Adaptive-Boundary-Clipping GRPO: Ensuring Bounded Ratios for Stable and Generalizable Training

Anonymous ACL submission

Abstract

Group Relative Policy Optimization (GRPO) has emerged as a popular algorithm for reinforcement learning with large language models (LLMs). However, upon analyzing its clipping mechanism, we argue that it is suboptimal in certain scenarios. With appropriate modifications, GRPO can be significantly enhanced to improve both flexibility and generalization. To this end, we propose Adaptive-Boundary-Clipping GRPO (ABC-GRPO), an asymmetric and adaptive refinement of the original GRPO framework. We demonstrate that ABC-GRPO achieves superior performance over standard GRPO on mathematical reasoning tasks using the Qwen3 LLMs. Moreover, ABC-GRPO maintains substantially higher entropy throughout training, thereby preserving the model's exploration capacity and mitigating premature convergence.

1 Introduction

Reinforcement learning from human feedback (Christiano et al., 2017; Ouyang et al., 2022) has emerged as a core technology for enhancing the reasoning capabilities of large language models (LLMs), powering recent advances in mathematical reasoning and complex problem-solving (Guo et al., 2025; Wei et al., 2022). Group Relative Policy Optimization (GRPO) (Shao et al., 2024) has become a popular choice for training reasoning models, eliminating the value network overhead of Proximal Policy Optimization (PPO) (Schulman et al., 2017, 2015) by computing advantages at the sequence level using group-relative rewards. This simplification dramatically reduces computational costs and implementation complexity.

However, while investigating ways to improve GRPO's model performance, we discovered that its design—specifically, the removal of PPO's value network while retaining PPO's clipping mechanism—is suboptimal and can be misleading in certain scenarios. This combination may lead to excessive penalization of tokens, thereby degrading the model's generalization capability.

In this paper, we dissect the four quadrants of the clipping operation and identify the one that most significantly affects model performance. Through both theoretical analysis and empirical experiments, we propose Adaptive-Boundary-Clipping GRPO (ABC-GRPO)—a novel GRPO-based algorithm that introduces greater flexibility in the clipping mechanism, independent of the sign of the advantage. ABC-GRPO not only enhances the model's generalization capability but also mitigates entropy collapse during training. Our experiments on mathematical reasoning with Qwen3 large language models (Qwen Team, 2025) demonstrate that ABC-GRPO improves both sampling accuracy (Avg@64) and exploration diversity (Pass@64). Critically, while standard GRPO's Pass@64 decreases during training, ABC-GRPO achieves monotonic improvement, directly addressing the reasoning boundary limitation of current Reinforcement Learning with Verifiable Rewards (RLVR) methods. ABC-GRPO also maintains significantly higher entropy than standard GRPO, preserving exploration capacity throughout training. We envision ABC-GRPO as a robust and scalable algorithmic foundation that will enable the continued advancement of large-scale RL training for reasoning tasks.[1]

2 Related Work

Several recent methods address GRPO's clipping and advantage-estimation limitations. DAPO (Yu et al., 2025) introduces a "clip-higher" strategy with asymmetric thresholds ($\varepsilon_{\text{low}} < \varepsilon_{\text{high}}$) and token-level loss normalization, while

[1]Code: https://anonymous.4open.science/r/ABC-GRPO-4707

GSPO (Zheng et al., 2025) lifts clipping from the token to the sequence level to alleviate over-punishment of individual tokens. Empirically, Yue et al. (2025) document that RLVR methods *narrow the reasoning boundary*—Pass@k at large k degrades below the base model during training; our four-quadrant analysis traces this to GRPO's unbounded Q4 clipping. Orthogonal alignment directions such as DPO (Rafailov et al., 2023) and REINFORCE-style methods (Williams, 1992; Ahmadian et al., 2024) are complementary but not the focus of this work.

3 Preliminaries

Notation. We consider a policy π_θ generating responses conditioned on prompt x. For a response $y = (y_1, \ldots, y_T)$, we denote $r_t = \frac{\pi_\theta(y_t|x, y_{<t})}{\pi_{\text{old}}(y_t|x, y_{<t})}$ as the importance ratio at token t, and A_t as the advantage estimate. For group-based methods sampling G responses $\{y_1, \ldots, y_G\}$, we use $r_{i,t}$ and A_i for the token-level ratio and sequence-level advantage of the i-th response, respectively. We write $\text{clip}(v, a, b) = \max(\min(v, b), a)$.

PPO. PPO (Schulman et al., 2017) constrains policy updates via a clipped surrogate objective:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}_t\Big[\min\big(r_t A_t, \text{clip}(r_t, 1-\varepsilon, 1+\varepsilon)\, A_t\big)\Big] \quad (1)$$

where ε is the clipping threshold (typically 0.2). The min operator creates an asymmetric trust region: when $A_t > 0$, it clips r_t at $1+\varepsilon$; when $A_t \leq 0$, it clips at $1-\varepsilon$. This sign-dependent clipping assumes advantage signs are correct, which is reasonable given PPO's token-level advantage estimation via a learned value function.

GRPO. GRPO (Shao et al., 2024) eliminates the value function by computing advantages at the sequence level. For each prompt x, it samples G responses from π_{old} and computes:

$$A_i = r(x, y_i) - \frac{1}{G}\sum_{j=1}^{G} r(x, y_j) \quad (2)$$

This sequence-level advantage A_i is then applied uniformly to all tokens in y_i using PPO's clipped objective (KL regularization omitted for brevity):

$$\mathcal{L}^{\text{GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, \{y_i\}\sim\pi_{\text{old}}}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \min\Big(r_{i,t}A_i,\ \text{clip}(r_{i,t}, 1-\varepsilon, 1+\varepsilon)A_i\Big)\Bigg] \quad (3)$$

4 Motivation

As formalized in Sect. 3, GRPO inherits PPO's clipping mechanism but replaces token-level advantages with sequence-level ones. This combination introduces several fundamental issues, as also noted by Liu et al. (2025).

First, the advantage assignment problem. GRPO eliminates the value network by relying on sequence-level advantages and uniformly applies this scalar advantage to every token in the sequence. This coarse-grained signal leads to misattribution: tokens that are correct or helpful within a low-reward (i.e., "failed") sequence are incorrectly penalized, while flawed or irrelevant tokens in a high-reward (i.e., "successful") sequence receive undue credit. This issue is particularly pronounced in long-form generation, as illustrated in Fig. 1: the same intermediate tokens may receive opposite advantage values depending solely on whether the final answer is correct, despite being identical and potentially valid in both cases.

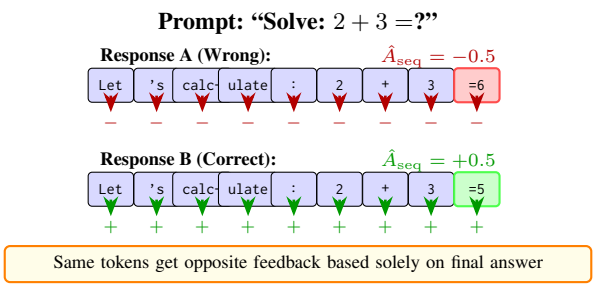

Figure 1: The credit assignment problem in GRPO. Two responses differ only in the final token, yet GRPO assigns opposite advantage signs to the identical first 8 tokens.

Second, the error reinforcement problem. When a response receives a positive sequence-level advantage, typically due to a correct final outcome (e.g., a few tokens at the end), all preceding tokens, including "nonsense" or "erroneous reasoning," are equally reinforced. Conversely, if the final result is incorrect and the sequence receives a negative advantage, even sound and logically valid

reasoning steps earlier in the response are negatively reinforced. While this may be tolerable for short responses, it becomes severely detrimental in long reasoning chains, where the signal-to-noise ratio of the learning signal degrades significantly. This is because in long reasoning trajectories, only a small subset of tokens directly determines the final outcome, yet GRPO's sequence-level credit assignment treats all tokens equally, introducing substantial noise into the learning signal.

Third, the over-punishment problem. GRPO adapts PPO's clipping mechanism, which is inherently asymmetric and imposes no upper bound on the policy loss in the following two cases:

- Case 1: $A < 0$ and $r > 1 + \varepsilon$
- Case 2: $A > 0$ and $r < 1 - \varepsilon$

In Case 1, the absence of an upper bound on r can lead to unbounded punishment when a large positive likelihood ratio r is multiplied by a large negative advantage A. While GRPO may intend such strong penalization for low-reward sequences, this becomes problematic when combined with the error reinforcement issue: all tokens in a negatively advantaged sequence, including those that are actually correct or reasonable, are subjected to potentially excessive and unbounded suppression—a phenomenon analogous to reward overoptimization (Gao et al., 2023). Notably, PPO does not suffer from this pathology because its advantage estimates are derived from a token-level value function, making them more accurate and localized.

The cumulative effect of this unbounded suppression is a progressive narrowing of the model's reasoning repertoire: as valid reasoning tokens are suppressed, the policy collapses onto fewer solution strategies, ultimately degrading Pass@k at large k below the pre-trained base model—a phenomenon empirically documented by Yue et al. (2025).

In Case 2, while over-punishment is not a concern, we still advocate for bounding the clipping operation. From a theoretical standpoint, such a bound acts as a form of regularization—analogous to maximum entropy objectives in policy optimization (Haarnoja et al., 2018)—which not only improves the model's generalization capability but also mitigates entropy collapse during training.

5 Method

Notation. We use $r_{i,t}$ to denote the importance ratio for token t in sequence i, and $\hat{A}_i$ for the sequence-level advantage estimate. In this section, we distinguish between A^*, the true token-level advantage (reflecting whether a token genuinely contributes positively or negatively to the outcome), and $\hat{A}$, the estimated advantage used in GRPO, which is computed at the sequence level and uniformly applied to all tokens in that sequence. A sign error occurs when $\text{sign}(\hat{A}) \neq \text{sign}(A^*)$. For example, a correct token ($A^* > 0$) may receive negative feedback ($\hat{A} < 0$) simply because it appeared in a low-reward (i.e., "failed") sequence.

5.1 The Four Quadrants of the $(r, \hat{A})$ Space

To better understand how GRPO's inherited clipping mechanism behaves, we classify all possible policy update scenarios along two orthogonal dimensions: (1) the sign of the estimated advantage ($\hat{A} \lesseqgtr 0$), indicating whether the token should be encouraged or discouraged; and (2) the direction of policy change, captured by the likelihood ratio $r \lesseqgtr 1$, which reflects whether the updated policy increases or decreases the probability of the token. These two binary distinctions partition the $(r, \hat{A})$ space into four quadrants (Table 1, Fig. 2). As we will demonstrate, GRPO's clipping, adopted from PPO, only provides protection in two of these quadrants (Q1 and Q3) while leaving the other two entirely unbounded.

Table 1: Four-quadrant analysis of GRPO's clipping. GRPO only protects Q1 and Q3. Q2 and Q4 are blind spots where no clipping occurs. Q4 ($\hat{A} < 0$, $r > 1$) is particularly problematic: unbounded updates suppress good tokens.

Quadrant	$\hat{A}$	r	GRPO Clipping Rule	Status
Q1	> 0	> 1	$\min(r, 1{+}\varepsilon)$	✓ Clipped
Q2	> 0	< 1	$\min(r, 1{+}\varepsilon){=}r$	✗ Blind Spot
Q3	< 0	< 1	$\max(r, 1{-}\varepsilon)$	✓ Clipped
Q4	< 0	> 1	$\max(r, 1{-}\varepsilon){=}r$	✗ Unbounded

Q4 ($\hat{A} < 0$, $r > 1$): In this quadrant, the unclipped update can aggressively suppress a token that should be encouraged, according to its true advantage A^*. Because the policy ratio r is unbounded above, the update can drive the token's probability close to zero. This excessive suppression forces the lost probability mass to redistribute across the entire vocabulary, inadvertently

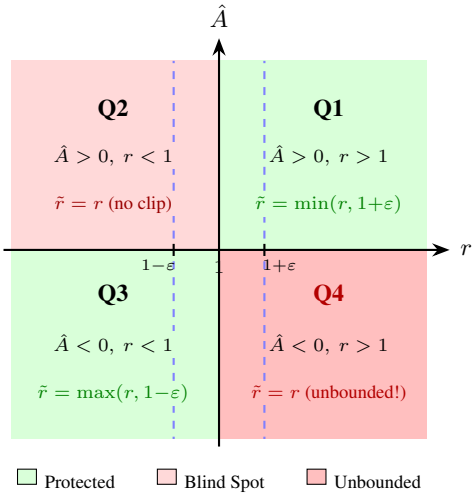

Figure 2: Four-quadrant analysis of GRPO's clipping in $(r, \hat{A})$ space. Q1 and Q3 are protected by clipping. Q2 is a mild blind spot ($r < 1$ bounds magnitude). Q4 is particularly problematic: when $\hat{A} < 0$ and $r \gg 1$, GRPO's $\max(r, 1-\varepsilon)$ provides no protection, enabling unbounded policy updates.

amplifying many irrelevant or suboptimal alternatives (Gao et al., 2025).

Q2 ($\hat{A} > 0,\ r < 1$): This quadrant also represents a blind spot in GRPO's clipping mechanism, but its impact is bounded. Since $r \in (0, 1)$ in this case, the magnitude of the update is inherently limited, even if the direction is incorrect (e.g., discouraging a correct token). Consequently, while Q2 may introduce bias, it cannot lead to the wrong, unbounded updates observed in Q4.

5.2 Adaptive-Boundary Clipping GRPO

Motivated by the above observation, we propose Adaptive-Boundary Clipping GRPO (ABC-GRPO), which introduces four unconditional clipping boundaries, one for each quadrant (Fig. 3). This allows independent clipping thresholds for all four scenarios:

$$\tilde{r}_{i,t} = \begin{cases} \text{clip}(r_{i,t}, 1-\varepsilon_2, 1+\varepsilon_1), & \text{if } \hat{A}_i > 0 \\ \text{clip}(r_{i,t}, 1-\varepsilon_4, 1+\varepsilon_3), & \text{if } \hat{A}_i \leq 0 \end{cases} \tag{4}$$

Here, ε_1 and ε_2 control the upper and lower clipping bounds for positive advantages, while ε_3 and ε_4 govern those for negative advantages. This design provides four independent parameters, three more than standard GRPO, enabling fine-grained control over policy updates in each quadrant.

Our experiments show that, although this more flexible clipping mechanism does not resolve the fundamental granularity limitation of sequence-level advantages, it effectively mitigates the adverse effects of incorrect updates through smarter, quadrant-aware clipping. We note that GSPO (Zheng et al., 2025) takes a complementary approach by extending clipping to the sequence level (see Sect. 2 for a detailed comparison).

This yields the ABC-GRPO objective:

$$\mathcal{J}_{\text{ABC-GRPO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}\sim\pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \tilde{r}_{i,t}(\theta)\, \hat{A}_i \right], \tag{5}$$

where $r_{i,t} = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}$ is the token-level importance ratio and $\hat{A}_i$ is the sequence-level advantage from GRPO.

Key insight. By clipping the probability ratio $r_{i,t}$ before multiplying by the advantage $\hat{A}_i$, the update magnitude remains bounded in all four quadrants, regardless of the sign of the advantage. Notably, GRPO's original two conditional boundaries emerge as a special case of ABC-GRPO: the unbounded clipping in quadrant Q4 (where $\hat{A}_i \leq 0$ and $r_{i,t} > 1$) corresponds to setting $\varepsilon_3 \to \infty$ (or effectively, no upper bound).

Table 2 summarizes the algorithm differences:

Table 2: Architectural comparison: GRPO's 2 conditional boundaries vs ABC-GRPO's 4 unconditional boundaries.

Property	GRPO	ABC-GRPO
Number of boundaries	2 (conditional)	4 (unconditional)
Sign-conditional blind spots	Yes (Q2, Q4)	No
Number of parameters	1 (ε)	4 (ε_1–ε_4)
Bounded in all quadrants	No (Q2, Q4 unbounded)	Yes

5.3 Gradient Analysis

In ABC-GRPO, the policy is updated by maximizing the clipped surrogate objective:

$$\mathcal{L}^{\text{ABC-GRPO}} = \mathbb{E}\left[\tilde{r}_{i,t} \cdot \hat{A}_i\right], \tag{6}$$

where $\tilde{r}_{i,t}$ is the adaptively clipped probability ratio defined as:

$$\tilde{r}_{i,t} = \begin{cases} \text{clip}(r_{i,t}, 1-\varepsilon_2, 1+\varepsilon_1), & \text{if } \hat{A}_i > 0, \\ \text{clip}(r_{i,t}, 1-\varepsilon_4, 1+\varepsilon_3), & \text{if } \hat{A}_i \leq 0, \end{cases} \tag{7}$$

and $r_{i,t} = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}$ is the probability ratio between the current and old policies.

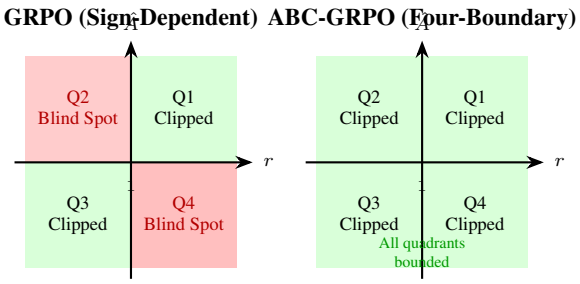

Figure 3: Closing GRPO's blind spots. Left: GRPO's inherited sign-dependent clipping leaves two quadrants unprotected (red). Right: ABC-GRPO applies unconditional clipping with four independent parameters $(\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4)$, closing all blind spots. For $\hat{A} > 0$, ε_1 and ε_2 control upper and lower bounds; for $\hat{A} < 0$, ε_3 and ε_4 provide the bounds.

The gradient with respect to the policy parameters θ is computed via automatic differentiation:

$$\nabla_\theta \mathcal{L}^{\text{ABC-GRPO}} = \mathbb{E}\left[\nabla_\theta \tilde{r}_{i,t} \cdot \hat{A}_i\right]. \tag{8}$$

Because the clipping operation is piecewise constant at its boundaries, gradients are propagated only when the ratio $r_{i,t}$ lies within the active clipping interval. Specifically:

- If $r_{i,t}$ is not clipped (i.e., it falls within $[1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}]$ for the corresponding advantage sign), then $\nabla_\theta \tilde{r}_{i,t} = \nabla_\theta r_{i,t}$, and the full gradient flows through.
- If $r_{i,t}$ is clipped to a boundary (e.g., $r_{i,t} > 1+\varepsilon_1$ or $r_{i,t} < 1-\varepsilon_4$), then $\tilde{r}_{i,t}$ becomes constant with respect to θ, and the gradient is zero at that point.

This behavior ensures that policy updates are restrained in all four quadrants of the $(r_{i,t}, \hat{A}_i)$ space, preventing excessively large or destabilizing gradient steps—particularly in Quadrant 4 ($\hat{A}_i \le 0$, $r_{i,t} > 1$), which is often left unbounded in standard GRPO. By introducing independent clipping thresholds $\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4$, ABC-GRPO enables finer control over the magnitude and direction of updates while preserving differentiability in the non-clipped regions.

5.4 Theoretical Guarantees

To analyze the boundedness of gradients, we consider the per-token gradient contribution. From Eq. (8), the overall gradient is the expectation over per-token gradients:

$$\nabla_\theta \mathcal{L}^{\text{ABC-GRPO}} = \mathbb{E}\left[\nabla_\theta \mathcal{L}_t^{\text{ABC}}\right], \tag{9}$$

where $\nabla_\theta \mathcal{L}_t^{\text{ABC}} = \hat{A}_i \cdot \nabla_\theta \tilde{r}_{i,t}$.

Because the clipping operator is piecewise constant, its derivative is:

$$\nabla_\theta \tilde{r}_{i,t} = \begin{cases} \nabla_\theta r_{i,t}, & r_{i,t} \in (1-\varepsilon_{\text{low}}, 1+\varepsilon_{\text{high}}), \\ 0, & \text{otherwise}, \end{cases} \tag{10}$$

where $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}})$ depends on the sign of $\hat{A}_i$.

Substituting Eq. (10) into the per-token gradient, we obtain:

$$\nabla_\theta \mathcal{L}_t^{\text{ABC}} = \begin{cases} \hat{A}_i \cdot \nabla_\theta r_{i,t}, & r_{i,t} \in (1-\varepsilon_l, 1+\varepsilon_h), \\ 0, & \text{otherwise}, \end{cases} \tag{11}$$

where $(\varepsilon_l, \varepsilon_h)$ denotes the active clipping interval. Using the log-derivative identity (where $\pi_{\theta_{\text{old}}}$ is constant w.r.t. θ):

$$\begin{aligned} \nabla_\theta r_{i,t} &= \nabla_\theta \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})} \\ &= r_{i,t} \cdot \nabla_\theta \log \pi_\theta(y_{i,t}|x, y_{i,<t}), \end{aligned} \tag{12}$$

we have:

$$\nabla_\theta \mathcal{L}_t^{\text{ABC}} = \begin{cases} \hat{A}_i \cdot r_{i,t} \cdot \nabla_\theta \log \pi_\theta, & \text{if unclipped}, \\ 0, & \text{if clipped}. \end{cases} \tag{13}$$

Boundedness Argument. Assume:

1. The advantage $\hat{A}_i$ is bounded: $|\hat{A}_i| \le A_{\max} < \infty$. (This holds in practice, as rewards are typically normalized or capped.)
2. The clipped ratio satisfies $\tilde{r}_{i,t} \in [1-\varepsilon_{\min}, 1+\varepsilon_{\max}]$, where
$$\varepsilon_{\min} = \min(\varepsilon_2, \varepsilon_4), \quad \varepsilon_{\max} = \max(\varepsilon_1, \varepsilon_3). \tag{14}$$
Hence, when unclipped, $r_{i,t} \le 1+\varepsilon_{\max}$.
3. The policy is smooth and differentiable (e.g., a softmax-parameterized neural network). While $\|\nabla_\theta \log \pi_\theta\|$ is not globally bounded in theory, in any finite-precision training run with bounded inputs and weights, the gradient norm is effectively bounded. Denote this practical bound as $G_{\max}$.

Then, for any token t in sequence i:

$$\begin{aligned} &\|\nabla_\theta \mathcal{L}_t^{\text{ABC}}\| \le |\hat{A}_i| \cdot r_{i,t} \cdot \|\nabla_\theta \log \pi_\theta\| \\ &\le A_{\max} \cdot (1+\varepsilon_{\max}) \cdot G_{\max} \quad \text{(when unclipped)}, \end{aligned} \tag{15}$$

and zero otherwise.

Thus, the per-token gradient in ABC-GRPO is uniformly bounded:

$$\|\nabla_\theta \mathcal{L}_t^{\mathrm{ABC}}\| \le C \triangleq A_{\max}(1+\varepsilon_{\max})G_{\max}. \quad (16)$$

The full implementation pseudocode is given in Appendix A.

6 Experiments

6.1 Experimental Setup

We train Qwen3-1.7B-Base and Qwen3-4B-Base (Qwen Team, 2025) on DAPO-Math-17k-Processed (Yu et al., 2025), sampling $G = 8$ responses per prompt. ABC-GRPO uses $\varepsilon_1{=}\varepsilon_2{=}\varepsilon_3{=}\varepsilon_4{=}0.2$ (selected by hyperparameter search; see Appendix B); baselines are GRPO ($\varepsilon{=}0.2$), DAPO with clip-higher, and GSPO ($\varepsilon{=}4{\times}10^{-4}$), all sharing data and optimizer settings apart from their clipping mechanism. We evaluate on AIME 2024, AIME 2025 (H4, 2025; OpenCompass, 2025), and AMC 2023 (math ai, 2025), sampling 64 responses per problem and reporting Avg@64 and Pass@64 (Chen et al., 2021). For each method we report numbers from the checkpoint with the best Avg@64, using the same checkpoint for the corresponding Pass@64.

6.2 Main Results

Table 3 compares ABC-GRPO against three baselines—GRPO, DAPO (Yu et al., 2025), and GSPO (Zheng et al., 2025)—across two model scales. Several patterns emerge from the results:

Gains increase with k. On Qwen3-4B AIME24, gains over GRPO grow from $+5.6$ at Avg@64 to $+23.4$ at Pass@64; Fig. 4(b,c) shows ABC-GRPO maintains monotonic Pass@32 and Pass@64 throughout training while GRPO degrades. The gap widens with k, indicating ABC-GRPO preserves multiple solution paths rather than collapsing to a narrow distribution.

Robust across benchmarks and scales. ABC-GRPO leads on Avg@64 at both scales (20.7 at 1.7B, 38.3 at 4B; $+1.7$ and $+1.8$ over the next-best baseline), with gains spread across the challenging AIME problems and the easier AMC23.

Addressing the reasoning-boundary narrowing. Comparing Pass@64 against the *base model* (rather than GRPO) reveals a known pathology: GRPO's Pass@64 falls below base on Qwen3-1.7B (49.7 vs 53.3, -3.6) and Qwen3-4B (59.4 vs 61.1, -1.7), with AIME24 dropping -6.7 (43.3 vs 50.0)—confirming the *reasoning-boundary narrowing* reported by Yue et al. (2025). DAPO and GSPO also recover the boundary, but ABC-GRPO uniquely achieves the best joint accuracy–diversity tradeoff at both scales: at 4B, the highest Pass@64 (70.3) co-occurs with the highest Avg@64 (38.3); at 1.7B, it ties DAPO on Pass@64 (55.3) while leading on Avg@64 (20.7 vs 18.1).

6.3 Ablation Study: Quadrant Importance

To directly validate our central claim that Q4 is the critical blind spot while Q2 is mild (Sect. 4), we ablate each unprotected quadrant in isolation on Qwen3-4B. "Qx-only" denotes clipping in Q1, Q3, and Qx while leaving the other previously unprotected quadrant unbounded; the four-quadrant ABC-GRPO and vanilla GRPO are reported as the upper and lower references. Table 4 reports the results.

Q4 protection recovers most of ABC-GRPO's gains. Q4-only achieves average Pass@64 of 67.2 ($+7.8$ over GRPO), recovering roughly $7.8/10.9 \approx 72\%$ of the full four-quadrant ABC-GRPO's gain (70.3, $+10.9$). The ratio of Q4-only's improvement to Q2-only's ($+7.8$ vs $+2.5$ on Pass@64) quantitatively confirms that Q4 is the dominant unbounded quadrant, contributing roughly $3\times$ more than Q2 to the headline improvement. The AIME24 Pass@64 sequence is particularly diagnostic: Base $\rightarrow$ GRPO $\rightarrow$ Q2-only $\rightarrow$ Q4-only $\rightarrow$ ABC-GRPO yields 50.0, 43.3, 50.0, 63.3, 66.7—vanilla GRPO degrades the reasoning boundary below the base model, Q2-only merely recovers to the base level, Q4-only crosses the base by $+13.3$, and the full ABC-GRPO adds another $+3.4$ on top.

Q2 protection requires Q4 to be bounded first. Q2-only's Avg@64 (32.6) falls -1.9 below vanilla GRPO. This is fully consistent with our framework: when Q4 remains unbounded, its suppression of correct tokens inside failed sequences persists as the dominant pathology, and adding Q2 protection in isolation does not address it. Only after Q4 is bounded does Q2 clipping provide useful refinement—adding Q2 protection to Q4-only yields a further $+2.6$ Avg@64 (23.3 $\rightarrow$ 25.9) and $+3.4$ Pass@64 (63.3 $\rightarrow$ 66.7) on AIME24, lifting performance to its highest level.

Justifying the four-parameter design. Q4 protection consistently improves or matches GRPO across all three benchmarks (strict gains on AIME24 and AIME25; tied AMC23 Pass@64

Table 3: Comparison of clipping methods across two model scales. Avg@64 measures sampling accuracy; Pass@64 measures reasoning diversity (the reasoning boundary). ΔBase shows change relative to the pre-trained base model. Best results in **bold**.

Scale	Method	Avg@64					Pass@64				
		AIME24	AIME25	AMC23	Avg	ΔBase	AIME24	AIME25	AMC23	Avg	ΔBase
Qwen3-1.7B	Base	3.4	3.2	24.5	10.4	–	30.0	40.0	90.0	53.3	–
	GRPO	12.3	5.4	34.7	17.5	+7.1	30.0	26.7	**92.5**	49.7	−3.6
	DAPO	9.3	6.2	38.9	18.1	+7.7	**36.7**	36.7	**92.5**	**55.3**	**+2.0**
	GSPO	9.3	6.5	**41.3**	19.0	+8.6	30.0	**40.0**	**92.5**	54.2	+0.9
	ABC-GRPO	**14.4**	**8.2**	39.5	**20.7**	**+10.3**	**36.7**	36.7	**92.5**	**55.3**	**+2.0**
Qwen3-4B	Base	8.1	5.9	37.9	17.3	–	50.0	43.3	90.0	61.1	–
	GRPO	20.3	20.0	63.1	34.5	+17.2	43.3	40.0	95.0	59.4	−1.7
	DAPO	22.6	**21.0**	65.9	36.5	+19.2	56.7	**56.7**	95.0	69.5	+8.4
	GSPO	20.7	**21.0**	64.9	35.5	+18.2	53.3	43.3	95.0	63.9	+2.8
	ABC-GRPO	**25.9**	20.6	**68.4**	**38.3**	**+21.0**	**66.7**	46.7	**97.5**	**70.3**	**+9.2**

Table 4: Ablation on Qwen3-4B: contribution of each unprotected quadrant. "Qx-only" clips in Q1, Q3, and Qx; the other previously unprotected quadrant remains unbounded. ΔGRPO is the change relative to vanilla GRPO. Best result per column in **bold**.

Method	Avg@64					Pass@64				
	AIME24	AIME25	AMC23	Avg	ΔGRPO	AIME24	AIME25	AMC23	Avg	ΔGRPO
Base	8.1	5.9	37.9	17.3	–	50.0	43.3	90.0	61.1	–
GRPO	20.3	20.0	63.1	34.5	–	43.3	40.0	95.0	59.4	–
Q2-only	21.2	16.4	60.3	32.6	−1.9	50.0	43.3	92.5	61.9	+2.5
Q4-only	23.3	**22.5**	67.3	37.7	+3.2	63.3	43.3	95.0	67.2	+7.8
ABC-GRPO	**25.9**	20.6	**68.4**	**38.3**	**+3.8**	**66.7**	**46.7**	**97.5**	**70.3**	**+10.9**

with a $+4.2$ Avg@64 gain), while Q2 protection plays a smaller, complementary role whose effect varies modestly by metric and task: on AIME25, for instance, adding Q2 to Q4-only improves Pass@64 by $+3.4$ but slightly reduces Avg@64 by -1.9. This metric- and task-dependent behavior is precisely why ABC-GRPO introduces four independent thresholds rather than a single shared ε: each quadrant plays a distinct role in entropy preservation, and decoupling the bounds allows separate calibration of Q4's dominant effect and Q2's complementary refinement per domain.

6.4 Training Dynamics and Diagnostic Analysis

Fig. 4 traces the training-time dynamics underlying the static comparisons in Sect. 6.2 and the causal ablation in Sect. 6.3, providing per-step evidence that ABC-GRPO addresses GRPO's degradation.

As shown in panels (a-c), ABC-GRPO maintains monotonic improvement across Avg@64, Pass@64, and Pass@32 while GRPO degrades. Panel (d) shows ABC-GRPO preserves substantially higher entropy throughout training: the steady-state entropy (dashed reference lines) is ≈ 0.39 nats for ABC-GRPO versus ≈ 0.036 nats for GRPO, a ratio of $\approx 10.9\times$. To validate our four-quadrant analysis, panel (f) tracks clipping events during training—Q4 accounts for 41.4% of all clipping events, confirming it as the dominant failure mode in GRPO's mechanism.

6.5 Clipping Behavior Analysis

As shown in Fig. 4(f), we observe a pronounced asymmetry in how GRPO's clipping events distribute across the four quadrants of the $(r, \hat{A})$ space. Q4 ($\hat{A} < 0, r > 1$) accounts for 41.4% of all clipping events—the single largest category—while Q3 accounts for 22.9%, Q1 for 16.8%, and Q2 for 7.3%. Notably, Q2 and Q4—the two quadrants left unprotected by GRPO—together comprise 48.7% of all clipping events. The resulting Q4:Q2 frequency ratio ($\approx 5.7\times$) is of the same order as the causal contribution ratio ($\approx 3.1\times$ on Pass@64) from the ablation in Sect. 6.3; descriptive and causal evidence converge on Q4 as the dominant blind spot predicted by our analysis in

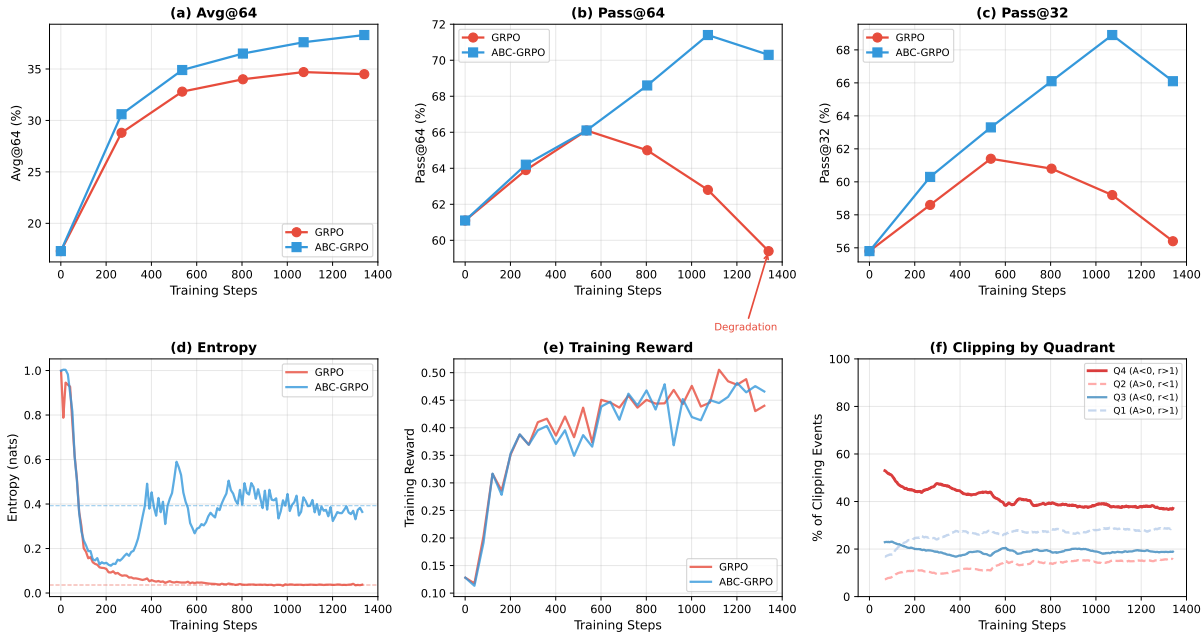

Figure 4: Training dynamics for GRPO vs ABC-GRPO (Qwen3-4B). Top row shows performance metrics: (a) Avg@64 progression, (b) Pass@64 evolution showing GRPO degradation, (c) Pass@32 curves. Bottom row shows diagnostic metrics: (d) Entropy preservation, (e) Training reward curves, (f) Clipping distribution by quadrant, validating Q4 as the critical blind spot (41.4% of events).

Sect. 4.

Combined with the causal evidence from Sect. 6.3, this frequency pattern explains why bounding Q4 alone recovers most of ABC-GRPO's gains: the sequence-level/token-level mismatch detailed in Sect. 4 produces a continual stream of high-ratio correct tokens inside failed sequences, and GRPO's $\max(r, 1-\varepsilon)$ leaves precisely this stream unbounded. ABC-GRPO's unconditional clipping addresses this by ensuring $\tilde{r}_{i,t} \in [1-\varepsilon_4, 1+\varepsilon_3]$ even in Q4, eliminating the entropy collapse visible in Fig. 4(d).

7 Conclusion

We propose ABC-GRPO, a refined reinforcement learning algorithm that addresses fundamental limitations in GRPO's clipping mechanism. Through systematic four-quadrant analysis of the $(r, \hat{A})$ space, we identify two critical blind spots (Q2 and Q4) left unprotected by GRPO's asymmetric, conditional clipping, with Q4 alone accounting for 41.4% of all clipping events and, under causal ablation, recovering roughly 72% of ABC-GRPO's Pass@64 gain. Combined with sequence-level advantage estimation, this asymmetry leads to unbounded gradient updates that suppress correct reasoning steps and cause entropy collapse.

ABC-GRPO introduces four independent clipping parameters that provide unconditional bounds across all quadrants, with theoretical guarantees of bounded gradient magnitudes. Empirical evaluation on Qwen3 models shows ABC-GRPO maintains entropy $10.9\times$ higher than GRPO while achieving superior Pass@k metrics across benchmarks.

ABC-GRPO shows that careful analysis of policy optimization yields principled improvements, providing a robust foundation for stable, high-performance RLHF training.

Limitations

Our work has several limitations that we wish to acknowledge openly. First, ABC-GRPO does not resolve the coarse-grained credit assignment problem inherent to GRPO's value-free, sequence-level advantage estimation: correct reasoning steps inside failed sequences still receive uniformly negative signals, and our unconditional clipping only bounds the resulting gradient magnitudes rather than fixing the underlying attribution issue. Second, although we identified the uniform setting ($\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 = 0.2$) as effective through hyperparameter search, we do not provide a principled method for choosing the clipping thresholds across different model families, task domains, or training regimes; our recommended con-

figuration remains empirical rather than theoretically derived. Third, our empirical validation is limited in scope: we evaluate only on mathematical reasoning benchmarks (AIME 2024, AIME 2025, AMC 2023) with two model scales (1.7B and 4B parameters), and we therefore cannot make claims about generalization to other reasoning domains (e.g., code generation, commonsense reasoning) or larger model scales (7B+) without further evidence. Fourth, while our analysis isolates clipping behavior, we do not control for potential interactions with other GRPO design choices (e.g., group size, reward normalization, KL regularization strength), so the magnitude of the reported improvements may shift under different training recipes.

Ethical Considerations

This work focuses on improving the clipping mechanism of reinforcement learning algorithms for LLM training. Our method uses publicly available models and mathematical benchmarks, involves no human subjects or private data, and does not introduce new risks beyond those inherent to large language model training.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 12248–12267.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 30.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tri Dao. 2023. FlashAttention-2: Faster attention with better parallelism and work partitioning. *arXiv preprint arXiv:2307.08691*.

Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. 2025. Soft adaptive policy optimization. *Preprint*, arXiv:2511.20347.

Leo Gao, John Schulman, and Jacob Hilton. 2023. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning (ICML)*.

Daya Guo, Dejian Yang, He Zhang, Junxiao Song, Runxin Zhang, Ruoyu Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Nature*, 645:633–638.

Hugging Face H4. 2025. AIME 2024 Dataset (Hugging Face). https://huggingface.co/datasets/HuggingFaceH4/aime_2024. Accessed: 2025-09-18.

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 *International Conference on Machine Learning (ICML)*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 34.

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. *Preprint*, arXiv:2503.20783.

math ai. 2025. AMC23 Dataset (Hugging Face). https://huggingface.co/datasets/math-ai/amc23. Accessed: 2025-09-18.

OpenCompass. 2025. AIME 2025 Dataset (Hugging Face). https://huggingface.co/datasets/opencompass/AIME2025. Accessed: 2025-09-18.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35, pages 27730–27744.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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 (NeurIPS)*, volume 36.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. 2015. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, pages 1889–1897.

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

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. *arXiv preprint arXiv:2402.03300*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 35.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *Advances in Neural Information Processing Systems (NeurIPS)*, volume 38.

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. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.

A Algorithm Pseudocode

Algorithm 1 Adaptive-Boundary-Clipping GRPO (ABC-GRPO)

Input: Prompt x, old policy π_{old}, current policy π_θ, clip thresholds $\{\varepsilon_1, \varepsilon_2, \varepsilon_3, \varepsilon_4\}$
Sample group of responses: $\{y_1, \ldots, y_G\} \sim \pi_{\text{old}}(\cdot|x)$
Compute rewards: $r_i = r(x, y_i)$ for $i = 1, \ldots, G$
Compute advantages: $\hat{A}_i = r_i - \frac{1}{G}\sum_j r_j$
for each sequence y_i **do**
  for each token t in y_i **do**
    Compute importance ratio: $r_{i,t} = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\text{old}}(y_{i,t}|x, y_{i,<t})}$
    4-BOUNDARY CLIP: $\tilde{r}_{i,t} \leftarrow \begin{cases} \text{clip}(r_{i,t}, 1-\varepsilon_2, 1+\varepsilon_1), & \hat{A}_i > 0 \\ \text{clip}(r_{i,t}, 1-\varepsilon_4, 1+\varepsilon_3), & \hat{A}_i \le 0 \end{cases}$
    Compute loss: $\ell_t = -\tilde{r}_{i,t}\hat{A}_i$
  end for
end for
Accumulate losses and compute gradients

B Experimental Setup Details

Infrastructure and framework. All training runs use a single 8-GPU node with DeepSpeed ZeRO-3, mixed-precision `bfloat16`, and FlashAttention-2 (Dao, 2023). The trainer is a TRL-based GRPO implementation modified only to swap GRPO's clipping rule for the method-specific clipping under comparison; all other code paths, data loaders, reward functions, and logging are shared across GRPO, DAPO, GSPO, and ABC-GRPO.

Data and prompt template. We train on the open-r1/DAPO-Math-17k-Processed dataset (Yu et al., 2025) for 5 epochs with seed 42. Each prompt is wrapped with the system instruction *"Please reason step by step, and put your final answer within* `\boxed{}`*"*, matching the format used at evaluation time. The reward function is a single binary correctness signal (`accuracy`) with group-wise reward scaling enabled (`scale_rewards=True`); no format or process reward is added.

Optimization. We use AdamW with peak learning rate 1×10^{-6}, cosine schedule, and warmup ratio 0.1. Each device processes a per-device train batch of 4 prompts with gradient accumulation 16, gradient checkpointing, and `steps_per_generation`=64, so each rollout batch is reused for 64/16=4 inner optimizer passes before fresh sampling. We do not override the trainer's default KL coefficient; the same value is used for every method, isolating the clipping mechanism as the only intervention. Truncated completions are kept (`mask_truncated_completions=false`), and token-level importance sampling is used throughout (`importance_sampling_level=token`);

for GSPO we additionally enable sequence-level aggregation as prescribed in the original paper (Zheng et al., 2025).

Rollouts and context length. We sample $G{=}8$ responses per prompt via vLLM (colocated, 40% GPU memory budget) with temperature 1.0, top-$p{=}1.0$, and top-k disabled. Context budgets differ slightly by model scale: Qwen3-1.7B uses (prompt $\leq$ 2048, completion $\leq$ 4096) and Qwen3-4B uses (prompt $\leq$ 1024, completion $\leq$ 3072); both budgets are shared across all baselines at the same scale. Method-specific clipping thresholds are: GRPO $\varepsilon{=}0.2$; DAPO $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}}){=}(0.2, 0.28)$ following the clip-higher recipe; GSPO $\varepsilon{=}4{\times}10^{-4}$ at the sequence level; ABC-GRPO $\varepsilon_1{=}\varepsilon_2{=}\varepsilon_3{=}\varepsilon_4{=}0.2$.

Hyperparameter search for ABC-GRPO. We searched over both asymmetric configurations (e.g., $\varepsilon_3 < \varepsilon_1$ to more aggressively bound Q4) and uniform settings $\varepsilon_1 = \varepsilon_2 = \varepsilon_3 = \varepsilon_4 \in \{0.1, 0.2, 0.3\}$. The uniform setting at 0.2 matches GRPO's standard threshold and gave the strongest Pass@64 across both model scales; asymmetric configurations did not consistently outperform it within our search budget.

Evaluation protocol. Checkpoints are saved every epoch (4B) or every 100 optimizer steps (1.7B) and evaluated with vLLM following the standard Qwen3-math decoding convention (Qwen Team, 2025): temperature 0.6, top-$p{=}0.95$, top-$k{=}20$, min-$p{=}0$, and decoding context length 4096. For each problem we draw $N{=}64$ independent completions and apply the `qwen25-math-cot` answer-extraction template; Avg@64 is the mean accuracy over the 64 samples and Pass@k is the standard unbiased estimator of $\Pr[\text{any of } k \text{ samples is correct}]$ (Chen et al., 2021). Checkpoint selection follows the rule stated in Sect. 6.1.

Benchmark details. AIME 2024 and AIME 2025 each contain 30 problems; AMC 2023 contains 40 problems. These complement established benchmarks such as GSM8K (Cobbe et al., 2021) and MATH (Hendrycks et al., 2021) by targeting competition-level difficulty. Pass@k is computed over $k \in \{2, 4, 8, 16, 32, 64\}$; the main paper reports $k{=}64$ to track reasoning-boundary preservation, with the full k sweep available in our anonymous code release.