WHEN CLIPPING BECOMES A LEARNING SIGNAL: A PERMUTATION BASELINE FOR POLICY OPTIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has improved mathematical reasoning in large language models. Methods such as Group Relative Policy Optimization (GRPO) use clipping to stabilize training by discouraging large policy changes. However, we identify a component of the clipped objective whose gradient acts as an additional learning signal outside the clipping interval and can counteract reward-guided updates. This component equals the average clipped objective over all within-group reward permutations, which we call the permutation baseline. In this work, we propose **Per**mutation-**C**orrected **P**olicy **O**ptimization (**PerCPO**), which removes this source of interference with reward-guided updates by subtracting this baseline from the original clipped objective in closed form. For binary rewards, the resulting updates follow the direction indicated by rewards and are naturally attenuated outside the clipping interval, restoring gradients suppressed by clipping. We further show that the PerCPO objective increases along a path that improves accuracy with minimal Kullback–Leibler divergence from the sampling policy, even where the standard clipped objective plateaus. Extensive experiments on mathematical reasoning benchmarks demonstrate the effectiveness of PerCPO. For instance, on Qwen2.5-Math-7B, PerCPO outperforms GRPO by 3.47% in average accuracy across seven benchmarks.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has driven recent progress in the reasoning capabilities of large language models (DeepSeek-AI, 2025). For tasks such as mathematical problem solving, answer correctness can be checked automatically to provide rewards for training. Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is widely used for this purpose. It estimates advantages by comparing the rewards of responses to the same prompt, without requiring a separate critic.

To reduce rollout costs, GRPO commonly reuses each batch of generated responses for multiple policy updates. Repeated updates can move the current policy away from the policy that generated the responses. GRPO adopts the clipped objective of Proximal Policy Optimization (PPO) (Schulman et al., 2017) to stabilize training by discouraging large policy changes. The clipping rule uses the ratio of a token's probability under the current policy to that under the sampling policy. For a token with positive advantage, clipping sets its gradient contribution to zero once this ratio exceeds the upper threshold, even though the advantage still favors increasing its probability. This can limit how much the policy learns from each batch.

Recent methods address this limitation by changing how clipping controls policy updates. DAPO raises the upper clipping threshold to promote exploration (Yu et al., 2025). CISPO clips importance-sampling weights while retaining token gradients (MiniMax, 2025), and CE-GPPO restores clipped gradients and adjusts their strength to regulate policy entropy (Su et al., 2026). However, these studies do not explicitly examine whether clipping itself can introduce a learning signal beyond the verifier's feedback.

To the best of our knowledge, we are the first to derive an exact decomposition of the standard clipped objective and explicitly identify a component whose gradient acts as an additional learning signal. Even with only this component contributing to the expected gradient of the clipped objective, random-reward training can substantially alter model behavior and performance (Shao et al., 2025).

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.

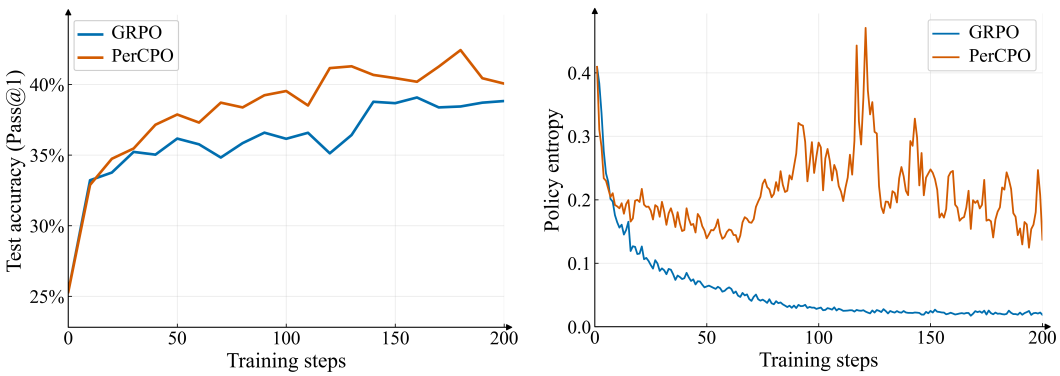


Figure 1: Training dynamics for Qwen2.5-Math-7B under GRPO and PerCPO: average Pass@1 across seven mathematical reasoning benchmarks (left) and policy entropy (right).

In standard GRPO, this signal does not reflect the verifier's assessment of individual responses and can oppose reward-guided updates. To resolve this conflict between clipping and reward feedback, we propose *Permutation-Corrected Policy Optimization* (PerCPO), which subtracts this component from the standard clipped objective. The component is exactly the average clipped objective over all reassignments of a group's rewards to its responses. We call this average the *permutation baseline* and compute it in closed form. On Qwen2.5-Math-7B, this correction yields a 3.47% absolute improvement over GRPO in average accuracy across the seven benchmarks and mitigates the entropy collapse observed with GRPO (Figure 1).

For binary rewards, this subtraction restores the gradients suppressed by clipping and reduces those that clipping retains beyond the bounds. The proportion of correct responses in the group determines how much of the unclipped gradient remains, while its direction continues to follow the advantage. In an idealized analysis, we further show that the standard clipped objective can remain unchanged as accuracy improves. This can happen even when each improvement is achieved with the smallest possible change to the sampling policy, measured by Kullback–Leibler (KL) divergence (Kullback & Leibler, 1951). PerCPO continues to assign higher objective values to these more accurate policies.

In summary, our main contributions are:

- We identify a component of the standard clipped objective whose gradient acts as an additional learning signal introduced by clipping and can oppose reward-guided updates. This signal remains unchanged under within-group reward reassignment.
- We derive a closed-form permutation baseline and propose PerCPO, which subtracts it from the standard clipped objective to remove this extra signal and better align policy optimization with reward feedback.
- We conduct extensive experiments on three language models across seven mathematical reasoning benchmarks. The results demonstrate the effectiveness of PerCPO, which consistently outperforms GRPO and alternative clipping strategies in average accuracy.

2 Related Work

Reinforcement Learning for LLM Reasoning. Reinforcement learning has become an important approach to LLM post-training, from alignment with human preferences (Stiennon et al., 2020; Ouyang et al., 2022) to reasoning with verifiable outcome rewards (Lambert et al., 2025; DeepSeek-AI, 2025; Kimi Team, 2025). For tasks such as mathematics and code generation, automatically checking final outputs provides scalable feedback without requiring annotations of individual reasoning steps. PPO commonly combines a clipped surrogate with a learned value function (Schulman et al., 2017; Hu et al., 2025), while GRPO and RLOO construct baselines from multiple responses to the same prompt without a separate critic (Shao et al., 2024; Ahmadian et al., 2024). Subsequent work explores credit assignment through Monte Carlo estimation or improved value-based training (Kazemnejad et al., 2025; Yue et al., 2025), and persistent prompt-level baselines for single-sample

learning (Xu & Ding, 2026). Other work revisits the weighting and aggregation of training signals. Dr. GRPO removes normalization by response length and within-group reward standard deviation to address length and difficulty biases (Liu et al., 2025), while GMPO uses geometric aggregation to reduce sensitivity to token-level outliers (Zhao et al., 2026). To reduce training costs, CPPO prunes completions with small absolute advantages before policy updates (Lin et al., 2025), while GRESO uses reward histories to skip likely uninformative prompts before rollout (Zheng et al., 2025b). DPPO combines pruning at both levels with importance weighting, deriving unbiased gradient estimates for the GRPO surrogate (Zhu et al., 2026).

Clipping in Policy Optimization. While RLVR methods differ in advantage estimation and data selection, many rely on clipping to control policy updates (Schulman et al., 2017). Research on clipping has examined adaptive bounds (Wang et al., 2019), the granularity of importance ratios, and gradient preservation. DAPO increases the upper clipping threshold to promote exploration (Yu et al., 2025), while BAPO adapts the bounds to balance positive and negative contributions to the objective (Xi et al., 2026). GSPO instead applies clipping to a length-normalized sequence-level ratio (Zheng et al., 2025a). For highly stale rollouts, M2PO filters extreme deviations using a batch-level constraint on the second moment of log importance ratios (Zheng et al., 2026). To preserve learning signals beyond the clipping bounds, CISPO clips detached importance weights while retaining token gradients (MiniMax, 2025), and CE-GPPO restores and scales clipped gradients to regulate policy entropy (Su et al., 2026). To explain reasoning gains from random rewards in some models, Shao et al. (2025) analyze how GRPO's clipping bias can amplify behaviors already favored by the model. Chen et al. (2026) analyze the entropy effects of clipping and show that entropy reduction alone does not explain these gains. However, these studies do not provide an objective correction for clipping-induced signals that can oppose reward feedback. By isolating and subtracting the underlying component in closed form, PerCPO allows learning from binary rewards to continue beyond the clipping bounds while still scaling down the updates.

3 BACKGROUND

Group Relative Policy Optimization (GRPO; Shao et al., 2024) simplifies policy optimization by estimating advantages from the rewards of multiple responses to the same prompt, without training a separate critic. GRPO draws a prompt q from the training distribution $\mathcal{D}$ and samples G responses $\{y_1, \ldots, y_G\}$ using the behavior policy $\pi_{\text{old}}(\cdot \mid q)$. Given their rewards $\{R_1, \ldots, R_G\}$, the current policy π_θ parameterized by θ is optimized by maximizing

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \left\{\min\{\rho_{i,t}(\theta)A_i, \text{clip}(\rho_{i,t}(\theta), L, U)A_i\} - \beta\widehat{D}_{\text{KL},i,t}\right\}\right]. \tag{1}$$

The token probability ratio $\rho_{i,t}(\theta)$ and group-normalized advantage A_i are

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid q, y_{i,<t})}{\pi_{\text{old}}(y_{i,t} \mid q, y_{i,<t})}, \qquad A_i = \frac{R_i - \text{mean}(R_1, \ldots, R_G)}{\text{std}(R_1, \ldots, R_G) + \delta}, \tag{2}$$

where $\delta > 0$ is a small constant for numerical stability. The clipping bounds satisfy $0 < L < 1 < U$, with $L = 1 - \epsilon$ and $U = 1 + \epsilon$ in standard GRPO. The term $\widehat{D}_{\text{KL},i,t}$ penalizes divergence from a fixed reference policy. Following DAPO and CISPO (Yu et al., 2025; MiniMax, 2025), we set $\beta = 0$ in both our analysis and experiments.

GRPO assigns the same advantage to all tokens in a response, while applying PPO clipping to their probability ratios individually (Schulman et al., 2017). With π_{old} held fixed, the unclipped term $A_i\rho_{i,t}$ encourages π_θ to raise the sampled token's probability for positive advantages and lower it for negative advantages. Clipping makes this term flat above U for positive advantages and below L for negative advantages. A token in either region therefore contributes no gradient to the overall objective.

4 METHOD

In this section, we first analyze the standard clipped objective through an exact decomposition and within-group reward permutations, identifying a learning signal that can counteract verifier-guided updates. To this end, we then introduce **Permutation-Corrected Policy Optimization (PerCPO)**, which subtracts the corresponding permutation baseline in closed form, and explain how it restores and attenuates token updates under binary rewards. Finally, we show that the corrected objective continues to increase as accuracy improves along the minimum-KL path, even where the standard clipped objective plateaus.

4.1 CLIPPING AS A LEARNING SIGNAL

With $\beta = 0$, each term inside the token sum of Equation (1) is the clipped objective for an individual token. For token t in response y_i, we denote its clipped token objective by

$$\ell_{\text{clip}}(A_i, \rho_{i,t}(\theta)) = \min\{A_i\rho_{i,t}(\theta), A_i \,\text{clip}(\rho_{i,t}(\theta), L, U)\}\,. \tag{3}$$

The corresponding unclipped token objective is denoted by

$$\ell_{\text{unclip}}(A_i, \rho_{i,t}(\theta)) = A_i\rho_{i,t}(\theta). \tag{4}$$

For each sampled group, A_i is held fixed during policy optimization, whereas $\rho_{i,t}(\theta)$ varies with the current policy.

In GRPO, the learning signal supplied by the verifier depends on the pairing between each response y_i and its reward R_i. We remove this pairing by averaging over all within-group reward permutations to examine whether clipping leaves a systematic update when reward assignment no longer reflects response correctness. Let $\mathfrak{S}_G$ denote the set of permutations of $\{1, \ldots, G\}$, and draw σ uniformly from $\mathfrak{S}_G$. Under permutation σ, response y_i receives reward $R_{\sigma(i)}$. Since a permutation preserves the group reward statistics, $R_{\sigma(i)}$ yields the normalized advantage $A_{\sigma(i)}$. The probability ratio $\rho_{i,t}(\theta)$ is unchanged by the reward permutation.

For the unclipped objective, the integrand depends on σ only through $\sigma(i)$, which is uniformly distributed over $\{1, \ldots, G\}$. Hence,

$$\begin{aligned}&\mathbb{E}_{\sigma\sim\text{Uniform}(\mathfrak{S}_G)}\left[\ell_{\text{unclip}}(A_{\sigma(i)}, \rho_{i,t}(\theta))\right] \\ &\quad= \frac{1}{G}\sum_{j=1}^{G}\ell_{\text{unclip}}(A_j, \rho_{i,t}(\theta)) = \frac{1}{G}\sum_{j=1}^{G}A_j\rho_{i,t}(\theta) = \rho_{i,t}(\theta)\frac{1}{G}\sum_{j=1}^{G}A_j = 0.\end{aligned} \tag{5}$$

The last equality follows from the group centering of the advantages. Since policy updates are determined by objective gradients, differentiating Equation (5) with respect to θ gives

$$\nabla_\theta\mathbb{E}_{\sigma\sim\text{Uniform}(\mathfrak{S}_G)}\left[\ell_{\text{unclip}}(A_{\sigma(i)}, \rho_{i,t}(\theta))\right] = \nabla_\theta 0 = 0. \tag{6}$$

The unclipped objective therefore provides no learning signal after permutation averaging removes the association between assigned rewards and response correctness.

We next examine the clipped objective under the same reward permutations. Applying the identity $\min\{u, v\} = \frac{u+v-|u-v|}{2}$ to Equation (3) gives the exact decomposition

$$\ell_{\text{clip}}(A, \rho) = \underbrace{\frac{A}{2}\left[\rho + \text{clip}(\rho, L, U)\right]}_{\text{signed-advantage term}} - \underbrace{\frac{|A|}{2}\left|\rho - \text{clip}(\rho, L, U)\right|}_{\text{magnitude-weighted clipping term}}\,. \tag{7}$$

Like the unclipped objective, the signed-advantage term is linear in A. Because its probability-ratio factor is fixed across reward permutations and $\sum_{j=1}^{G} A_j = 0$, this term averages to zero. Unlike the signed advantages, the advantage magnitudes cannot cancel: $\sum_{j=1}^{G}|A_j| > 0$ whenever the group contains a nonzero advantage. Consequently, permutation averaging eliminates the signed-advantage term and leaves only the magnitude-weighted clipping term:

$$\begin{aligned}\mathbb{E}_{\sigma\sim\text{Uniform}(\mathfrak{S}_G)}\left[\ell_{\text{clip}}(A_{\sigma(i)}, \rho_{i,t}(\theta))\right] &= \frac{1}{G}\sum_{j=1}^{G}\ell_{\text{clip}}(A_j, \rho_{i,t}(\theta)) \\ &= -\frac{1}{2G}\sum_{j=1}^{G}|A_j|\left|\rho_{i,t}(\theta) - \text{clip}(\rho_{i,t}(\theta), L, U)\right|.\end{aligned} \tag{8}$$

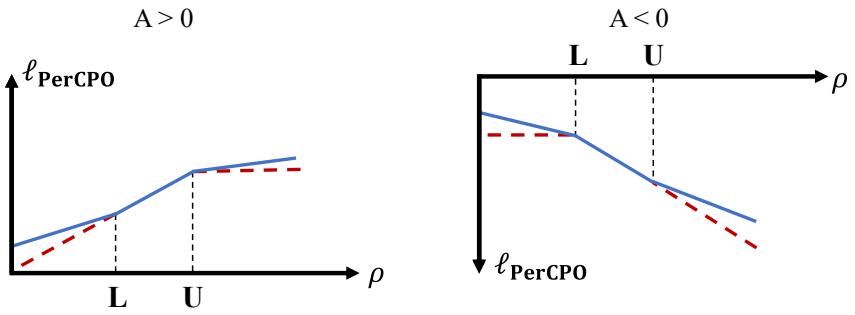


Figure 2: PerCPO token objectives (blue) under binary rewards for positive (left) and negative (right) advantages. Dashed red segments show the GRPO token objective outside $[L, U]$, while the two objectives coincide inside the interval. In both panels, the PerCPO slopes are $(1-p)A_i$ for $\rho < L$, A_i for $L < \rho < U$, and pA_i for $\rho > U$.

The resulting permutation mean is determined by the group statistic $\sum_{j=1}^{G} |A_j|$ and the token probability ratio, not by which reward in the group is assigned to the response. Let $\mathbf{A} = (A_1, \ldots, A_G)$ denote the group advantage vector. We denote the permutation mean by $B(\rho_{i,t}(\theta); \mathbf{A})$, where

$$B(\rho; \mathbf{A}) \equiv -\frac{1}{2G} \sum_{j=1}^{G} |A_j| \, |\rho - \text{clip}(\rho, L, U)| \, . \tag{9}$$

Because $B(\rho_{i,t}(\theta); \mathbf{A})$ is the permutation mean of the clipped token objective, the deviation of the permuted objective from this mean has zero expectation over reward permutations:

$$\mathbb{E}_{\sigma \sim \text{Uniform}(\mathfrak{S}_G)} \left[\ell_{\text{clip}}(A_{\sigma(i)}, \rho_{i,t}(\theta)) - B(\rho_{i,t}(\theta); \mathbf{A}) \right] = 0. \tag{10}$$

We therefore call B the permutation baseline. Equation (9) gives this baseline in closed form using the group advantages and the token probability ratio. Because the corresponding permutation mean of the unclipped objective is zero, $B(\rho_{i,t}(\theta); \mathbf{A})$ is precisely the component introduced by clipping under reward permutations.

Although $B(\rho_{i,t}(\theta); \mathbf{A})$ does not depend on the reward permutation σ, it depends on θ through $\rho_{i,t}(\theta)$. For $\rho_{i,t}(\theta) \notin \{L, U\}$, differentiating $B(\rho_{i,t}(\theta); \mathbf{A})$ with respect to θ gives

$$\nabla_\theta B(\rho_{i,t}(\theta); \mathbf{A}) = \begin{cases} \dfrac{1}{2G} \displaystyle\sum_{j=1}^{G} |A_j| \, \nabla_\theta \rho_{i,t}(\theta), & \rho_{i,t}(\theta) < L, \\ 0, & L < \rho_{i,t}(\theta) < U, \\ -\dfrac{1}{2G} \displaystyle\sum_{j=1}^{G} |A_j| \, \nabla_\theta \rho_{i,t}(\theta), & \rho_{i,t}(\theta) > U. \end{cases} \tag{11}$$

Thus, for a group containing responses with different rewards, a token inside the clipping interval contributes zero baseline gradient, whereas a token outside it contributes $\nabla_\theta \rho_{i,t}(\theta)$ scaled by a nonzero coefficient. The full objective applies the same token and response averages as GRPO to these terms. Consequently, a single token beyond either clipping bound is sufficient for the clipping-induced signal to enter the full objective, while tokens inside the interval contribute nothing to it.

Under the actual reward assignment, verifier feedback contributes $A_i \nabla_\theta \rho_{i,t}(\theta)$ to the token update. When $A_i > 0$ and $\rho_{i,t}(\theta) > U$, or when $A_i < 0$ and $\rho_{i,t}(\theta) < L$, the permutation-baseline gradient points in the opposite direction. These are precisely the two regimes where clipping suppresses the reward-supported update.

Thus, after permutation averaging removes the association between assigned rewards and response correctness, the clipped objective retains the policy-dependent component B. The gradient of this component is the learning signal induced by clipping.

4.2 PERMUTATION-CORRECTED POLICY OPTIMIZATION

To remove the clipping-induced learning signal identified in Section 4.1, we propose PerCPO, which subtracts this closed-form permutation baseline from each clipped token objective:

$$\begin{aligned}\ell_{\mathrm{PerCPO}}(A_i, \rho_{i,t}(\theta); \mathbf{A}) &= \ell_{\mathrm{clip}}(A_i, \rho_{i,t}(\theta)) - B(\rho_{i,t}(\theta); \mathbf{A}) \\ &= \ell_{\mathrm{clip}}(A_i, \rho_{i,t}(\theta)) + \frac{1}{2G}\sum_{j=1}^{G} |A_j| \, |\rho_{i,t}(\theta) - \mathrm{clip}(\rho_{i,t}(\theta), L, U)| \, .\end{aligned} \tag{12}$$

By Equation (10), the corrected token objective has zero mean over within-group reward permutations. Moreover, $B(\rho_{i,t}(\theta); \mathbf{A}) = 0$ when $\rho_{i,t}(\theta) \in [L, U]$, so PerCPO leaves the original token objective unchanged inside the clipping interval and applies the correction only outside it.

For binary verifier rewards, let p denote the fraction of correct responses in the group. As derived in Appendix A, Equation (12) can be written equivalently as

$$\ell_{\mathrm{PerCPO}}(A_i, \rho_{i,t}(\theta); \mathbf{A}) = A_i \left[(1-p)\min(\rho_{i,t}(\theta), U) + p \max(\rho_{i,t}(\theta), L)\right]. \tag{13}$$

The piecewise slopes derived in Appendix B show that the PerCPO token objective preserves the GRPO update inside the clipping interval, restores the updates suppressed by clipping, and attenuates the updates that clipping retains when $A_i > 0$ and $\rho < L$ or when $A_i < 0$ and $\rho > U$. Figure 2 visualizes these effects for positive and negative advantages.

Aggregating the corrected token objective in Equation (13) over tokens and responses gives the full PerCPO objective:

$$\begin{aligned}J_{\mathrm{PerCPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\mathrm{old}}(\cdot|q)} \Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} \\ A_i\Big[(1-p)\min\big(\rho_{i,t}(\theta), U\big) + p\max\big(\rho_{i,t}(\theta), L\big)\Big]\Bigg].\end{aligned} \tag{14}$$

PerCPO retains GRPO's sampling and aggregation procedure. Reward permutations are used only to derive the baseline, after which training directly optimizes the closed-form objective in Equation (14). More generally, PerCPO can be incorporated into other methods that use the clipped token objective with group-centered advantages. By replacing this objective before aggregation, the correction can be combined with alternative sampling strategies, group-level advantage scaling, and token weighting schemes.

4.3 THEORETICAL ANALYSIS

We examine whether the objective continues to favor higher response accuracy along a path with the smallest KL change from the sampling policy. For a fixed prompt q, consider a population model that treats each complete response as one action. Let $R(y) \in \{0, 1\}$ indicate correctness and let $p_0 \in (0, 1)$ denote the accuracy of $\pi_{\mathrm{old}}(\cdot \mid q)$. The fixed advantage is $A(y) = (R(y) - p_0)/Z_0$, where $Z_0 = \sqrt{p_0(1-p_0)} + \delta$.

To attain accuracy $x \in [p_0, 1)$ while preserving the distribution within each reward class, define

$$\pi_x(y \mid q) = \begin{cases} \dfrac{x}{p_0}\pi_{\mathrm{old}}(y \mid q), & R(y) = 1, \\ \dfrac{1-x}{1-p_0}\pi_{\mathrm{old}}(y \mid q), & R(y) = 0. \end{cases} \tag{15}$$

Let $J_{\mathrm{clip}}(x)$ and $J_{\mathrm{PerCPO}}(x)$ denote the expectations under $y \sim \pi_{\mathrm{old}}(\cdot \mid q)$ of the respective objectives evaluated at π_x. The latter uses $p = p_0$ in the binary PerCPO objective of Equation (13).

Proposition 1 *Among policies with accuracy x, π_x minimizes $D_{\mathrm{KL}}(\pi_{\mathrm{old}}(\cdot \mid q) \,\|\, \pi(\cdot \mid q))$. Along this path, $J_{\mathrm{PerCPO}}(x)$ is strictly increasing and satisfies*

$$\frac{dJ_{\mathrm{PerCPO}}(x)}{dx} \geq \frac{2p_0(1-p_0)}{Z_0} > 0 \tag{16}$$

Table 1: Comparison of PerCPO and baseline methods on seven mathematical reasoning benchmarks in terms of Pass@1 accuracy (%). Initial denotes the starting checkpoint before RL training. Avg. is the unweighted mean across the seven benchmarks. Bold indicates the best result within each model group, including ties.

Model	Method	AIME24	AIME25	AMC23	MATH-500	Minerva.	Olympiad.	HMMT	Avg.↑
Qwen2.5-Math -1.5B	*Initial*	5.83	4.17	22.50	43.00	12.13	22.02	0.00	15.66
	GRPO	**15.00**	11.67	62.50	75.40	29.41	38.10	0.00	33.15
	Clip-Higher	13.33	9.17	65.00	75.20	**34.93**	38.99	**1.67**	34.04
	CISPO	**15.00**	10.83	62.50	76.20	29.41	39.58	0.83	33.48
	CE-GPPO	14.17	7.50	55.00	73.00	26.84	34.08	0.00	30.08
	PerCPO	13.33	**13.33**	**72.50**	**77.20**	30.88	**40.48**	0.83	**35.51**
Qwen2.5-Math -7B	*Initial*	10.83	5.83	47.50	61.00	21.32	29.02	1.67	25.31
	GRPO	32.50	14.17	65.00	79.20	34.56	44.79	3.33	39.08
	Clip-Higher	**34.17**	12.50	72.50	**82.00**	**38.60**	45.54	1.67	41.00
	CISPO	30.00	14.17	75.00	81.00	37.50	44.49	**5.00**	41.02
	CE-GPPO	24.17	15.00	67.50	79.40	37.50	46.28	2.50	38.91
	PerCPO	**34.17**	**15.83**	**77.50**	**82.00**	**38.60**	**46.43**	3.33	**42.55**
Qwen3-4B (non-thinking)	*Initial*	19.17	22.50	57.50	82.60	39.71	50.15	10.00	40.23
	GRPO	26.67	25.83	82.50	87.20	40.81	58.78	11.67	47.64
	Clip-Higher	35.83	31.67	**92.50**	88.80	45.22	64.43	12.50	52.99
	CISPO	35.83	32.50	82.50	89.80	**46.32**	63.54	10.00	51.50
	CE-GPPO	38.33	35.00	90.00	92.00	**46.32**	63.10	12.50	53.89
	PerCPO	**39.17**	**35.83**	**92.50**	**92.20**	45.59	**64.73**	**16.67**	**55.24**

at every differentiable point $x \in (p_0, 1)$. In contrast, $J_{\text{clip}}(x)$ is constant in any region where the correct-response ratio $x/p_0 \geq U$ and the incorrect-response ratio $(1-x)/(1-p_0) \leq L$, so that clipping makes both classes' objective terms constant.

PerCPO therefore continues to favor more accurate policies along this path, even when the clipped objective assigns them identical values. The proof is provided in Appendix C.

5 Experiments

5.1 Experimental Setup

Models and Training Data. We use Qwen2.5-Math-1.5B, Qwen2.5-Math-7B (Yang et al., 2024), and Qwen3-4B (Yang et al., 2025) as the initial policies for all methods. Qwen3-4B is used in non-thinking mode. Across all models and methods, each prompt explicitly instructs the model to solve the problem step by step. All models are trained on DAPO-Math-17K (Yu et al., 2025), with 17K mathematical prompts. Rewards depend only on correctness assessed by Math-Verify 0.7.0, with $+1$ for correct answers and -1 otherwise.

Baselines and Training Details. We compare PerCPO with GRPO (Shao et al., 2024), GRPO with the Clip-Higher strategy from DAPO (Yu et al., 2025), CISPO (MiniMax, 2025), and CE-GPPO (Su et al., 2026). GRPO, CE-GPPO, and PerCPO use clipping bounds $L = 0.8$ and $U = 1.2$. Clip-Higher increases U to 1.28 while keeping $L = 0.8$. For CE-GPPO, we set $\beta_1 = 0.75$ and $\beta_2 = 1.0$. CISPO instead clips importance-sampling weights at an upper bound of 5, without a PPO-style lower clipping bound. We implement all methods in the VeRL framework (Sheng et al., 2025) and run each experiment on 8 NVIDIA H100 GPUs. Training runs for 200 rollout steps with a rollout batch size of 64 prompts and a group size of 16. Rollouts use temperature 1.0 and top-p 1.0, with a maximum response length of 3,072 tokens. We use one PPO epoch per rollout step. The mini-batch sizes are 1, 4, and 8 prompt groups for Qwen2.5-Math-1.5B, Qwen2.5-Math-7B, and Qwen3-4B, respectively, giving 64, 16, and 8 policy updates per rollout step. All methods use a constant learning rate of 10^{-6} and are trained without a KL penalty ($\beta = 0$).



(a) Qwen2.5-Math-1.5B (b) Qwen2.5-Math-7B (c) Qwen3-4B

Figure 3: Training reward curves for PerCPO and baseline methods, smoothed with a centered 10-step moving average.

Evaluation. We evaluate on seven mathematical reasoning benchmarks: AIME 2024 (Mathematical Association of America, 2024), AIME 2025 (Mathematical Association of America, 2025), AMC 2023 (Mathematical Association of America, 2023), OlympiadBench (He et al., 2024), HMMT February 2025 (Dekoninck et al., 2026), Minerva Math (Lewkowycz et al., 2022), and MATH-500 (Hendrycks et al., 2021; Lightman et al., 2023). Evaluation responses are sampled with temperature 1.0 and top-p 0.7, with a maximum length of 3,072 tokens. To reduce sampling variance, we independently repeat Pass@1 evaluation four times on each benchmark and report the mean. Additional experimental details are provided in Appendix D.

5.2 Main Results

As shown in Table 1, PerCPO achieves the highest average Pass@1 on all three models, with absolute accuracy gains over GRPO of 2.36% on Qwen2.5-Math-1.5B, 3.47% on Qwen2.5-Math-7B, and 7.60% on Qwen3-4B. It also surpasses the strongest competing baseline for each model by absolute margins of 1.35%–1.53%.

On Qwen3-4B, PerCPO improves upon GRPO on all seven benchmarks, including absolute accuracy gains of 12.50% on AIME 2024 and 10.00% on AIME 2025. On HMMT, it reaches 16.67%, compared with 12.50% for the strongest competing baselines, Clip-Higher and CE-GPPO showing its effectiveness.

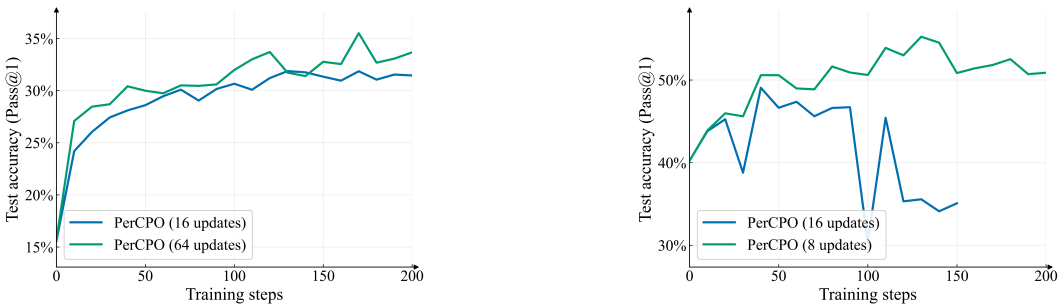
Figure 3 compares the training reward curves of PerCPO and the baseline methods. Under our settings, CE-GPPO training collapses on Qwen2.5-Math-1.5B and Qwen2.5-Math-7B. For these two runs, we plot only the rewards recorded before collapse. PerCPO generally achieves higher training rewards than GRPO across the three models, with the clearest separation on Qwen3-4B, consistent with its larger improvement in benchmark accuracy.

The ordering of methods by training reward does not always match their benchmark performance. On both Qwen2.5-Math models, CISPO and PerCPO reach similar training reward levels, yet PerCPO achieves higher average benchmark accuracy. On Qwen3-4B, CE-GPPO generally attains higher rewards during the later stages of training, while PerCPO achieves a higher average Pass@1 of 55.24% compared with 53.89%. These comparisons suggest that PerCPO generalizes more effectively to the evaluation benchmarks, even when competing methods attain comparable or higher rewards on the training data.

5.3 Ablation Studies

Effects of Restoration and Attenuation. We evaluate Restore-Only and Attenuation-Only to isolate the effects of restoration and attenuation. Bounded PerCPO further tests whether limiting restoration improves performance.

Restore-Only applies the PerCPO correction only where GRPO suppresses the token update, namely when $A_i > 0$ and $\rho > U$ or when $A_i < 0$ and $\rho < L$. All other token updates follow GRPO. *Attenuation-Only* applies the PerCPO correction only where GRPO retains an update outside the clipping interval, namely when $A_i > 0$ and $\rho < L$ or when $A_i < 0$ and $\rho > U$. Updates suppressed by GRPO remain zero. *Bounded PerCPO* retains both restoration and attenuation but stops the



(a) Qwen2.5-Math-1.5B (b) Qwen3-4B

Figure 4: Comparison of PerCPO's test Pass@1 accuracy with different numbers of policy updates per rollout.

Table 2: Comparison of Restore-Only, Attenuation-Only, Bounded PerCPO, and PerCPO on Qwen2.5-Math-7B in terms of Pass@1 accuracy (%).

Method	AIME 2024	AIME 2025	AMC 2023	MATH-500	Minerva Math	OlympiadBench	HMMT	Average
Restore-only	29.17	**15.83**	70.00	81.00	36.03	43.30	**4.17**	39.93
Attenuation-only	**35.00**	**15.83**	67.50	81.00	34.19	44.79	2.50	40.12
Bounded	32.50	14.17	75.00	80.60	**40.07**	44.05	**4.17**	41.51
PerCPO	34.17	**15.83**	**77.50**	**82.00**	38.60	**46.43**	3.33	**42.55**

restored updates beyond additional thresholds. For the symmetric clipping bounds $L = 1 - \epsilon$ and $U = 1 + \epsilon$, these thresholds are $L^\star = L - \frac{\epsilon}{1-p}$ and $U^\star = U + \frac{\epsilon}{p}$. Restored updates stop when $\rho \leq L^\star$ for negative advantages or when $\rho \geq U^\star$ for positive advantages. These thresholds limit the additional objective change from restoration to $\epsilon|A_i|$, while leaving attenuation unchanged.

We conduct this ablation on Qwen2.5-Math-7B with all other settings unchanged from the main experiments. Table 2 shows that PerCPO achieves the highest average accuracy, which suggests that restoration and attenuation are more effective when combined. Bounded PerCPO's lower overall accuracy further suggests that its additional cutoffs can remove useful reward-guided updates.

Policy Updates per Rollout. We examine the sensitivity of PerCPO to the number of policy updates per rollout by varying the mini-batch size while keeping the rollout batch size fixed. We compare 16 and 64 updates on Qwen2.5-Math-1.5B, and 8 and 16 updates on Qwen3-4B. All other settings follow the main experiments.

Figure 4 shows that Qwen2.5-Math-1.5B generally benefits from 64 updates compared with 16. Qwen3-4B instead performs better with 8 updates, while 16 updates lead to pronounced accuracy drops later in training. With π_{old} fixed within each rollout, more updates can move π_θ further from the behavior policy, potentially placing more ratios $\rho_{i,t}(\theta)$ outside $[L, U]$, where the PerCPO correction takes effect. These results highlight PerCPO's sensitivity to the number of policy updates per rollout.

6 Conclusion

Through an exact decomposition and within-group reward permutations, we showed that the standard clipped objective contains a component whose gradient acts as a clipping-induced learning signal that can counteract verifier-guided updates. We proposed PerCPO to subtract this component as a closed-form permutation baseline. Our theoretical analysis shows that PerCPO's objective increases as accuracy improves along the minimum-KL path, even where the standard clipped objective plateaus. Experiments on three models across seven mathematical reasoning benchmarks demonstrate the effectiveness of this correction. Ablation studies further support the complementary roles of restoration and attenuation, while highlighting PerCPO's sensitivity to the number of policy updates per rollout.

AI USE STATEMENT

We used generative AI tools to improve the language and clarity of the manuscript, identify relevant literature, and assist with mathematical derivations and proofs. The authors independently checked the AI-assisted mathematical arguments and verified the cited literature against the original sources. The authors take full responsibility for the accuracy and integrity of the final manuscript.

REPRODUCIBILITY STATEMENT

The PerCPO objective and its derivation are presented in Section 4. Detailed derivations for binary rewards are provided in Appendices A and B. The assumptions underlying our theoretical analysis are stated in Section 4.3, with the proof provided in Appendix C. Section 5.1 describes the initial models, training data, reward computation, implementation framework, hardware, training hyperparameters, and evaluation protocol.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12248–12267, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.662. URL https://aclanthology.org/2024.acl-long.662/.

Peter Chen, Xiaopeng Li, Ziniu Li, Wotao Yin, Xi Chen, and Tianyi Lin. Exploration vs exploitation: Rethinking RLVR through clipping, entropy, and spurious reward. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d99e8e80a6c41e148db686918dd7eab3-Abstract-Conference.html.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs, 2026. URL https://arxiv.org/abs/2605.00674.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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, 2024. URL https://arxiv.org/abs/2402.14008.

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 Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-Reasoner-Zero: An open source approach to scaling up reinforcement learning on the base model. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ed873d79e7c268c020c4b4db13a2812a-Abstract-Conference.html.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. VinePPO: Refining credit assignment in RL training of LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 29557–29590. PMLR, 2025. URL https://proceedings.mlr.press/v267/kazemnejad25a.html.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs, 2025. URL https://arxiv.org/abs/2501.12599.

S. Kullback and R. A. Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. doi: 10.1214/aoms/1177729694.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tülu 3: Pushing frontiers in open language model post-training. In *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2411.15124.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 3843–3857, 2022.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. CPPO: Accelerating the training of group relative policy optimization-based reasoning models. In *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/57ec371599f3d4123c9b054ffa3ef0ca-Abstract-Conference.html.

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 *Conference on Language Modeling*, 2025. URL https://arxiv.org/abs/2503.20783.

Mathematical Association of America. American mathematics competitions, 2023.

Mathematical Association of America. American invitational mathematics examination, 2024.

Mathematical Association of America. American invitational mathematics examination, 2025.

MiniMax. MiniMax-M1: Scaling test-time compute efficiently with lightning attention, 2025. URL https://arxiv.org/abs/2506.13585.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017. URL https://arxiv.org/abs/1707.06347.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, Yulia Tsvetkov, Hannaneh Hajishirzi, Pang Wei Koh, and Luke Zettlemoyer. Spurious rewards: Rethinking training signals in RLVR, 2025. URL https://arxiv.org/abs/2506.10947.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Nisan Stiennon, Long Ouyang, Jeff Wu, Daniel M. Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul Christiano. Learning to summarize from human feedback. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html.

Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. CE-GPPO: Coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 37747–37760, San Diego, California, United States, July 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1752. URL https://aclanthology.org/2026.acl-long.1752/.

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/a666587afda6e89aec274a3657558a27-Abstract.html.

Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, Xun Deng, Zhikai Lei, Miao Zheng, Guoteng Wang, Shuo Zhang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, Qi Zhang, and Xuanjing Huang. BAPO: Stabilizing off-policy reinforcement learning for LLMs via balanced policy optimization with adaptive clipping. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/cca4df598d8066ac9e985852dd1d48b5-Abstract-Conference.html.

Zhongwen Xu and Zihan Ding. Single-stream policy optimization. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/e46fc33e80e9fa2febcdb058fba4beca-Abstract-Conference.html.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Kai Dang, Mingfeng Xue, Rui Men, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement, 2024. URL https://arxiv.org/abs/2409.12122.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, 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 *Advances in Neural Information Processing Systems*, volume 38, 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/a4277440d50f1f15d2cb4c14f7e0c0d2-Abstract-Conference.html.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, Tiantian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Ru Zhang, Xin Liu, Mingxuan Wang, Yonghui Wu, and Lin Yan. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks, 2025. URL https://arxiv.org/abs/2504.05118.

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 *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/44a1f18afd6d5cc34d7e5c3d8a80f63b-Abstract-Conference.html.

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, 2025a. URL https://arxiv.org/abs/2507.18071.

Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. In *Advances in Neural Information Processing Systems*, volume 38, 2025b. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/b42fb82fc3ffb88b872c4714f093875d-Abstract-Conference.html.

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy RL reach with stale data on LLMs? In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/8590b946e4f0c2b1262cb5b047924653-Abstract-Conference.html.

Haodong Zhu, Yangyang Ren, Yanjing Li, Mingbao Lin, Linlin Yang, Xuhui Liu, Xiantong Zhen, Haiguang Liu, and Baochang Zhang. Unbiased dynamic pruning for efficient group-based policy optimization. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2603.04135.

A Derivation of the Binary-Reward PerCPO Objective

This section derives the compact form of the PerCPO token objective for binary verifier rewards in Section 4.2. Consider binary verifier rewards with values $r_- < r_+$, where r_+ is assigned to a correct response and r_- to an incorrect response. Let $p \in (0, 1)$ denote the fraction of correct responses in a sampled group. The group mean reward is

$$\bar{R} = pr_+ + (1-p)r_-. \tag{17}$$

Define $\Delta = r_+ - r_- > 0$ and

$$Z = \mathrm{std}(R_1, \ldots, R_G) + \delta. \tag{18}$$

The normalized advantages of correct and incorrect responses are

$$A^+ = \frac{r_+ - \bar{R}}{Z} = \frac{(1-p)\Delta}{Z}, \qquad A^- = \frac{r_- - \bar{R}}{Z} = -\frac{p\Delta}{Z}. \tag{19}$$

Let

$$C(\mathbf{A}) = \frac{1}{2G}\sum_{j=1}^{G} |A_j| \tag{20}$$

denote the coefficient of the correction term in Equation (12). Since the group contains pG correct responses and $(1-p)G$ incorrect responses,

$$\begin{aligned} C(\mathbf{A}) &= \frac{1}{2G}\left[pG\frac{(1-p)\Delta}{Z} + (1-p)G\frac{p\Delta}{Z}\right] \\ &= \frac{p(1-p)\Delta}{Z} = pA^+ = -(1-p)A^-. \end{aligned} \tag{21}$$

For compactness, define

$$m(\rho) = \min(\rho, U), \qquad M(\rho) = \max(\rho, L). \tag{22}$$

The clipping distance satisfies

$$|\rho - \mathrm{clip}(\rho, L, U)| = M(\rho) - m(\rho). \tag{23}$$

For a correct response, $A_i = A^+ > 0$ and $\ell_{\mathrm{clip}}(A_i, \rho) = A_i m(\rho)$. Using $C(\mathbf{A}) = pA_i$ from Equation (21), the corrected objective becomes

$$\begin{aligned} \ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A}) &= A_i m(\rho) + C(\mathbf{A})\big[M(\rho) - m(\rho)\big] \\ &= A_i\big[(1-p)m(\rho) + pM(\rho)\big]. \end{aligned} \tag{24}$$

For an incorrect response, $A_i = A^- < 0$ and $\ell_{\mathrm{clip}}(A_i, \rho) = A_i M(\rho)$. Since $C(\mathbf{A}) = -(1-p)A_i$,

$$\begin{aligned} \ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A}) &= A_i M(\rho) + C(\mathbf{A})\big[M(\rho) - m(\rho)\big] \\ &= A_i\big[(1-p)m(\rho) + pM(\rho)\big]. \end{aligned} \tag{25}$$

The same expression therefore holds for both reward outcomes:

$$\ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A}) = A_i\left[(1-p)\min(\rho, U) + p\max(\rho, L)\right]. \tag{26}$$

Equation (26) shows that, for any binary reward values $r_- < r_+$, the corrected token objective depends on the reward encoding only through the normalized advantage A_i and on the group through its fraction p of correct responses. The same form therefore applies to both $\{0, 1\}$ and $\{-1, 1\}$ rewards. If all responses receive the same reward, their centered advantages are zero and the group contributes no update.

B Piecewise Slopes of the Binary-Reward PerCPO Objective

This section derives the piecewise slopes shown in Figure 2 and compares the resulting token updates with those of GRPO. For a fixed sampled group, A_i and p remain fixed during policy optimization, while the probability ratio depends on the current policy.

Resolving the minimum and maximum in Equation (13) over the three ratio regions gives

$$\ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A}) = \begin{cases} A_i\big[(1-p)\rho + pL\big], & \rho < L, \\ A_i\rho, & L \le \rho \le U, \\ A_i\big[(1-p)U + p\rho\big], & \rho > U. \end{cases} \tag{27}$$

Therefore, for $\rho \notin \{L, U\}$, the slope of the token objective is

$$\frac{\partial \ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A})}{\partial \rho} = \begin{cases} (1-p)A_i, & \rho < L, \\ A_i, & L < \rho < U, \\ pA_i, & \rho > U. \end{cases} \tag{28}$$

Since $\rho = \rho_{i,t}(\theta)$ during policy optimization, the corresponding parameter gradient follows from

$$\nabla_\theta \ell_{\mathrm{PerCPO}}(A_i, \rho_{i,t}(\theta); \mathbf{A}) = \left.\frac{\partial \ell_{\mathrm{PerCPO}}(A_i, \rho; \mathbf{A})}{\partial \rho}\right|_{\rho = \rho_{i,t}(\theta)} \nabla_\theta \rho_{i,t}(\theta). \tag{29}$$

Inside the clipping interval, both GRPO and PerCPO have slope A_i. When $A_i > 0$ and $\rho > U$, GRPO has zero slope, whereas PerCPO has slope pA_i; when $A_i < 0$ and $\rho < L$, the corresponding slopes are zero and $(1-p)A_i$. PerCPO therefore restores a nonzero update in the two regions suppressed by clipping. In contrast, when $A_i > 0$ and $\rho < L$, the slope changes from A_i under GRPO to $(1-p)A_i$ under PerCPO; when $A_i < 0$ and $\rho > U$, it changes from A_i to pA_i. These are the two updates retained by clipping, and PerCPO attenuates their magnitudes without changing their directions.

Thus, the three slopes in Equation (28) establish the behavior illustrated in Figure 2: PerCPO preserves the GRPO update inside the clipping interval, restores the updates suppressed by clipping, and attenuates the retained updates in the opposite out-of-range regions.

C Proof of Proposition 1

We use the population setting of Section 4.3. For the policy π_x, let $\rho_x(y) = \pi_x(y \mid q)/\pi_{\text{old}}(y \mid q)$ denote its response probability ratio. The two population objectives are

$$\begin{aligned} J_{\text{clip}}(x) &= \mathbb{E}_{y\sim\pi_{\text{old}}(\cdot|q)}\left[\ell_{\text{clip}}(A(y), \rho_x(y))\right], \\ J_{\text{PerCPO}}(x) &= \mathbb{E}_{y\sim\pi_{\text{old}}(\cdot|q)}\left[A(y)\Big((1-p_0)\min(\rho_x(y), U) + p_0 \max(\rho_x(y), L)\Big)\right]. \end{aligned} \tag{30}$$

The policy π_x has accuracy x and preserves the conditional response distribution within each reward class. For any policy π with the same accuracy, the KL chain rule gives

$$\begin{aligned} D_{\text{KL}}(\pi_{\text{old}}(\cdot \mid q) \,\|\, \pi(\cdot \mid q)) = p_0 \log \frac{p_0}{x} + (1-p_0)\log\frac{1-p_0}{1-x} \\ + p_0 D_{\text{KL}}(\pi_{\text{old}}(\cdot \mid q, R=1) \,\|\, \pi(\cdot \mid q, R=1)) \\ + (1-p_0) D_{\text{KL}}(\pi_{\text{old}}(\cdot \mid q, R=0) \,\|\, \pi(\cdot \mid q, R=0)). \end{aligned} \tag{31}$$

The first two terms are fixed by p_0 and x. Both conditional KL terms are nonnegative and vanish for π_x, establishing its minimum-KL property.

Correct and incorrect responses have advantages $A^+ = (1-p_0)/Z_0$ and $A^- = -p_0/Z_0$, and probability ratios $\rho_+ = x/p_0$ and $\rho_- = (1-x)/(1-p_0)$, respectively. Since $x \geq p_0$, we have $\rho_+ \geq 1$ and $\rho_- \leq 1$. Averaging over the two reward classes gives

$$\begin{aligned} J_{\text{clip}}(x) &= p_0 A^+ \min(\rho_+, U) + (1-p_0) A^- \max(\rho_-, L) \\ &= \frac{1-p_0}{Z_0}\min(x-p_0, p_0(U-1)) + \frac{p_0}{Z_0}\min(x-p_0, (1-p_0)(1-L)). \end{aligned} \tag{32}$$

For PerCPO, using $\max(\rho_+, L) = \rho_+$ and $\min(\rho_-, U) = \rho_-$ yields

$$\begin{aligned} J_{\text{PerCPO}}(x) &= p_0 A^+ \left[(1-p_0)\min(\rho_+, U) + p_0\rho_+\right] \\ &\quad + (1-p_0) A^- \left[(1-p_0)\rho_- + p_0 \max(\rho_-, L)\right] \\ &= \frac{2p_0(1-p_0)}{Z_0}(x-p_0) + \frac{(1-p_0)^2}{Z_0}\min(x-p_0, p_0(U-1)) \\ &\quad + \frac{p_0^2}{Z_0}\min(x-p_0, (1-p_0)(1-L)). \end{aligned} \tag{33}$$

Both minimum terms are nondecreasing in x, while the first term increases linearly. Hence, for any $p_0 \leq x_1 < x_2 < 1$,

$$J_{\text{PerCPO}}(x_2) - J_{\text{PerCPO}}(x_1) \geq \frac{2p_0(1-p_0)}{Z_0}(x_2 - x_1) > 0. \tag{34}$$

This establishes strict monotonicity and the derivative bound in Equation (16).

When $\rho_+ \geq U$ and $\rho_- \leq L$, both minimum terms in Equation (32) are constant. The clipped objective therefore plateaus, whereas PerCPO retains the strictly increasing first term in Equation (33). This completes the proof.

D Additional Experimental Details

Prompt template. We use the following prompt template, replacing {`problem`} with the problem statement.

```
Solve the following math problem step by step. The last line of your response
should be of the form \boxed{Answer} where Answer is the answer to the problem.

{problem}

Remember to put your answer in \boxed{...} format on its own line.
```

Software versions. Table 4 lists the package versions recorded in the environment lockfile.

Table 3: Additional training and evaluation settings.

Setting	Value
Fine-tuning method	Full-parameter fine-tuning
Optimizer	AdamW
Adam momentum parameters	$(0.9, 0.999)$
Weight decay	0.01
Learning-rate schedule	Constant, without warmup
Gradient norm clipping threshold	1.0
Entropy bonus coefficient	0
Distributed training	PyTorch FSDP
Parameter computation dtype	BF16
Gradient reduction dtype	FP32
Gradient checkpointing	Enabled
Padding removal	Enabled
Maximum prompt length (training and evaluation)	1,024 tokens
Overlength prompts	Filtered using the model's tokenizer, without truncation
Rollout engine	vLLM
Rollout tensor parallelism	1
vLLM maximum sequence length	4,096 tokens
Top-k sampling	Disabled (`top_k=-1`)
Evaluation batch size	128
Evaluation frequency	Before training and every 10 rollout steps

Table 4: Software versions recorded in the environment lockfile.

Package	Version
PyTorch	`2.6.0+cu124`
Transformers	`4.53.2`
vLLM	`0.8.5.post1`
Ray	`2.43.0`
FlashAttention	`2.7.4.post1`
Math-Verify	`0.7.0`