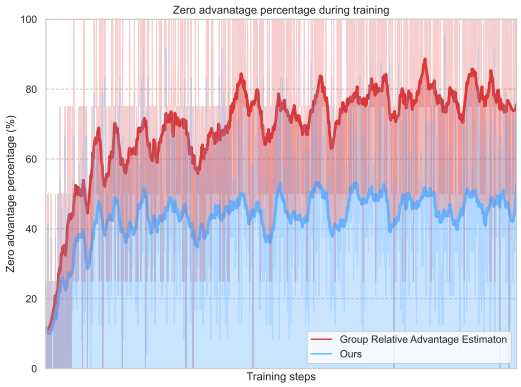

Figure 1: Statistical analysis of zero advantage proportion during training.

Its effectiveness is further evidenced by the strong performance of modern reasoning models such as DeepSeek-R1 (Guo et al., 2025), which highlights the effectiveness of group relative advantage estimation in balancing computational efficiency and performance robustness. Several extensions have further refined this paradigm. Decoupled Clipping and Dynamic sAmpling Policy Optimization (DAPO) (Yu et al., 2026) improves training on long CoT sequences through token-level gradient estimation and relaxed clipping strategies. Dr. GRPO (Liu et al., 2025) introduces an unbiased optimization method that enhances token efficiency, while Group Policy Gradient (GPG) (Chu et al., 2026) further simplifies the learning process by removing surrogate losses and eliminating the reference model. While these methods vary in implementation specifics, they all share a common foundation in the principle of group relative advantage estimation for effective LLMs post-training.

Although the above approaches have significantly advanced RL in the post-training stage and enhanced the reasoning capabilities of LLMs beyond what SFT can achieve, the practical limitations associated with the group relative advantage estimation method remain unresolved. When the group relative advantage estimation method (Shao et al., 2024; Chu et al., 2026; Liu et al., 2025) is adopted, the advantage may approach zero when the rewards within a group exhibit low variance, resulting in zero gradient and, consequently, no parameter updates. As shown in Figure 1, our statistical analysis of zero advantage proportion revealed that it is severe during training and approached even 100% in the later training steps more frequently. Conversely, when rewards among responses within the group vary significantly, the resulting advantages can exhibit high variance, potentially leading to unstable or unintended gradient ascent. Both scenarios deviate from the desired optimization trajectories. In this work, to address the challenges of policy optimization from the group relative advantage estimation method mentioned above, we propose a novel RL algorithm Advantage-Augmented Policy Optimization (AAPO), which mitigates the issues we address by estimating the advantage with advantage margin. Advantage margin is defined as the distance between the rewards of responses from the policy model and those of the responses from the reference model. This approach incorporates a reference gradient into the original gradient, thus providing a reliable optimization signal that reflects the overall direction of improvement, even when the advantages approach zero. Experiments on several representative mathematical reasoning benchmarks demonstrate the effectiveness and robustness of AAPO.

Our main contributions are as follows:

- We delve into the optimization behavior of current RL algorithms adopting the group relative advantage estimation method, in the context of post-training LLMs with RL, with a particular emphasis on potential issues related to advantage estimation during the optimization.
- We propose Advantage-Augmented Policy Optimization (AAPO), which mitigates the issues of advantage estimation with advantage margin by taking a comparison with the reference model.
- Extensive experimental evaluation has demonstrated that AAPO achieves superior performance across different model series and mathematical reasoning benchmarks.

2 Related Work

2.1 Reinforcement Learning for LLMs

Early RL-based post-training methods (Christiano et al., 2017; Song et al., 2024; Ouyang et al., 2022; Ziegler et al., 2020) mainly relied on human-labeled preference data and reward models to evaluate responses. PPO (Schulman et al., 2017) estimates state values via a value model for advantage estimation while using a reward model to assign rewards. DPO (Rafailov et al., 2023) uses paired preference data to encourage preferred responses and suppress undesired ones. However,

annotating preferences and training reward models are resource intensive. Moreover, the scarcity of explicit reasoning data limits the enhancement of models' reasoning capabilities via these RL methods. DeepSeek-R1 is the first to report the use of RL to enable extended CoT generation and the emergence of the so-called "aha moment" (Guo et al., 2025). Specifically, DeepSeek-R1 adopts GRPO (Shao et al., 2024), which estimates advantages through in-group comparisons, thereby enhancing the feasibility of aligning models for reasoning tasks. Yet, the limitations of existing advantage estimation remain underexplored. In this work, we introduce AAPO, which redefines advantage estimation by incorporating advantage margin, effectively addressing the issues of the group relative advantage estimation method.

2.2 Advantage estimation in RL

Previous RL approaches (Schulman et al., 2017, 2015; Haarnoja et al., 2018) estimate the advantage using either Monte Carlo returns (Williams, 1992), Temporal-Difference (TD) (Sutton et al., 1999) errors, or Generalized Advantage Estimation (GAE) (Schulman et al., 2016). While these critic-based estimators form the backbone of algorithms such as PPO, maintaining a separate value network becomes computationally expensive when the policy is implemented as an LLM. Group relative advantage estimation method proposed by GRPO (Shao et al., 2024) computes the advantage by comparing each response's reward to the group mean, reducing the resources needed. However, it introduces new optimization challenges, as discussed in the introduction section, that have yet to be investigated. In this work, we propose a novel advantage estimation method that incorporates advantage margin and optimizes the cross-entropy loss to better enhance the reasoning capabilities of LLMs.

3 Preliminary

This section provides a preliminary overview of group-based relative advantage estimation methods for RL-based training by briefly introducing GRPO (Shao et al., 2024) and GPG (Chu et al., 2026). In the context of the GRPO algorithm, it circumvents the dependency on the value model which is commonly required in PPO (Schulman et al., 2017) by estimating advantage within grouped samples. It utilizes a rule-based reward system to score responses generated by LLMs. Furthermore, GRPO retains the clipping strategy from PPO to prevent excessively large policy updates and leverages the reference model π_{ref} to compute the current Kullback-Leibler (KL) divergence, thereby ensuring the stability and integrity of the model during training. Specifically, for each question q, GRPO samples a group of responses $O = \{o_1, o_2, \cdots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and optimizes the policy model π_θ by minimizing the following objective:

$$\begin{aligned}\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \\ - \Bigg[\min\left[r_{i,t}(\theta)\hat{A}_{i,t}^{\text{GRPO}}, \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}^{\text{GRPO}}\right] \\ - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\Bigg]\end{aligned} \tag{1}$$

where ε and β are hyper-parameters to control the clip boundary and the KL divergence penalty coefficient, respectively. In this context, $\hat{A}_{i,t}^{\text{GRPO}}$ represents the advantage, derived exclusively from the relative reward of the responses within the same group. $r_{i,t}(\theta)$ represents the likelihood ratio between the current policy π_θ and the old policy $\pi_{\theta_{\text{old}}}$. Typically, the likelihood ratio $r_{i,t}(\theta)$ and the advantage $\hat{A}_{i,t}^{\text{GRPO}}$ are calculated as:

$$r_{i,t}(\theta) = \frac{\pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)}{\pi_{\theta_{\text{old}}}\left(o_{i,t} \mid q, o_{i,<t}\right)}, \quad \hat{A}_{i,t}^{\text{GRPO}} = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}. \tag{2}$$

Recent work GPG (Chu et al., 2026) proposes directly optimizing the original RL objective, yielding improved performance over GRPO. While the advantage estimation method used in GPG is similar to that of GRPO, it further emphasizes the advantage of responses that are valid to gradient estimation. Generally speaking, the core objective of GPG is to optimize the following objective:

$$\begin{aligned}\mathcal{J}_{\text{GPG}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}} \\ \left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\left(-\log\pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)\hat{A}_{i,t}^{\text{GPG}}\right)\right],\end{aligned} \tag{3}$$

where $\hat{A}_{i,t}^{\text{GPG}} = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^{G})}{F_{norm}}$ and F_{norm} could be 1 or $\text{std}(\{R_i\}_{i=1}^{G})$. However, the rewards of the responses within a group may exhibit low variance. This suggests that $\hat{A}_{i,t}^{\text{GPG}}$ in GPG encounters the same challenges as $\hat{A}_{i,t}^{\text{GRPO}}$ in GRPO.

In this work, we propose a novel algorithm AAPO to address the issues caused by the group relative advantage estimation method in the policy optimization process. We also provide an in-depth analysis of both the prevailing advantage estimation method adopted in (Shao et al., 2024; Yu et al., 2026; Chu et al., 2026) and our proposed AAPO.

4 Advantage-Augmented Policy Optimization

Inspired by the optimization behavior found in current RL algorithms (Shao et al., 2024; Liu et al., 2025; Yu et al., 2026; Chu et al., 2026), our objective is to mitigate the issues that arise in the policy optimization process, overcoming the challenge where advantage estimation tends to approach zero or bad advantage estimation in the later steps of RL training. Drawing further inspiration from the well-established Adam (Kingma and Ba, 2015) and the recent GPG algorithm (Chu et al., 2026), which optimizes the RL objective directly, thus avoiding the surrogate loss function, we propose a novel algorithm, Advantage-Augmented Policy Optimization (AAPO), which directly optimizes the cross-entropy (CE) loss enhanced by augmented advantage, which is driven by the advantage margin. In contrast to previous approaches (Shao et al., 2024; Chu et al., 2026; Liu et al., 2025), AAPO leverages advantage amplification by performing group-based sampling for both the policy model π_θ and the reference model π_{ref}. Specifically, we calculate the reward for each response generated by the policy model π_θ, evaluate the relative advantage of each response within its group G, and compare these rewards r_{θ_i} with those rewards r_{ref_i} obtained from the reference model π_{ref}. This method improves the effectiveness of the RL training step by preventing the advantage from approaching zero and exhibiting high variance. Formally, AAPO optimizes the policy model π_θ by minimizing the following objective:

$$\mathcal{J}_{\text{AAPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \left(-\log \pi_\theta \left(o_{i,t} \mid q, o_{i,<t}\right) \hat{A}^*_{i,t} \right) \right], \tag{4}$$

where $\hat{A}^*_{i,t}$ is computed using Equation (5) and clip operation is added to improve stability, which is discussed in Section 6.3.

$$\hat{A}^*_{i,t} = \frac{r_{\theta_i} - \text{mean}(r_\theta)}{\text{std}(r_\theta)} + \text{clip}(\underbrace{r_{\theta_i} - r_{ref_i}}_{\textbf{Advantage margin}}, \delta_{\text{low}}, \delta_{\text{high}}). \tag{5}$$

By Equation (4), we can derive its gradient:

$$\nabla_\theta \mathcal{J}_{\text{AAPO}}(\theta) = -\mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \hat{A}^*_{i,t} \cdot \nabla_\theta \log \pi_\theta \left(o_{i,t} \mid q, o_{i,<t}\right) \right], \tag{6}$$

where the augmented advantage $\hat{A}^*_{i,t}$ functions as a constant coefficient that scales the gradient.

To provide theoretical guarantees for the training dynamics of AAPO, we explicitly analyze its stability and convergence properties. We begin by formalizing the empirical loss function over a sampled group, which serves as the foundation for our theoretical derivation.

Definition *For a group $\mathcal{G}$ containing sampled responses $O = \{o_1, o_2, \cdots, o_\mathcal{G}\}$, the empirical AAPO loss is defined as $\mathcal{L}_\mathcal{G}(\theta) = \frac{1}{N_\mathcal{G}} \sum_{o\in\mathcal{G}}[-\log \pi_\theta(o)\hat{A}^*]$, where π_θ is the policy model, $N_\mathcal{G} = \sum_{o\in\mathcal{G}}|o|$ is the total number of tokens in the group. We further define the expected objective as the expectation of the empirical loss over all possible groups $\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{G}\sim\pi_\theta}[\mathcal{L}_\mathcal{G}(\theta)]$.*

First, we establish the stability of the AAPO training process. It is crucial to ensure that the introduction of the augmented advantage term does not lead to unbounded parameter updates.

Theorem 1. *(Stability) Since the rewards are bounded, the group standard deviation satisfies $0 \le \sigma_{min} \le \sigma$, and the log-likelihood gradients are bounded as $||\nabla_\theta \log \pi_\theta(o)|| \le M$. Then, each gradient step with learning rate η_k satisfies $||\theta_{k+1} - \theta_k|| \le \eta_k MB$, where $B = \frac{R_{max}-R_{min}}{\sigma_{min}} + \max(|\delta_{\text{low}}|, |\delta_{\text{high}}|)$ is the uniform bound on the AAPO weights. The expected objective is bounded from $\mathcal{L}(\theta) \ge -B \log |\mathcal{V}|$, where $|\mathcal{V}|$ is the vocabulary size. Hence, AAPO training is stable: the objective cannot diverge to $-\infty$ and parameter updates are always finite. Proof in Appendix B.*

Beyond stability, we ensure that AAPO effectively minimizes the loss and locates a stationary point. The following theorem guarantees the convergence of AAPO under standard stochastic approximation assumptions.

Theorem 2. *(Convergence) Assume that the stochastic gradient is unbiased and that the per-sample gradient has bounded second moment. Let the step sizes satisfy the Robbins–Monro conditions $\eta_k > 0$, $\sum_k \eta_k = \infty$, $\sum_k \eta_k^2 < \infty$. AAPO converges to a stationary point of its expected objective $\liminf_{k\to\infty} \mathbb{E}\left[\|\nabla\mathcal{L}(\theta_k)\|^2\right] = 0$. Moreover, if a constant step size $\eta < \frac{1}{BL_0}$ is used, where L_0 is the smoothness constant of $-\log \pi_\theta(o)$, then the iterates converge to a neighborhood of stationarity $\limsup_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[\|\nabla\mathcal{L}(\theta_k)\|^2\right] \lesssim \mathcal{O}(\eta) + \mathcal{O}\left(\frac{1}{N_\mathcal{G}}\right)$. Proof in Appendix B.*

From the formulation of the augmented advantage $\hat{A}^*_{i,t}$, it is evident that as the policy is optimized, the reward r_{θ_i} of the responses sampled

from the policy π_θ within a group increases. Consequently, the distance between these rewards r_{θ_i} and those r_{ref_i} of the responses sampled from the reference model π_{ref} will also widen, since the parameters of the reference model remain frozen throughout the training process of AAPO. In later steps of RL, the responses sampled from the policy tend to be of high quality, causing the relative advantages $\hat{A}_{i,t}^{\text{GRPO}}$ within the group to approach zero. If training continues using the original advantage estimation method in Equation (2), the resulting gradients will approach zero, leading to significantly reduced training efficiency. However, under the proposed method augmenting advantage with advantage margin in Equation (5), even when the group relative advantage approaches zero, the rewards of the policy samples remain higher than those of the reference samples. This ensures a nonzero advantage $\hat{A}_{i,t}^{*}$ in AAPO as discussed in Section 5.2, thereby maintaining informative gradients for continued policy optimization.

5 Analysis of AAPO

5.1 Deep Analysis of Group Relative Advantage Estimation

As shown in Equation (2), current advantage estimation methods (Shao et al., 2024; Yu et al., 2026; Liu et al., 2025) that eliminate the dependency on a value model predominantly adopt this form of computation. We now present a rigorous analysis into the underlying phenomena induced by this advantage estimation method. **Phenomenon 1**: What are the implications when all responses within a group are similarly good (or all of high quality)? **Phenomenon 2**: What are the implications when all responses within a group are similarly bad (or all of low quality)?

To rigorously address the aforementioned questions, we proceed with a systematic, step-by-step analysis. For the sake of mathematical convenience in the proof, we limit our discussion to cases where all responses are similarly good in Phenomenon 1 and similarly bad in Phenomenon 2, respectively, as the proof for all of high quality and low quality responses follows the same structure.

Phenomenon 1 Considering that all responses are similarly good, which implies that the reward for each response is nearly the same, this indicates $\forall i, j \in \{1, 2, 3, \cdots, G\} \wedge i \neq j,\ r_i \approx r_j$, their respective advantage, as estimated according to Equation (2), can be expressed as following:

$$\hat{A}_{i,t}^{\text{GRPO}} = \frac{r_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})} \to 0. \quad (7)$$

As expressed in Equation (7), the advantage of each response approaches zero in this case. For illustrative purposes, we use the loss function of GRPO (Yu et al., 2026) as an example. As mentioned in DAPO (Yu et al., 2026), removing the KL divergence in GRPO could further improve the optimization. The KL divergence, summation and averaging operators in Equation (1) are omitted to facilitate clarity in this context, the gradient formula of GRPO is formally given by:

$$\nabla_\theta \mathcal{J}_{\text{GRPO}}(\theta) = A_{i,t}^{\text{GRPO}} \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}). \quad (8)$$

As illustrated in the computation of Equation (8), when the advantage of each response approaches zero, the corresponding gradient also decreases to zero. This implies that the gradient update for the policy becomes negligible for this training step, resulting in very low training efficiency. Similar issues are observed in methods such as GRPO, DAPO, GPG (Chu et al., 2026), and Dr. GRPO (Liu et al., 2025). Nonetheless, the observation that all responses are associated with similarly high reward does not unequivocally imply that the policy has been sufficiently optimized; it may alternatively reflect a high variance (Roelofs et al., 2019; Yu et al., 2022) in the current policy, suggesting that the policy has converged to a sub-optimum, performing well only on a narrow category of questions. However, this phenomenon is frequently observed during the later steps of RL training if the advantage estimation method in GRPO is adopted.

Phenomenon 2 Similar to the scenario in Phenomenon 1 where all responses are similarly good, when all responses are similarly bad, the reward assigned to each response tends to be similar. As a result, the relative advantage of each response approaches zero, leading to a zero gradient during policy updates, which is computed using Equation (8). Consequently, the efficiency of this training step becomes significantly low. Such a phenomenon is frequently observed when the policy is confronted with input samples that exhibit inherently high complexity or ambiguous representation.

Analysis Conclusion Based on our in-depth analysis of the two phenomena where generated responses are similarly good and similarly bad within a group, we observed that the gradient tends to approach zero. This results in extremely low training

efficiency during RL training.

Above phenomena can be generalized When the rewards of responses within any given group are identical or highly similar, regardless of whether the responses are all good or all bad, the gradient approaches zero, rendering the gradient update nearly ineffective in training. This issue becomes particularly pronounced in the later steps of RL training, where generated responses are consistently of high quality, and the corresponding advantage approaches zero. This indicates that the training efficiency progressively declines as RL training progresses. Motivated by this insight, we propose an advantage-augmented RL algorithm AAPO to address the aforementioned phenomena.

5.2 Understanding the effectiveness of AAPO

As discussed in Section 5.1, the previous advantage estimation method (as expressed in Equation (2)) in RL algorithms (Shao et al., 2024; Chu et al., 2026; Liu et al., 2025) can lead to the advantage value approaching zero, which in turn causes the magnitude of gradient updates to diminish accordingly. In this section, we provide a comprehensive analysis of why our proposed advantage-augmented method can effectively mitigate these issues.

Analysis 1 Consider a general situation, which naturally encompasses the two phenomena in Section 5.1. In the later steps of RL training, once the policy has already acquired relatively easier-to-learn features, it often struggles to learn more complex ones. During this phase, when the policy π_θ samples a group of responses, the reward associated with each sample tends to be similar. Consequently, the relative advantage computed according to Equation (2) approaches zero. To address this issue, we propose AAPO, which estimates the advantage $\hat{A}^*_{i,t}$ with advantage margin following Equation (5). Since the capability of the reference model π_{ref} remains unchanged while the policy model π_θ improves progressively throughout AAPO training, the quality of responses generated by the policy model π_θ exceeds that of the reference model π_{ref} over time. By measuring the distance between the rewards of two groups of responses $O_\theta = \{o_{\theta_1}, o_{\theta_2}, \cdots, o_{\theta_G}\}$ and $O_{ref} = \{o_{ref_1}, o_{ref_2}, \cdots, o_{ref_G}\}$ from π_{ref} generated by the policy model π_θ and the reference model π_{ref}, respectively, we can calculate the augmented advantages of each response in O_θ via Equation (5). This prevents the advantages from approaching zero and ensures that the gradients used to update the policy remain informative and effective. This analysis can be generalized to any situation where the rewards of the responses in the group are similar, whether good or not.

Analysis 2 Group relative advantage estimation (Equation (2)) can be problematic in the presence of two types of asymmetric response distributions. In one case, where most responses from the policy π_θ are high quality except for a single outlier, the relatively worse response may receive a negative advantage and contribute an opposing gradient, increasing variance. Although this response may still be correct under multi-dimensional reward rules, such as both format and correctness, it is penalized due to its format. This misalignment between reward attribution and the true value of a response increases the risk of reward hacking(Everitt et al., 2021; Pan et al., 2022). In the opposite case, where most responses are low quality and one is relatively better, the better one receives an excessively high advantage despite possibly low absolute quality, leading to biased updates and suboptimal convergence. AAPO augments advantage estimation with advantage margin, addresses both issues: it boosts underappreciated yet valuable responses and suppresses misleading ones that exhibit a high estimated advantage. As a result, AAPO reduces variance and the risk of reward hacking, and improves the optimization stability of the policy.

Analysis 3 When policy optimization with AAPO reaches a global optimum, which means $\forall i, j \in \{1, 2, 3, \cdots, G\} \wedge i \neq j,\ r_i \approx r_j$, the objective in Equation (6), which consists of the CE loss and the augmented advantage, exhibits a specific and predictable behavior as detailed in our derivation below (for clarity and notational convenience, we omit the expectation operator). The optimization exhibits slight oscillations near the optimum, avoiding large gradient updates.

$$\begin{aligned}\mathcal{J}_{\text{AAPO}}(\theta) \approx & \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \\ & \left[-\log \pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right) \cdot \text{clip}(r_{\theta_i} - r_{ref_i}, \delta_{\text{low}}, \delta_{\text{high}})\right] \\ & \leq \delta_{\text{high}} \cdot \frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\left(-\log \pi_\theta\left(o_{i,t} \mid q, o_{i,<t}\right)\right)\end{aligned}$$

6 Experiments

6.1 Experimental setup

Most recent RL algorithms (Shao et al., 2024; Yu et al., 2026; Hu et al., 2025a; Liu et al., 2025) struggle to make LLMs perform better at solving

Model	Training Samples	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
Llama 1B Models							
Llama-3.2-1B-Instruct†	–	0.0	11.8	2.5	1.8	3.7	4.0
+GRPO† (Shao et al., 2024)	8,523	0.0	19.4	12.5	3.7	4.3	8.0
+GPG† (Chu et al., 2026)	8,523	0.0	21.2	17.5	1.8	4.7	9.0
+AAPO (Ours)	8,523	10.0	25.0	12.5	4.0	6.1	**11.5**
Llama 3B Models							
Llama-3.2-3B-Instruct†	–	3.3	28.6	7.5	3.7	7.6	10.1
+GRPO† (Shao et al., 2024)	8,523	0.0	32.8	22.5	8.1	7.7	14.2
+GPG† (Chu et al., 2026)	8,523	6.7	40.0	15.0	8.8	11.6	16.2
+AAPO (Ours)	8,523	6.7	43.8	22.5	9.6	11.4	**18.8**
Qwen 1.5B Models							
DeepSeek-R1-Distill-Qwen-1.5B†	–	33.3	84.4	70.0	30.9	50.8	53.9
GRPO-1.5B† (Dang and Ngo, 2026)	7,000	26.7	86.2	82.5	27.6	52.6	55.2
GPG-1.5B† (Chu et al., 2026)	7,000	36.7	83.4	75.0	29.8	53.2	55.6
Still-3-1.5B-Preview† (Chen et al., 2025)	30K	40.0	85.5	72.5	30.5	53.9	56.5
AAPO-1.5B (Ours)	7,000	33.3	86.0	80.0	30.9	53.3	**56.7**
Qwen 7B Models							
Qwen2.5-Math-7B† (Yang et al., 2024)	–	6.7	56.2	47.5	14.0	23.4	39.6
SimpleRL-Zero-7B† (Zeng et al., 2025)	8,523	30.0	77.4	57.5	30.5	38.1	46.7
GPG-7B† (Chu et al., 2026)	8,523	23.3	80.2	55.0	36.0	42.8	47.5
OpenReasoner-Zero-7B† (Hu et al., 2025b)	57K	20.0	80.8	65.0	29.4	46.2	48.3
Eurus-2-7B-PRIME† (Cui et al., 2025)	230K + 150K	16.7	81.8	65.0	37.5	44.6	49.1
Oat-Zero-7B† (Liu et al., 2025)	8,523	30.0	81.2	65.0	34.9	43.4	50.3
AAPO-7B (Ours)	8,523	30.0	82.4	70.0	35.3	44.3	**52.4**

Table 1: Zero-shot pass@1 performance on mathematical reasoning benchmarks. † represents reproduced results with our best effort under the same settings. **Bold** and Underline indicate the best and the second-best performance in the corresponding category, respectively.

mathematical problems, which requires the model to think in the CoT (Wei et al., 2022) format before deciding the final answer. We choose open-rs (Dang and Ngo, 2026) as our training dataset for DeepSeek-R1-Distill-Qwen-1.5B base model, because the data in this dataset cover various types and difficulty levels of mathematical problems that are highly representative. To further provide a fair and rigorous evaluation of the effectiveness of our proposed AAPO, Qwen2.5-Math-7B model is chosen as the base model and subsequently trained on more challenging simplelr_qwen_level3to5 dataset (Zeng et al., 2025). In addition, we also validate the effectiveness of AAPO on Llama series models.

In our experiment setting, we set the clip parameters δ_{low} and δ_{high} to be -0.2 and 0.28, respectively. We train all base models under the AAPO following the training process depicted in Algorithm 1. All rule-based reward functions adopted in our experiments are simple. More training details about rule-based reward functions and the system prompt are provided in Appendix C. To evaluate the extent to which our proposed AAPO algorithm can enhance the reasoning capabilities of the model, we select AIME24 (30 questions), MATH-500 (500 questions) (Hendrycks et al., 2021b; Lightman et al., 2024), AMC23 (40 questions), Minerva (272 questions) (Lewkowycz et al., 2022), and OlympiadBench (674 questions) (Huang et al., 2024) as evaluation benchmarks. Our evaluation framework utilizes a well-established and community-vetted codebase, maintaining consistency with widely adopted implementations.

6.2 Results

As shown in Table 1, taking model DeepSeek-R1-Distill-Qwen-1.5B as our base model, the application of our proposed AAPO enables the base model to achieve the SOTA performance on Minerva, the second-best performance on MATH-500, AMC23 and OlympiadBench. When averaging scores across all benchmarks, the resulting AAPO-1.5B model achieves an overall SOTA performance. By direct comparison under the same training data, AAPO-1.5B achieves improvements of **2.7%** and **2.0%** over GRPO-1.5B and GPG-1.5B, respectively. It is worth noting that our AAPO-1.5B achieves performance superior to Still-

3-1.5B-Preview (Team, 2025), despite the fact that Still-3-1.5B-Preview benefits from a larger training dataset that consists of long CoT reasoning data distilled from the DeepSeek-R1 (Guo et al., 2025) model and performs RL training with specified reward strategies. After employing our proposed AAPO on the Qwen2.5-Math-7B model, AAPO-7B achieves the overall SOTA performance compared to other methods. AAPO-7B achieves improvements of **12.2%**, **10.3%** over SimpleRL-Zero-7B (Zeng et al., 2025), GPG-7B (Chu et al., 2026), respectively, under identical training data. Furthermore, AAPO-7B also outperforms other models such as Eurus-2-7B-PRIME (Cui et al., 2025), OpenReasoner-Zero-7B (Hu et al., 2025b), despite the fact that these baselines are trained with more high-quality data or data distilled from the DeepSeek-R1 model. This suggests that the effectiveness can be mainly attributed to the design of our proposed AAPO rather than to the scale or quality of the training data. AAPO also demonstrates superior performance compared to GRPO and GPG on Llama series models under the same training and evaluation settings as reported in Table 1. Compared to GRPO and GPG, AAPO achieves absolute improvements of **3.5%**, **2.5%**, and **4.6%**, **2.6%** on Llama 1B and 3B models, respectively. More experimental results can be found in Appendix D.

6.3 Ablation study

Clip operation To investigate the contribution of the clip operation to AAPO, we conduct additional ablation studies by removing the clip operation on both the 1.5B and 7B models. The results presented in Table 2 indicate that the performance of the optimized model without clip is inferior to that with the clip, suggesting that incorporating the clip operation effectively contributes to further improvements in optimization results. As illustrated in Appendix Figure 3, the optimization process becomes more stable with the incorporation of the clip operation compared to the optimization process without it. The resulting AAPO-7B exhibits better performance on the benchmarks when the clip operation is adopted.

Choice of the reference model Since the AAPO requires using the reference model to generate responses and obtain rewards r_{ref_i}, it is necessary to investigate the selection of the reference model. We further explored two scenarios: using the base model (initial model) as the reference model and replacing the reference model with the policy model during training. For replacing reference model, the reference model was continuously updated by replacing it with the current policy model, we update the reference model every 20 steps and 100 steps in AAPO-1.5B and AAPO-7B training process, respectively. Experimental results indicate that using the base model as the reference model yields the best performance. The experimental results are in Table 3. Keeping the reference model as the initial model yields superior results because AAPO's core mechanism relies on the advantage margin in Equation (5). Updating the reference model during training reduces the value of advantage margin, making AAPO less effective.

Model	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
AAPO-1.5B	33.3	86.0	80.0	30.9	53.3	**56.7**
AAPO-1.5B *w/o* clip	33.3	85.0	82.5	29.0	53.3	56.6
AAPO-7B	30.0	82.4	70.0	35.3	44.3	**52.4**
AAPO-7B *w/o* clip	30.0	79.6	70.0	34.9	42.5	51.2

Table 2: Ablation study results. *w/o* indicates without clip operation on the advantage margin. Zero-shot pass@1 performance on different benchmarks.

Model	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
AAPO-1.5B	33.3	86.0	80.0	30.9	53.3	**56.7**
AAPO-1.5B *w* reference update	36.7	84.2	75.0	27.9	53.2	55.4
AAPO-7B	30.0	82.4	70.0	35.3	44.3	**52.4**
AAPO-7B *w* reference update	26.7	80.6	70.0	33.8	44.0	51.0

Table 3: Ablation on the choice of the reference model.

6.4 Training dynamics of advantage

To further illustrate how our method AAPO mitigates the zero advantage phenomenon compared to the group relative advantage estimation method, we present the proportion of zero advantage per device during training across different group sizes. As shown in Figure 2, as the number of training steps increases, the vanilla group relative advantage estimation method A (calculated by Equation (2)) exhibits a higher proportion of zero advantage per device. In the later steps of training, the proportion of zero advantage increases significantly, and the frequency of all advantages being zero also increases markedly, leading to zero gradients and consequently reducing training efficiency. However, during training with AAPO, regardless of the group size, no instance of all responses having all zero advantage occurred. These training dynamics demonstrate that our A^* (calculated by Equation (5)) effectively mitigates the occurrence of all zero advantage, transforming what would oth-

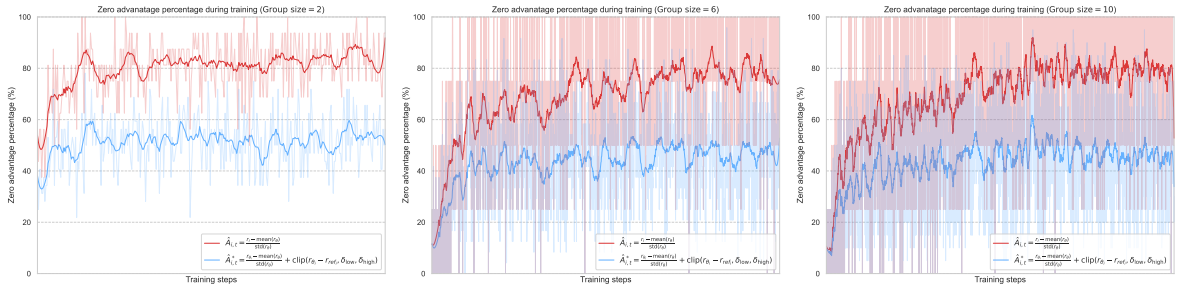

Figure 2: Training dynamics of the proportion of zero advantage with varying group size. We calculated the percentage of zero advantage under vanilla A and our A^* across different group size, presenting the average zero advantage percentage per device.

erwise be ineffective gradient updates into meaningful ones.

6.5 Computational efficiency

Since AAPO requires sampling on the reference model, it introduces additional training overhead. We measured the training time and GPU memory usage per step using AAPO compared to GRPO on the same computational device. The additional experimental information are shown in Table 4. When training 1.5B model, AAPO and GRPO both use 52.2GiB GPU memory per device, AAPO takes 826s per step and GRPO takes 567s per step. When training 7B model, AAPO use 78.9GiB GPU memory per device and GRPO use 78.7GiB GPU memory per device, AAPO takes 384s per step and GRPO takes 299s per step. It is clear that AAPO does not introduce extra GPU memory overhead and remains on par with GRPO. The reason for why 1.5B model takes longer training time per step is that 1.5B model samples responses with much longer length (i.e., average_response_length $\approx$ 3000 for 1.5B model and average_response_length $\approx$ 700 for 7B model during training). Since we use vanilla inference method (i.e., model.generate()) to generate responses, the extra time consumption could be significantly reduced using an advanced inference engine (e.g., VLLM) during training.

Method	Memory	Time
	1.5B Model	
GRPO	52.2GiB	567s
AAPO	52.2GiB	826s
	7B Model	
GRPO	78.7GiB	299s
AAPO	78.9GiB	384s

Table 4: Extra experimental information about computational efficiency.

6.6 More experimental results

Experimental results about Out-of-Domain performance, more training dynamics of AAPO, and more ablation studies can be found in Appendix D.

7 Conclusion

In this paper, we conduct an in-depth analysis of the limitations inherent in the group relative advantage estimation method used by mainstream RL algorithms, such as GRPO, which would lead to optimization issues such as zero gradient and gradient ascent. To address these issues, we propose a novel RL algorithm Advantage-Augmented Policy Optimization (AAPO). By augmenting the group relative advantage estimation method with advantage margin, our method effectively improves policy optimization performance in experimental benchmarks. Experimental results across several mathematical reasoning benchmarks and model series demonstrate that AAPO achieves the overall superior performance.

Limitations

Our proposed AAPO can effectively mitigate the phenomenon that the estimated advantage approaches zero, but AAPO cannot eliminate this phenomenon, leaving this problem for further research. Another limitation of our work lies in the long training time required, since AAPO depends on sampling responses from the reference model. However, this could be optimized by using advanced inference library (e.g. vLLM (Kwon et al., 2023)) to generate all responses.

References

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. 2025. An empirical study on eliciting and improving r1-like reasoning models. *Preprint*, arXiv:2503.04548.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2023. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems*.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2026. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *The Fourteenth International Conference on Learning Representations*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, and 6 others. 2025. Process reinforcement through implicit rewards. *Preprint*, arXiv:2502.01456.

Quy-Anh Dang and Chris Ngo. 2026. Reinforcement learning for reasoning in small LLMs: What works and what doesn't. In *Logical and Symbolic Reasoning in Language Models @ AAAI 2026*.

Tom Everitt, Marcus Hutter, Ramana Kumar, and Victoria Krakovna. 2021. Reward tampering problems and solutions in reinforcement learning: A causal influence diagram perspective. *Synthese*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *Proceedings of the 35th International Conference on Machine Learning*.

Nathan Habib, Clémentine Fourrier, Hynek Kydlíček, Thomas Wolf, and Lewis Tunstall. 2023. Lighteval: A lightweight framework for llm evaluation.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *International Conference on Learning Representations*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025a. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization. *Preprint*, arXiv:2501.03262.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025b. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *Preprint*, arXiv:2503.24290.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, Yikai Zhang, Yuqing Yang, Ting Wu, Binjie Wang, Shichao Sun, Yang Xiao, Yiyuan Li, Fan Zhou, Steffi Chern, and 9 others. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. In *Advances in Neural Information Processing Systems*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *The Third International Conference on Learning Representations*.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*.

Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM Comput. Surv.*

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. In *Proceedings of the Conference On Language Modeling*.

OpenAI. 2024. Introducing OpenAI O1 Preview. https://openai.com/index/introducing-openai-o1-preview/.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. 2022. The effects of reward misspecification: Mapping and mitigating misaligned models. In *International Conference on Learning Representations*.

Qwen Team. 2024. QwQ: Reflect Deeply on the Boundaries of the Unknown. https://qwenlm.github.io/blog/qwq-32b-preview/.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, and 1 others. 2018. Improving language understanding by generative pre-training.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems*.

Herbert Robbins and David Siegmund. 1971. A convergence theorem for non negative almost supermartingales and some applications. In *Optimizing methods in statistics*. Elsevier.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. 2019. A meta-analysis of overfitting in machine learning. In *Advances in Neural Information Processing Systems*.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning*.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. In *The Fourth International Conference on Learning Representations*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *Preprint*, arXiv:1707.06347.

Claude E Shannon. 1948. A mathematical theory of communication. *The Bell system technical journal*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. *Proceedings of the AAAI Conference on Artificial Intelligence*.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems*.

RUCAIBoxSTILL Team. 2025. Still-3-1.5b-preview: Enhancing slow thinking abilities of small models through reinforcement learning. https://github.com/RUCAIBox/Slow_Thinking_with_LLMs.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Chaojian Yu, Bo Han, Li Shen, Jun Yu, Chen Gong, Mingming Gong, and Tongliang Liu. 2022. Understanding robust overfitting of adversarial training and beyond. In *Proceedings of the 39th International Conference on Machine Learning*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2026. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2026. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Proceedings of the Conference On Language Modeling*.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, and 3 others. 2026. A survey of large language models. *Preprint*, arXiv:2303.18223.

Daniel M. Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B. Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2020. Fine-tuning language models from human preferences. *Preprint*, arXiv:1909.08593.

A Algorithm

AAPO Training The training process of AAPO can be summarized as follows: 1) Sample two groups of responses, O_θ and O_{ref}, of equal size from both the policy model π_θ and the reference model π_{ref}, respectively. 2) For each response in O_θ, compute its group relative advantage and the advantage margin with the corresponding response in O_{ref}. The values, calculated by (5), are then used to perform gradient updates according to (6). It is important to note that during the training process, the parameters of the reference model π_{ref} remain frozen and do not undergo gradient updates. The training procedure is described in Algorithm 1.

B Proof

Assumption *Here, we state all assumptions we use in our proof of theorem. (1) The gradient of the log-policy is bounded:* $||\nabla_\theta \log \pi_\theta(o)|| \leq M$, *and* $-\log \pi_\theta(o)$ *is* L_0*-Lipschitz. (2) Assume that the stochastic gradient is unbiased and that the per-sample gradient has bounded second moment. (3) For all* θ, *the per-sample loss* $\ell(q, o; \theta) = -\log \pi_\theta(o \mid q)\, \hat{A}^*$ *has a gradient with bounded second moment:* $\sup_\theta \mathbb{E}_{(q,o)\sim\pi_\theta}\left[\|\nabla_\theta \ell(q, o; \theta)\|^2\right] \leq \sigma^2$, *where* σ^2 *is a constant.*

Theorem 1. *(Stability) Since the rewards are bounded, the group standard deviation satisfies* $0 \leq \sigma_{min} \leq \sigma$, *and the log-likelihood gradients are bounded as* $||\nabla_\theta \log \pi_\theta(o)|| \leq M$. *Then, each gradient step with learning rate* η_k *satisfies* $||\theta_{k+1} - \theta_k|| \leq \eta_k MB$, *where* $B = \frac{R_{max} - R_{min}}{\sigma_{min}} + \max(|\delta_{\text{low}}|, |\delta_{\text{high}}|)$ *is the uniform bound on the AAPO weights. The expected objective is bounded from* $\mathcal{L}(\theta) \geq -B \log |\mathcal{V}|$, *where* $|\mathcal{V}|$ *is the vocabulary size. Hence, AAPO training is stable: the objective cannot diverge to* $-\infty$ *and parameter updates are always finite.*

Proof. We restate the definitions used: For a group $\mathcal{G}$, the gradient of the empirical loss is:

$$\nabla_\theta \mathcal{L}_\mathcal{G}(\theta) = \frac{1}{N_\mathcal{G}} \sum_{o \in O} \nabla_\theta [-\log \pi_\theta(o)] \hat{A}^*.$$

By assumption (1), we have

$$||\nabla_\theta \mathcal{L}_\mathcal{G}(\theta)|| \leq \frac{1}{N_\mathcal{G}} \sum_{o \in O} M \cdot B = MB.$$

Thus, one gradient update with learning rate η_k yields

$$||\theta_{k+1} - \theta_k|| = \eta_k ||\nabla_\theta \mathcal{L}_\mathcal{G}(\theta)|| \leq \eta_k MB.$$

For any response o,

$$\mathbb{E}_{a\sim\pi_\theta}\left[-\log \pi_\theta(o)\right] = H(\pi_\theta(\cdot \mid o_{<t})) \leq \log |\mathcal{V}|.$$

where $H(\cdot) = -\sum_x p(x) \log p(x)$ denotes Shannon entropy (Shannon, 1948). Multiplying by the

Algorithm 1: Advantage-Augmented Policy Optimization

Input: policy model π_θ, reference model π_{ref}, group size G, reward functions F=$\{\text{format},\text{accuracy},\cdots\}$, reward functions' corresponding weights W=$\{w_{\text{format}},w_{\text{accuracy}},\cdots\}$, data batch $\mathcal{B}$, total training steps $\mathcal{S}_{gloabl}$.

for *global training step* $\mathcal{S} < \mathcal{S}_{gloabl}$ **do**
  for *data in data batch* $\mathcal{B}$ **do**
    Sample $O_\theta = \{o_{\theta_1}, o_{\theta_2}, \cdots, o_{\theta_G}\}$ from π_θ;
    Sample $O_{ref} = \{o_{ref_1}, o_{ref_2}, \cdots, o_{ref_G}\}$ from π_{ref};
    Compute rewards R_θ=$\{R_\theta^{\text{format}},R_\theta^{\text{accuracy}},\cdots,R_\theta^{\cdots}\}$ and R_{ref}=$\{R_{ref}^{\text{format}},R_{ref}^{\text{accuracy}},\cdots,R_{ref}^{\cdots}\}$ for each response in O_θ and O_{ref} using the reward functions in F;
    Compute the weighted rewards $R_\theta^{\text{weighted}}$=$R_\theta W^\top$ and $R_{ref}^{\text{weighted}}$=$R_{ref} W^\top$;
    Compute the augmented advantage $\hat{A}^*_{i,t}$ following Equation (5) for each sample in O_θ;
    Compute loss for O_θ following Equation (4);
  end
  Update π_θ following Equation (6);
end

bounded advantage $|\hat{A}^*| \le B$ and averaging over tokens in the batch, we obtain

$$\mathcal{L}(\theta) \ge -B \log |\mathcal{V}|.$$

we conclude that parameter updates are bounded and the objective is lower bounded (which means the objective cannot diverge to $-\infty$). Therefore, AAPO training is stable and cannot diverge. □

Theorem 2. *(Convergence) By assumption (2), let the step sizes satisfy the Robbins–Monro conditions* $\eta_k > 0$, $\sum_k \eta_k = \infty$, $\sum_k \eta_k^2 < \infty$. *AAPO converges to a stationary point of its expected objective* $\liminf_{k\to\infty} \mathbb{E}\left[\|\nabla \mathcal{L}(\theta_k)\|^2\right] = 0$. *Moreover, if a constant step size* $\eta < \frac{1}{BL_0}$ *is used, where* L_0 *is the smoothness constant of* $-\log \pi_\theta(o)$*, then the iterates converge to a neighborhood of stationarity* $\limsup_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\left[\|\nabla \mathcal{L}(\theta_k)\|^2\right] \lesssim \mathcal{O}(\eta) + \mathcal{O}\left(\frac{1}{N_\mathcal{G}}\right)$.

Proof. We adopt the same definitions as in the Proof of Theorem 1. Since $-\log \pi_\theta(o \mid q)$ is L_0-smooth (i.e., its gradient is L_0-Lipschitz) and the advantage is bounded by B, the empirical loss $\mathcal{L}_\mathcal{G}(\theta)$ is $L = BL_0$-smooth:

$$\left\|\nabla \mathcal{L}_\mathcal{G}(\theta) - \nabla \mathcal{L}_\mathcal{G}(\theta')\right\|_2 \le L \left\|\theta - \theta'\right\|_2.$$

By the descent lemma for L-smooth functions, for any step size η_k,

$$\mathcal{L}_\mathcal{G}(\theta - \eta_k g) \le \mathcal{L}_\mathcal{G}(\theta) - \eta_k \langle g, \nabla \mathcal{L}_\mathcal{G}(\theta)\rangle + \tfrac{L}{2}\eta_k^2 \|g\|^2.$$

Choosing $g = \nabla \mathcal{L}_\mathcal{G}(\theta_k)$ and requiring $\eta_k \le 1/L$, we obtain

$$\mathcal{L}_\mathcal{G}(\theta_{k+1}) \le \mathcal{L}_\mathcal{G}(\theta_k) - \tfrac{\eta_k}{2} \left\|\nabla \mathcal{L}_\mathcal{G}(\theta_k)\right\|^2.$$

By assumption (3), for all θ, the per-sample loss $\ell(q, o; \theta) = -\log \pi_\theta(o \mid q)\, \hat{A}^*$ has a gradient with bounded second moment:

$$\sup_\theta \mathbb{E}_{(q,o)\sim\pi_\theta}\left[\|\nabla_\theta \ell(q, o; \theta)\|^2\right] \le \sigma^2,$$

where σ^2 is a constant. Since $\nabla \mathcal{L}_\mathcal{G}(\theta_k)$ is the average of $N_\mathcal{G}$ i.i.d. samples conditional on θ_k, we have

$$\mathbb{E}[\nabla \mathcal{L}_\mathcal{G}(\theta_k)] = \nabla \mathcal{L}(\theta_k),$$
$$\mathbb{E}\left[\|\nabla \mathcal{L}_\mathcal{G}(\theta_k) - \nabla \mathcal{L}(\theta_k)\|^2\right] \le \frac{\sigma^2}{N_\mathcal{G}}.$$

Consequently (by variance decomposition),

$$\mathbb{E}\left[\|\nabla \mathcal{L}_\mathcal{G}(\theta_k)\|^2\right] \le \|\nabla \mathcal{L}(\theta_k)\|^2 + \frac{\sigma^2}{N_\mathcal{G}}.$$

Taking total expectation and noting $\mathbb{E}[\mathcal{L}_\mathcal{G}(\theta)] = \mathcal{L}(\theta)$, we obtain

$$\mathbb{E}[\mathcal{L}(\theta_{k+1})] \le \mathbb{E}[\mathcal{L}(\theta_k)] - \tfrac{\eta_k}{2}\mathbb{E}\left[\|\nabla \mathcal{L}(\theta_k)\|^2\right] + \tfrac{L}{2}\frac{\eta_k^2 \sigma^2}{N_\mathcal{G}}.$$

Since $\mathcal{L}(\theta)$ is lower bounded (Theorem 1), applying the Robbins–Siegmund theorem (Robbins and Siegmund, 1971) gives

$$\sum_{k=0}^{\infty} \eta_k \,\mathbb{E}\left[\|\nabla \mathcal{L}(\theta_k)\|^2\right] < \infty.$$

Given $\sum_k \eta_k = \infty$, it follows that

$$\liminf_{k\to\infty} \mathbb{E}\Big[\|\nabla \mathcal{L}(\theta_k)\|^2\Big] = 0.$$

If $\eta_k \equiv \eta \le 1/L$ is fixed, the residual is of order $\mathcal{O}(\eta)$ (from smoothness) and $\mathcal{O}(1/N_{\mathcal{G}})$ (from gradient variance). Hence

$$\limsup_{K\to\infty} \frac{1}{K}\sum_{k=1}^{K} \mathbb{E}\Big[\|\nabla \mathcal{L}(\theta_k)\|^2\Big] \;\lesssim\; \mathcal{O}(\eta) + \mathcal{O}\Big(\tfrac{1}{N_{\mathcal{G}}}\Big).$$

With diminishing step sizes, the algorithm converges to a stationary point of the expected objective; with a small constant step size, it converges to a neighborhood of stationarity whose size depends on η and the group size. The advantage margin does not affect the asymptotic rates, as it only changes the constant B (and hence $L = BL_0$). □

C Experiment details

C.1 Reward rules

Format Reward To encourage adherence to structured reasoning, we adopt a binary format reward $R_{format}(o) \in \{0, 1\}$, which assigns a reward of 1 if the model response o conforms to the expected template by containing the delimiter sequence "\n </think>\n", and 0 otherwise.

Cosine Scaled Reward We adopt the $R_{cosine\_scaled\_accuracy} \in \{0, 1\}$ as expressed in equation (9), which encourages correct outputs with shorter lengths and penalizes incorrect outputs with reduced severity as their length increases, following a cosine annealing schedule in equation (10).

$$R_{cosine\_scaled\_accuracy}(o) = \begin{cases} R_{\text{correct}}(l), & \text{if correct} \\ R_{\text{wrong}}(l), & \text{if wrong} \end{cases}, \quad (9)$$

where

$$\begin{aligned} R_{\text{correct}}(l) &= \alpha^c_{\min} + \frac{1}{2}(\alpha^c_{\max} - \alpha^c_{\min})\left[1 + \cos\left(\pi \frac{l}{L}\right)\right], \\ R_{\text{wrong}}(l) &= \alpha^w_{\max} + \frac{1}{2}(\alpha^w_{\min} - \alpha^w_{\max})\left[1 + \cos\left(\pi \frac{l}{L}\right)\right], \end{aligned} \quad (10)$$

Accuracy Reward We adopt a standard accuracy reward $R_{accuracy} \in \{0, 1\}$, which assigns a binary reward of 1 for correct response and 0 otherwise, providing a sparse but direct reward signal.

C.2 Training setup

Training DeepSeek-R1-Distill-Qwen-1.5B For training DeepSeek-R1-Distill-Qwen-1.5B on openrs experiment (Dang and Ngo, 2026), we adopt the R_{format} and $R_{cosine\_scaled\_accuracy}$ reward functions, with respective weights of 1 and 2. We directly perform our AAPO training from the base model without any SFT with the group size of 6 and the per_device_batch_size of 6 with gradient_accumulation_steps of 4 on 2 Nvidia A800 GPUs with 80G VRAM. The system prompt adopted in the RL training is provided below.

Training prompt AAPO-1.5B model

A conversation between User and Assistant. The user asks a question, and the Assistant solves it. The assistant first thinks about the reasoning process in the mind and then provides the user with the answer, and put your final answer within \\boxed . The reasoning process and answer are enclosed within <think></think >and <answer></answer>tags, respectively, i.e., <think>reasoning process here </think ><answer>answer here </answer>. Note that respond by English, NOT use other languages.

Training Qwen2.5-Math-7B For training Qwen2.5-Math-7B (Yang et al., 2024) on simplelr_qwen_level3to5 (Zeng et al., 2025), we adopt the $R_{accuracy}$ reward function with a weight of 1. We directly perform our AAPO training without any SFT with the group size of 8 and the per_device_batch_size 8 with gradient_accumulation_steps of 4 on 8 Nvidia A800 GPUs with 80G VRAM. The system prompt adopted in the RL training is provided below.

Training prompt for AAPO-7B model

You are a helpful AI Assistant, designed to provided well-reasoned anddetailed responses. You FIRST think about the reasoning process as an internal monologue and then provide the user with the answer. The reasoning process MUST BE enclosed within <think>and </think>tags.

Training Llama series models For training Llama-3.2-1B-Instruct and Llama-3.2-3B-Instruct models, we adopt the simplelr_able_level3to5 dataset (Zeng et al., 2025) and the $R_{accuracy}$ reward function with a weight of 1. We directly perform our AAPO training without any SFT with the group size of 8 and the per_device_batch_size of 16 on 4

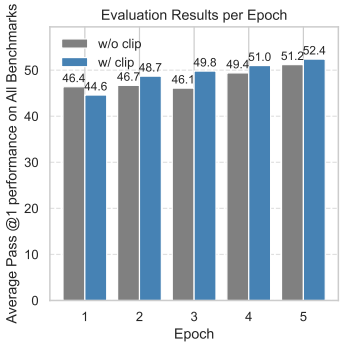

Figure 3: Ablation study on clip operation, reported by average pass@1 scores across five benchmarks.

Nvidia A800 GPUs with 80G VRAM. There is no extra system prompt added when training Llama series models.

C.3 Evaluation setup

Evaluation of Qwen series models All the reproduced results in Table 1 are evaluated using the lighteval framework (Habib et al., 2023) with vllm backend proposed by Hugging Face. Our evaluation experiments on all benchmarks are conducted on a single Nvidia A800 GPU with 80G VRAM, with system prompt provided in below.

System prompt for Qwen series evaluation on all benchmarks

Solve the following math problem efficiently and clearly. The last line of your response should be of the following format: 'Therefore, the final answer is: $\\boxed{{ ANSWER}}$. I hope it is correct' (without quotes) where ANSWER is just the final number or expression that solves the problem. Think step by step before answering.

Evaluation of Llama series models All the reported results in Table 1 are evaluated using vllm backend for generation and the evaluation script provided by Zeng et al. (2025). Our evaluation experiments on all benchmarks are conducted on 4 Nvidia A800 GPUs with 80G VRAM.

D More experiment results

D.1 Performance on Out-of-Domain benchmarks

Beyond mathematical reasoning tasks, we evaluated AAPO-1.5B and AAPO-7B on a more general benchmark: MMLU (Hendrycks et al., 2021a), MMLU encompasses 57 subdomains spanning STEM, social science, and other domains. As reported in Table 9, results show that even when our method trained models using only a small subset of math problems, AAPO-1.5B achieved a slight improvement (**+0.27**) over DeepSeek-R1-Distill-Qwen-1.5B model, attributing to the base model's inherent strength. However, AAPO-7B demonstrated substantial gains (**+13.69**) over Qwen2.5-Math-7B model. Except for MMLU benchmark, we evaluate AAPO-1.5B on LiveCodeBench (Jain et al., 2025), which evaluates models on a variety of code-related scenarios, such as code generation, self-repair, test output prediction, and code execution. Since the max_model_length of AAPO-7B is 4,096, it is not appropriate to evaluate AAPO-7B on LiveCodeBench. As shown in Table 5, AAPO-1.5B also outperforms its base model by **1.23** on code-related benchmark. These findings demonstrate that our AAPO does not compromise model's generalization. Instead, it extends model's reasoning capabilities to other domains to a certain extent.

LiveCodeBench	DeepSeek-R1-Distill-Qwen-1.5B	AAPO-1.5B	Δ
Average	25.78	27.01	+1.23

Table 5: Performance on LiveCodeBench benchmark.

D.2 The effect of training group size

To examine the effect of group size G on training dynamics, we conducted a set of additional experiments with varying group sizes on DeepSeek-R1-Distill-Qwen-1.5B and Qwen2.5-Math-7B models. The results are summarized in Table 6. The experimental results indicate that AAPO is robust to group size across model size. Based on this observation, we use group size $G = 6$ as our main results in Table 1, which is consistent to other baselines.

Model	G	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
AAPO-1.5B	2	36.7	83.0	80.0	32.0	51.4	56.6
	6	33.3	86.0	80.0	30.9	53.3	56.7
	10	36.7	85.6	77.5	27.6	52.0	55.9
AAPO-7B	2	33.3	78.8	62.5	32.4	39.4	49.3
	6	30.0	82.4	70.0	35.3	44.3	52.4
	10	36.7	81.8	65.0	35.3	44.6	52.7

Table 6: Additional ablation study on group size G.

D.3 Pass@k performance

Since our main experiments are reported using the pass@1 metric, we conducted additional pass@k

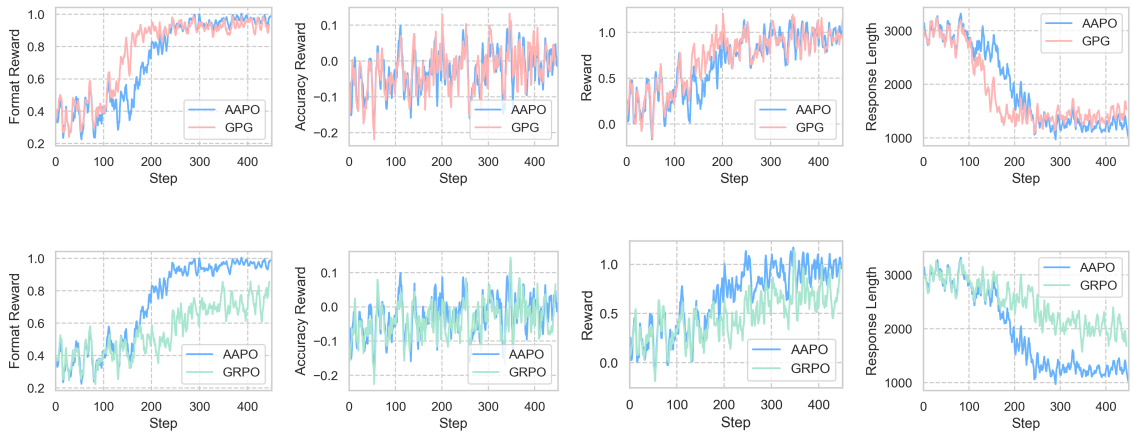

Figure 4: Training process with DeepSeek-R1-Distill-Qwen-1.5B on open-rs (Dang and Ngo, 2026) utilizing our proposed AAPO algorithm. Compared to GPG (Chu et al., 2026) and GRPO (Shao et al., 2024), AAPO demonstrates better stability during training and achieves superior performance in the final results as shown in Table 1.

Model	K	AIME24	MATH-500	AMC23	Minerva	OlympiadBench
DeepSeek-R1-Distill-Qwen-1.5B	8	64.3	95.6	93.5	48.2	72.2
	16	73.2	97.1	95.4	52.1	76.4
	32	78.9	98.1	97.4	55.2	79.7
	64	83.3	99.0	100.0	57.1	81.9
AAPO-1.5B	8	62.6	95.6	93.2	47.5	72.9
	16	70.9	96.9	94.4	50.9	77.3
	32	77.3	97.6	94.9	53.5	80.5
	64	80.0	98.0	95.0	55.9	83.1
Qwen2.5-Math-7B	8	39.5	86.1	80.1	32.5	50.9
	16	46.2	90.0	84.7	39.3	58.2
	32	51.6	92.8	89.1	45.4	64.1
	64	56.7	94.8	95.0	50.4	68.7
AAPO-7B	8	50.7	91.9	85.4	48.0	61.9
	16	70.9	93.6	90.4	50.9	66.3
	32	77.3	95.1	95.5	53.7	70.0
	64	80.0	96.2	100.0	56.6	72.7

Table 7: Additional results with pass@k metrics.

experiments. In these extra experiments, we sampled 64 samples to compute pass@k with $k = 8/16/32/64$. Results are reported in Table 7. As shown in the results table, both the base model and the trained model exhibit improved performance as the value of k increases. Consistent with the findings presented by Yue et al. (2026), reinforcement learning with verifiable rewards (RLVR) for LLMs primarily enhances the models' pass@1 performance. In some cases, the pass@k performance of the RLVR-trained model even falls short of the base model. For instance, DeepSeek-R1-Distill-Qwen-1.5B, trained using a large scale of high-quality CoT data distilled from DeepSeek-R1, shows limited improvement after training with RLVR in pass@1 performance in Table 1. Even as k increases, AAPO-1.5B's pass@k performance remains slightly inferior to the base model except for OlympiadBench. However, AAPO-7B consistently outperforms its base model Qwen2.5-Math-7B for $k = 1/8/16/32/64$ on all benchmarks, since Qwen2.5-Math-7B retains considerable potential for improvement.

D.4 Training Analysis

Training process analysis As illustrated by the training curves in Figure 4, AAPO achieves optimization performance comparable to that of GPG, while exhibiting further improvements during the later steps of RL training. In the Format Reward and Reward figures, AAPO consistently attains higher reward than GPG in the later steps of the training. In addition, three Reward figures demonstrate significantly reduced fluctuations during the training process. AAPO also outperforms GRPO. Moreover, in the Response Length figure, our proposed AAPO demonstrates superior optimization, indicating better training effectiveness. The final results presented in Table 1 demonstrate that our proposed AAPO achieves superior performance across five mathematical reasoning benchmarks. To straightly and effectively analyze the training stability of AAPO, we calculated the variance of the training loss for AAPO as well as methods GRPO and GPG. $\mathrm{Var}(\mathcal{L}_{\mathrm{AAPO}}) = 3.6 \times 10^{-4}$, $\mathrm{Var}(\mathcal{L}_{\mathrm{GPG}}) = 3.9 \times 10^{-4}$ and $\mathrm{Var}(\mathcal{L}_{\mathrm{GRPO}}) = 3.53 \times 10^{-3}$. The results show that AAPO exhibits the lowest variance, indicating relatively stable training. Additionally, the results of the ablation experiments on clip operation demonstrate that AAPO can steadily improve model performance as training epochs progress.

Training loss analysis We plot the loss curves of AAPO and GPG (Chu et al., 2026) during train-

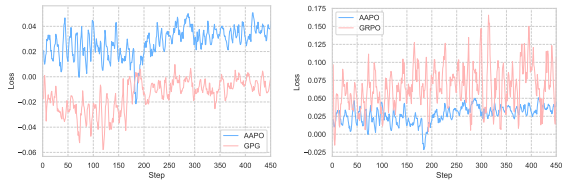

Figure 5: A comparative analysis of training loss between AAPO and GPG, AAPO and GRPO.

ing DeepSeek-R1-Distill-Qwen-1.5B on open-rs (Dang and Ngo, 2026) in Figure 5. As observed, the loss values for AAPO remain predominantly positive throughout the training process. This indicates that AAPO primarily optimizes the policy by encouraging diverse responses better than the reference, assigning different gradient magnitudes according to the advantage estimated by $\hat{A}^*_{i,t}$, thus leading to performance improvements. In contrast, GPG exhibits mostly negative loss values, suggesting that it focuses on suppressing suboptimal responses as its main optimization strategy. It can be clearly seen from the Figure 5 that the training process of AAPO is more stable than GRPO (Shao et al., 2024). These results of the analysis imply that models trained with our proposed AAPO demonstrate stronger generalization, resulting in superior performance as shown in Table 1.

E Extra comparison with original results

It is worth noting that evaluation results may be influenced by the computational device type. We have also provided the original results of each model from the corresponding papers. Here we present the original results reported in the corresponding papers for Qwen series models in Table 8.

Model	AIME24	MATH-500	AMC23	Minerva	OlympiadBench	Avg.
Qwen 1.5B Models						
DeepSeek-R1-Distill-Qwen-1.5B	28.9	83.9	–	–	–	–
GRPO-1.5B (Dang and Ngo, 2026)	46.7	84.4	72.5	26.8	51.3	56.3
GPG-1.5B (Chu et al., 2026)	33.3	85.0	80.0	26.8	52.4	55.5
Still-3-1.5B-Preview (Chen et al., 2025)	39.3	85.5	–	–	–	–
AAPO-1.5B (Ours)	33.3	86.0	80.0	30.9	53.3	**56.7**
Qwen 7B Models						
Qwen2.5-Math-7B (Yang et al., 2024)	–	55.4	–	–	–	–
SimpleRL-Zero-7B (Zeng et al., 2025)	20.0	78.2	62.5	38.6	40.4	47.9
GPG-7B(Chu et al., 2026)	33.3	80.0	65.0	34.2	42.4	51.0
OpenReasoner-Zero-7B (Hu et al., 2025b)	–	–	–	–	–	–
Eurus-2-7B-PRIME (Cui et al., 2025)	20.0	78.2	50.6	39.3	40.3	45.7
Oat-Zero-7B(Liu et al., 2025)	43.3	80.0	62.7	30.1	41.0	51.4
AAPO-7B (Ours)	30.0	82.4	70.0	35.3	44.3	**52.4**

Table 8: Zero-shot pass@1 performance on mathematical reasoning benchmarks. All reported results in this table are directly adopted from the corresponding papers. Dashes (–) denote unavailable official score.

F Disclosure of AI assistants usage

AI assistants are utilized to assist in the drafting, refinement, and wording adjustments of portions of the paper's text. All content was ultimately reviewed and revised by the authors, who are solely responsible for the facts, assertions, and arguments presented herein.

MMLU task-domain	DeepSeek-R1-Distill-Qwen-1.5B	AAPO-1.5B	Δ	Qwen2.5-Math-7B	AAPO-7B	Δ
abstract_algebra	23.00	22.00	-1.00	21.00	22.00	+1.00
anatomy	22.96	24.44	+1.48	25.19	39.26	+14.07
astronomy	21.71	22.37	+0.66	19.47	34.87	+15.13
business_ethics	21.00	23.00	+2.00	33.00	50.00	+17.00
clinical_knowledge	30.19	28.68	-1.51	23.02	40.00	+16.98
college_biology	25.00	23.61	-1.39	27.78	50.69	+22.91
college_chemistry	29.00	26.00	-3.00	21.00	27.00	+6.00
college_computer_science	25.00	25.00	0.00	27.00	37.00	+10.00
college_mathematics	20.00	20.00	0.00	21.00	25.00	+4.00
college_medicine	32.95	32.95	0.00	20.81	42.2	+21.39
college_physics	22.55	21.57	-0.98	21.57	27.45	+5.88
computer_security	26.00	27.00	+1.00	29.00	50.00	+21.00
conceptual_physics	26.81	26.38	-0.43	27.23	53.19	+25.96
econometrics	24.56	25.44	+0.88	24.56	28.95	+4.39
electrical_engineering	27.59	27.59	0.00	24.14	48.28	+24.14
elementary_mathematics	21.16	20.90	-0.26	21.16	31.48	+10.32
formal_logic	24.60	25.40	+0.80	28.57	35.71	+7.14
global_facts	20.00	21.00	+1.00	19.00	21.00	+2.00
high_school_biology	23.87	23.55	-0.32	21.29	47.42	+26.13
high_school_chemistry	15.27	15.27	0.00	16.75	41.38	+24.63
high_school_computer_science	26.00	26.00	0.00	28.00	51.00	+23.00
high_school_european_history	29.70	30.30	+0.60	21.81	40.00	+18.19
high_school_geography	30.30	28.79	-1.51	22.73	50.51	+27.78
high_school_government_and_politics	27.46	26.94	-0.52	30.57	46.63	+16.06
high_school_macroeconomics	27.95	27.95	0.00	22.56	42.82	+20.26
high_school_mathematics	21.85	21.85	0.00	21.11	22.59	+1.48
high_school_microeconomics	28.57	26.89	-1.68	27.73	48.32	+20.59
high_school_physics	19.21	19.87	+0.66	19.87	36.42	+16.55
high_school_psychology	24.40	24.40	0.00	25.69	57.43	+31.74
high_school_statistics	16.20	15.74	-0.46	16.20	27.78	+11.58
high_school_us_history	25.98	26.47	+0.49	25.00	38.73	+13.73
high_school_world_history	24.05	23.63	-0.42	27.34	38.82	+11.39
human_aging	18.83	20.18	+1.35	33.63	38.57	+4.94
human_sexuality	22.90	27.48	+4.58	29.77	38.93	+9.16
international_law	19.84	20.66	+0.82	28.10	40.50	+12.40
jurisprudence	24.07	25.93	+1.86	37.04	42.59	+5.55
logical_fallacies	28.22	29.45	+1.23	26.38	47.24	+20.86
machine_learning	34.82	35.71	+0.89	31.25	36.61	+5.36
management	18.45	18.45	0.00	25.24	48.54	+23.30
marketing	26.50	27.35	+0.85	31.62	66.23	+34.61
medical_genetics	20.00	23.00	+3.00	30.00	33.00	+3.00
miscellaneous	23.24	23.88	+0.64	32.70	52.11	+19.41
moral_disputes	29.19	29.19	0.00	39.31	40.46	+1.15
moral_scenarios	24.81	24.81	0.00	23.80	23.91	+0.11
nutrition	26.80	26.80	0.00	30.39	42.16	+11.77
philosophy	20.26	18.01	-2.25	18.97	31.19	+12.22
prehistory	24.69	23.77	-0.92	25.93	38.58	+12.65
professional_accounting	23.76	24.11	+0.35	23.76	30.14	+6.38
professional_law	24.58	25.55	+0.97	24.58	29.01	+4.43
professional_medicine	16.18	15.81	-0.37	18.38	27.57	+9.19
professional_psychology	26.63	25.33	-1.30	26.14	41.50	+15.36
public_relations	17.27	20.91	+3.64	22.73	35.46	+12.73
security_studies	35.29	34.69	-1.23	20.00	35.10	+15.10
sociology	25.37	28.86	+3.49	27.86	44.78	+16.92
us_foreign_policy	26.00	25.00	-1.00	39.00	50.00	+11.00
virology	19.88	21.08	+1.20	28.92	42.17	+13.25
world_religions	20.47	21.64	+1.17	42.69	49.71	+7.02
Average	24.27	24.54	+0.27	25.96	39.65	+13.69

Table 9: Performance on MMLU benchmark.