LEARNING FROM HARD PROMPTS: DIFFICULTY-AWARE ADVANTAGE AMPLIFICATION IN DYNAMIC SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) is a prominent variant of Group Relative Policy Optimization (GRPO). DAPO introduces several improvements over GRPO. Among these, *Dynamic Sampling* contributes the most to DAPO's accuracy improvement, which has also been adopted in training frontier models. To improve accuracy, Dynamic Sampling enhances training stability by eliminating zero policy gradients caused by zero advantages. Specifically, it avoids zero gradients by filtering out prompts where sampled responses are either correct or incorrect. However, our theoretical analysis shows that Dynamic Sampling decrease training efficiency as it cannot effectively utilize hard-to-sample correct responses on hard prompts. Formally, it asymmetrically amplifies the advantages of distinct responses to the same prompts. On hard prompts, incorrect responses undergo much greater amplification than correct ones. This leads the model to avoid generating the observed incorrect responses rather than capitalizing on the hard-to-sample correct ones on hard prompts, resulting in low training efficiency. To improve training efficiency, we propose *Direct Advantage Amplification* (DAA), which amplifies the advantages of hard-to-sample correct responses retained by Dynamic Sampling on hard prompts. This ensures that, these hard-to-sample responses can be effectively capitalized on when Dynamic Sampling is used, implying higher training efficiency. By integrating DAA into DAPO, we obtain *Difficulty-aware Advantage Amplification Policy Optimization* (DA3PO), which is implemented with fewer than 30 lines of code from DAPO. Experiments show that DA3PO significantly outperforms GRPO and other classical GRPO variants on various model families and tasks.

1 INTRODUCTION

The release of DeepSeek-R1 (Guo et al., 2025) has drawn significant attention to Reinforcement Learning from Verifiable Rewards (RLVR), establishing Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as a prevailing approach for training Large Language Models (LLMs). Then, to enhance the accuracies across a variety of tasks, various GRPO variants have been proposed, such as Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO) (Yu et al., 2026b) and Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025), which have further extended GRPO. Among them, DAPO is one of the most popular variants (Liu et al., 2026a; Yu et al., 2026a; Yuchen et al., 2026; Zhang et al., 2026; Ma et al., 2026).

Compared to GRPO, DAPO introduces several significant improvements. Among these, Dynamic Sampling contributes the most to DAPO's accuracy enhancements from GRPO, as presented by the experimental results of the original paper of DAPO (Yu et al., 2026b). More importantly, Dynamic Sampling has also been adopted in the training of many frontier models, such as MiMo-V2.6 (Xiaomi MiMo Team, 2026). To improve the accuracy, Dynamic Sampling enhances training stability, a critical factor for achieving accuracy enhancements, by eliminating zero policy gradients caused by zero advantages (Yu et al., 2026b). Specifically, such zero policy gradients come from prompts where sampled responses are entirely correct or incorrect. Therefore, to avoid producing such zero policy gradients, Dynamic Sampling over-samples prompts and filters out prompts where sampled responses are entirely correct or incorrect.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

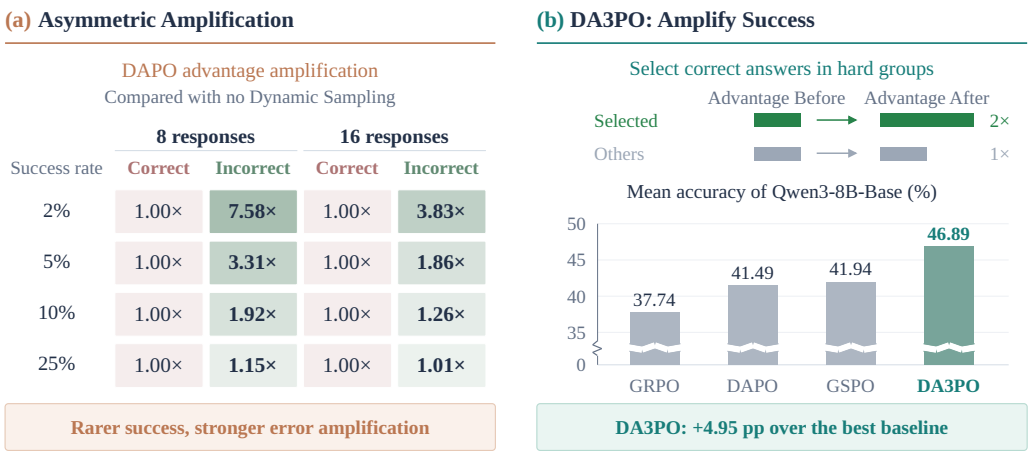


Figure 1: Overview of DA3PO. (a) Dynamic Sampling makes incorrect responses undergo much greater amplification relative to correct ones on hard prompts . (b) DAA doubles advantages of correct responses and DA3PO significantly outperforms other baselines.

Unfortunately, while Dynamic Sampling improves training stability, our theoretical analysis indicates that it will reduce training efficiency, another crucial aspect for attaining accuracy enhancements. Specifically, the decline in training efficiency primarily stems from Dynamic Sampling's inability to effectively capitalize on hard-to-sample correct responses on hard prompts. More precisely, we demonstrate that Dynamic Sampling will asymmetrically amplify the advantages of different responses to the same prompts, with this amplification varying based on prompt difficulty. On hard prompts, incorrect responses undergo much greater amplification relative to correct ones, as shown in Figure 1(a). This asymmetric amplification causes the model to avoid generating observed incorrect responses, rather than capitalizing on the correct ones on hard prompts, thereby reducing training efficiency as the correct responses on hard prompts are exceedingly hard to sample.

To improve training efficiency, we propose *Direct Advantage Amplification* (DAA). Its key insight is to amplify the advantages of hard-to-sample correct responses on hard prompts, as retained by Dynamic Sampling. Formally, after obtaining such advantages via Dynamic Sampling, we multiply these advantages by a positive constant greater than one to amplify them. This ensures that, when Dynamic Sampling is used, these hard-to-sample correct responses on hard prompts can be effectively capitalized on, implying higher training efficiency. Then, by integrating DAA into DAPO, we obtain *Difficulty-aware Advantage Amplification Policy Optimization* (DA3PO). More surprisingly, since the implementation of DAA is straightforward, DA3PO requires fewer than 30 lines of code derived from DAPO (see more details in Section 4).

To evaluate DA3PO, we conduct experiments on mathematical reasoning and code generation benchmarks using different base models of different sizes. Experimental results demonstrate that DA3PO yields consistent accuracy enhancements across all tested models and benchmarks, significantly outperforming GRPO and classical GRPO variants on various model families and tasks.

2 Preliminary

2.1 Group Relative Policy Optimization (GRPO)

GRPO (Shao et al., 2024) computes advantages purely from group-level rewards, making it an effective approach for optimizing a policy π_θ. At each training step, a prompt x is sampled from the dataset $\mathcal{D}$, and the behavior policy $\pi_{\theta_{\text{old}}}$ generates a group of G responses $\{o_1, \ldots, o_G\}$. Here, $o_{i,t}$ denotes the t-th token of response o_i, while $o_{i,<t}$ denotes the sequence of tokens generated before position t. The conditional probability of response o_i under π_θ and the importance sampling ratio at token position t are given by:

$$\pi_\theta(o_i \mid x) = \prod_{t=1}^{|o_i|} \pi_\theta(o_{i,t} \mid x,\, o_{i,<t}), \qquad r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid x,\, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid x,\, o_{i,<t})}.$$

A rule-based verifier assigns each response a correctness reward $R_{\text{acc}}(o_i)$, defined as:

$$R_{\text{acc}}(o_i) = \begin{cases} 1, & \text{if } o_i \text{ is correct,} \\ -1, & \text{otherwise.} \end{cases} \tag{1}$$

Using the correctness rewards $R_{\text{acc}}(o_j)$ from Eq. (1), the advantage for the i-th response at token position t is computed as:

$$\hat{A}_{i,t} = \hat{A}_i := \frac{R_{\text{acc}}(o_i) - \text{mean}(\{R_{\text{acc}}(o_j)\}_{j=1}^{G})}{\text{std}(\{R_{\text{acc}}(o_j)\}_{j=1}^{G})}. \tag{2}$$

Here, $\hat{A}_i$ denotes the response-level advantage shared by every valid token position t in response o_i. The policy is updated by maximizing a clipped objective with per-sample loss averaging:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \mathcal{L}_{i,t}^{\text{GRPO}}(\theta) \right],$$

where the per-token clipped loss uses $\hat{A}_{i,t}$ from Eq. (2):

$$\mathcal{L}_{i,t}^{\text{GRPO}}(\theta) = \min\Big(r_{i,t}(\theta)\, \hat{A}_{i,t},\ \text{clip}(r_{i,t}(\theta),\, 1-\epsilon,\, 1+\epsilon)\, \hat{A}_{i,t} \Big), \tag{3}$$

and ϵ is the symmetric clipping range.

2.2 Decoupled Clip and Dynamic Sampling Policy Optimization (DAPO)

Compared to GRPO, DAPO (Yu et al., 2026b) introduces several improvements to enhance training efficiency and training stability, resulting in higher accuracy than GRPO.

Clip-Higher. Firstly, to mitigate training inefficiency arising from low exploration, DAPO applies Clip-Higher to increase the probability of low-probability exploration tokens. Formally, Clip-Higher decouples the symmetric clipping range in Eq. (3) into asymmetric bounds:

$$\text{clip}(r_{i,t}(\theta),\, 1-\epsilon_l,\, 1+\epsilon_h), \epsilon_h > \epsilon_l.$$

Overlong Reward Shaping. Then, to discourage overly long responses and thereby improve training efficiency, DAPO adds a length penalty to the GRPO correctness reward:

$$R_i = R_{\text{acc}}(o_i) + R_{\text{len}}(o_i),$$

where the correctness reward $R_{\text{acc}}(o_i)$ follows Eq. (1), and the length penalty $R_{\text{len}}(o_i)$ is defined as:

$$R_{\text{len}}(o_i) = \begin{cases} 0, & |o_i| \le L_m - L_c, \\ \dfrac{L_m - L_c - |o_i|}{L_c}, & L_m - L_c < |o_i| \le L_m, \\ -1, & |o_i| > L_m. \end{cases}$$

Here, L_m is the maximum generation length and L_c is the width of the soft penalty interval, with $0 < L_c \le L_m$. Prior studies suggest that overlong reward shaping provides limited performance improvements (Yu et al., 2026b; He et al., 2025), and many subsequent works do not adopt overlong reward shaping (Yang et al., 2025b; He et al., 2025; Xiaomi MiMo Team, 2026). Following this practice, we also omit overlong reward shaping in this paper unless otherwise stated. Our experimental results in Section E further support this choice, showing only minor differences in average performance between DAPO with and without overlong reward shaping.

Token-Level Loss. In addition, to improve training stability, DAPO replaces the per-sample normalizer of GRPO with a single global token-level normalizer in Eq. (4):

$$\mathbb{E}_{x\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \mathcal{L}_{i,t}^{\text{DAPO}}(\theta) \right].$$

This formulation ensures that every token contributes equally to the gradient, thereby preventing an unhealthy increase in entropy (Yu et al., 2026b), which would otherwise degrade training stability.

Dynamic Sampling. Lastly, to address low training stability caused by zero policy gradients, DAPO utilizes Dynamic Sampling. Specifically, when a group of sampled responses $\{o_i\}_{i=1}^{G}$ of a prompt are entirely correct or incorrect, the advantage is zero, which yields zero policy gradients. These zero policy gradients shrink the magnitude and increase the noise sensitivity of the batch gradient, thereby degrading sample utilization (Yu et al., 2026b). To address this problem, Dynamic Sampling ensures that for all training prompts, the sampled responses $\{o_i\}_{i=1}^{G}$ are neither entirely correct nor entirely incorrect.

The response success probability p is the probability that a sampled response o_i to prompt x is correct. The group success rate s_G is the fraction of correct responses among G samples:

$$p := \Pr_{o_i \sim \pi_\theta(\cdot|x)}(R_{\text{acc}}(o_i) = 1) \in (0, 1), \qquad s_G := \frac{1}{G}\sum_{i=1}^{G} \mathbb{I}[R_{\text{acc}}(o_i) = 1],$$

where $\mathbb{I}[\cdot]$ is the indicator function. Dynamic Sampling will over-sample and filter out prompts, leaving all prompts in the batch with $0 < s_G < 1$ and keeping a consistent number of prompts.

Integrating all four improvements, the DAPO objective takes the form:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \mathcal{L}_{i,t}^{\text{DAPO}}(\theta) \right], \quad \text{s.t.} \quad 0 < s_G < 1, \quad (4)$$

where the per-token clipped surrogate loss uses $\hat{A}_{i,t}$ from Eq. (2):

$$\mathcal{L}_{i,t}^{\text{DAPO}}(\theta) = \min\Big(r_{i,t}(\theta)\,\hat{A}_{i,t},\ \text{clip}(r_{i,t}(\theta),\, 1-\epsilon_l,\, 1+\epsilon_h)\,\hat{A}_{i,t} \Big).$$

3 THEORY

As presented in the experimental results of the original DAPO paper, among the four improvements, Dynamic Sampling contributes the most to DAPO's accuracy gain over GRPO (Xiaomi MiMo Team, 2026). However, in this section, we demonstrate that while Dynamic Sampling enhances accuracy by improving training stability, it reduces training efficiency. Specifically, Dynamic Sampling induces an asymmetric amplification of advantages (Theorem 3.1), leading the model to avoid generating observed incorrect responses rather than strengthening the hard-to-sample correct ones on hard prompts (Corollary 3.2), resulting in low training efficiency.

We now characterize the asymmetric amplification induced by Dynamic Sampling. Specifically, the expected advantages of the correct and incorrect responses are multiplied by the coefficients α^+ and α^-, respectively.

Theorem 3.1 (Proof is in Appendix B)**.** *For any prompt x with response success probability $p \in (0, 1)$ and $G \geq 2$ independently sampled responses, conditioning on a group success rate $0 < s_G < 1$ gives:*

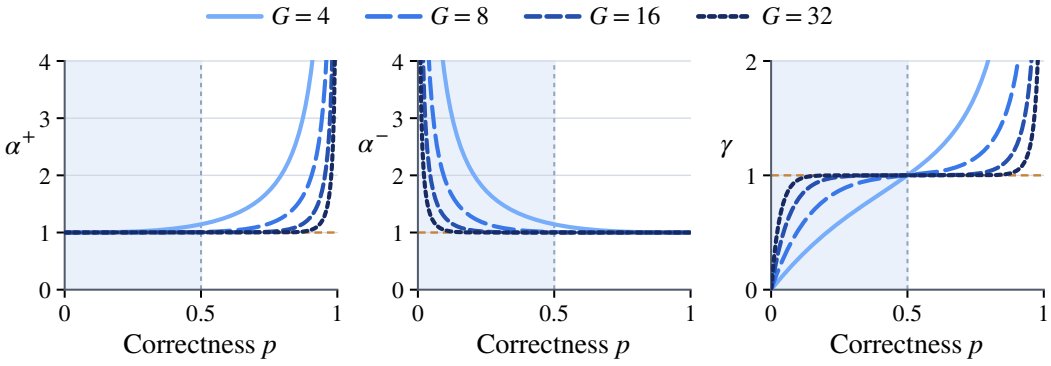
$$\mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\, R_{\text{acc}}(o_i){=}1] = \alpha^+ \cdot \mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i){=}1], \quad (5)$$

$$\mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\, R_{\text{acc}}(o_i){=}{-}1] = \alpha^- \cdot \mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i){=}{-}1]. \quad (6)$$

where the amplification factors are

$$\alpha^+ := \frac{1}{1-p^{G-1}}, \qquad \alpha^- := \frac{1}{1-(1-p)^{G-1}}.$$

Remark. The factors α^+ and α^- amplify the conditional expected advantage for correct and incorrect responses, respectively. The amplification factors reveal a striking asymmetry as prompt difficulty varies. As shown in Figure 2(a) and Figure 2(b), α^+ remains close to 1 throughout the hard-prompt regime ($p \ll 0.5$) while α^- grows large precisely there, indicating that Dynamic Sampling disproportionately amplifies the penalty signal for incorrect responses.



(a) Correct responses: α^+. (b) Incorrect responses: α^-. (c) Amplification ratio γ.

Figure 2: Asymmetric amplification under Dynamic Sampling for $G \in \{4, 8, 16, 32\}$ (Theorem 3.1). Blue shading marks hard prompts ($p < 0.5$), where $\gamma < 1$ (Corollary 3.2); dashed amber lines mark unity. Curves extending beyond the displayed vertical range are clipped.

Corollary 3.2. *Let* $\gamma(p, G) := \alpha^+/\alpha^-$. *Then:*

$$\gamma(p, G) < 1, \quad \textit{if } p < 0.5;$$
$$\gamma(p, G) > 1, \quad \textit{if } p > 0.5;$$
$$\gamma(p, G) = 1, \quad \textit{if and only if } p = 0.5.$$

This corollary follows directly from Theorem 3.1: for $p < 0.5$ we have $p < 1 - p$, so $p^{G-1} < (1-p)^{G-1}$, giving $\alpha^+ < \alpha^-$ and hence $\gamma < 1$. Symmetrically, for $p > 0.5$ we have $p > 1 - p$, giving $\alpha^+ > \alpha^-$ and hence $\gamma > 1$. The equality $\gamma = 1$ holds if and only if $p = 1 - p = 0.5$.

Remark. Figure 2c confirms Corollary 3.2: the ratio $\gamma = \alpha^+/\alpha^-$ falls below 1 for all $p < 0.5$ and all group sizes. Specifically, for $G = 8$ and $p = 0.1$: $\alpha^+ \approx 1.00$, $\alpha^- \approx 1.92$, $\gamma \approx 0.52$. For $G = 16$ and $p = 0.1$: $\alpha^+ \approx 1.00$, $\alpha^- \approx 1.26$, $\gamma \approx 0.79$. In both cases the advantages on incorrect responses is amplified substantially more than correct ones, and a smaller group size intensifies this asymmetry.

Corollary 3.2 reveals that for hard prompts ($p < 0.5$), the ratio $\gamma = \alpha^+/\alpha^-$ is strictly less than 1. As $p \to 0$, this asymmetry becomes extreme: α^- grows unboundedly, while α^+ remains close to 1, implying Dynamic Sampling asymmetrically amplifies the advantage of incorrect responses more than that of correct ones. Consequently, rather than capitalizing on the hard-to-sample correct responses on hard prompts, the model learns to avoid generating the observed incorrect responses, without strengthening the correct responses, thereby reducing training efficiency.

4 Algorithm

As mentioned in Section 3, Dynamic Sampling induces an asymmetric amplification. It causes the model to learn to avoid generating observed incorrect responses, rather than to capitalize on the hard-to-sample correct ones on hard prompts, which reduces training efficiency. To address this issue, we propose Direct Advantage Amplification (DAA). Its key insight is to amplify the advantage of hard-to-sample correct responses on hard prompts retained by Dynamic Sampling by a positive constant. This allows such correct responses to be capitalized on more effectively, leading to higher training efficiency. Then, by integrating DAA into DAPO, we obtain *Difficulty-aware Advantage Amplification Policy Optimization* (DA3PO). Algorithm 1 summarises the full DA3PO training procedure. More surprisingly, since the implementation of DAA is straightforward, DA3PO requires fewer than 30 lines of code derived from DAPO.

4.1 Details

From Theorem 3.1, Dynamic Sampling applies asymmetric amplification factors α^+ and α^- to the advantage of correct and incorrect responses, respectively. As shown in the subsequent remark of Theorem 3.1, α^+ remains close to 1 on very hard prompts ($p \ll 0.5$) while α^- is much larger than

Algorithm 1 Difficulty-aware Advantage Amplification Policy Optimization

Require: Policy π_θ; prompt distribution $\mathcal{D}$; prompt batch size B; group size G; clipping bounds ϵ_l, ϵ_h; amplification factor $\lambda > 1$; difficulty threshold $\tau \in (0, 1)$

for each training step **do**
  $\theta_{\text{old}} \leftarrow \theta$; initialize an empty batch $\mathcal{B}$
  while $|\mathcal{B}| < B$ **do** *(Dynamic Sampling)*
    Sample $x \sim \mathcal{D}$ and G responses $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$
    Compute accuracy rewards $\{R_{\text{acc}}(o_i)\}_{i=1}^{G}$ and success rate $s_G \leftarrow \frac{1}{G}\sum_{i=1}^{G} \mathbb{I}[R_{\text{acc}}(o_i) > 0]$
    if $0 < s_G < 1$ **then**
      Compute advantages $\hat{A}_{i,t}$ via Eq. (2)
      Initialize $\tilde{A}_{i,t} \leftarrow \hat{A}_{i,t}$ for all response tokens
      if $s_G < \tau$ **then**
        $\tilde{A}_{i,t} \leftarrow \lambda \hat{A}_{i,t}$ for all i with $R_{\text{acc}}(o_i) > 0$ and all valid t ▷ *DAA*
      end if
      Append $(x, \{o_i\}_{i=1}^{G}, \{\tilde{A}_{i,t}\}_{i,t})$ to $\mathcal{B}$
    end if
  end while
  Update θ by maximizing $\mathcal{J}_{\text{DA3PO}}(\theta)$ over $\mathcal{B}$ using Eq. (8)
end for

α^+. The asymmetric amplification prevents the model from effectively capitalizing on hard-to-sample correct responses, ultimately reducing training efficiency on hard prompts. without strengthening the correct responses, thereby reducing training efficiency.

To address this problem, we can scale the advantage of hard-to-sample correct responses on hard prompts by a factor of $1/\gamma$. However, p varies as the policy updates during training, making the true value of γ almost impossible to track, so we use a substituting way to improve the training efficiency. We propose Direct Advantage Amplification (DAA), which adopts a constant $\lambda > 1$ to amplify the advantage of hard-to-sample correct responses on hard prompts retained by Dynamic Sampling. This allows hard-to-sample correct responses on hard prompts to be effectively capitalized on, improving training efficiency without requiring estimation of p. The advantage $\hat{A}_{i,t}$ is modified as:

$$\tilde{A}_{i,t} = \begin{cases} \lambda \cdot \hat{A}_{i,t}, & \text{if } R_{\text{acc}}(o_i) > 0 \text{ and } s_G < \tau, \\ \hat{A}_{i,t}, & \text{otherwise}, \end{cases} \tag{7}$$

where $\lambda > 1$ is a constant amplification factor and $\tau \in (0, 1)$ a difficulty threshold. DAA is activated only when the response is correct ($R_{\text{acc}}(o_i) > 0$) and the prompt is hard ($s_G < \tau$). Following Corollary 3.2, we set $\tau = 0.5$, since the amplification ratio $\gamma = \alpha^+/\alpha^-$ is less than 1 when $p < 0.5$. By integrating DAA into the DAPO objective, we get DA3PO, whose objective is formulated as:

$$\mathcal{J}_{\text{DA3PO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \mathcal{L}_{i,t}^{\text{DA3PO}}(\theta) \right], \quad \text{s.t.} \quad 0 < s_G < 1, \tag{8}$$

where $\mathcal{L}_{i,t}^{\text{DA3PO}}(\theta)$ denotes the per-token loss computed with the DAA advantage $\tilde{A}_{i,t}$ from Eq. (7):

$$\mathcal{L}_{i,t}^{\text{DA3PO}}(\theta) = \min\Big(r_{i,t}(\theta)\, \tilde{A}_{i,t},\ \text{clip}\big(r_{i,t}(\theta),\, 1-\epsilon_l,\, 1+\epsilon_h\big)\, \tilde{A}_{i,t} \Big).$$

4.2 Discussion

To improve the training efficiency of Dynamic Sampling, our DAA amplifies the advantage of hard-to-sample correct responses on hard prompts, as retained by Dynamic Sampling. In addition, since the implementation of DAA is straightforward, DA3PO requires fewer than 30 lines of code derived from DAPO. Moreover, although we only apply DAA to DAPO, DAA can be applied to GRPO and its variants. This is because we only modify Dynamic Sampling, ensuring compatibility with any other enhancements made to GRPO and its variants. Lastly, our DAA still has significant room for improvement. For example, the constant λ and τ used in Eq. (7) can be replaced by adaptive factors. We leave such extensions to future work.

Model	Algorithm	AIME24 avg@32	AIME25 avg@32	AIME26 avg@32	AMC avg@32	Minerva avg@8	Olympiad avg@8	MATH avg@8	AVG
Qwen3-4B-Base	Base Model	5.21	3.23	3.65	20.18	12.22	19.56	41.23	15.04
	GRPO	15.31	12.81	11.56	55.50	23.99	39.35	77.22	33.68
	DAPO	18.96	15.10	12.71	52.07	24.17	40.65	78.20	34.55
	GSPO	18.85	13.12	14.27	51.51	22.61	41.96	79.87	34.60
	DA3PO	**23.96**	**19.17**	**17.19**	**60.50**	**26.84**	**44.52**	**85.20**	**39.63**
	vs. GRPO	↑8.65%	↑6.36%	↑5.63%	↑5.00%	↑2.85%	↑5.17%	↑7.98%	↑5.95%
	vs. DAPO	↑5.00%	↑4.07%	↑4.48%	↑8.43%	↑2.67%	↑3.87%	↑7.00%	↑5.08%
	vs. GSPO	↑5.11%	↑6.05%	↑2.92%	↑8.99%	↑4.23%	↑2.56%	↑5.33%	↑5.03%
Qwen3-8B-Base	Base Model	5.52	5.73	4.69	26.13	14.71	26.78	59.62	20.45
	GRPO	18.54	17.29	16.77	61.11	24.77	42.74	82.95	37.74
	DAPO	24.58	20.52	17.19	64.83	29.50	48.30	85.52	41.49
	GSPO	24.27	21.56	17.92	65.44	30.06	48.83	85.52	41.94
	DA3PO	**30.52**	**23.23**	**24.17**	**69.69**	**37.91**	**54.26**	**88.45**	**46.89**
	vs. GRPO	↑11.98%	↑5.94%	↑7.40%	↑8.58%	↑13.14%	↑11.52%	↑5.50%	↑9.15%
	vs. DAPO	↑5.94%	↑2.71%	↑6.98%	↑4.86%	↑8.41%	↑5.96%	↑2.93%	↑5.40%
	vs. GSPO	↑6.25%	↑1.67%	↑6.25%	↑4.25%	↑7.85%	↑5.43%	↑2.93%	↑4.95%

Table 1: Avg@k (%) of all compared algorithms across seven mathematical reasoning benchmarks, with absolute improvement of DA3PO over each baseline. We report avg@32 for AIME and AMC, and avg@8 for Minerva, Olympiad and MATH. Best results within each benchmark are **bolded**.

5 Experiments

5.1 Experimental Setup

Datasets and Benchmarks. We evaluate DA3PO across model scales, task domains, and model families through three groups of experiments. For the main mathematical reasoning experiments, we train Qwen3-4/8B-Base on DAPO-Math-17K following Yu et al. (2026b). We evaluate the trained models on seven benchmarks: AIME24/25/26, AMC, Minerva, OlympiadBench (He et al., 2024), and MATH (Hendrycks et al., 2021). To assess generalization to code generation, we train Qwen3-8B-Base on the full code subset of Eurus-2-RL-Data and evaluate it on HumanEval+, MBPP+, and LiveCodeBench v6. To assess generalization to different model, we train DeepSeek-R1-Distill-Qwen-1.5B on DeepScaleR and evaluate it on the same seven mathematical benchmarks.

Baselines and Parameters. We compare DA3PO with GRPO, DAPO, GSPO. All algorithms train for 200 steps. We fix $\lambda = 2$ and $\tau = 0.5$ for DA3PO unless otherwise stated. Detailed training and evaluation configurations are provided in Appendices C and D, respectively.

Metrics. We report avg@k and pass@k. We use $k = 32$ for AIME and AMC, $k = 8$ for Minerva, OlympiadBench, and MATH, and $k = 4$ for code generation. For each algorithm, we report results from its best-performing checkpoint.

5.2 Main Results

Finding 1. DA3PO achieves the highest average accuracy on mathematical task.

To verify the effectiveness of our algorithm, we evaluate the avg@k of all algorithms on the seven mathematical benchmarks for Qwen3-4B-Base and Qwen3-8B-Base. The detailed experimental results are illustrated in Table 1. Firstly, DA3PO outperforms all other GRPO variants by at least 5.03% in average accuracy on Qwen3-4B-Base and by at least 4.95% on Qwen3-8B-Base. Secondly, DA3PO achieves the best performance on the hardest benchmarks. Specifically, DA3PO outperforms all baseline algorithm by up to 5.00% and 6.25% on AIME24 benchmark compared to the best baseline for Qwen3-4B-Base and Qwen3-8B-Base, respectively. Lastly, on easy benchmarks such as

Model	Algorithm	AIME24 pass@32	AIME25 pass@32	AIME26 pass@32	AMC pass@32	Minerva pass@8	Olympiad pass@8	MATH pass@8	AVG
Qwen3-4B-Base	Base Model	30.00	23.33	16.67	78.31	28.31	42.67	77.60	42.41
	GRPO	36.67	**46.67**	30.00	86.75	36.03	56.00	90.20	54.62
	DAPO	43.33	43.33	30.00	84.34	39.71	59.56	91.00	55.90
	GSPO	50.00	40.00	30.00	**91.57**	35.66	57.33	89.20	56.25
	DA3PO	**60.00**	**46.67**	**40.00**	**91.57**	**41.18**	**61.19**	**93.80**	**62.06**
	vs. GRPO	↑23.33%	0.00	↑10.00%	↑4.82%	↑5.15%	↑5.19%	↑3.60%	↑7.44%
	vs. DAPO	↑16.67%	↑3.34%	↑10.00%	↑7.23%	↑1.47%	↑1.63%	↑2.80%	↑6.16%
	vs. GSPO	↑10.00%	↑6.67%	↑10.00%	0.00	↑5.52%	↑3.86%	↑4.60%	↑5.81%
Qwen3-8B-Base	Base Model	23.33	33.33	23.33	83.13	31.62	46.81	85.80	46.76
	GRPO	43.33	40.00	43.33	90.36	38.24	61.63	93.20	58.58
	DAPO	56.67	46.67	46.67	90.36	43.01	63.70	93.40	62.93
	GSPO	50.00	40.00	46.67	89.16	45.59	**71.41**	**95.80**	62.66
	DA3PO	**70.00**	**50.00**	**53.33**	**92.77**	**49.26**	**71.41**	94.80	**68.80**
	vs. GRPO	↑26.67%	↑10.00%	↑10.00%	↑2.41%	↑11.02%	↑9.78%	↑1.60%	↑10.22%
	vs. DAPO	↑13.33%	↑3.33%	↑6.66%	↑2.41%	↑6.25%	↑7.71%	↑1.40%	↑5.87%
	vs. GSPO	↑20.00%	↑10.00%	↑6.66%	↑3.61%	↑3.67%	0.00	↓1.00%	↑6.14%

Table 2: Pass@k (%) of all compared algorithms across seven mathematical reasoning benchmarks, with absolute improvement of DA3PO over each baseline. We report pass@32 for AIME and AMC, and pass@8 for Minerva, Olympiad and MATH. Best results within each benchmark are **bolded**.

AMC, Olympiad and MATH, DA3PO also outperforms baselines across both model scales. These results indicates that DA3PO not only improves performance on hard prompts but also improves performance on easy ones.

Finding 2. DA3PO broadens the coverage of correct responses rather than collapsing onto a single solution mode.

To verify that DA3PO does not merely raise average accuracy by collapsing onto a single solution mode, we report the pass@k of all algorithms, as shown in Table 2. Firstly, DA3PO outperforms baselines by at least 5.81% in average pass@k on Qwen3-4B-Base and by at least 5.87% on Qwen3-8B-Base. Then, DA3PO exceeds all baselines on five of seven benchmarks on both model scales. Besides, on the easiest dataset MATH DA3PO trails by only 1.00% on Qwen3-8B-Base, as baselines even base model already achieve pass@k above 90% on MATH, leaving limited room for further performance gains. These results confirm that DA3PO broadens the space of correct responses across the evaluated benchmarks, rather than collapsing onto a single solution mode.

Finding 3. DA3PO consistently outperforms the other methods throughout RL training.

To verify that DA3PO brings stable performance improvement throughout the training, we plot the avg@32 on AIME24 over training on Qwen3-4B-Base and Qwen3-8B-Base, respectively. The experimental results are shown in Figure 3. On both model scales, the avg@32 on AIME24 of DA3PO curve sits visibly above those of GRPO, DAPO, and GSPO throughout RL training, showing that DA3PO consistently outperforms the other methods throughout RL training rather than appearing only at a selected checkpoint.

Finding 4. DA3PO demonstrates strong generalization on other tasks.

To evaluate whether DA3PO extends beyond mathematical reasoning, we conduct code generation experiments on Qwen3-8B-Base. Results show that DA3PO ranks first on all three benchmarks. Its average avg@4 reaches 61.39%, outperforming the strongest baseline by 2.00 percentage points. DA3PO also achieves the highest pass@4 on every benchmark. These consistent gains in both accuracy and solution coverage demonstrate DA3PO's generalization to other tasks. Details are shown in Appendix F.1.

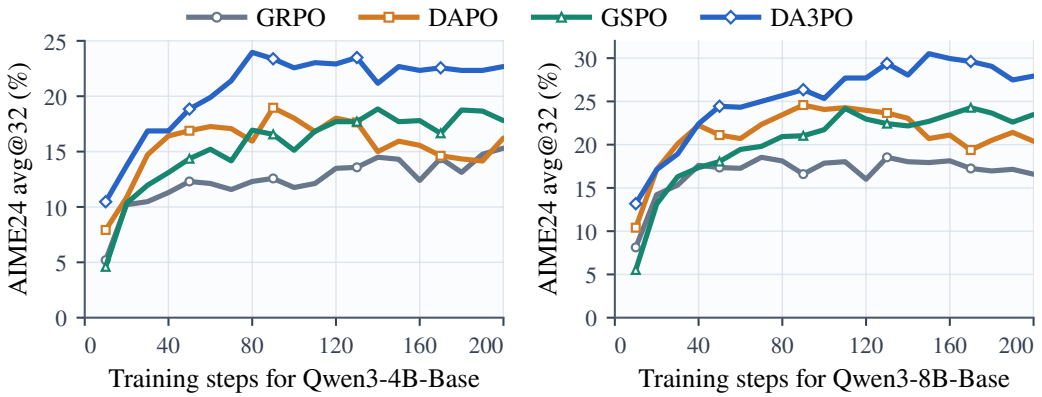


Figure 3: AIME24 avg@32 over the course of RL training. Due to computational constraints, AIME24 is used as the validation benchmark.

Finding 5. DA3PO demonstrates strong generalization on other model families.

To examine whether DA3PO generalizes beyond the Qwen3 model family, we evaluate DeepSeek-R1-Distill-Qwen-1.5B on seven mathematical reasoning benchmarks using the same reinforcement learning algorithm. DA3PO achieves the highest average avg@k and pass@k on seven benchmarks, reaching a mean of 52.09% avg@k and exceeding the strongest baseline GSPO by 2.48 percentage points. These results demonstrate that DA3PO also generalizes to other model families. Details are provided in Appendix F.2.

Finding 6. DA3PO is robust to the amplification factor.

We examine DA3PO's sensitivity to the amplification factor and its effectiveness under comparable resource budgets. Across $\lambda \in \{1.5, 2.0, 2.5\}$, DA3PO consistently improves mean accuracy over DAPO on both model scales, with $\lambda = 2$ performing best. These observations support moderate amplification and our default choice of $\lambda = 2$. Details are provided in Appendix F.3.

Finding 7. DA3PO retains its improvements under comparable sampling and training budgets.

Dynamic Sampling requires extra generations because some sampled responses are discarded. For a fair efficiency comparison, we therefore evaluate validation accuracy under similar generated responses and similar training-time budgets. DA3PO remains more accurate on both model scales. Moreover, even when GRPO and GSPO are given larger budgets than DA3PO, they do not reach its validation accuracy within our runs. Thus, DA3PO's gains do not depend on additional sampling or training time. Details are provided in Appendix F.4.

5.3 Ablation Study

To isolate the contribution of the difficulty threshold, we compare DA3PO with a variant that removes the threshold and amplifies every correct response retained by Dynamic Sampling. Restricting amplification to hard prompts further increases mean avg@k and mean pass@k on the 4B and 8B models. Thus, amplifying correct responses only for hard prompts works better than amplifying every correct response, which validates the effectiveness of DAA. Details are provided in Appendix G.

6 Conclusion

In this paper, we show that Dynamic Sampling induces asymmetrical amplification, which reduces training efficiency in DAPO. To address this issue, we proposed Direct Advantage Amplification (DAA). Its key insight is to amplify the advantage of correct responses on hard prompts so that the hard-to-sample correct responses on hard prompts retained by Dynamic Sampling can be effectively capitalized on. By integrating DAA into DAPO, we obtain Difficulty-aware Advantage Amplification Policy Optimization (DA3PO). Experiments demonstrate that DA3PO significantly outperforms GRPO and other classical GRPO variants.

AI USE STATEMENT

We used generative AI tools to assist with language editing and improve the readability of the manuscript. These tools were not used to formulate research hypotheses, design the methodology or experiments, derive theoretical results, interpret experimental results, or generate scientific claims. All AI-assisted edits were reviewed and verified by the authors. The authors take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

The DA3PO training procedure and objective are specified in Section 4. Section 5.1 describes the models, datasets, benchmarks, baselines, and evaluation metrics. The training and evaluation configurations, including sampling settings and seeds, are detailed in Appendices C and D.

REFERENCES

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://ojs.aaai.org/index.php/AAAI/article/view/40290.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=inccdtfx8x.

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, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, September 2025. ISSN 1476-4687. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific

problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the Sixty-second Annual Meeting of the Association for Computational Linguistics*, August 2024. URL https://aclanthology.org/2024.acl-long.211/.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report, 2025. URL https://arxiv.org/abs/2505.22312.

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 *Proceedings of the Thirty-fifth Conference on Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=7Bywt2mQsCe.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. Reinforce++: Stabilizing critic-free policy optimization with global advantage normalization, 2025. URL https://arxiv.org/abs/2501.03262.

Mengqi Liao, Xiangyu Xi, Chen Ruinian, Jia Leng, Yangen Hu, Ke Zeng, Shuai Liu, and Huaiyu Wan. Enhancing efficiency and exploration in reinforcement learning for LLMs. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, November 2025. URL https://aclanthology.org/2025.emnlp-main.75/.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. ProRL: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026a. URL https://openreview.net/forum?id=YPsJha5HXQ.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. Gdpo: Group reward-decoupled normalization policy optimization for multi-reward rl optimization, 2026b. URL https://arxiv.org/abs/2601.05242.

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. In *Proceedings of the Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Chiyu Ma, Shuo Yang, Kexin Huang, Jinda Lu, Haoming Meng, Shangshang Wang, Bolin Ding, Soroush Vosoughi, Guoyin Wang, and Jingren Zhou. Fipo: Eliciting deep reasoning with future-kl influenced policy optimization, 2026. URL https://arxiv.org/abs/2603.19835.

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

Abdullah Vanlioglu. Entropy-guided sequence weighting for efficient exploration in rl-based llm fine-tuning, 2025. URL https://arxiv.org/abs/2503.22456.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=yfcpdY4gMP.

Xiaomi MiMo Team. MiMo-V2.6 technical report, 2026. URL https://huggingface.co/XiaomiMiMo/MiMo-V2.6-Pro-RL/blob/main/MiMo_V2_6_technical_report.pdf.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. Dcpo: Dynamic clipping policy optimization, 2025a. URL https://arxiv.org/abs/2509.02333.

Ziyi Yang, Weizhou Shen, Chenliang Li, Ruijun Chen, Fanqi Wan, Ming Yan, Xiaojun Quan, and Fei Huang. SPELL: Self-play reinforcement learning for evolving long-context language models, 2025b. URL https://arxiv.org/abs/2509.23863.

Linhao Yu, Tianmeng Yang, Siyu Ding, Renren Jin, Naibin Gu, Xiangzhao Hao, Shuaiyi Nie, Deyi Xiong, Weichong Yin, Yu Sun, and Hua Wu. Knowrl: Boosting llm reasoning via reinforcement learning with minimal-sufficient knowledge guidance, 2026a. URL https://arxiv.org/abs/2604.12627.

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, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Proceedings of the Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026b. URL https://openreview.net/forum?id=2a36EMSSTp.

Cai Yuchen, Ding Cao, Xin Xu, Zijun Yao, Yuqing Huang, Benyi Zhang, Zhenyu Tan, Guiquan Liu, and Junfeng Fang. On predictability of reinforcement learning dynamics for large language models. In *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=SdHmA6BYVJ.

Wenchuan Zhang, Penghao Zhang, Jingru Guo, Tao Cheng, Jie Chen, Shuwan Zhang, Zhang Zhang, Yuhao Yi, and Hong Bu. Patho-r1: A multimodal reinforcement learning-based pathology expert reasoner. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Proceedings of the Fortieth AAAI Conference on Artificial Intelligence*, 2026. URL https://doi.org/10.1609/aaai.v40i33.40071.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. Geometric-mean policy optimization. In *Proceedings of the Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=nCEs0tSwc2.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

A Related Work

As the original group relative reinforcement learning algorithm, GRPO (Shao et al., 2024) has attracted considerable research attention. However, instabilities in GRPO accumulate over long-term training, preventing the policy from consistently reinforcing correct responses (Liu et al., 2025; Chu et al., 2026; Yu et al., 2026b). To address this, several works (Liu et al., 2025; Chu et al., 2026; Hu et al., 2025; Zheng et al., 2025; Yu et al., 2026b; Yang et al., 2025a; Zhao et al., 2026; Liu et al., 2026b; Ma et al., 2026; Yuchen et al., 2026) focus on improving the objective function itself to enhance training stability. Among them, DAPO (Yu et al., 2026b) is one of the most representative works. DAPO introduces four improvements over GRPO, and Dynamic Sampling contributes the most to its accuracy gain over GRPO (Yu et al., 2026b), which has also been adopted in frontier models such as MiMo-V2.6 (Xiaomi MiMo Team, 2026). Specifically, when all responses sampled for a prompt are correct, or when all are incorrect, their advantages are zero, yielding zero policy gradients. These zero gradients shrink the magnitude of the batch gradient and increase its sensitivity to noise, thereby degrading sample utilization. DAPO's Dynamic Sampling addresses this issue by oversampling prompts and retaining only groups that contain both correct and incorrect responses.

Building upon these stability improvements, several works further aim to improve training efficiency in LLMs reinforcement learning. These works (Cheng et al., 2026; Vanlioglu, 2025; Wang et al., 2026; Liao et al., 2025) design algorithms to explore GRPO that regulate entropy dynamics, operating on the clipping, token masking, temperature scheduling, advantage reweighting, etc. However, these works largely overlook the impact of Dynamic Sampling on training efficiency. Most of these approaches determine how strongly to update the policy after responses have been sampled. Dynamic Sampling acts at an earlier stage, it determines whether a sampled group enters the batch at all.

In this work, we reveal that Dynamic Sampling, while enhancing accuracy through improved training stability, actually reduces training efficiency. Building on this analysis, we propose DA3PO, which improves the training efficiency of Dynamic Sampling while preserving the original training stability benefits and achieves significant improvements in accuracy across both model scales and all benchmarks. Our analysis shows that, on hard prompts, Dynamic Sampling amplifies the negative advantages of incorrect responses more than the positive advantages of correct ones. Our DA3PO multiplies the advantages of hard-to-sample correct responses on hard prompts retained by Dynamic Sampling by a positive constant greater than one. This allows the model to capitalize on these rare correct responses more effectively, thereby improving training efficiency. Besides, DA3PO requires only fewer than 30 lines of code derived from DAPO. Moreover, the same advantage amplification can be applied to GRPO and its variants while remaining compatible with their other improvements.

B Proof of Theorem 3.1

B.1 Proof of Eq. (5)

Proof. Since $R_{\text{acc}}(o_i)=1$ guarantees that at least one of the G responses is correct, the event $s_G = 0$ (all responses incorrect) is impossible. Therefore, the sample space conditioned on $R_{\text{acc}}(o_i)=1$ is partitioned into two mutually exclusive cases: (i) $s_G = 1$, where all G responses are correct, and (ii) $0 < s_G < 1$, where at least one response is incorrect.

We first compute $P(s_G = 1 \mid R_{\text{acc}}(o_i)=1)$. Since $R_{\text{acc}}(o_i)=1$ already fixes one response as correct, by the independence of responses, the remaining $G-1$ responses must each be independently correct. It follows that

$$P(s_G = 1 \mid R_{\text{acc}}(o_i)=1) = p^{G-1}, \tag{9}$$

which implies

$$P(0 < s_G < 1 \mid R_{\text{acc}}(o_i)=1) = 1 - p^{G-1}. \tag{10}$$

When $s_G = 1$, all G responses are correct and hence all correctness rewards are identical. As a result, z-score normalisation yields zero variance, and we obtain

$$\mathbb{E}[\hat{A}_i \mid s_G = 1,\ R_{\text{acc}}(o_i)=1] = 0. \tag{11}$$

By the law of total expectation over the partition established above, we have

$$\begin{aligned}\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i)=1] = {} & P(s_G = 1 \mid R_{\text{acc}}(o_i)=1) \cdot \mathbb{E}[\hat{A}_i \mid s_G = 1,\ R_{\text{acc}}(o_i)=1] \\ & + P(0 < s_G < 1 \mid R_{\text{acc}}(o_i)=1) \cdot \mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\ R_{\text{acc}}(o_i)=1].\end{aligned} \tag{12}$$

By substituting Eq. (9), (10), and (11) into Eq. (12), we get

$$\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i)=1] = (1 - p^{G-1}) \cdot \mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\ R_{\text{acc}}(o_i)=1]. \tag{13}$$

Since $1 - p^{G-1} > 0$ for $p \in (0, 1)$, rearranging Eq. (13) yields

$$\begin{aligned}\mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\ R_{\text{acc}}(o_i)=1] &= \frac{\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i)=1]}{1 - p^{G-1}} \\ &= \alpha^{+} \cdot \mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i)=1],\end{aligned}$$

which is Eq. (5). We complete the proof. □

B.2 Proof of Eq. (6)

Proof. Since $R_{\text{acc}}(o_i)=-1$ guarantees that at least one of the G responses is incorrect, the event $s_G = 1$ (all responses correct) is impossible. Therefore, the sample space conditioned on $R_{\text{acc}}(o_i)=-1$ is partitioned into two mutually exclusive cases: (i) $s_G = 0$, where all G responses are incorrect, and (ii) $0 < s_G < 1$, where at least one response is correct.

We first compute $P(s_G = 0 \mid R_{\text{acc}}(o_i)=-1)$. Since $R_{\text{acc}}(o_i)=-1$ already fixes one response as incorrect, by the independence of responses, the remaining $G-1$ responses must each be independently incorrect. It follows that

$$P(s_G = 0 \mid R_{\text{acc}}(o_i)=-1) = (1 - p)^{G-1}, \tag{14}$$

which implies

$$P(0 < s_G < 1 \mid R_{\text{acc}}(o_i)=-1) = 1 - (1 - p)^{G-1}. \tag{15}$$

When $s_G = 0$, all G responses are incorrect and hence all correctness rewards are identical. As a result, z-score normalisation yields zero variance, and we obtain

$$\mathbb{E}[\hat{A}_i \mid s_G = 0,\ R_{\text{acc}}(o_i)=-1] = 0. \tag{16}$$

By the law of total expectation over the partition established above, we have

$$\begin{aligned}\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i)=-1] = {} & P(s_G = 0 \mid R_{\text{acc}}(o_i)=-1) \cdot \mathbb{E}[\hat{A}_i \mid s_G = 0,\ R_{\text{acc}}(o_i)=-1] \\ & + P(0 < s_G < 1 \mid R_{\text{acc}}(o_i)=-1) \cdot \mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\ R_{\text{acc}}(o_i)=-1].\end{aligned} \tag{17}$$

By substituting Eq. (14), (15), and (16) into Eq. (17), we get

$$\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i){=}{-}1] = (1-(1-p)^{G-1}) \cdot \mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\, R_{\text{acc}}(o_i){=}{-}1]. \tag{18}$$

Since $1-(1-p)^{G-1} > 0$ for $p \in (0,1)$, rearranging Eq. (18) yields

$$\begin{aligned}\mathbb{E}[\hat{A}_i \mid 0 < s_G < 1,\, R_{\text{acc}}(o_i){=}{-}1] &= \frac{\mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i){=}{-}1]}{1-(1-p)^{G-1}} \\ &= \alpha^{-} \cdot \mathbb{E}[\hat{A}_i \mid R_{\text{acc}}(o_i){=}{-}1],\end{aligned}$$

which is Eq. (6). We complete the proof. □

C TRAINING SETTINGS

Tables 3 and 4 summarize the training and evaluation configurations, respectively. Qwen3-4B-Base and Qwen3-8B-Base use identical training settings in the main mathematical reasoning experiments. Differences for code generation and DeepSeek-R1-Distill-Qwen-1.5B are included in the same tables. We focus on experiment-specific settings and omit routine framework defaults. All lengths are measured in tokens; 1K denotes 1,024 tokens.

Setting	**Main experiments**	**Code / DeepSeek differences**
Model	Qwen3-4B-Base; Qwen3-8B-Base	Code: Qwen3-8B-Base. DeepSeek: DeepSeek-R1-Distill-Qwen-1.5B.
Training dataset	DAPO-Math-17K	Code: Eurus-2-RL-Data, full code subset (25,276 problems). DeepSeek: DeepScaleR (approximately 40K problems).
Validation dataset	AIME 2024	Code: LiveCodeBench v6, with each problem repeated four times.
Maximum prompt length	2,048	Same for both.
Maximum training response length	4,096	Code: 8,192. DeepSeek: 16,384.
Training context budget (prompt + response)	6,144	Code: 10,240. DeepSeek: 18,432.
Validation response limit	8,192	Code: 16,384. DeepSeek: 32,768.
Training batch size	512 prompts	Same for both.
Generation batch size	512×3 prompts	Same for both.
Rollout group size G	16 responses per prompt	Same for both.
PPO mini-batch size	32 prompts	Same for both.
Learning-rate warmup	10 steps	Same for both.
Weight decay	0.1	Same for both.
Advantage estimator	GRPO	Same for both.
Upper clipping parameter	$\epsilon_{\text{high}} = 0.28$	Same for both.
Dual-clip parameter	10.0	Same for both.
Dynamic sampling	Accuracy-based group filtering; at most 10 generation batches	Same for both.
Advantage amplification	Enabled (DA3PO)	Same for both.
Training budget	200 steps	Code: 200 steps. DeepSeek: 200 steps.
Validation interval	Every 10 steps	Same for both.

Table 3: Training settings shared by the two main model scales, with task- and model-specific differences. Unchanged entries in the last column inherit the main setting. The learning rate is 10^{-6}, and DA3PO uses $\lambda = 2$ and $\tau = 0.5$, as specified in Section 5.1. All experiments use a training budget of 200 steps, as specified in Sections 5.1 and F.

D EVALUATION SETTINGS

Setting	Main experiments	Code / DeepSeek differences
Evaluation benchmarks	AIME 2024/2025/2026, AMC, Minerva, OlympiadBench, MATH	Code: HumanEval+, MBPP+, LiveCodeBench v6. DeepSeek: the same seven mathematics benchmarks.
Mathematical answer checking	Math-Verify	DeepSeek: same.
Code evaluation implementation	—	HumanEval+ and MBPP+: EvalPlus. LiveCodeBench v6: the official LiveCodeBench code-generation evaluator. We use locally integrated copies of these implementations.
Maximum prompt length	2,048	DeepSeek: 2,048.
Maximum response length	4,096	Code: 16,384. DeepSeek: 32,768.
Mathematics context window (prompt + response)	6,144	DeepSeek: 34,816.
Sampling configuration	Temperature 1.0; top-p 0.95	Same for code and DeepSeek.
Samples per problem	32 for AIME and AMC; 8 for Minerva, OlympiadBench, and MATH	Code: 4. DeepSeek: the same benchmark-specific counts as the main experiments.
Sampling seeds	Seed $42 + i$ for sample i, $i = 0, \ldots, k - 1$; e.g., 42–49 for 8 samples	Same rule for code and DeepSeek; increment the seed by 1 for each new sample.
Reported metrics	Avg@k and pass@k; $k = 32$ for AIME and AMC, $k = 8$ for Minerva, OlympiadBench, and MATH	Code: avg@4 and pass@4. DeepSeek: the same benchmark-specific k as the main results.

Table 4: Evaluation settings and code-evaluation provenance, consistent with Sections 5.1 and F. For the main experiments, we evaluate the best-performing checkpoint for each algorithm.

E Overlong Reward Shaping in DAPO

Enabling overlong reward shaping has only a small effect on DAPO's average performance in our experiments. As discussed in Section 2.2, our default setting omits overlong reward shaping, and the DAPO results in Tables 1 and 2 use this setting. To examine the effect of this choice, we compare DAPO with and without overlong reward shaping on Qwen3-4B-Base and Qwen3-8B-Base. We report avg@k and pass@k on the same seven mathematical reasoning benchmarks, retaining the benchmark-specific sample counts from Section 5.1.

Model	DAPO variant	AIME24 avg@32	AIME25 avg@32	AIME26 avg@32	AMC avg@32	Minerva avg@8	Olympiad avg@8	MATH avg@8	AVG
Qwen3-4B -Base	w/o overlong	**18.96**	15.10	12.71	52.07	**24.17**	**40.65**	78.20	34.55
	w/ overlong	18.33	**15.62**	**14.27**	**53.01**	23.02	37.74	**80.33**	**34.62**
Qwen3-8B -Base	w/o overlong	**24.58**	20.52	17.19	64.83	**29.50**	48.30	**85.52**	41.49
	w/ overlong	22.92	**21.04**	**21.34**	**66.72**	26.84	**48.94**	85.02	**41.83**

Table 5: Avg@k (%) of DAPO with and without overlong reward shaping. We report avg@32 for AIME and AMC, and avg@8 for Minerva, Olympiad, and MATH. Best results within each model scale are **bolded**.

Model	DAPO variant	AIME24 pass@32	AIME25 pass@32	AIME26 pass@32	AMC pass@32	Minerva pass@8	Olympiad pass@8	MATH pass@8	AVG
Qwen3-4B -Base	w/o overlong	43.33	**43.33**	30.00	**84.34**	**39.71**	59.56	91.00	55.90
	w/ overlong	**50.00**	36.67	**33.33**	83.13	36.03	**61.04**	**92.60**	**56.11**
Qwen3-8B -Base	w/o overlong	**56.67**	46.67	**46.67**	**90.36**	**43.01**	63.70	**93.40**	62.93
	w/ overlong	**56.67**	**50.00**	**46.67**	86.75	40.07	**67.41**	**93.40**	**63.00**

Table 6: Pass@k (%) of DAPO with and without overlong reward shaping. We report pass@32 for AIME and AMC, and pass@8 for Minerva, Olympiad, and MATH. Best results within each model scale, including ties, are **bolded**.

Tables 5 and 6 show similar aggregate performance with and without overlong reward shaping. Enabling it improves average avg@k by only 0.07 and 0.34 percentage points on the 4B and 8B models, respectively, while average pass@k changes by 0.21 and 0.07 points. All four aggregate changes are below 0.35 percentage points, indicating that overlong reward shaping has only a limited effect on overall performance.

These results leave the main comparison with DA3PO unchanged. Relative to DAPO with overlong reward shaping, DA3PO retains average avg@k gains of 5.01 and 5.06 percentage points on the 4B and 8B models, respectively, and average pass@k gains of 5.95 and 5.80 points. Under the evaluated settings, enabling overlong reward shaping therefore has a limited effect on DAPO's overall performance and does not close the gap to DA3PO.

F Additional Experimental Results

F.1 Generalization to Code Generation

Model	Algorithm	HumanEval+ avg@4	MBPP+ avg@4	LiveCodeBench v6 avg@4	AVG
Qwen3-8B-Base	Base Model	18.14	13.16	16.14	15.81
	GRPO	74.09	69.91	25.43	56.48
	DAPO	78.20	71.10	28.86	59.39
	GSPO	77.44	71.96	27.43	58.94
	DA3PO	**81.25**	**73.35**	**29.57**	**61.39**
	vs. GRPO	↑7.16%	↑3.44%	↑4.14%	↑4.91%
	vs. DAPO	↑3.05%	↑2.25%	↑0.71%	↑2.00%
	vs. GSPO	↑3.81%	↑1.39%	↑2.14%	↑2.45%

Table 7: Avg@k (%) across three code generation benchmarks with Qwen3-8B-Base. We report avg@4 on all benchmarks. Best results are **bolded**.

Model	Algorithm	HumanEval+ pass@4	MBPP+ pass@4	LiveCodeBench v6 pass@4	AVG
Qwen3-8B-Base	Base Model	48.78	41.27	28.00	39.35
	GRPO	82.93	79.89	28.00	63.61
	DAPO	89.02	78.84	32.00	66.62
	GSPO	87.20	80.42	31.43	66.35
	DA3PO	**91.46**	**81.22**	**32.57**	**68.42**
	vs. GRPO	↑8.53%	↑1.33%	↑4.57%	↑4.81%
	vs. DAPO	↑2.44%	↑2.38%	↑0.57%	↑1.80%
	vs. GSPO	↑4.26%	↑0.80%	↑1.14%	↑2.07%

Table 8: Pass@k (%) across three code generation benchmarks with Qwen3-8B-Base. We report pass@4 on all benchmarks. Best results are **bolded**.

Tables 7 and 8 report the code generation results. Using the same $\lambda = 2$ and $\tau = 0.5$ as in the mathematical reasoning experiments, DA3PO achieves an average avg@4 of 61.39%. This exceeds GRPO, DAPO, and GSPO by 4.91, 2.00, and 2.45 percentage points, respectively. DA3PO also ranks first on every benchmark, outperforming the strongest baseline by 3.05 points on HumanEval+, 1.39 points on MBPP+, and 0.71 points on LiveCodeBench v6. The same pattern holds for pass@4. DA3PO achieves the highest result on all three benchmarks, surpassing the strongest baseline by 1.80 points. Its gains in both avg@4 and pass@4 show that DA3PO improves the likelihood of generating correct programs as well as the coverage of problems solved within four attempts.

Together, these results show that the benefits observed in the main experiments extend to code generation. The same amplification factor $\lambda = 2$ and difficulty threshold $\tau = 0.5$ yield improvements in average accuracy and pass@k in both settings, supporting the generality of DA3PO beyond the mathematical reasoning experiments.

F.2 Generalization to a Smaller Distilled Reasoning Model

Model	Algorithm	AIME24 avg@32	AIME25 avg@32	AIME26 avg@32	AMC avg@32	Minerva avg@8	Olympiad avg@8	MATH avg@8	AVG
DeepSeek-R1-Distill-Qwen-1.5B	Base Model	23.54	19.06	13.13	58.40	25.37	50.41	81.88	38.83
	GRPO	26.77	22.81	17.19	67.06	28.31	53.54	86.52	43.17
	DAPO	34.69	28.12	21.56	71.12	30.24	59.09	88.85	47.67
	GSPO	38.44	30.63	23.96	71.80	33.00	59.91	89.55	49.61
	DA3PO	**40.21**	**34.38**	**26.98**	**75.38**	**35.11**	**61.57**	**91.00**	**52.09**
	vs. GRPO	↑13.44%	↑11.57%	↑9.79%	↑8.32%	↑6.80%	↑8.03%	↑4.48%	↑8.92%
	vs. DAPO	↑5.52%	↑6.26%	↑5.42%	↑4.26%	↑4.87%	↑2.48%	↑2.15%	↑4.42%
	vs. GSPO	↑1.77%	↑3.75%	↑3.02%	↑3.58%	↑2.11%	↑1.66%	↑1.45%	↑2.48%

Table 9: Avg@k (%) across seven mathematical reasoning benchmarks with DeepSeek-R1-Distill-Qwen-1.5B. We report avg@32 for AIME and AMC, and avg@8 for Minerva, Olympiad, and MATH. Best results are **bolded**.

Model	Algorithm	AIME24 pass@32	AIME25 pass@32	AIME26 pass@32	AMC pass@32	Minerva pass@8	Olympiad pass@8	MATH pass@8	AVG
DeepSeek-R1-Distill-Qwen-1.5B	Base Model	63.33	43.33	20.00	91.57	43.75	71.70	94.80	61.21
	GRPO	70.00	40.00	46.67	93.98	43.75	70.22	94.20	65.55
	DAPO	63.33	43.33	50.00	92.77	46.32	72.15	95.60	66.21
	GSPO	**73.33**	56.67	**56.67**	92.77	48.16	75.41	96.00	71.29
	DA3PO	**73.33**	**60.00**	**56.67**	**95.18**	**49.63**	**75.56**	**96.40**	**72.40**
	vs. GRPO	↑3.33%	↑20.00%	↑10.00%	↑1.20%	↑5.88%	↑5.34%	↑2.20%	↑6.85%
	vs. DAPO	↑10.00%	↑16.67%	↑6.67%	↑2.41%	↑3.31%	↑3.41%	↑0.80%	↑6.19%
	vs. GSPO	0.00	↑3.33%	0.00	↑2.41%	↑1.47%	↑0.15%	↑0.40%	↑1.11%

Table 10: Pass@k (%) across seven mathematical reasoning benchmarks with DeepSeek-R1-Distill-Qwen-1.5B. We report pass@32 for AIME and AMC, and pass@8 for Minerva, Olympiad, and MATH. Best results are **bolded**.

Tables 9 and 10 show that DA3PO remains effective on DeepSeek-R1-Distill-Qwen-1.5B. It achieves an average avg@k of 52.09%, exceeding GRPO, DAPO, and GSPO by 8.92, 4.42, and 2.48 percentage points, respectively. On AIME24, AIME25, and AIME26, it outperforms GSPO by 1.77, 3.75, and 3.02 points, respectively, showing that the gains on challenging mathematical problems also hold for this smaller distilled model. Improvements over GSPO are also consistent on the remaining benchmarks, ranging from 1.45 points on MATH to 3.58 points on AMC. DA3PO also achieves the highest average pass@k, at 72.40%. This exceeds GRPO, DAPO, and GSPO by 6.85, 6.19, and 1.11 points, respectively. It is strictly best on five benchmarks and ties GSPO on AIME24 and AIME26, without reducing pass@k on any benchmark.

Together, these results show that the benefits observed in the main experiments extend to DeepSeek-R1 model family. The same amplification factor $\lambda = 2$ and difficulty threshold $\tau = 0.5$ yield improvements in average accuracy and pass@k in both settings, supporting the generality of DA3PO beyond the original Qwen3-Base model family.

F.3 Robustness to the Amplification Factor

Model	Factor λ	AIME24 avg@32	AIME25 avg@32	AIME26 avg@32	AMC avg@32	Minerva avg@8	Olympiad avg@8	MATH avg@8	AVG
Qwen3-4B-Base	$\lambda=1.0$	18.96	15.10	12.71	52.07	24.17	40.65	78.20	34.55
	$\lambda=1.5$	22.08	**19.17**	15.94	60.35	**26.98**	44.50	83.80	38.97
	$\lambda=2.0$	**23.96**	**19.17**	**17.19**	**60.50**	26.84	**44.52**	**85.20**	**39.63**
	$\lambda=2.5$	23.02	15.62	13.33	58.32	25.64	39.91	78.33	36.31
Qwen3-8B-Base	$\lambda=1.0$	24.58	20.52	17.19	64.83	29.50	48.30	85.52	41.49
	$\lambda=1.5$	28.65	21.56	21.35	66.72	35.57	53.19	87.02	44.87
	$\lambda=2.0$	**30.52**	**23.23**	24.17	69.69	**37.91**	**54.26**	**88.45**	**46.89**
	$\lambda=2.5$	29.69	21.04	**24.69**	**69.95**	30.24	48.94	86.85	44.49

Table 11: Ablation on the amplification factor λ. We report avg@32 for AIME and AMC, and avg@8 for Minerva, Olympiad and MATH. Setting $\lambda=1.0$ disables DAA and recovers the default DAPO baseline. Best results within each benchmark are **bolded**.

To study sensitivity and generalization of the amplification factor, we evaluate $\lambda \in \{1.5, 2.0, 2.5\}$ on both model scales. As defined in Eq. (7), λ controls how strongly DAA amplifies the advantages of correct responses on hard prompts. Setting $\lambda = 1$ disables DAA and recovers DAPO, providing the reference point in Table 11.

Every tested value of $\lambda > 1$ improves average accuracy over DAPO on both model scales. Even $\lambda = 1.5$ yields gains of 4.42 and 3.38 percentage points on Qwen3-4B-Base and Qwen3-8B-Base, respectively. Among the tested values, $\lambda = 2$ performs best, reaching average accuracies of 39.63% and 46.89% on the two models. These results confirm that the amplification mechanism is robust and effective regardless of the specific choice of λ, and support our default choice of $\lambda = 2$.

Increasing the factor to $\lambda = 2.5$ reduces average accuracy by 3.32 points on the 4B model and 2.40 points on the 8B model relative to $\lambda = 2$. However, it still produces the best 8B result on AIME26 and AMC. Thus, stronger amplification can benefit individual benchmarks, but moderate amplification provides the best overall performance. One possible explanation is that smaller models may generate fewer correct responses on hard prompts, which could make them more sensitive to overly aggressive amplification, whereas larger models may be more tolerant of higher amplification.

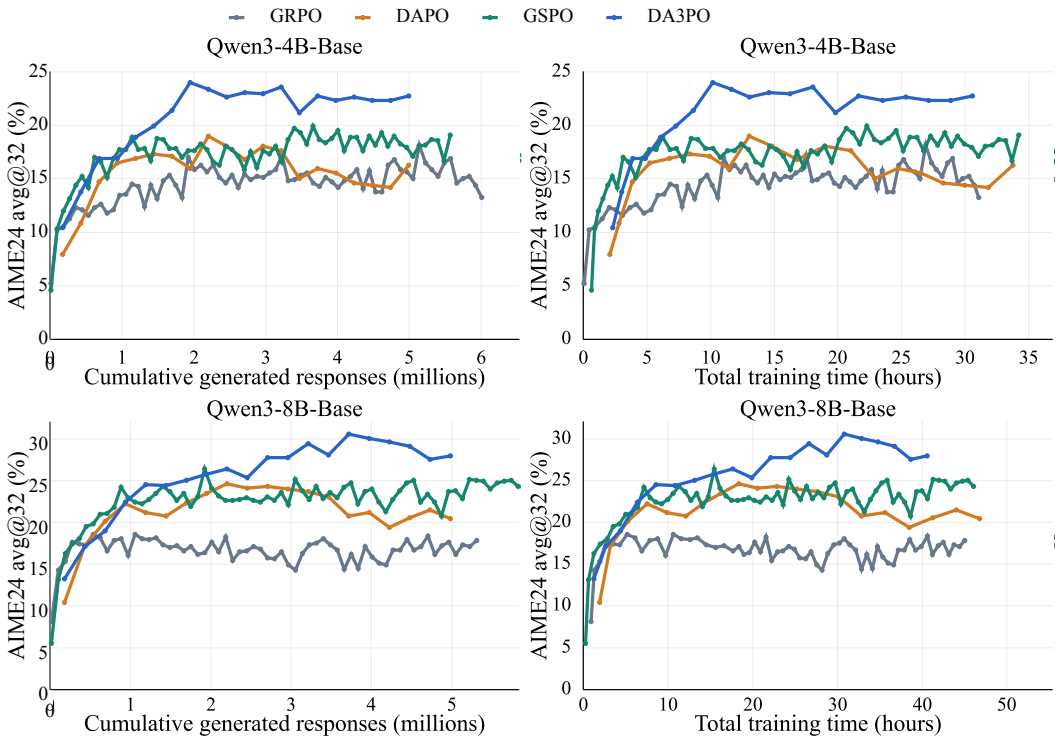


Figure 4: AIME24 avg@32 as a function of the total number of generated responses (left) and total training time in hours (right) for Qwen3-4B-Base (top) and Qwen3-8B-Base (bottom). Response counts include all generations, including those discarded by Dynamic Sampling, while training time covers the entire training procedure. DA3PO achieves the highest validation accuracy under similar sampling and time budgets on both model scales.

F.4 Efficiency under Comparable Sampling and Training Budgets

Comparing methods at the same training step can be misleading because Dynamic Sampling discards response groups that are entirely correct or incorrect and generates replacements. Consequently, two methods trained for the same number of steps may use different numbers of generated responses and different amounts of time. Figure 4 therefore reports AIME24 avg@32 against two direct measures of cost: the total number of generated responses, including discarded ones, and the total training time under the same hardware and software environment.

DAPO and DA3PO are trained for 200 steps, with DA3PO using $\lambda = 2$ and $\tau = 0.5$. To test whether longer training allows the baselines to catch up, we extend GRPO and GSPO to 700 and 650 steps on Qwen3-4B-Base, and to 620 and 680 steps on Qwen3-8B-Base, respectively.

The left panels compare methods by sampling cost. DA3PO achieves the highest validation accuracy on both model scales at similar numbers of generated responses. GRPO and GSPO still do not close the accuracy gap after generating more responses than DA3PO within the evaluated runs. The right panels compare methods by total training time. DA3PO again remains more accurate on both model scales, even when GRPO and GSPO run longer than DA3PO.

Thus, DA3PO's improvements cannot be attributed simply to using more generations or additional training time. In other words, DA3PO's gains do not depend on additional sampling or training time. These results show that DA3PO improves accuracy by capitalizing on these hard-to-sample responses more effectively and improving training efficiency.

G Ablation Study

We isolate the effect of restricting advantage amplification to hard prompts by comparing three settings on both model scales. DAPO uses no advantage amplification. DA3PO (ACR) amplifies all correct responses in groups retained by Dynamic Sampling, removing the condition $s_G < \tau$ from Eq. (7). The default DA3PO applies amplification only when $s_G < \tau$, with $\tau = 0.5$. All three settings retain Dynamic Sampling and use the same seven benchmarks and evaluation sample counts.

Model	Algorithm	AIME24 avg@32	AIME25 avg@32	AIME26 avg@32	AMC avg@32	Minerva avg@8	Olympiad avg@8	MATH avg@8	AVG
Qwen3-4B -Base	DAPO	18.96	15.10	12.71	52.07	24.17	40.65	78.20	34.55
	DA3PO (ACR)	23.02	15.62	15.94	58.32	25.64	39.91	83.80	37.46
	DA3PO	**23.96**	**19.17**	**17.19**	**60.50**	**26.84**	**44.52**	**85.20**	**39.63**
Qwen3-8B -Base	DAPO	24.58	20.52	17.19	64.83	29.50	48.30	85.52	41.49
	DA3PO (ACR)	28.65	21.56	**24.17**	**69.69**	30.24	48.94	86.85	44.30
	DA3PO	**30.52**	**23.23**	**24.17**	**69.69**	**37.91**	**54.26**	**88.45**	**46.89**

Table 12: Avg@k (%) of DAPO, DA3PO (ACR), and DA3PO. ACR denotes all correct responses: this variant amplifies the advantages of all correct responses in groups retained by Dynamic Sampling, without the condition $s_G < \tau$ in Eq. (7). Best results within each model scale, including ties, are **bolded**.

Model	Algorithm	AIME24 pass@32	AIME25 pass@32	AIME26 pass@32	AMC pass@32	Minerva pass@8	Olympiad pass@8	MATH pass@8	AVG
Qwen3-4B -Base	DAPO	43.33	43.33	30.00	84.34	39.71	59.56	91.00	55.90
	DA3PO (ACR)	50.00	43.33	**43.33**	86.75	40.44	58.07	92.60	59.22
	DA3PO	**60.00**	**46.67**	40.00	**91.57**	**41.18**	**61.19**	**93.80**	**62.06**
Qwen3-8B -Base	DAPO	56.67	46.67	46.67	90.36	43.01	63.70	93.40	62.93
	DA3PO (ACR)	60.00	**53.33**	46.67	89.16	46.32	70.52	**96.40**	66.06
	DA3PO	**70.00**	50.00	**53.33**	**92.77**	**49.26**	**71.41**	94.80	**68.80**

Table 13: Pass@k (%) of DAPO, DA3PO (ACR), and DA3PO. ACR denotes all correct responses: this variant amplifies the advantages of all correct responses in groups retained by Dynamic Sampling, without the condition $s_G < \tau$ in Eq. (7). Best results within each model scale are **bolded**.

Tables 12 and 13 show that advantage amplification is beneficial even without the difficulty condition. Compared with DAPO, DA3PO (ACR) improves average avg@k by 2.91 and 2.81 percentage points on the 4B and 8B models, respectively. It also improves average pass@k by 3.32 and 3.13 points, showing gains in both average accuracy and solution coverage.

Restricting amplification to hard prompts provides a further improvement. Relative to DA3PO (ACR), the default DA3PO raises average avg@k by another 2.17 and 2.59 points on the 4B and 8B models. It improves avg@k on all seven benchmarks for the 4B model and on five benchmarks, with two ties, for the 8B model. Average pass@k also increases by 2.84 and 2.74 points, respectively. The default DA3PO therefore achieves the best aggregate result for both metrics on both model scales.

These results support both advantage amplification itself and the use of prompt difficulty to determine when amplification is applied.

H CASE STUDIES

H.1 QWEN3-4B: MAXIMIZING A COMPLEX EXPRESSION

Question (AIME 2024). Find the largest possible real part of $(75+117i)z+(96+144i)/z$, where z is complex and $|z|=4$. **Reference answer:** 540.

Base Model (Incorrect)

Let $z = 4(\cos\theta + i\sin\theta)$. It expands the first product, then states: "The real part is $300\cos\theta - 468\sin\theta$." It rewrites this as a shifted cosine with amplitude

$$\sqrt{300^2+468^2}.$$

[... intermediate reasoning omitted ...]

"Since $-1 \le \cos(\theta+\phi) \le 1$," it concludes that this amplitude is the maximum, then gives a different numerical value.

Model answer: 480.

Check. The reciprocal term contributes $24\cos\theta + 36\sin\theta$ and cannot be dropped.

GRPO (Incorrect)

The rollout correctly sets $z = 4e^{i\theta}$ and obtains

$$w = (300+468i)e^{i\theta} + (24+36i)e^{-i\theta}.$$

After expansion, it collects only the cosine terms as the real part: "Real parts: $300\cos\theta + 24\cos\theta = 324\cos\theta$."

[... intermediate reasoning omitted ...]

It checks the imaginary terms again, but leaves out the real sine terms. "Since $\cos\theta$ has a maximum value of 1," it returns 324.

Model answer: 324.

Check. The missing real term is $-432\sin\theta$.

DAPO (Incorrect)

The response incorrectly obtains

$$396\cos\theta - 324\sin\theta.$$

It then applies the amplitude formula:

$$R = \sqrt{396^2+324^2} = 36\sqrt{202}.$$

[... intermediate reasoning omitted ...]

"Rounding to the nearest whole number, it's approximately 512." It debates the answer format without correcting the coefficients.

Model answer: 512.

Check. It miscalculates $(96+144i)/4$ as $24(4+6i)$; the correct coefficients are 324 and -432.

GSPO (Incorrect)

The response substitutes $z = 4e^{i\theta}$ but loses parentheses in the reciprocal term:

$$E = 4(75+117i)e^{i\theta} + 24 + 36i\,e^{-i\theta}.$$

It consequently obtains

$$\mathrm{Re}(E) = 300\cos\theta - 432\sin\theta + 24.$$

[... intermediate reasoning omitted ...]

It evaluates $\sqrt{300^2+432^2}$ as 526, then adds 24.

Model answer: 550.

Check. The entire factor $(24+36i)$ must multiply $e^{-i\theta}$; the 24 must also carry a cosine. Its square-root calculation is also inaccurate.

DA3PO (Correct)

"Let's denote $z = x+yi$," with $x^2+y^2=16$. The response rationalizes the reciprocal term and keeps both real contributions:

$$\mathrm{Re}((75+117i)z) = 75x - 117y, \qquad \mathrm{Re}\left(\frac{96+144i}{z}\right) = 6x+9y.$$

Thus the objective is $81x - 108y$.

[... intermediate reasoning omitted ...]

With $x = 4\cos\theta$ and $y = 4\sin\theta$, it maximizes $324\cos\theta - 432\sin\theta$ using the amplitude formula: $\sqrt{324^2+432^2} = \sqrt{291600} = 540$.

Model answer: 540.

Check. Both terms are retained and the amplitude is evaluated exactly.

H.2 QWEN3-8B: SATISFYING THE MEDIAN AND MODE CONSTRAINTS

Question (AIME 2024). A list of positive integers sums to 30, has unique mode 9, and has a positive integer median that does not appear in the list. Find the sum of the squares of its items. **Reference answer:** 236.

Base Model (Incorrect)

The response correctly reasons that the list must have an even number of elements. It tries six elements, with two 9s and four remaining numbers summing to 12. It first tests $1, 2, 3, 6$.

[... intermediate reasoning omitted ...]

After rejecting $[1, 2, 3, 6, 9, 9]$ and $[1, 2, 4, 5, 9, 9]$ for noninteger medians, it proposes $[1, 3, 4, 4, 9, 9]$: "The median would be $(4 + 4)/2 = 4$, which is not in the list. This works!" It computes

$$1^2 + 3^2 + 4^2 + 4^2 + 9^2 + 9^2 = 204.$$

Model answer: 204.

Check. 4 is in the list; moreover, 4 and 9 tie as modes. Only the sum condition holds.

GRPO (Incorrect)

The response claims the list must have odd length "so there's a single middle number," even though that middle element necessarily belongs to the list. It starts with $n = 5$ and places two 9s alongside a proposed median 7.

[... intermediate reasoning omitted ...]

With the remaining pair $(1, 4)$, it sorts the list as $[1, 4, 7, 9, 9]$, then states: "Median is 7, which is not in the list." It checks the sum and mode, and computes

$$1 + 16 + 49 + 81 + 81 = 228.$$

Model answer: 228.

Check. The median is indeed 7, but 7 is explicitly present.

DAPO (Incorrect)

The response considers four elements with two 9s and a remaining pair summing to 12. For the pair $(1, 11)$, it writes: "List $9, 9, 1, 11$. Median $(9 + 1)/2 = 5$. 5 is not in the list. Valid." It then calculates

$$9^2 + 9^2 + 1^2 + 11^2 = 284.$$

[... intermediate reasoning omitted ...]

It rejects a six-element candidate $[9, 9, 9, 1, 1, 1]$ because 9 and 1 tie as modes. After considering other lengths, it concludes that "the list $1, 9, 9, 11$ satisfies all the given conditions."

Model answer: 284.

Check. After sorting, the middle entries are $9, 9$, so the median is 9, not 5.

GSPO (Incorrect)

The response also assumes an odd length. It tries several three-, five-, and seven-element lists. For $[9, 9, 12]$, it correctly rejects the median 9 because it is present. It also rejects longer candidates whose sums exceed 30.

[... intermediate reasoning omitted ...]

It finally selects $[2, 3, 7, 9, 9]$ and states: "Median: 7 (not in the list)." It then asserts that all three conditions are satisfied and evaluates

$$2^2+3^2+7^2+9^2+9^2 = 4+9+49+81+81 = 224.$$

Model answer: 224.

Check. The same membership error remains: the third element, 7, is the median and is in the list. The sum and mode are insufficient.

DA3PO (Correct)

The response initially makes the same odd-length assumption, but notices that its proposed middle elements appear in the list. It rejects $[9, 9, 12]$ after sorting and revisits the number of elements. It later considers an even length, where the median is the average of two middle entries.

[... intermediate reasoning omitted ...]

It moves to four elements with two 9s. Trying $c = 7$ gives the other value $d = 5$, and the response explicitly sorts the result: "So, the list is $5, 7, 9, 9$. Wait, ordered: $5, 7, 9, 9$." It checks that the median is $(7+9)/2 = 8$, absent from the list; the sum is 30 and 9 is the unique mode. Finally,

$$5^2 + 7^2 + 9^2 + 9^2 = 25 + 49 + 81 + 81 = 236.$$

Model answer: 236.

Check. The final list satisfies all three constraints: sum 30, a unique mode 9, and absent integer median 8.

H.3 QWEN3-8B CODE: CONSTRUCTING THE SHORTEST PALINDROME

Question (HumanEval/10). Implement `make_palindrome(string)`: return the shortest palindrome that begins with the supplied string. For example, `'cat'` and `'cata'` both yield `'catac'`. The helper `is_palindrome(s)` tests `s == s[::-1]`. **Reference criterion:** pass both the base and extended HumanEval+ tests.

Base Model (Incorrect)

[Prose, docstrings and comments omitted.]

```
def make_palindrome(string: str) -> str:
    if not string:
        return string
    for i in range(len(string), 0, -1):
        if is_palindrome(string[:i]):
            prefix = string[i:]
            return string + prefix[::-1]

def is_palindrome(string: str) -> bool:
    return string == string[::-1]
```

Recorded verdict: base fail; extended fail.

Check. The code checks a prefix, `string[:i]`, instead of a suffix. For `'cat'`, it accepts `'c'` and appends `'ta'`, producing `'catta'`. This is not a palindrome.

GRPO (Incorrect)

[Prose, helper, docstring and comments omitted.]

```
def make_palindrome(string: str) -> str:
    if not string:
        return ""
    for i in range(len(string), -1, -1):
        suffix = string[i:]
        if is_palindrome(suffix):
            prefix = string[:i]
            reversed_prefix = prefix[::-1]
            return string + reversed_prefix
    return string
```

Recorded verdict: base fail; extended fail.

Check. Although the prose calls for the longest palindromic suffix, the loop starts at the empty suffix. It returns `'cattac'` for `'cat'`. The palindrome condition holds, but minimality fails.

DAPO (Incorrect)

[Prose, helper, docstring and comments omitted.]

```
def make_palindrome(string: str) -> str:
    if string == "":
        return ""
    if is_palindrome(string):
        return string
    for i in range(len(string), -1, -1):
        suffix = string[i:]
        if is_palindrome(suffix):
            prefix = string[:i]
            return string + prefix[::-1]
    return string
```

Recorded verdict: base fail; extended fail.

Check. The early return fixes already-palindromic inputs, but other inputs still reach the empty suffix first. `'cat'` yields `'cattac'`.

GSPO (Incorrect)

[Prose, helper, docstring and comments omitted.]

```
def make_palindrome(string: str) -> str:
    if len(string) == 0:
        return ''
    if is_palindrome(string):
        return string
    for i in range(len(string), -1, -1):
        suffix = string[i:]
        if is_palindrome(suffix):
            prefix_to_reverse = string[:len(
 string) - len(suffix)]
            reversed_prefix = prefix_to_reverse
 [::-1]
            return string + reversed_prefix
    return string
```

Recorded verdict: base fail; extended fail.

Check. For `'cat'`, it selects the empty suffix and returns `'cattac'`.

DA3PO (Correct)

[Prose, docstrings and comments omitted.]

```
def is_palindrome(string: str) -> bool:
    return string == string[::-1]
def make_palindrome(string: str) -> str:
    if string == "":
        return ""
    for i in range(len(string)):
        if is_palindrome(string[i:]):
            prefix = string[:i]
            return string + prefix[::-1]
    return string + string[::-1]
```

Recorded verdict: base pass; extended pass.

Check. Increasing i examines suffixes from longest to shortest. The first palindromic suffix therefore requires the fewest appended characters. For `'cat'`, the suffix is `'t'` and the appended prefix is `'ac'`; the result is `'catac'`. The same code leaves an already-palindromic input unchanged and returns the empty string for empty input.

H.4 DEEPSEEK-R1-DISTILL-QWEN-1.5B: TRACKING LOGARITHMIC EXPONENTS

Question (AIME 2024). Positive x, y, z satisfy $\log_2(x/(yz)) = 1/2$, $\log_2(y/(xz)) = 1/3$, and $\log_2(z/(xy)) = 1/4$. Write $|\log_2(x^4y^3z^2)| = m/n$ in lowest terms and find $m+n$. **Reference answer:** 33.

Base Model (Incorrect)

The base model converts the equations to exponential form and correctly obtains

$$x = 2^{-7/24}, \quad y = 2^{-9/24}, \quad z = 2^{-10/24}.$$

It uses $\log_2(x^4y^3z^2) = 4\log_2 x + 3\log_2 y + 2\log_2 z$.

[... intermediate reasoning omitted ...]

It then adds the numerators incorrectly:

$$\frac{-28-27-20}{24} = \frac{-65}{24}.$$

The response takes the absolute value, declares $65/24$ reduced, and adds $65+24$.

Model answer: 89.

Check. The numerator is -75, so the reduced fraction is $25/8$.

GRPO (Incorrect)

The response correctly derives $z = 2^{-5/12}$, $xy = 2^{-2/3}$ and $x = 2^{1/12}y$. It then writes

$$2^{1/12+8/12} = 2^{9/12},$$

but immediately changes the denominator: "Thus, $2^{9/2}y^2 = 1$," giving $y = 2^{-9/4}$ and $x = 2^{-13/6}$.

[... intermediate reasoning omitted ...]

It reports $65+4 = 69$ after these exponents lead to

$$|\log_2(x^4y^3z^2)| = 65/4.$$

Model answer: 69.

Check. The exponent $9/12$ must stay $3/4$, giving $y = 2^{-3/8}$. The erroneous exponents give $\log_2(z/(xy)) = 4$, contradicting the required $1/4$.

DAPO (Incorrect)

After substitution, the response obtains $x = 2^{-7/24}$. A later substitution only yields $y = y$, which it calls a consistency check. It then asserts, "Solving, we find: $z = 2^{-1/4}$." No valid constraint is given to justify this new value.

[... intermediate reasoning omitted ...]

Using $x = 2^{-7/24}$, $y = 2^{-9/24}$, and $z = 2^{-6/24}$, it calculates

$$x^4y^3z^2 = 2^{(-28-27-12)/24} = 2^{-67/24}.$$

"Since 67 and 24 are coprime," it returns their sum.

Model answer: 91.

Check. The first two equations imply $z^2 = 2^{-5/6}$, contradicting $z = 2^{-1/4}$.

GSPO (Incorrect)

The response reaches $z = 2^{-5/12}$ but incorrectly obtains $xy = 2^{-1/6}$, followed by

$$x = 2^{-1/24}, \qquad y = 2^{-1/8}.$$

It attempts to verify the third equation and computes $z/(xy) = 2^{-1/4}$, then labels this "Correct," although the required ratio is $2^{1/4}$.

[... intermediate reasoning omitted ...]

The incorrect exponents give

$$|\log_2(x^4y^3z^2)| = 11/8.$$

Model answer: 19.

Check. The verification has the wrong sign and should reject the candidate. Agreement with the first two equations alone is insufficient.

DA3PO (Correct)

Let $a = \log_2 x$, $b = \log_2 y$, and $c = \log_2 z$. The response transforms the problem into

$$a-b-c = \tfrac{1}{2}, \qquad -a+b-c = \tfrac{1}{3}, \qquad -a-b+c = \tfrac{1}{4}.$$

Adding the first two equations gives $c = -5/12$; adding the last two gives $a = -7/24$. Then $a-b = 1/12$ gives $b = -3/8$. The response explicitly substitutes $a = -7/24$ into this last relation: $b = -7/24 - 2/24 = -9/24$.

[... intermediate reasoning omitted ...]

It keeps the common denominator throughout: $4a+3b+2c = (-28-27-20)/24 = -75/24$. Thus the absolute value is $25/8$ in lowest terms and $m+n = 25+8 = 33$.

Model answer: 33.

Check. The exponents satisfy all three original equations, and the final step answers the requested sum rather than the intermediate fraction.