RETHINKING THE DIVERGENCE REGULARIZATION IN LLM RL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become a key component of post-training large language models (LLMs). In practice, LLM RL is often off-policy because of training-inference mismatch and policy staleness, making trust-region control essential for stable optimization. Mainstream methods such as PPO and GRPO approximate this control with a ratio-clipping mechanism, but the importance ratio can be a poor proxy for distributional shift in long-tailed vocabularies. Recent work such as DPPO addresses this mismatch by replacing ratio-based clipping with a divergence-based mask, yielding a trust region defined by the sampled token's absolute probability shift. However, DPPO still relies on a hard mask: once a token crosses the trust-region boundary in a harmful direction, its gradient is discarded entirely. To address this, we propose Divergence Regularized Policy Optimization (DRPO), which replaces the hard mask with a smooth advantage-weighted quadratic regularizer on policy shift. DRPO keeps DPPO's trust-region geometry while inducing bounded, continuous gradient weights that attenuate diverging updates by their absolute probability shift and can reverse them beyond the boundary. Experiments across model scales, architectures, and precision settings show that DRPO improves the stability and efficiency of LLM RL training.

1 INTRODUCTION

Reinforcement learning (RL) has become a central component of post-training large language models (LLMs), enabling models to better align with human preferences and improve performance on complex reasoning tasks (Ouyang et al., 2022; Rafailov et al., 2023; Guo et al., 2025; Liu et al., 2025c). During training, an LLM is optimized as an autoregressive token-level policy that generates a response and receives a scalar reward from either a learned reward model (Ouyang et al., 2022) or a rule-based verifier (Guo et al., 2025; Yu et al., 2025). In practice, modern LLM RL is typically *off-policy*: rollouts are generated by inference engines whose numerical behavior differs from training engines (Qi et al., 2025; Yao et al., 2025), and collected trajectories are commonly split into multiple mini-batches or gradient steps (Liu et al., 2025a). As a result, the policy being updated is not identical to the behavior policy that generated the data.

In such off-policy settings, Trust Region Policy Optimization (TRPO) provides a principled solution by maximizing a surrogate objective under an explicit divergence constraint between the current and behavior policy (Schulman et al., 2015; Achiam et al., 2017). However, its second-order optimization makes TRPO impractical to scale. Proximal Policy Optimization (PPO) (Schulman et al., 2017) replaces the constrained optimization with a simple ratio-clipping heuristic and has become the dominant recipe in modern LLM RL training. Building on PPO, GRPO improves practicality by replacing a learned critic with group-relative reward normalization (Shao et al., 2024; Ahmadian et al., 2024; Liu et al., 2025c). More recently, Simple Policy Optimization (SPO) (Xie et al., 2024) replaces hard clipping with a smooth quadratic regularizer that preserves the same ratio boundary while avoiding the zero-gradient issue outside the clipping range. These methods differ in implementation details, but they share the same trust-region geometry: the per-token update is controlled through its importance ratio.

The importance ratio, however, is a poor proxy for distributional shift for LLMs due to large and long-tailed vocabularies (Qi et al., 2026). A small increase on a low-probability token can produce a very large ratio while changing little probability mass. Conversely, a moderate ratio change on

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.

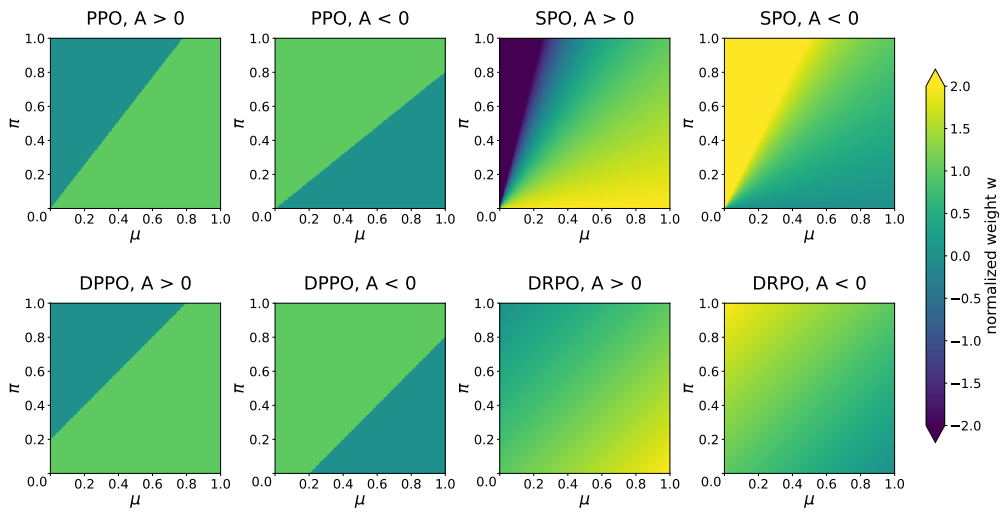

Figure 1: Per-token gradient weights of different algorithms as a function of the current probability $\pi(y_t|s_t)$ and behavior probability $\mu(y_t|s_t)$. For SPO, $\epsilon = 1$; for DRPO, $\delta = 1$; for PPO, $\varepsilon_{\text{low}} = 0.2$ and $\varepsilon_{\text{high}} = 0.28$; for DPPO, $\delta = 0.2$. SPO's weight grows without bound as $\mu(y_t|s_t) \to 0$, while the weight of DRPO remains bounded for all tokens.

a high-probability token can move substantial mass and meaningfully alter the policy. A fixed ratio window therefore tends to over-constrain low-probability tokens and under-constrain high-probability tokens (Qi et al., 2026; Yu et al., 2025; Chen et al., 2025).

DPPO addresses this issue by replacing ratio-based clipping with a divergence-based mask (Qi et al., 2026). When the policy divergence exceeds a prescribed threshold and the current update would increase it further, DPPO disables the corresponding token gradient. Its Binary-TV variant, which we refer to as DPPO unless otherwise stated, measures the absolute probability shift of the sampled token. This quantity aligns more closely with total variation (TV) geometry than the importance ratio in long-tailed vocabularies. However, DPPO still enforces the trust region with a binary mask. Once a token moves outside the trust region in a harmful direction, its gradient is set to zero. This prevents further movement away from the behavior policy, but it discards the gradient regardless of how far the token has moved and can introduce abrupt changes near the threshold.

We propose DRPO, a divergence-regularized policy optimization method that replaces the hard mask while preserving the Binary-TV geometry of DPPO. Our method is motivated by SPO, which places the per-token optimum exactly at PPO's trust-region boundary through an advantage-weighted χ^2 regularizer. We rewrite the Binary-TV constraint as a token-adaptive ratio bound and apply the same construction as SPO, which yields an advantage-weighted ℓ_2^2 regularizer. The resulting regularizer changes the trust-region geometry from a fixed ratio constraint to an absolute probability-shift constraint, combining the smoothness of SPO with the divergence-based geometry of DPPO.

Our method also gives a simple and stable gradient form. Each token's policy-gradient contribution is multiplied by a continuous weight determined by its Binary-TV shift and by whether the current update moves away from or toward the behavior policy. When the update moves away from the behavior policy, the weight decreases with the absolute probability shift and, for thresholds below one, turns negative beyond the boundary. When the update moves back toward the behavior policy, the weight is amplified. Because this weight depends on an absolute probability shift rather than an importance ratio, it better captures the geometry of policy change and remains bounded even in the low-probability tail where SPO's ratio-based weight can grow without bound (Figure 1).

Beyond the specific algorithm, our results motivate a gradient-centered view of regularizer design for LLM RL. Our ablations show that standard KL or TV penalties can underperform because their gradients reintroduce ratio-based geometry. They also show that the per-token penalty should be weighted by the absolute-advantage because it keeps the trust-region boundary independent of reward scale. These findings suggest three practical criteria for an effective regularizer: it should induce a

stable boundary aligned with distributional shift, keep per-token gradient weights bounded in the long-tailed vocabulary, and smoothly attenuate updates that move the policy away. DRPO satisfies these criteria with a simple Binary-TV-aligned regularizer, offering an empirical lens for designing stable policy-optimization objectives for LLMs.

2 BACKGROUND

We defer a broader discussion of related work to Section A.

The generation process of LLMs can be formulated as a token-level MDP (Bellman, 1957) $\mathcal{M} = (\mathcal{S}, \mathcal{A}, R, p_{\mathcal{X}})$. Given a prompt $x \sim p_{\mathcal{X}}$, a response $y = (y_1, \ldots, y_T)$ is autoregressively sampled by a conditional stochastic policy $\pi(y_t|s_t)$ over the vocabulary $\mathcal{A}$, where the state $s_t = (x, y_1, \ldots, y_{t-1}) \in \mathcal{S}$ is the concatenation of the prompt and the generated tokens so far. The generation terminates upon producing the `[eos]` token or reaching the token limit. A scalar reward $R(x, y)$ is then provided, either from a reward model (Ouyang et al., 2022) or a rule-based verifier (Guo et al., 2025). The policy objective is to maximize the expected reward:

$$\mathcal{J}(\pi) = \mathbb{E}_{x\sim p_{\mathcal{X}}}\left[\mathcal{J}(x, \pi)\right] = \mathbb{E}_{x\sim p_{\mathcal{X}}}\left[\mathbb{E}_{y\sim\pi(\cdot|x)}[R(x, y)]\right].$$

Modern RL frameworks for LLM fine-tuning rely on highly optimized training and inference engines to maximize throughput, which inevitably introduces subtle but non-negligible numerical discrepancies (Qi et al., 2025; Yao et al., 2025). A further common practice is to collect a large batch of rollouts and split it into multiple mini-batches for multiple gradient updates (Liu et al., 2025a). Both cases bring RL training into an *off-policy* paradigm, where the data is sampled from a behavior policy μ and the objective becomes:

$$\mathcal{J}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\prod_{t=1}^{|y|}\frac{\pi(y_t|s_t)}{\mu(y_t|s_t)} \cdot R(x, y)\right]. \tag{1}$$

2.1 TRUST REGION POLICY OPTIMIZATION

Directly optimizing Equation (1) often suffers from high variance due to the product of importance sampling ratios. TRPO (Schulman et al., 2015) handles this with a token-level surrogate objective:

$$\mathcal{L}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|}\frac{\pi(y_t|s_t)}{\mu(y_t|s_t)} \cdot \hat{A}_t\right], \tag{2}$$

where $\hat{A}_t = R(x, y) - V(s_t)$ is the advantage estimate, and $V(s_t)$ is a variance-reduction baseline that does not change the expected policy gradient. Typically, $V(s_t)$ is set to the expected reward conditioned on state s_t. TRPO and later work (Achiam et al., 2017; Qi et al., 2026) have shown that this surrogate is a first-order approximation[1] of Equation (1), and a monotonic performance improvement can be guaranteed within a *trust region* defined by the KL divergence or TV distance. Formally, TRPO solves the following constrained optimization problem:

$$\max_{\pi}\ \mathcal{L}(x, \pi) \qquad \text{s.t. } \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} D_{\mathrm{TV}}\left(\mu(\cdot|s_t)\|\pi(\cdot|s_t)\right)\right] \le \delta. \tag{3}$$

2.2 PROXIMAL POLICY OPTIMIZATION

TRPO requires second-order methods that are computationally prohibitive at scale. PPO (Schulman et al., 2017) was introduced as a simple alternative that approximates the trust region via a ratio-clipping mechanism. Letting $r_t \triangleq \frac{\pi(y_t|s_t)}{\mu(y_t|s_t)}$ denote the per-token importance ratio, PPO optimizes:

$$\mathcal{L}_{\mathrm{PPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|}\min\Big(r_t \cdot \hat{A}_t,\ \mathrm{clip}(r_t, 1-\epsilon, 1+\epsilon) \cdot \hat{A}_t\Big)\right]. \tag{4}$$

[1] We adapt TRPO to the LLM setting and ignore a constant term; see Qi et al. (2026) for a rigorous derivation.

The clipping mechanism deactivates the gradient whenever r_t leaves the interval $[1-\epsilon,\, 1+\epsilon]$ and further increase the loss, thereby enforcing a per-token, ratio-based trust region, i.e., $|r_t - 1| \leq \epsilon$.

Group Relative Policy Optimization. In traditional RL settings, $V(s_t)$ is typically estimated by a critic model. Learning such a critic is, however, expensive and noisy for LLMs. To address this, Shao et al. (2024); Ahmadian et al. (2024); Liu et al. (2025c) propose sampling a group of responses $\{y_i\}_{i=1}^{G}$ per prompt and estimating the advantage as $\hat{A}_{t,i} = R(x, y_i) - \frac{1}{G}\sum_{j=1}^{G} R(x, y_j)$. This critic-free approach is widely known as Group Relative Policy Optimization (GRPO).

2.3 Simple Policy Optimization

While effective in practice, PPO enforces its trust region through a hard clipping rule. This mechanism is brittle near the clipping boundary: a small change in r_t can abruptly switch a token's gradient from active to zero. Moreover, once a token has moved outside the clip range in a harmful direction, PPO removes its gradient entirely and provides no corrective signal back toward the trust region. SPO (Xie et al., 2024) addresses these issues by replacing the hard clip with a smooth quadratic regularizer:

$$\mathcal{L}_{\mathrm{SPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|}\left(r_t \cdot \hat{A}_t - \frac{|\hat{A}_t|}{2\epsilon}\,(r_t - 1)^2\right)\right]. \tag{5}$$

For each token, the integrand is a concave quadratic in r_t. Setting its derivative $\hat{A}_t - \frac{|\hat{A}_t|}{\epsilon}(r_t - 1)$ to zero gives the unique maximizer $r_t^\star = 1 + \mathrm{sign}(\hat{A}_t)\epsilon$, which exactly matches PPO's clipping boundary in Equation (4). SPO therefore preserves the same ratio-based trust region as PPO, but enforces it through a continuous gradient weight.

2.4 Divergence Proximal Policy Optimization

PPO, GRPO, and SPO all derive their trust region from the per-token ratio r_t. Qi et al. (2026) argues that this estimator is poorly behaved over LLMs' long-tailed vocabulary: a low-probability token can produce an enormous ratio (e.g., $10^{-5} \to 10^{-3}$) while contributing negligibly to the actual distributional shift, whereas a high-probability token may exhibit a modest ratio (e.g., $0.99 \to 0.80$) that nevertheless induces a substantial change in policy. Ratio-based trust regions thus over-penalize low-probability tokens, which are often exploratory, and under-penalize high-probability ones, harming both efficiency and stability.

DPPO (Qi et al., 2026) replaces the ratio-based clip with a divergence-based mask M_t^{DPPO} conditioned on the policy divergence $D_t \triangleq D\big(\mu(\cdot|s_t)\,\|\,\pi(\cdot|s_t)\big)$, where D is either the TV or KL divergence over the full per-state token distributions. The DPPO objective and mask are

$$\begin{aligned}\mathcal{L}_{\mathrm{DPPO}}(x, \pi) &= \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} M_t^{\mathrm{DPPO}} \cdot r_t \cdot \hat{A}_t\right], \\ M_t^{\mathrm{DPPO}} &= \begin{cases} 0, & \mathrm{sign}\big(\hat{A}_t \cdot (r_t - 1)\big) > 0 \text{ and } D_t > \delta, \\ 1, & \text{otherwise,} \end{cases}\end{aligned} \tag{6}$$

with divergence threshold δ. The mask zeros the gradient only when the policy has already moved outside the trust region in a direction that would push it further away. For tractability over large vocabularies, DPPO approximates D_t with binary or top-k surrogates. Most relevant to our method is the Binary-TV approximation, which collapses the per-state distribution into a Bernoulli over the sampled token versus the rest, yielding

$$D_t^{\text{Bin-TV}} \triangleq \big|\pi(y_t|s_t) - \mu(y_t|s_t)\big|. \tag{7}$$

The corresponding trust region $\big|\pi(y_t|s_t) - \mu(y_t|s_t)\big| \leq \delta$ constrains the *absolute* probability shift on the sampled token, in contrast to the *relative* ratio constraint $|r_t - 1| \leq \epsilon$ shared by PPO and SPO.

3 Methodology

We derive DRPO from the Binary-TV view of DPPO. For a sampled token y_t, the Binary-TV proxy in Equation (7) satisfies $D_t^{\text{Bin-TV}} = \big|\pi(y_t|s_t) - \mu(y_t|s_t)\big| = \mu(y_t|s_t)\,|r_t - 1|$. Thus the Binary-TV trust

region $D_t^{\text{Bin-TV}} \leq \delta$ is equivalent to a token-adaptive ratio constraint, $|r_t - 1| \leq \frac{\delta}{\mu(y_t|s_t)}$. Under this view, DPPO can be represented by a PPO-style clipped surrogate with the same gradient behavior:

$$\mathcal{L}_{\text{DPPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} \min\left(r_t\hat{A}_t,\ \text{clip}\left(r_t, 1 - \frac{\delta}{\mu(y_t|s_t)}, 1 + \frac{\delta}{\mu(y_t|s_t)}\right)\hat{A}_t\right)\right].$$

Compared with PPO in Equation (4), DPPO replaces the fixed ratio interval with an adaptive one whose width is inversely proportional to the behavior probability of the sampled token. Low-probability tokens therefore receive a looser ratio tolerance, while high-probability tokens receive a tighter one.

However, DPPO still enforces this divergence-based trust region through a binary mask, which makes the update brittle near the boundary: a small change in the estimated divergence can abruptly switch a token's gradient from full strength to zero. The key lesson from SPO is that the same trust-region boundary can be enforced by a smooth regularizer instead of a discontinuous cutoff. Such a regularizer induces a continuous gradient weight that varies with both the magnitude and direction of the probability shift. Inside the boundary, it smoothly reweights the policy gradient; outside the boundary, it can also pull the policy back toward the trust region. We apply this principle to the Binary-TV trust region by replacing DPPO's mask with a quadratic regularizer on the sampled token's absolute probability shift. The resulting objective, *Divergence Regularized Policy Optimization* (DRPO), is

$$\mathcal{L}_{\text{DRPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\delta}\,\mu(y_t|s_t)\,(r_t - 1)^2\right]. \tag{8}$$

The first term is the token-level surrogate in Equation (2). The second term is a quadratic regularizer whose curvature is scaled by the behavior probability of the sampled token. This single factor changes the equilibrium from a fixed ratio shift, as in PPO and SPO, to a fixed absolute probability shift, as required by DPPO. Taking the gradient of Equation (8) gives (see Appendix B for a full derivation)

$$\nabla\mathcal{L}_{\text{DRPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|}\left(1 - \text{sign}(\hat{A}_t(r_t - 1))\frac{D_t^{\text{Bin-TV}}}{\delta}\right) r_t\hat{A}_t\nabla\log\pi(y_t|s_t)\right]. \tag{9}$$

Relative to the unregularized gradient of Equation (2), DRPO multiplies each token's policy-gradient contribution by a continuous weight

$$w_t = 1 - \text{sign}(\hat{A}_t(r_t - 1))\,\frac{D_t^{\text{Bin-TV}}}{\delta}. \tag{10}$$

The sign term indicates whether the current update moves the sampled probability away from or toward the behavior policy. The magnitude term measures the Binary-TV shift that should be controlled. Together, these terms make the weight vary smoothly with both the size and direction of the sampled token's probability shift.

Table 1: Comparison of trust-region mechanisms. DPPO and DRPO share the Binary-TV geometry of the sampled token's absolute probability shift. Because TV is bounded in $[0, 1]$, DRPO produces bounded gradient weights, whereas the ratio-based constraint in SPO does not.

Method	Mechanism	Trust-region constraint	Gradient weight w_t	Range of w_t
PPO	hard clip	$\lvert r_t - 1\rvert \leq \epsilon$	0 or 1	$\{0, 1\}$
SPO	smooth regularizer	$\lvert r_t - 1\rvert \leq \epsilon$	$1 - \text{sign}(\hat{A}_t(r_t - 1))\,\lvert r_t - 1\rvert/\epsilon$	$(-\infty, +\infty)$
DPPO	hard mask	$\lvert\pi(y_t\lvert s_t) - \mu(y_t\lvert s_t)\rvert \leq \delta$	0 or 1	$\{0, 1\}$
DRPO	smooth regularizer	$\lvert\pi(y_t\lvert s_t) - \mu(y_t\lvert s_t)\rvert \leq \delta$	$1 - \text{sign}(\hat{A}_t(r_t - 1))\,\lvert\pi(y_t\lvert s_t) - \mu(y_t\lvert s_t)\rvert/\delta$	$[1 - \frac{1}{\delta}, 1 + \frac{1}{\delta}]$

3.1 Trust Region Analysis

We now examine how the smooth gradient weight in Equation (10) encodes the trust-region boundary.

Diverging update ($\text{sign}(\hat{A}_t(r_t - 1)) > 0$)**.** When the update moves $\pi(y_t|s_t)$ away from $\mu(y_t|s_t)$, the weight becomes $w_t = 1 - D_t^{\text{Bin-TV}}/\delta$. Thus the gradient is gradually attenuated as the Binary-TV

shift approaches the boundary. Inside the trust region, where $D_t^{\text{Bin-TV}} < \delta$, the weight remains positive and the update still follows the reward-improving direction. Outside the trust region, where $D_t^{\text{Bin-TV}} > \delta$ (possible only if $\delta < 1$), the weight is negative, so the gradient reverses back toward the trust region. Since the per-token objective in Equation (8) is a concave quadratic in r_t, the zero-weight condition gives the stationary point

$$\pi(y_t|s_t)^\star = \mu(y_t|s_t) + \text{sign}(\hat{A}_t)\,\delta, \tag{11}$$

which matches DPPO's boundary for the same δ and is feasible only if $\delta < 1$.

Converging update ($\text{sign}(\hat{A}_t(r_t - 1)) < 0$)**.** When the update moves $\pi(y_t|s_t)$ toward $\mu(y_t|s_t)$, the weight becomes $w_t = 1 + D_t^{\text{Bin-TV}}/\delta$. The gradient is therefore amplified rather than suppressed, encouraging the policy to move smoothly back toward the behavior policy.

Takeaway. The two cases show that **DRPO preserves the same trust-region geometry as DPPO**, but replaces the brittle hard mask with continuous gradient reweighting whose strength is set by δ. For $\delta < 1$, the weight vanishes at the boundary and reverses beyond it, rolling back over-shifted tokens; for $\delta \geq 1$ (all our experiments), $w_t \in [1 - \frac{1}{\delta}, 1 + \frac{1}{\delta}]$ is never negative, and DRPO attenuates diverging and amplifies converging updates by each token's absolute probability shift.

3.2 Comparison with SPO

To justify why the probability factor in Equation (8) is essential, we compare DRPO and SPO from two perspectives: the divergence each method implicitly regularizes, and the stability of the resulting per-token gradient weight. Table 1 summarizes the key design differences across the four objectives.

Implicit regularizer: ℓ_2^2 versus χ^2. For a fixed state s_t, write $\hat{A}_t(a)$ for the advantage that would be assigned when the sampled token is a. The regularization term in DRPO has expectation

$$\mathbb{E}_{y_t\sim\mu(\cdot|s_t)}\left[|\hat{A}_t(y_t)|\,\mu(y_t|s_t)(r_t-1)^2\right] = \sum_{a\in\mathcal{A}} |\hat{A}_t(a)|\,\big(\pi(a|s_t) - \mu(a|s_t)\big)^2.$$

Thus DRPO penalizes an advantage-weighted squared ℓ_2 distance between $\pi(\cdot|s_t)$ and $\mu(\cdot|s_t)$. In contrast, SPO uses the same quadratic form without the factor $\mu(y_t|s_t)$, giving

$$\mathbb{E}_{y_t\sim\mu(\cdot|s_t)}\left[|\hat{A}_t(y_t)|\,(r_t-1)^2\right] = \sum_{a\in\mathcal{A}} |\hat{A}_t(a)|\,\frac{\big(\pi(a|s_t) - \mu(a|s_t)\big)^2}{\mu(a|s_t)}.$$

This is an advantage-weighted Pearson-χ^2 penalty. The advantage weights modulate which tokens matter more for learning, but the key geometric difference comes from the denominator $\mu(a|s_t)$. SPO scales each squared probability shift by $1/\mu(a|s_t)$, making the penalty highly sensitive to deviations on low-probability tokens. DRPO instead penalizes the absolute probability shift directly: at a fixed advantage value, the same shift $|\pi(a|s_t) - \mu(a|s_t)|$ receives the same cost regardless of the token's behavior probability. In this sense, the ℓ_2^2-type penalty is symmetric in π and μ, whereas the χ^2-type penalty is tied to the behavior policy and can be dominated by the low-probability tail of μ.

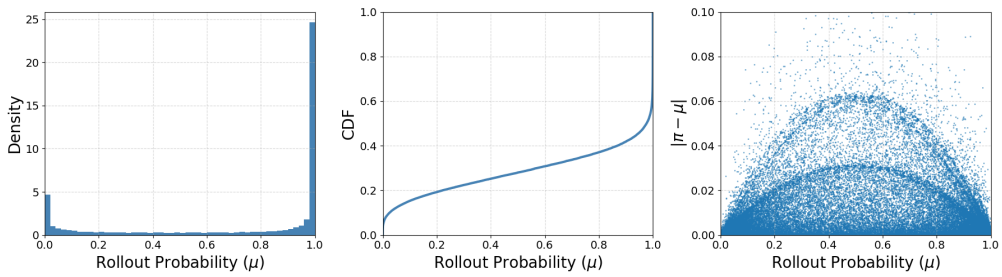

Figure 2: Histogram, cumulative distribution, and absolute probability shift $|\pi - \mu|$ of rollout probabilities $\mu(y_t|s_t)$ for tokens sampled from Qwen3-30B-A3B-Base (Yang et al., 2025). The shift $|\pi - \mu|$ reflects training-inference mismatch. Tokens with $\mu(y_t|s_t) \leq 0.01$ account for 7.8% of all sampled tokens, showing that the low-probability tail is sampled non-negligibly often.

Gradient stability in the long tail. A similar distinction appears in the gradient weights. From Table 1, SPO weights each token by a term involving $|r_t - 1|$. Under $y_t \sim \mu(\cdot|s_t)$, this quantity is an unbiased single-sample Monte Carlo estimator of the unnormalized TV distance (Qi et al., 2026):

$$\mathbb{E}_{y_t \sim \mu(\cdot|s_t)}[|r_t - 1|] = \sum_{a \in \mathcal{A}} |\pi(a|s_t) - \mu(a|s_t)| = 2\, D_{\mathrm{TV}}\big(\mu(\cdot|s_t) \,\|\, \pi(\cdot|s_t)\big).$$

Its variance, however, is

$$\mathrm{Var}_{y_t \sim \mu(\cdot|s_t)}(|r_t - 1|) = \chi^2(\pi(\cdot|s_t) \,\|\, \mu(\cdot|s_t)) - \big(2D_{\mathrm{TV}}\big(\mu(\cdot|s_t) \,\|\, \pi(\cdot|s_t)\big)\big)^2.$$

The χ^2 term contains the factor $1/\mu(a|s_t)$, so the variance can become arbitrarily large when probability mass shifts on tokens with very small behavior probability. This is the typical long-tail regime of LLM sampling. As Figure 2 shows, tokens with $\mu(y_t|s_t) \leq 0.01$ account for 7.8% of all sampled tokens. For these tokens, even a modest absolute probability shift can induce a large ratio change, causing the SPO weight $1 \pm |r_t - 1|/\epsilon$ to dominate the gradient despite a small contribution to the actual distributional shift.

DRPO avoids this instability by replacing $|r_t - 1|$ with $|\pi(y_t|s_t) - \mu(y_t|s_t)|$, which directly measures absolute probability shift and more faithfully reflects the geometry of TV divergence (compare the right panel of Figure 2 with Figure 1 of Qi et al. (2026)). Since it is bounded in $[0, 1]$ for every token, its variance is bounded by $1/4$, and the gradient weight of DRPO is confined to $1 - \frac{1}{\delta} \leq w_t \leq 1 + \frac{1}{\delta}$. Figure 1 illustrates this contrast. SPO's weight grows without bound along the low-μ axis, whereas DRPO remains bounded everywhere. Thus DRPO realizes a smooth version of DPPO's divergence-based trust region while avoiding the high-variance weighting induced by ratio-based regularization.

Why this form. The factor $\mu(y_t|s_t)$ is not a heuristic choice: under a per-token penalty $\frac{c_t}{2}(r_t - 1)^2$, the optimal shift $\pi(y_t|s_t) - \mu(y_t|s_t) = \mu(y_t|s_t)\hat{A}_t/c_t$ equals $\mathrm{sign}(\hat{A}_t)\,\delta$ for every token if and only if $c_t = \mu(y_t|s_t)|\hat{A}_t|/\delta$, i.e., exactly Equation (8).

4 Experiments

Models, Data, and Benchmarks. We perform RL fine-tuning on Qwen3-4B-Base, Qwen3-30B-A3B-Base, and Qwen3.5-35B-A3B-Base (Yang et al., 2025), using a filtered subset of the original DAPO dataset (Yu et al., 2025) that contains approximately 13K math problems with rule-based verification. In addition, we fine-tune DeepSeek-R1-Distill-Qwen-1.5B (R1D) (Guo et al., 2025) on a small sanity test dataset of 1,460 solvable questions (Qi et al., 2025). During training, we evaluate on AIME 2024 and AIME 2025 (MAA, 2025). For each problem, we sample 16 responses and report the average score.

Experimental Settings. We use the VeRL framework (Sheng et al., 2024) for RL training, with BF16 precision by default. For Qwen3-30B-A3B-Base, we additionally consider two low-precision settings: FP8 for rollout only, and FP8 for both training and rollout (FP8-E2E). These settings make optimization more challenging because FP8 precision, together with the MoE architecture, can increase the numerical mismatch between training and inference. Across all settings, we evaluate the unregularized trust-region-free surrogate (Equation (2)), GRPO (Equation (4)), SPO (Equation (5)), DPPO (Equation (6)), and our proposed DRPO (Equation (8)). For GRPO, we adopt the clip-higher trick with $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$, following Yu et al. (2025). For DPPO, we use a mask threshold of 0.2. For SPO and DRPO, we set $\epsilon = \delta = 12.5$ (DRPO: $w_t \in [0.92, 1.08]$). For other hyperparameters and hardware requirements, please refer to Appendix D and Table 2.

4.1 Main Results

We present the main results in Figure 3 (see Appendix D.3 for comparing with KL regularization). Across all six settings, our DRPO consistently enables stable and efficient training, matching or exceeding the best evaluation accuracy achieved by the baselines.

Instability of ratio-based methods. We find that ratio-based methods, namely GRPO and SPO, generally suffer from unstable training. This issue is especially severe in the low-precision settings, where they often collapse before reaching reasonable performance. Even in their strongest settings, their training efficiency and final accuracy lag behind their divergence-based counterparts. Since SPO

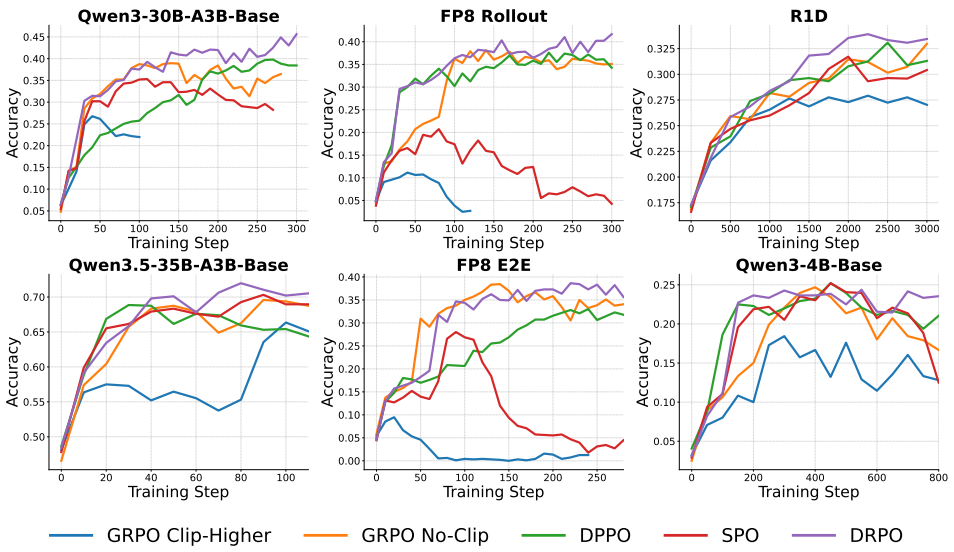

Figure 3: Average accuracy across all main experiment settings on AIME24 and AIME25.

and DRPO share the same coefficient and differ only in the factor $\mu(y_t|s_t)$, the gap between them isolates ratio-based versus Binary-TV geometry, consistent with Qi et al. (2026).

Limitations of a hard mask. Another observation is that hard-mask methods, such as GRPO and DPPO, often underperform their counterparts with smooth regularization. For example, although DPPO trains stably on Qwen3-30B-A3B-Base, it often converges more slowly and reaches lower final accuracy than DRPO. This suggests that, within the same geometry, a weight that varies smoothly with the shift is more effective than a hard mask.

The need for a proper trust region. In some cases, the unregularized trust-region-free surrogate in Equation (2) already achieves strong performance, while a hard mask or ratio-based trust region can degrade performance. However, this unregularized surrogate is not reliable across settings, suffering a performance drop in three of the six settings. The most notable example is in the Qwen3-4B-Base experiment, where the accuracy decreases from 0.25 to 0.17. These results support the claim of Qi et al. (2026) that a trust region remains necessary, but suggest that its geometry is crucial.

4.2 Ablation Studies

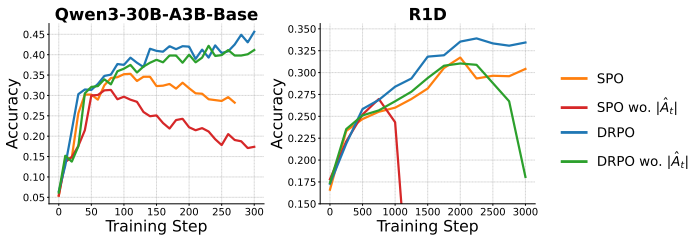

Figure 4: Ablation on $|\hat{A}_t|$. Removing this term degrades performance and destabilizes training.

Advantage weight. In both SPO and DRPO, the regularization term is weighted by the absolute advantage $|\hat{A}_t|$. This weighting ensures that the per-token optimum lies on a stable trust-region boundary that does not depend on the magnitude of the advantage. Without this weighting, the trust-region boundary in Equation (11) would be coupled with $|\hat{A}_t|$, making it sensitive to token-level advantage noise and group-level advantage variance. However, this choice also makes the regularizer advantage-weighted rather than a pure divergence, as used in many prior works (Luo et al., 2026; Becker et al., 2025).

To examine whether $|\hat{A}_t|$ is necessary, we conduct ablations on Qwen3-30B-A3B-Base FP8-E2E and R1D by removing this factor from SPO and DRPO (see Appendix D.5 for this ablation on other alternative regularizations and further discussion). As shown in Figure 4, removing $|\hat{A}_t|$ consistently causes a performance drop and leads to training instability. These results suggest that maintaining a stable trust-region boundary is more important than enforcing a pure divergence form for the regularizer.

Other alternative regularizations. As shown in Section 3.2, the regularizer in DRPO can be interpreted as an advantage-weighted ℓ_2^2 penalty, whereas the regularizer in SPO corresponds to an advantage-weighted χ^2 divergence. This raises a natural question: can other divergence measures yield better performance?

To answer this question, we compare DRPO with several alternatives, including commonly used forward KL and TV penalties (Equations (12) and (14)). As shown in Figure 5, all of these alternatives underperform DRPO. We argue that this result is expected because their per-token gradient weights are binary or ratio-based rather than bounded and smooth in the Binary-TV shift, with the detailed analysis deferred to Appendix C.

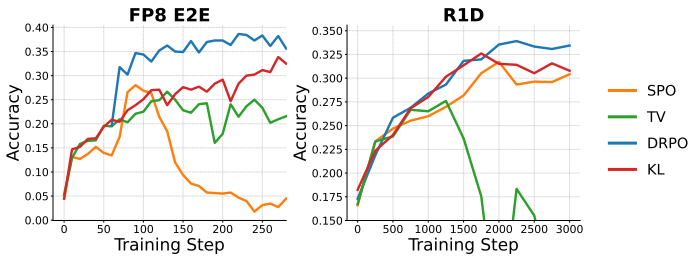

Figure 5: Ablation on alternative divergence metrics. DRPO achieves the best performance.

Applying the regularizer only outside DPPO's trust region. To examine where the performance gain of DRPO primarily comes from, we conduct an experiment in which the regularizer is applied only outside the DPPO trust region. We refer to this variant as Mask-DRPO. Within the DPPO trust-region boundary, Mask-DRPO has the same gradient as DPPO; outside this boundary, it has the same gradient as DRPO. As shown in Figure 11, Mask-DRPO achieves performance comparable to DRPO, suggesting that the main performance gain comes from the corrective regularization outside the trust region. In addition, other regularizer alternatives still do not match DRPO's performance, further supporting the effectiveness of our design. See Appendix D.9 for more details.

5 Conclusion

Many prior works design regularizers from the objective perspective, typically by adopting standard divergence measures such as KL, JS, or related variants. Our empirical results suggest that the induced gradient form is more critical than the nominal divergence in the objective. For example, although the absolute-advantage term in Equations (5) and (8) prevents the regularizer from being a pure divergence, we find that it is essential for maintaining a stable trust-region boundary and enabling stable training.

The choice of regularizer therefore requires careful consideration. A regularization term that appears reasonable at the objective level can perform poorly if its gradient induces undesirable geometry. In particular, we identify a common failure mode in which the gradient induces a ratio-based trust region, whose weights can have high variance and become unbounded under the long-tailed vocabularies of LLMs. In contrast, the absolute probability shift, namely Binary-TV, provides a better alternative: it is bounded and better captures the geometry of TV divergence.

This observation is consistent with DPPO (Qi et al., 2026), which replaces ratio-based clipping in PPO with a divergence-based mask. However, DPPO still relies on a hard mask, which changes the gradient abruptly near the boundary and discards it for diverging tokens beyond it. To address this, we propose DRPO, which replaces the hard mask with a smooth quadratic regularizer while preserving the same trust-region geometry. Across dense and MoE architectures, reasoning and non-reasoning models, and BF16 and FP8 precision settings, DRPO improves training stability and achieves stronger performance than a diverse set of baselines.

AI USE STATEMENT

In this work, we used generative AI tools to polish the writing, including correcting grammatical errors and typos. We also used AI tools to provide feedback on the methodology and experimental design, to assist in implementing parts of the experimental codebase, and to help retrieve related work. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work studies optimization objectives for reinforcement-learning fine-tuning of LLMs on mathematical reasoning and does not involve human subjects or personally identifiable data. All external assets, namely the open-weight Qwen3, Qwen3.5, and DeepSeek-R1-Distill-Qwen-1.5B models, the VeRL framework, the DAPO-Math-17k dataset from which our training data is filtered, and the sanity-test set of Qi et al. (2025), are used for research purposes only and under their respective licenses and terms of use. We evaluate on AIME 2024 and AIME 2025, whose problems are copyrighted by the Mathematical Association of America; we use them only for research evaluation, report aggregate metrics, and do not redistribute the problems.

REPRODUCIBILITY STATEMENT

Section 3 specifies the DRPO objective and its gradient, with the full derivation in Section B and the analysis of the trust regions induced by alternative regularizers in Section C. Section 4 describes the models, training data, benchmarks, and baselines, and Section D (including Table 2) lists all hyperparameters, training steps, compute resources, and software used. All base models, source datasets, and frameworks used in our experiments are publicly available, and our training data is a filtered subset of the public DAPO-Math-17k dataset. We will release our code upon publication.

REFERENCES

Joshua Achiam, David Held, Aviv Tamar, and Pieter Abbeel. Constrained policy optimization. In *International conference on machine learning*, pp. 22–31. PMLR, 2017.

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, 2024.

Amir Beck and Marc Teboulle. Mirror descent and nonlinear projected subgradient methods for convex optimization. *Operations Research Letters*, 31(3):167–175, 2003.

Philipp Becker, Niklas Freymuth, Serge Thilges, Fabian Otto, and Gerhard Neumann. Troll: Trust regions improve reinforcement learning for large language models. *arXiv preprint arXiv:2510.03817*, 2025.

Richard Bellman. A markovian decision process. *Journal of mathematics and mechanics*, 1957.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Dan Hendrycks, Steven Basart, Saurav Kadavath, Mantas Mazeika, Akul Arora, Ethan Guo, Collin Burns, Samir Puranik, Horace He, Dawn Song, et al. Measuring coding challenge competence with apps. *arXiv preprint arXiv:2105.09938*, 2021.

Naman Jain, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, volume 2025, pp. 58791–58831, 2025.

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

Rongao Li, Jie Fu, Bo-Wen Zhang, Tao Huang, Zhihong Sun, Chen Lyu, Guang Liu, Zhi Jin, and Ge Li. Taco: Topics in algorithmic code generation dataset. *arXiv preprint arXiv:2312.14852*, 2023.

Yujia Li, David Choi, Junyoung Chung, Nate Kushman, Julian Schrittwieser, Rémi Leblond, Tom Eccles, James Keeling, Felix Gimeno, Agustin Dal Lago, et al. Competition-level code generation with alphacode. *Science*, 378(6624):1092–1097, 2022.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, et al. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*, 2025a.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying rl collapse from the inference-training mismatch, 2025b. https://yingru.notion.site/When-Speed-Kills-Stability-Demystifying-RL-Collapse-from-the-Inference-Training-Mismatch-271211a558b7808d8b12d403fd15edda.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

Yu Luo, Shuo Han, Yihan Hu, Dong Li, and Jianye Hao. Ratio-variance regularized policy optimization for efficient llm fine-tuning. *arXiv preprint arXiv:2601.03320*, 2026.

MAA. American invitational mathematics examination - aime. `https://maa.org/`, 2025.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Penghui Qi, Zichen Liu, Xiangxin Zhou, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Defeating the training-inference mismatch via fp16. *arXiv preprint arXiv:2510.26788*, 2025.

Penghui Qi, Xiangxin Zhou, Zichen Liu, Tianyu Pang, Chao Du, Min Lin, and Wee Sun Lee. Rethinking the trust region in llm reinforcement learning. *arXiv preprint arXiv:2602.04879*, 2026.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *International conference on machine learning*, pp. 1889–1897. PMLR, 2015.

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

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

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. *arXiv preprint arXiv: 2409.19256*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-lm: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025a.

Kimi Team, Tongtong Bai, Yifan Bai, Yiping Bao, SH Cai, Yuan Cao, Y Charles, HS Che, Cheng Chen, Guanduo Chen, et al. Kimi k2. 5: Visual agentic intelligence. *arXiv preprint arXiv:2602.02276*, 2026.

Ling Team, Anqi Shen, Baihui Li, Bin Hu, Bin Jing, Cai Chen, Chao Huang, Chao Zhang, Chaokun Yang, Cheng Lin, et al. Every step evolves: Scaling reinforcement learning for trillion-scale thinking model. *arXiv preprint arXiv:2510.18855*, 2025b.

Manan Tomar, Lior Shani, Yonathan Efroni, and Mohammad Ghavamzadeh. Mirror descent policy optimization. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=aBO5SvgSt1.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.

Yuhui Wang, Hao He, Xiaoyang Tan, and Yaozhong Gan. Trust region-guided proximal policy optimization. *Advances in Neural Information Processing Systems*, 32, 2019.

Yuhui Wang, Hao He, and Xiaoyang Tan. Truly proximal policy optimization. In *Uncertainty in artificial intelligence*, pp. 113–122. PMLR, 2020.

Zhengpeng Xie, Qiang Zhang, Fan Yang, Marco Hutter, and Renjing Xu. Simple policy optimization. *arXiv preprint arXiv:2401.16025*, 2024.

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. *arXiv preprint arXiv:2505.09388*, 2025.

Feng Yao, Liyuan Liu, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Your efficient rl framework secretly brings you off-policy rl training, August 2025. https://fengyao.notion.site/off-policy-rl.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, et al. Stabilizing reinforcement learning with llms: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025a.

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. Prosperity before collapse: How far can off-policy rl reach with stale data on llms? *arXiv preprint arXiv:2510.01161*, 2025b.

A Related Work

A.1 Traditional RL based on Trust Region Methods

Trust region methods ensure stable policy optimization by limiting how much the policy can change in each update. TRPO (Schulman et al., 2015) derives a policy improvement bound penalized by TV divergence and solves the resulting KL-constrained optimization via conjugate gradient, guaranteeing monotonic improvement. CPO (Achiam et al., 2017) extends this to constrained MDPs. However, both require second-order optimization that is prohibitive at scale.

PPO (Schulman et al., 2017) replaces the explicit KL constraint with a ratio-clipping heuristic, enabling first-order optimization. Despite its success, the clipping mechanism neither strictly bounds the likelihood ratio nor enforces a well-defined divergence constraint (Wang et al., 2020). Truly PPO (Wang et al., 2020) addresses this by introducing a rollback clipping function with a KL-based triggering condition. Trust Region-Guided PPO (Wang et al., 2019) proposes adaptive clipping thresholds guided by KL divergence, providing stronger guarantees than fixed-width clipping. MDPO (Tomar et al., 2022) connects trust-region policy optimization with mirror descent (Beck & Teboulle, 2003), approximately solving the trust-region subproblem via multiple gradient steps on a Bregman divergence objective rather than enforcing a hard constraint.

Most relevant to our work, SPO (Xie et al., 2024) replaces PPO's hard clipping with a smooth quadratic regularizer on the importance ratio. The per-token optimum of the resulting concave quadratic exactly matches PPO's clipping boundary, while providing non-zero corrective gradients outside the trust region. Our method adopts SPO's smooth regularization principle but changes the trust-region geometry from ratio-based to divergence-based. Specifically, we weight SPO's quadratic penalty by the behavior probability $\mu(y_t|s_t)$, which transforms the implicit regularization from a χ^2-type penalty to an ℓ_2^2-type penalty on probability shifts. This single modification changes the per-token optimum from the ratio boundary $|r_t - 1| = \epsilon$ to the Binary-TV boundary $|\pi(y_t|s_t) - \mu(y_t|s_t)| = \delta$, inheriting the smooth gradient structure of SPO while aligning the trust region with the TV geometry of DPPO.

A.2 RL for LLM Reasoning

Reinforcement learning has become a key technique for improving reasoning in LLMs (Guo et al., 2025; Team et al., 2025a). In practice, LLM RL is inherently off-policy due to training-inference mismatch (Yao et al., 2025; Qi et al., 2025) and mini-batch policy staleness (Liu et al., 2025a), making trust-region optimization essential for stable training.

The dominant approach uses PPO-style hard clipping to impose ratio-based trust regions. GRPO (Shao et al., 2024; Liu et al., 2025c) retains this objective while replacing critic-based advantages with group-relative advantages (Liu et al., 2025c; Zeng et al., 2025). DAPO (Yu et al., 2025) asymmetrically widens the upper clipping bound, CISPO (Chen et al., 2025) removes clipping through truncated importance sampling, and M2PO (Zheng et al., 2025b) constrains the second moment of importance weights. To reduce variance under off-policy data, prior work has also proposed truncated (Yao et al., 2025; Zheng et al., 2025a) and masked (Liu et al., 2025b; Team et al., 2025b) importance sampling.

Another line of work uses regularization to enforce trust-region behavior instead of relying on hard clipping or masking. Kimi k1.5 (Team et al., 2025a) and Kimi K2.5 (Team et al., 2026) adopt online policy mirror descent. R^{2}VPO (Luo et al., 2026) replaces hard clipping with a smooth Lagrangian penalty on ratio variance, but it remains ratio-based and can induce unbounded gradient weights for low-probability tokens. TROLL (Becker et al., 2025) enforces per-token KL constraints through differentiable projections, but requires solving an optimization problem for each token.

These mask-based and regularizer-based methods either remain tied to the importance ratio or adjust the trust region heuristically, without directly resolving the mismatch between ratio change and distributional shift. DPPO (Qi et al., 2026) identifies this flaw in long-tailed vocabularies (Wang et al., 2025) and replaces ratio clipping with a divergence-based binary mask on TV or KL divergence. However, DPPO still changes gradients abruptly at the boundary and discards the gradient entirely once a token moves outside the trust region in a harmful direction.

DRPO combines the divergence-based geometry of DPPO with the smooth enforcement principle used by regularizer-based methods such as R^{2}VPO, while avoiding their main limitations. DRPO preserves the directional structure of PPO and DPPO: it attenuates updates that move the policy away from the behavior policy and amplifies updates that move it back. Through a lightweight advantage-weighted ℓ_2^2 regularizer, DRPO aligns the update with Binary-TV geometry, reweights gradients smoothly according to the absolute probability shift, and keeps per-token gradient weights bounded.

B Detailed Derivation of the Gradient of DRPO

The gradient of the objective in Equation (8) can be derived as follows:

$$
\begin{aligned}
&\nabla \mathcal{L}_{\text{DRPO}}(x, \pi) \\
&= \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} \nabla r_t \hat{A}_t - \frac{|\hat{A}_t|}{\delta}\,\mu(y_t|s_t)(r_t-1)\nabla r_t\right] \\
&= \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} \left(1 - \operatorname{sign}(\hat{A}_t)\frac{\mu(y_t|s_t)(r_t-1)}{\delta}\right)\nabla r_t \hat{A}_t\right] \\
&= \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} \left(1 - \operatorname{sign}(\hat{A}_t(r_t-1))\frac{|\pi(y_t|s_t) - \mu(y_t|s_t)|}{\delta}\right) r_t \hat{A}_t \nabla \log \pi(y_t|s_t)\right] \\
&= \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} \left(1 - \operatorname{sign}(\hat{A}_t(r_t-1))\frac{D_t^{\text{Bin-TV}}}{\delta}\right) r_t \hat{A}_t \nabla \log \pi(y_t|s_t)\right].
\end{aligned}
$$

C Induced Trust Regions of Alternative Regularizers

We analyze the trust region induced by each alternative regularizer through its per-token gradient. Fix a state s_t and a sampled token y_t. For compactness, denote

$$
\mu_t \triangleq \mu(y_t|s_t), \qquad \pi_t \triangleq \pi(y_t|s_t), \qquad r_t \triangleq \frac{\pi_t}{\mu_t}.
$$

This appendix is intended to clarify a subtle point in regularizer design. Two objectives can look similar at the loss level but induce very different gradient geometries after importance sampling. For LLM RL, this distinction is important because the optimization update is driven by sampled tokens from a highly long-tailed vocabulary. A useful trust-region regularizer should therefore be judged not only by the name of the divergence it resembles, but also by the scalar weight it applies to the token-level policy gradient.

We consider the following alternative regularizers:

$$
\mathcal{L}_{\text{KL}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \cdot \hat{A}_t + \frac{|\hat{A}_t|}{2\delta} \cdot \log r_t\right], \tag{12}
$$

$$
\mathcal{L}_{\text{K3}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \cdot \hat{A}_t - \frac{|\hat{A}_t|}{2\delta} \cdot (r_t - 1 - \log r_t)\right], \tag{13}
$$

$$
\mathcal{L}_{\text{TV}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \cdot \hat{A}_t - \frac{|\hat{A}_t|}{2\delta} \cdot |r_t - 1|\right], \tag{14}
$$

Since μ_t is fixed during the policy update, $\nabla r_t = r_t \nabla \log \pi_t$. We therefore write each gradient as the original policy-gradient term $r_t \hat{A}_t \nabla \log \pi_t$ multiplied by an induced weight. The zero of this weight gives the boundary at which the regularizer cancels the reward-improving gradient. When this

boundary is expressed as a fixed value of r_t, the regularizer inherits the same ratio-based geometry as PPO and SPO. When the boundary is expressed as a fixed value of $|\pi_t - \mu_t|$, it matches the Binary-TV geometry used by DRPO and DPPO.

Advantage-weighted KL regularizer. Consider the per-token KL-regularized objective

$$\ell_{\mathrm{KL}}(r_t) = r_t \hat{A}_t + \frac{|\hat{A}_t|}{2\delta} \log r_t. \tag{15}$$

This is the sampled contribution of the forward KL penalty $D_{\mathrm{KL}}(\mu\|\pi)$ under the behavior-policy expectation, up to the sign convention induced by maximizing the objective. Taking the gradient gives

$$\begin{aligned} \nabla \ell_{\mathrm{KL}}(r_t) &= \left(r_t \hat{A}_t + \frac{|\hat{A}_t|}{2\delta} \right) \nabla \log \pi_t \\ &= \left(1 + \frac{\mathrm{sign}(\hat{A}_t)}{2\delta r_t} \right) r_t \hat{A}_t \nabla \log \pi_t. \end{aligned} \tag{16}$$

Thus the KL-induced gradient weight is

$$w_{\mathrm{KL}}(r_t) = 1 + \frac{\mathrm{sign}(\hat{A}_t)}{2\delta r_t}.$$

A key observation is that the gradient weight only depends on r_t, which leads to a ratio-based geometry. Setting $w_{\mathrm{KL}}(r_t) = 0$ yields

$$r_t^\star = -\frac{\mathrm{sign}(\hat{A}_t)}{2\delta}. \tag{17}$$

For $\hat{A}_t > 0$, this equation has no feasible solution because $r_t > 0$ and the gradient weight is always positive. For $\hat{A}_t < 0$, the zero-gradient point is

$$r_t^\star = \frac{1}{2\delta}, \qquad \pi_t^\star = \frac{\mu_t}{2\delta}.$$

Therefore, whenever the KL penalty induces a finite cancellation boundary, that boundary is ratio-based. The stopping condition depends on π_t/μ_t, not on the absolute probability shift. This also explains why directly adding a KL penalty is not a drop-in replacement for DRPO. For positive-advantage tokens, the sampled forward-KL term does not create a finite rollback point in this one-sample gradient form; for negative-advantage tokens, the rollback point scales with μ_t. Consequently, a rare token and a frequent token can receive very different absolute probability tolerances even when their semantic effect on the next-token distribution should be judged by probability mass rather than by relative ratio.

Advantage-weighted KL regularizer with the K3 estimator. The previous objective uses the K1 estimator $-\log r_t$ for $D_{\mathrm{KL}}(\mu\|\pi)$, which can have high variance. A common lower-variance alternative is the K3 estimator

$$k_3(r_t) = r_t - 1 - \log r_t,$$

which has the same expectation under $y_t \sim \mu(\cdot|s_t)$ because $\mathbb{E}_{y_t\sim\mu}[r_t - 1] = 0$. The corresponding per-token objective is

$$\ell_{\mathrm{KL3}}(r_t) = r_t \hat{A}_t - \frac{|\hat{A}_t|}{2\delta}\big(r_t - 1 - \log r_t\big). \tag{18}$$

Taking the gradient gives

$$\begin{aligned} \nabla \ell_{\mathrm{KL3}}(r_t) &= \left(r_t \hat{A}_t - \frac{|\hat{A}_t|}{2\delta}(r_t - 1) \right) \nabla \log \pi_t \\ &= \left(1 - \frac{\mathrm{sign}(\hat{A}_t)(r_t - 1)}{2\delta r_t} \right) r_t \hat{A}_t \nabla \log \pi_t. \end{aligned} \tag{19}$$

Thus the K3-induced gradient weight is

$$w_{\mathrm{KL3}}(r_t) = 1 - \frac{\operatorname{sign}(\hat{A}_t)(r_t - 1)}{2\delta r_t},$$

which also gives a ratio-based geometry because it only depends on r_t. Setting $w_{\mathrm{KL3}}(r_t) = 0$ yields

$$r_t^\star = \begin{cases} \frac{1}{1-2\delta}, & \hat{A}_t > 0 \text{ and } \delta < \frac{1}{2}, \\ \frac{1}{1+2\delta}, & \hat{A}_t < 0. \end{cases}$$

For $\hat{A}_t > 0$ and $\delta \geq \frac{1}{2}$, the gradient weight remains positive for all feasible $r_t > 0$, so no finite cancellation boundary exists. When a finite boundary does exist, it is again expressed as a fixed value of the importance ratio $r_t = \pi_t/\mu_t$. The K3 estimator reduces the variance of the KL estimate, but it still induces a ratio-based trust region. In other words, K3 changes the estimator but not the relevant geometry. It can make the KL estimate numerically better behaved, yet the regularizing force is still calibrated in ratio space. This is the key mismatch for long-tailed language-model distributions: a small absolute movement on a low-probability token can dominate the gradient through the ratio factor r_t as r_t grows large, while a much larger movement on a high-probability token may appear modest in ratio terms.

Advantage-weighted TV regularizer. Now consider the per-token TV-regularized objective

$$\ell_{\mathrm{TV}}(r_t) = r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\delta}|r_t - 1|. \tag{20}$$

For $r_t \neq 1$, its gradient is

$$\begin{aligned} \nabla\ell_{\mathrm{TV}}(r_t) &= \left(r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\delta} r_t \operatorname{sign}(r_t - 1) \right) \nabla \log \pi_t \\ &= \left(1 - \frac{\operatorname{sign}(\hat{A}_t)\operatorname{sign}(r_t - 1)}{2\delta} \right) r_t\hat{A}_t \nabla \log \pi_t. \end{aligned} \tag{21}$$

Thus the TV-induced gradient weight is

$$w_{\mathrm{TV}}(r_t) = 1 - \frac{\operatorname{sign}(\hat{A}_t)\operatorname{sign}(r_t - 1)}{2\delta}. \tag{22}$$

This weight takes only two values:

$$w_{\mathrm{TV}}(r_t) = \begin{cases} 1 - \frac{1}{2\delta}, & \operatorname{sign}\big(\hat{A}_t(r_t - 1)\big) > 0, \\ 1 + \frac{1}{2\delta}, & \operatorname{sign}\big(\hat{A}_t(r_t - 1)\big) < 0. \end{cases}$$

It depends only on whether the current ratio shift has the same sign as the advantage. It does not depend on the magnitude of $|r_t - 1|$. The advantage-weighted TV penalty therefore induces a binary gradient weight, not a smooth trust-region boundary.

This behavior is undesirable for a different reason from KL. The TV penalty removes the unbounded ratio magnitude, but the sampled absolute-value form has a nondifferentiable kink at $r_t = 1$ and a piecewise-constant gradient weight away from that point. As a result, it distinguishes only whether the update is moving away from or toward the behavior policy, not how far the token has moved. It therefore cannot reproduce the displacement-proportional attenuation of diverging updates (and amplification of converging ones) that DRPO provides for any δ, nor, for $\delta < 1$, its zero crossing at the Binary-TV boundary.

Summary. The above derivations show that the nominal divergence in the objective is not sufficient to determine whether a method has the desired trust-region behavior. KL and K3 penalties induce ratio-based boundaries; the sampled TV penalty induces a two-level gradient weight; and none of them yields a smooth Binary-TV boundary. By contrast, the DRPO regularizer in Equation (8) produces the weight

$$1 - \operatorname{sign}(\hat{A}_t(r_t - 1))\frac{|\pi_t - \mu_t|}{\delta},$$

which depends continuously on the absolute probability shift. This is the property that lets DRPO preserve DPPO's divergence-based trust-region geometry while replacing the hard mask with a smooth update. The empirical comparisons in Appendix D and Appendix D.9 are consistent with this analysis: penalties whose gradients remain ratio-based or binary are less stable than the Binary-TV quadratic penalty.

D More Experimental Details

Table 2: Hyperparameters.

Hyperparameters	**Qwen3-4B-Base**	**Qwen3-30B-A3B-Base**	**Qwen3.5-35B-A3B-Base**	**R1D**
Learning Rate	1e-6	1e-6	1e-6	1e-6
PPO Epochs	1	1	1	1
Max Prompt Length	2048	2048	2048	2048
Max Response Length	8192	8192	8192	8192
Train Batch Size	64	256	256	64
PPO Mini Batch Size	32	32	32	16
Rollout Temperature	1.0	1.0	1.0	1.0
Group Size	8	16	16	8

We provide the detailed experiment configurations, more ablation studies, and results in this section as a complementary part of Section 4.

For the computation resources, we use 4×8 NVIDIA H20 to conduct most of the experiments. We build our codebase on VeRL (Sheng et al., 2024) and use Megatron (Shoeybi et al., 2019) as the training backend and vLLM (Kwon et al., 2023) as the inference backend to speed up rollout. To verify the correctness of the solutions in math reasoning tasks, we utilize the third-party library math-verify[2].

Besides, we have tried various kinds of objective functions, revealing the effects of the advantage scaling $|\hat{A}_t|$, different divergences, binary approximation, etc. Typically, we train Qwen3-4B-Base with 800 steps, Qwen3-30B-A3B-Base with 300 steps, Qwen3.5-35B-A3B-Base with 110 steps, and R1D with 3000 steps. Since we used Megatron (Shoeybi et al., 2019) as the training backend, and at the time we conducted experiments, it did not have sufficient support for efficiently training Qwen3.5, we chose to train fewer steps compared to Qwen3-30B-A3B-Base.

D.1 Final Accuracy of the Main Experiments

Table 3 lists the final accuracy of each run in Figure 3. DRPO achieves the highest final accuracy in all six settings, ahead of the strongest baseline by 0.4 to 7.2 points.

Table 3: Final accuracy (%, mean over AIME 2024 and AIME 2025, mean@16) of each run in Figure 3. Δ is DRPO minus the best baseline.

	GRPO					
Setting	Clip-Higher	No-Clip	DPPO	SPO	DRPO	Δ
Qwen3-4B-Base	12.8	16.7	21.0	12.5	**23.5**	+2.5
Qwen3-30B-A3B-Base	22.0	36.5	38.4	28.2	**45.6**	+7.2
+ FP8 Rollout	2.7	35.1	34.3	4.3	**41.7**	+6.6
+ FP8 E2E	1.2	34.1	31.8	4.5	**35.6**	+1.5
Qwen3.5-35B-A3B-Base	65.1	68.8	64.4	69.0	**70.5**	+1.5
R1D	27.0	33.0	31.3	30.4	**33.4**	+0.4

[2] https://github.com/huggingface/Math-Verify

D.2 CODE GENERATION

To test whether DRPO generalizes beyond mathematical reasoning, we train Qwen3-8B (Yang et al., 2025) with thinking mode disabled on 22,440 code problems from Eurus-2-RL-Data (Cui et al., 2025), drawn from CodeContests (Li et al., 2022), TACO (Li et al., 2023), APPS (Hendrycks et al., 2021), and Codeforces. The reward is computed by executing the generated program against test cases in SandboxFusion[3], and we evaluate on LiveCodeBench v6 (Jain et al., 2025). Table 4 summarizes the setup.

Table 4: Setup of the code-generation experiment.

Item	Configuration
Model	Qwen3-8B with thinking mode disabled
Training data	22,440 problems from Eurus-2-RL-Data: CodeContests (9,628), TACO (6,873), APPS (3,343), Codeforces (2,596)
Evaluation data	LiveCodeBench (LCB) v6
Reward	Execution-based, computed with SandboxFusion
Prompt / response length	2,048 / 4,096 tokens
Rollout	Train batch size 256, mini-batch size 64, 8 rollouts per prompt

As shown in Table 5, DRPO achieves the best LiveCodeBench v6 score (50.2, $+1.7$ over DPPO) and the highest weighted-mean training accuracy over the four sources (62.0, $+1.1$ over SPO). It ranks first on APPS and Codeforces and is within 0.2 and 0.4 points of the best result on CodeContests and TACO. Although the reward is still execution-based rather than a learned reward model, this setting differs from mathematical reasoning in both task and response structure, and the advantage of DRPO carries over.

Table 5: Code-generation results with Qwen3-8B. The first five columns report training accuracy (%), where "Weighted" is the weighted mean over the four sources; the last column reports LiveCodeBench v6 mean@8 (%).

Method	Weighted	APPS	CodeContests	Codeforces	TACO	LCB v6
GRPO Clip-Higher	55.2	59.5	57.0	56.7	49.9	44.8
GRPO No-Clip	55.9	59.4	57.7	59.5	50.2	44.7
DPPO	58.7	63.8	58.7	58.7	**56.3**	48.5
SPO	60.9	62.7	**64.6**	60.2	55.6	44.5
DRPO	**62.0**	**66.1**	64.4	**64.5**	55.9	**50.2**

D.3 COMPARING WITH KL REGULARIZATION

In addition to the baselines in Section 4.1, another common method is to use a pure KL regularizer (without the advantage weight) as in the Algorithm 1 of Schulman et al. (2015) and the Equation (8) of Schulman et al. (2017). We conduct an experiment to compare with this method. Specifically, we instantiate the below objective

$$\mathcal{L}_{\text{KL_wo_A}}(x, \pi) = \mathbb{E}_{y \sim \mu(\cdot|x)} \left[\sum_{t=1}^{|y|} r_t \cdot \hat{A}_t + \frac{1}{2\delta} \cdot \log r_t \right]$$

with the same hyperparameter $\delta = 12.5$ (see the hyperparameter tuning results in Figure 8).

As shown in Figure 6, DRPO consistently outperforms this KL regularizer across all six experiments. This gap can be explained from two complementary perspectives. First, the regularizer should adapt to the per-token advantage scale: as shown in Appendix D.5, removing the factor $|\hat{A}_t|$ degrades performance because the token-wise optimum depends on the current advantage magnitude. Second,

[3] https://github.com/bytedance/SandboxFusion



Figure 6: Training dynamics for DRPO and directly applying a KL penalty term without introducing the advantage weight $|\hat{A}_t|$.

even after setting the advantage-weight issue aside, the KL penalty still induces a ratio-based trust-region geometry as analyzed in Appendix C, which is less aligned with the desired constraint than DRPO.

Table 6 reports the final accuracies. DRPO outperforms the KL penalty in every setting, and the gap is especially large under FP8 precision and on R1D, where the KL penalty degrades sharply or collapses.

Table 6: Final accuracy (%, mean over AIME 2024 and AIME 2025, mean@16) of DRPO and the KL penalty without the advantage weight, both with $\delta = 12.5$.

Setting	DRPO	KL wo. $\lvert\hat{A}_t\rvert$	Δ
Qwen3-4B-Base	**23.5**	19.7	+3.8
Qwen3-30B-A3B-Base	**45.6**	39.5	+6.1
FP8 Rollout	**41.7**	13.6	+28.1
FP8 E2E	**35.6**	10.0	+25.6
Qwen3.5-35B-A3B-Base	**70.5**	68.4	+2.1
R1D	**33.4**	0.0	+33.4

D.4 Adding a Reference-KL Penalty

None of the math runs in Section 4 uses a KL penalty toward the reference or behavior policy. This follows recent reasoning-RL recipes (Yu et al., 2025; Liu et al., 2025c), which set GRPO's reference-KL coefficient (Shao et al., 2024) to zero, so that the comparison isolates each method's trust-region mechanism; in PPO (Schulman et al., 2017), the KL-penalty variant is an alternative to the clipped surrogate rather than a required component. The DRPO regularizer acts between the current and behavior policies and should not be confused with a KL-to-reference term.

To check that the comparison still holds when a reference-KL penalty is used, we rerun all methods on Qwen3-4B-Base with a KL penalty toward the reference policy in the loss. The KL coefficient is 10^{-3}, and the KL is estimated with the low-variance K3 estimator. All other configurations are the same as in the main Qwen3-4B-Base experiment. As shown in Table 7, DRPO remains the best method in this setting, outperforming GRPO Clip-Higher by 5.6 points. GRPO No-Clip collapses

even with the penalty, indicating that a reference-KL penalty at this coefficient does not substitute for the trust-region control that DRPO provides.

Table 7: Accuracy (%, mean over AIME 2024 and AIME 2025, mean@16) on Qwen3-4B-Base with a reference-KL penalty (coefficient 10^{-3}, K3 estimator) added to every method.

Method	Accuracy
GRPO Clip-Higher	17.6
GRPO No-Clip	collapsed
DPPO	21.6
SPO	21.9
DRPO	**23.2**

D.5 Extended Ablations on Advantage Weighting

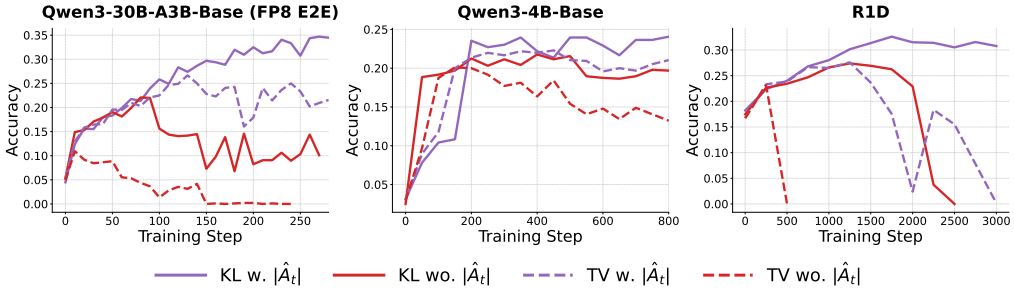

Figure 7: Comparison among experiments applying a KL penalty term or a TV penalty term, with or without the advantage weight $|\hat{A}_t|$.

To further isolate the role of the advantage weight $|\hat{A}_t|$, Figure 7 compares two penalty types, KL and TV, each evaluated both with and without this factor. The pattern is consistent across all settings: adding the advantage weight leads to clearly better performance, supporting the importance of weighting the regularizer by $|\hat{A}_t|$.

This behavior matches the analysis in Section 3.1 and the ablations in Section 4.2. Without $|\hat{A}_t|$, the effective trust-region boundary becomes entangled with the advantage magnitude rather than remaining stable. As a result, the update is overly restrictive for small-advantage tokens and too loose for large-advantage ones. Since token-level advantage estimates are also noisy in practice, this mismatch further hurts both training stability and final accuracy.

The results in Figure 4 are reasonable because $|\hat{A}_t|$ also determines the scale of the per-token policy gradient. Scaling the regularizer by $|\hat{A}_t|$ preserves the same relative regularization strength across tokens with different advantage magnitudes.

D.6 Hyperparameter Tuning of Advantage-Weighted KL Regularizer

To rule out the concern that the KL baseline may simply be under-tuned, we sweep a range of hyperparameters for the advantage-weighted KL regularizer (Equation (13)) under the R1D setting. Figure 8 shows that DRPO remains stronger across the full sweep, even when the KL baseline is equipped with the same advantage weight $|\hat{A}_t|$. As summarized in Table 8, the best non-collapsed KL setting remains about 2.6 points below DRPO, and the strongest KL regularization ($\delta = 2.5$) collapses after about 2,400 steps.

This robustness gap is consistent with our theoretical analysis. As discussed in Appendix C, the KL penalty fundamentally imposes a ratio-based trust-region geometry. By contrast, DRPO yields a Binary-TV geometry, which more faithfully reflects the intended divergence constraint and therefore produces more reliable optimization behavior (see Section 3.1).

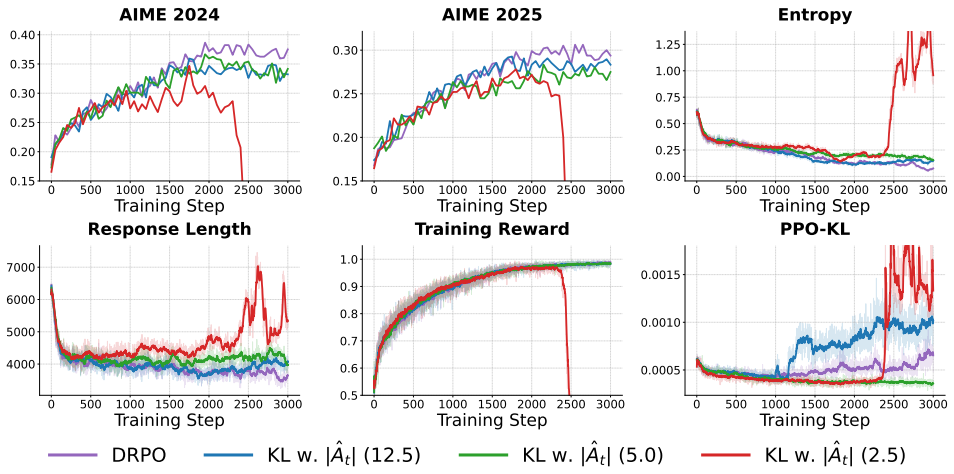

Figure 8: The hyperparameter tuning for KL with advantage weight $|\hat{A}_t|$ under the R1D setting.

Table 8: Final accuracy (%, mean@16) of the advantage-weighted KL regularizer under different δ in the R1D setting.

Objective	δ	AIME 2024	AIME 2025	Average
DRPO	12.5	**37.5**	**29.4**	**33.4**
KL w. $\|\hat{A}_t\|$	12.5	33.2	28.3	30.8
KL w. $\|\hat{A}_t\|$	5.0	34.2	27.5	30.8
KL w. $\|\hat{A}_t\|$	2.5	2.6	5.4	4.0 (collapsed after ≈2,400 steps)

D.7 Hyperparameter Tuning of DPPO Baseline

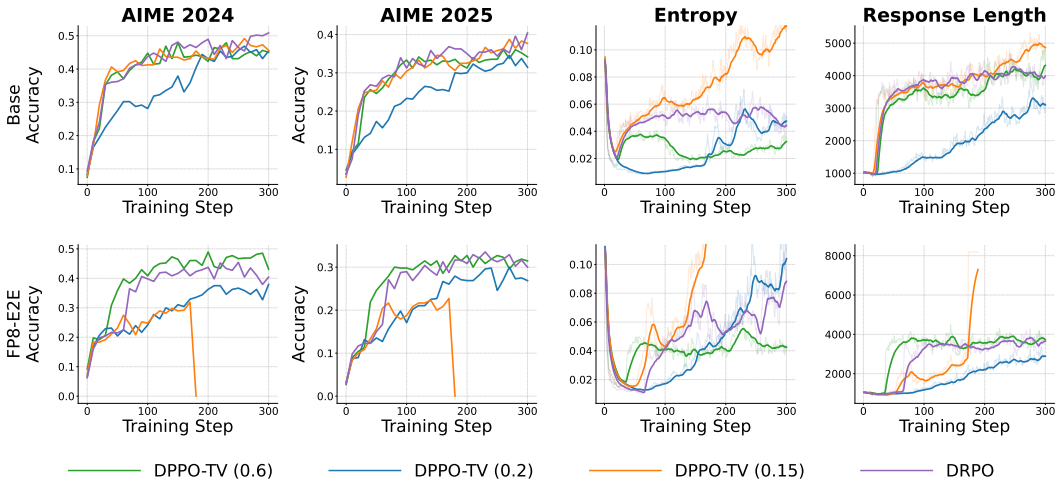

Figure 9: Training dynamics for different parameters under DPPO, compared to DRPO. **Top:** Qwen3-30B-A3B-Base; **Bottom:** Qwen3-30B-A3B-Base using FP8 precision for end-to-end training.

To compare DRPO against a carefully tuned DPPO baseline, we sweep several DPPO thresholds. Unlike DPPO, DRPO retains smoothly reweighted gradients for tokens outside the trust region. Figure 9 shows that DPPO needs a more fine-grained parameter tuning, and threshold 0.15 works best on the Qwen3-30B-A3B-Base setting, which still performs worse than DRPO, while threshold

0.6 works best on the FP8-E2E setting, which achieves similar performance with DRPO. So DRPO has a relatively universal hyperparameter $\delta = 12.5$ compared to DPPO.

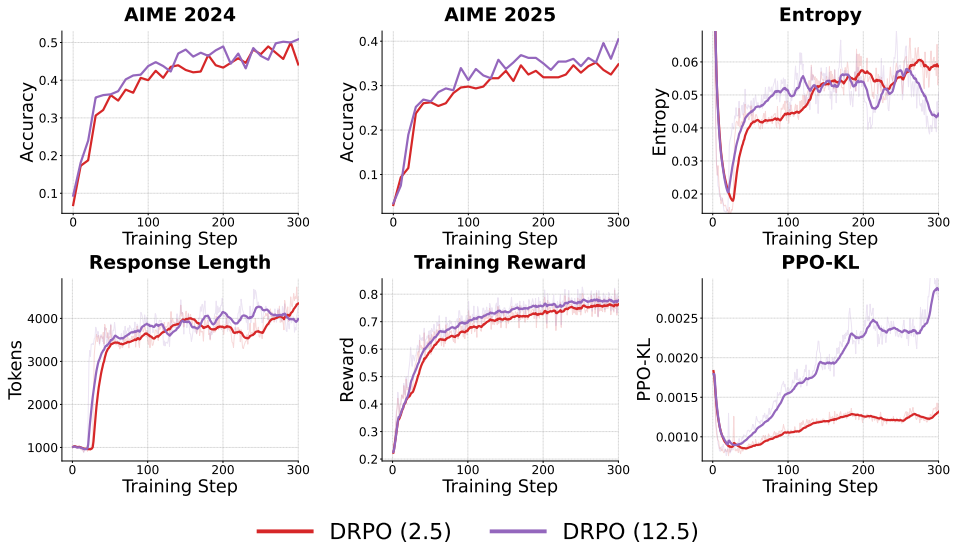

Figure 10: Hyperparameter tuning of the coefficient on DRPO.

D.8 Hyperparameter Tuning of DRPO

To assess the hyperparameter sensitivity of DRPO, we evaluate two values of the coefficient δ. As shown in Figure 10, reducing δ substantially from 12.5 to 2.5, which widens the weight range from $[0.92, 1.08]$ to $[0.6, 1.4]$, leads to only a minor drop in performance. This result suggests that DRPO is relatively robust to the choice of δ. Both values lie in the $\delta \geq 1$ regime of Section 3.1, where the weight never changes sign; the rollback regime $\delta < 1$ is left for future work.

D.9 Mask Ablation with Alternative Divergence Penalties

We further repeat the mask ablation with several choices of divergence penalty. Let $M_t^{\text{out}} = \mathbb{I}[D_t^{\text{Bin-TV}} > \delta]$ denote the indicator that the sampled token is outside the DPPO trust region. For a generic penalty Ω_t, the masked variant uses

$$\mathcal{L}_{\text{Mask-}\Omega}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \hat{A}_t - M_t^{\text{out}} \frac{|\hat{A}_t|}{2\delta}\,\Omega_t\right].$$

Inside the trust region, this objective reduces to the unregularized token surrogate, matching DPPO's active gradient. Outside the trust region, it restores the corrective penalty gradient that DPPO's hard mask discards. Under this framework, we instantiate three objectives as follows

$$\mathcal{L}_{\text{Mask-DRPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \hat{A}_t - M_t^{\text{out}} \frac{|\hat{A}_t|}{2\delta}\,\mu(y_t|s_t)(r_t - 1)^2\right], \tag{23}$$

$$\mathcal{L}_{\text{Mask-SPO}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \hat{A}_t - M_t^{\text{out}} \frac{|\hat{A}_t|}{2\delta}\,(r_t - 1)^2\right], \tag{24}$$

$$\mathcal{L}_{\text{Mask-KL}}(x, \pi) = \mathbb{E}_{y\sim\mu(\cdot|x)}\left[\sum_{t=1}^{|y|} r_t \hat{A}_t - M_t^{\text{out}} \frac{|\hat{A}_t|}{2\delta}\,\frac{(\log r_t)^2}{2}\right]. \tag{25}$$

Notably, $(\log r_t)^2$ in Equation (25) is exactly the penalty term in online policy mirror descent that Kimi series (Team et al., 2025a; 2026) utilized.

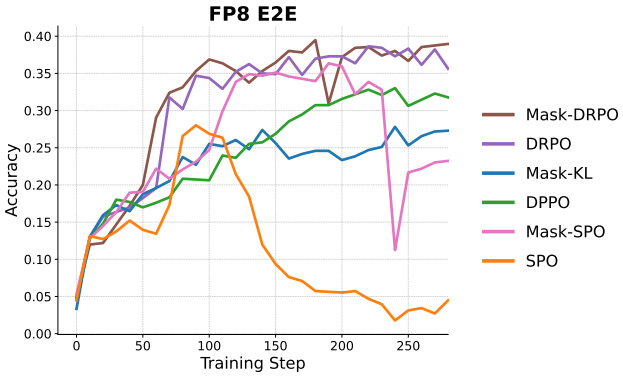

Figure 11: Ablation on applying the DRPO regularizer only outside DPPO's trust region.

As shown in Figure 11, applying the penalty only outside the trust region achieves performance close to applying it everywhere for the DRPO regularizer, confirming that the main gain comes from correcting tokens that have crossed the boundary. At the same time, the penalty choice still matters: ratio-space and KL-type penalties are harder to calibrate, whereas the Binary-TV quadratic penalty used by DRPO gives the best and most stable behavior because its gradient directly follows absolute probability displacement.