BOUNDING TOKEN DIVERGENCE TO UNLOCK STABLE REINFORCEMENT LEARNING FOR LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Decoupled reinforcement learning from verifiable rewards (RLVR) relies on separate inference and training engines that, despite sharing identical weights, assign different token-level probabilities due to floating-point non-determinism and stochastic expert routing. The resulting importance-sampling corruption destabilizes policy-gradient training, particularly for Mixture-of-Experts (MoE) models. We first examine *IcePop*, a token-level ratio mask that eliminates extreme discrepancies between inference and training probabilities and restores training stability. Analysis reveals, however, that any constant ratio threshold over-masks low-probability tokens because the softmax partition function attenuates perturbations for frequent tokens while leaving rare tokens disproportionately vulnerable to them. We then propose *KPop*, which replaces the fixed ratio window with a symmetric binary KL divergence criterion. The binary KL naturally widens its tolerance at low token probability and tightens it at high probability, matching the heteroscedastic noise structure. The symmetric formulation further eliminates directional leakage inherent in one-sided constraints. Experiments on dense and MoE architectures across mathematical reasoning benchmarks show that KPop retains significantly more rare-token gradient signal than existing methods while improving both training stability and final evaluation accuracy.

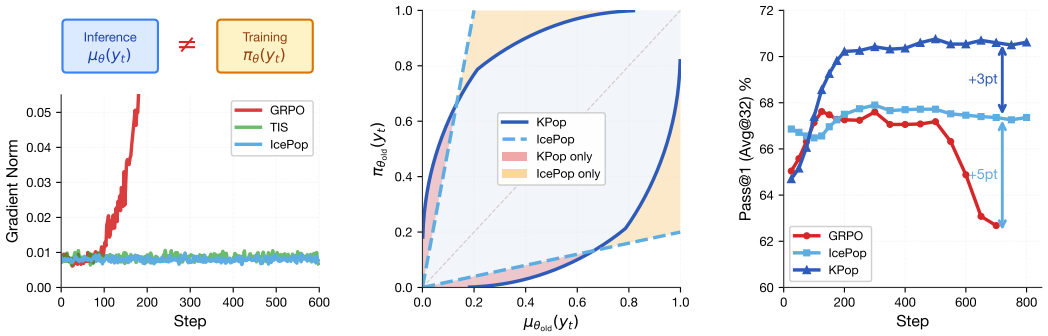


Figure 1: Overview. **(Left)** In decoupled RLVR, a separate inference engine (μ_θ) generates rollouts and a training engine (π_θ) computes gradients. Despite sharing identical weights, the two engines assign different token probabilities due to floating-point non-determinism and MoE routing differences. Vanilla GRPO's gradient norm explodes; IcePop and TIS remain stable. **(Middle)** Acceptance regions in the $(\mu_{\theta_{\text{old}}}(y_t), \pi_{\theta_{\text{old}}}(y_t))$ plane. IcePop's constant-ratio boundary (dashed) over-masks low-probability tokens; KPop's binary KL boundary (solid) is more permissive for rare tokens and stricter for common tokens. **(Right)** AIME 2025 accuracy on Ring-mini (MoE). KPop improves over IcePop by $\sim$3 points, and IcePop improves over vanilla GRPO by $\sim$5 points as GRPO collapses.

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.

1 INTRODUCTION

Reinforcement learning with verifiable rewards (RLVR) has become a standard approach for training large language models on reasoning tasks (Shao et al., 2024; Guo et al., 2025). To scale RLVR to large models, practitioners employ decoupled architectures in which a high-throughput inference engine (e.g., SGLang (Zheng et al., 2024)) generates rollouts and a separate training engine (e.g., Megatron-LM (Shoeybi et al., 2019)) computes gradients and updates weights. Even when both engines load identical model weights, they assign different token-level log-probabilities due to floating-point non-determinism (different reduction orders in parallel summations, fused kernel implementations) and non-deterministic routing in Mixture-of-Experts (MoE) models. We refer to the per-token discrepancy $|\log \pi_{\theta_{\text{old}}}(y_t) - \log \mu_{\theta_{\text{old}}}(y_t)|$ as *token divergence*. When token divergence is large, the importance-sampling (IS) weights that every policy-gradient algorithm relies on become corrupted, leading to severe training instability. Existing corrections such as TIS (Zheng et al., 2025), which caps the per-token ratio $\pi_{\theta_{\text{old}}}(y_t)/\mu_{\theta_{\text{old}}}(y_t)$, mitigate but do not eliminate this problem.

A straightforward mitigation is to mask tokens whose probability ratio between the two engines deviates beyond a fixed window. This approach, **IcePop** (Zhao et al., 2025), stabilizes large-scale MoE training in practice, as we confirm (Figure 1). However, IcePop disproportionately prunes low-probability tokens, which carry much of the gradient signal for exploration (Yu et al., 2025; Liu et al., 2025). We show that this over-masking is not a tuning artifact but a structural consequence of softmax normalization. The partition function absorbs perturbations for high-probability tokens while leaving low-probability tokens vulnerable to them, producing heteroscedastic noise that a constant threshold cannot accommodate.

To address this miscalibration, we propose **KPop** (KL-Pop), a framework that replaces the constant ratio threshold with a symmetric binary KL divergence criterion. The binary KL on the two-event space $\{y_t, \neg y_t\}$, where y_t denotes the actually generated token and $\neg y_t$ aggregates all other vocabulary entries, provides probability-adaptive tolerance that widens for rare tokens and tightens for common ones, matching the noise structure without over-masking. The symmetric formulation further eliminates directional leakage that arises from one-sided constraints. Empirically, KPop retains more rare-token gradient signal than IcePop and achieves higher evaluation accuracy on both dense and MoE architectures. We summarize our main contributions below.

(1) We deploy IcePop, a token-level ratio mask that stabilizes MoE training in practice, and identify through theoretical analysis that its constant cutoff inherently miscalibrates against the heteroscedastic noise induced by softmax normalization.

(2) We introduce a binary KL criterion between the training and inference engines that naturally adapts to token probability, providing a principled replacement for ratio thresholds.

(3) We present KPop rigorously proving that its symmetric binary KL formulation controls bidirectional system noise. KPop is compatible with existing IS correction methods and improves stability across dense and MoE architectures.[1]

2 PRELIMINARIES

2.1 RLVR OBJECTIVE

Given a prompt x and a group of sampled responses $\{y^{(i)}\}_{i=1}^{G}$, GRPO (Shao et al., 2024) optimizes the clipped policy-gradient objective

$$\mathcal{J}(\theta) = \mathbb{E}_{x,\,\{y^{(i)}\}_{i=1}^{G}} \left[\frac{1}{G} \sum_{i=1}^{G} \left(\min\Big(r_i(\theta)\hat{A}_i,\ \text{clip}(r_i(\theta),\, 1-\epsilon_-,\, 1+\epsilon_+)\, \hat{A}_i \Big) - \beta\, \mathbb{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}) \right) \right], \tag{1}$$

where $r_i(\theta) = \pi_\theta(y^{(i)}|x)/\pi_{\theta_{\text{old}}}(y^{(i)}|x)$ is the importance-sampling ratio, $\hat{A}_i = (R(x, y^{(i)}) - \bar{R})/\sigma_R$ is the group-relative advantage with reward R, group mean $\bar{R}$, and standard deviation σ_R, $\epsilon_\pm$ are the clipping bounds, and β weights a KL penalty toward a reference policy π_{ref}. In a single-engine system, the rollout policy and the training policy share the same forward pass, so $\pi_{\theta_{\text{old}}}$ is simply the policy snapshot at rollout time.

[1] Code will be released upon acceptance pending approval.

2.2 Two-Engine Architecture and the Mismatch Problem

In large-scale RLVR, an inference engine (e.g., SGLang) generates rollouts and records per-token log-probabilities, while a training engine (e.g., Megatron-LM) loads the same weights θ_{old}, recomputes a forward pass, and performs the gradient update. We write $\mu_{\theta_{\text{old}}}$ and $\pi_{\theta_{\text{old}}}$ for the distributions computed by the inference and training engines at the same parameters θ_{old}. The shared subscript is deliberate: the parameters are identical, but the two engines are different numerical programs (different kernels, precision, parallel layouts, and, for MoE, possibly different expert selections, §2.3), so for the same prefix they produce slightly different logits and hence $\pi_{\theta_{\text{old}}}(y_t) \neq \mu_{\theta_{\text{old}}}(y_t)$. For each generated token y_t we use the shorthand

$$q_t = \mu_{\theta_{\text{old}}}(y_t), \qquad p_t = \pi_{\theta_{\text{old}}}(y_t), \qquad u_t = \pi_{\theta}(y_t), \tag{2}$$

namely the rollout probability, the training-engine probability before the update, and the training-engine probability during the update. We define the *token divergence* as $|\log p_t - \log q_t|$.

In a single-engine system $p_t = q_t$ by construction; in a decoupled system this equality breaks. Since the tokens were sampled from $\mu_{\theta_{\text{old}}}$, the correct IS denominator in Eq. (1) is q_t, and using p_t instead introduces an uncontrolled error into every gradient step. The question this paper addresses is how to decide, for each token, whether to trust its gradient given p_t and q_t. The challenge is that the acceptable deviation is not constant across probability levels.

2.3 MoE Routing Instability

For dense models, the mismatch between p_t and q_t is mild, arising only from kernel-level floating-point differences such as non-deterministic reduction order and fused operations. For MoE models, the dominant source of mismatch is *non-deterministic top-K routing* (Fedus et al., 2022; Jiang et al., 2024). The router computes gating scores $g_k = \mathrm{softmax}(\mathbf{W}_r \mathbf{h}_t)_k$ and selects the top-K experts. Small hidden-state differences between engines can change the selected subset when expert scores are nearly tied, and a different expert set shifts the entire logit vector. Expert parallelism and low precision (Yan et al., 2026) amplify this, producing extreme deviations between p_t and q_t.

3 IcePop: Stabilization via Ratio Masking

3.1 Filtering Extreme Discrepancy Ratios

In MoE models the ratio p_t/q_t can deviate substantially from one because of routing instability. Most tokens satisfy $p_t/q_t \approx 1$, but a heavy tail extends beyond 5 or below 0.2. These outliers contribute disproportionately to gradient variance, compound across long reasoning sequences, and frequently trigger training collapse (Figure 1). A natural first response is to hard-mask tokens whose discrepancy ratio leaves a predefined window.

Definition 1 (IcePop (Zhao et al., 2025))**.** *IcePop applies the token-level mask $\mathcal{M}_{\text{IcePop}}(t) = \mathbb{1}[p_t/q_t \in [\alpha,\, \beta]]$, with bounds $[\alpha, \beta]$ chosen empirically to remove the heavy tail of the ratio distribution while retaining most tokens. The masked policy loss is*

$$\ell_t^{\text{IcePop}} = \frac{p_t}{q_t} \cdot \mathcal{M}_{\text{IcePop}}(t) \cdot \min\Big(A_t \, \tfrac{u_t}{p_t},\; A_t \, \mathrm{clip}\big(\tfrac{u_t}{p_t},\, 1{-}\epsilon_{-},\, 1{+}\epsilon_{+}\big)\Big), \tag{3}$$

where $A_t = \hat{A}_i$ is the sequence-level advantage applied to every token of response i.

Here p_t/q_t corrects the importance sampling from $\mu_{\theta_{\text{old}}}$ to $\pi_{\theta_{\text{old}}}$, the mask removes corrupted tokens, and the clipped term controls the policy-drift ratio u_t/p_t exactly as in Eq. (1). This thresholding stabilizes training where vanilla GRPO collapses (Figure 1, right; Appendix Figure 7).

3.2 The Over-Masking Problem

Figure 2 shows that tokens in the lowest probability bin are masked at roughly 1.2%, whereas tokens with $p_t > 0.4$ are virtually never masked. Low-probability tokens carry the gradient signal that moves the policy toward new high-reward trajectories (Yu et al., 2025; Liu et al., 2025), so this asymmetric pruning limits reward improvement. The next section shows that any fixed threshold on p_t/q_t removes rare tokens more aggressively, and that a useful masking rule must vary its acceptance width with token probability.

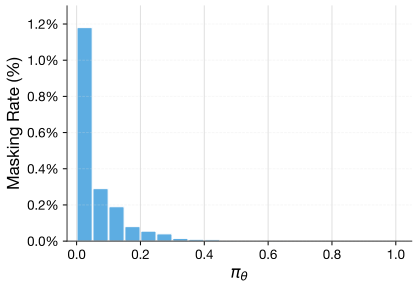


Figure 2: IcePop masking rate by $\pi_{\theta_{\text{old}}}$ probability bin. Tokens with $\pi_{\theta_{\text{old}}}(y_t) < 0.05$ are masked at about 1.2%, whereas tokens with $\pi_{\theta_{\text{old}}}(y_t) > 0.4$ are almost never masked.

4 Analysis of Token Divergence

This section explains why IcePop over-masks rare tokens by analyzing how engine differences propagate through softmax normalization.

4.1 Softmax Absorption

Both engines compute logits $\mathbf{z} = W_{\text{LM}}\,\mathbf{h} \in \mathbb{R}^V$ from a hidden state $\mathbf{h}$ and apply softmax. Because of the numerical differences in §2.3, the hidden states differ between engines. Let $\mathbf{z}$ and $\mathbf{z} + \boldsymbol{\xi}$ be the training- and inference-engine logits, where $\boldsymbol{\xi} = W_{\text{LM}}\,\Delta\mathbf{h}$ is the logit perturbation induced by the hidden-state difference $\Delta\mathbf{h}$, and write $p_j = \text{softmax}(\mathbf{z})_j$ and $q_j = \text{softmax}(\mathbf{z} + \boldsymbol{\xi})_j$. For the generated token $y = y_t$, the log-ratio satisfies the exact identity

$$\log \frac{p_y}{q_y} \;=\; -\xi_y + \log \sum\nolimits_j p_j e^{\xi_j} \;\approx\; -\xi_y + \bar{\xi}, \qquad \bar{\xi} := \textstyle\sum_j p_j \xi_j, \tag{4}$$

where the approximation linearizes the log-partition around $\boldsymbol{\xi} = \mathbf{0}$ (higher-order terms are given in Appendix B.2). When p_y is large, the token's own perturbation contributes heavily to the mean $\bar{\xi}$ and the two terms largely cancel. When p_y is small, it contributes almost nothing to $\bar{\xi}$ and the perturbation survives. Softmax normalization therefore attenuates discrepancy for frequent tokens much more than for rare ones.

Lemma 1 (Heteroscedastic token divergence)**.** *Assume i.i.d. perturbations ξ_j with mean zero and variance σ^2. For a token y with training-engine probability $p = p_y$, to first order,*

$$\text{Var}\Big[\log \tfrac{p_y}{q_y} \,\Big|\, \mathbf{p}\Big] = \sigma^2 \left\|\mathbf{e}_y - \mathbf{p}\right\|^2 = \sigma^2\big[(1-p)^2 + \mathcal{R}\big], \qquad \mathcal{R} = \textstyle\sum_{j \neq y} p_j^2, \tag{5}$$

where $\mathbf{e}_y$ is the indicator vector of y. When $\mathcal{R} \ll (1-p)^2$, $\text{Std}[\log(p_y/q_y) \mid \mathbf{p}] \approx \sigma(1-p)$.

The proof is in Appendix A.1. Note that $\mathbf{e}_y - \mathbf{p} = \nabla_{\mathbf{z}} \log p_y$ is exactly the logit gradient of the token's log-likelihood. The engine noise on a token's log-ratio is therefore proportional to the size of that token's own policy gradient. A constant threshold on $|\log(p_t/q_t)|$ rejects rare tokens far more often than common ones, not because they are more corrupted, but because their noise floor is structurally higher, and it does so in proportion to how much each token could change the policy. This is the over-masking in Figure 2.

Empirical verification. The i.i.d. assumption is a tractable approximation, since $\boldsymbol{\xi} = W_{\text{LM}}\,\Delta\mathbf{h}$ is correlated across the vocabulary. The predicted $(1-p)$ scaling nevertheless holds for synthetic noise ($R^2 \in [0.71, 0.97]$), LM-head perturbations of real dense and MoE models (R^2=0.79, 0.93), and deeper perturbations for MoE tokens with unchanged routing (R^2=0.89); routing-flipped tokens show a higher, probability-independent noise floor (Appendix B.3). The mask introduced next evaluates exact divergences and does not rely on the approximation.

4.2 Design Requirements

The analysis yields two requirements: (1) the tolerance must be probability-adaptive, since discrepancy noise scales as $\sigma(1-p)$; and (2) the criterion must use only p_t and q_t, since the full perturbation $\boldsymbol{\xi}$ is unobservable and the inference engine records only sampled-token probabilities.

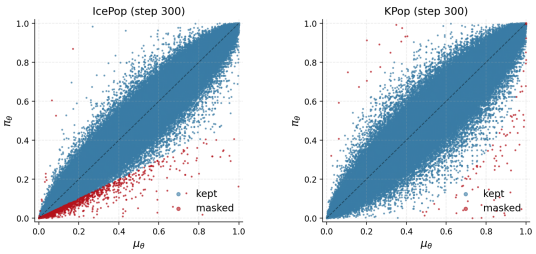


Figure 3: Token masking on Ring-mini. **(Left)** IcePop concentrates masked tokens (red) in the low-probability region. **(Right)** KPop spreads masked tokens along the acceptance boundary, filtering corrupted tokens at every probability level.

5 KPOP: SYMMETRIC BINARY KL MASKING

KPop satisfies both requirements of §4.2 using binary KL divergence. We define the mask (§5.1), show why it gives probability-adaptive tolerance (§5.2), justify symmetry (§5.3), explain why retaining rare tokens is beneficial (§5.4), and separate KPop from update trust regions such as DPPO (§5.5).

5.1 ALGORITHM

Definition 2 (KPop)**.** *Given a tolerance $\delta > 0$, the KPop mask is*

$$\mathcal{M}_{\mathrm{KPop}}(t) = \mathbb{1}\big[D^B_{\mathrm{KL}}(p_t \,\|\, q_t) \le \delta\big] \cdot \mathbb{1}\big[D^B_{\mathrm{KL}}(q_t \,\|\, p_t) \le \delta\big], \tag{6}$$

where $D^B_{\mathrm{KL}}(p \,\|\, q) = p \log \frac{p}{q} + (1-p) \log \frac{1-p}{1-q}$ is the binary KL divergence between the Bernoulli distributions $(p_t, 1-p_t)$ and $(q_t, 1-q_t)$ on the two-event space $\{y_t, \neg y_t\}$.

The token loss follows IcePop (Eq. 3) with $\mathcal{M}_{\mathrm{KPop}}$ in place of the constant ratio window:

$$\ell_t = \frac{p_t}{q_t} \cdot \mathcal{M}_{\mathrm{KPop}}(t) \cdot \min\Big(A_t \, \tfrac{u_t}{p_t},\ A_t \ \mathrm{clip}\big(\tfrac{u_t}{p_t},\ 1-\epsilon_-,\ 1+\epsilon_+\big)\Big). \tag{7}$$

Cost. The mask needs only p_t and q_t, which the training and inference engines already produce as per-token log-probabilities. It is a handful of element-wise operations on tensors that are already materialized: $O(1)$ time and memory per token, no extra forward pass, and no vocabulary-wide logits. Pseudocode is in Appendix D.

5.2 PROBABILITY-ADAPTIVE ACCEPTANCE VIA BINARY KL

Proposition 1 (Local log-ratio form of the binary KL bound)**.** *Let $\ell = \log(p_t/q_t)$ and $p = p_t$. For $|\ell| \ll 1$, $D^B_{\mathrm{KL}}(p \,\|\, q) = \frac{p}{2(1-p)}\,\ell^2 + O(|\ell|^3)$, so $D^B_{\mathrm{KL}}(p \,\|\, q) \le \delta$ implies, up to higher-order terms,*

$$|\log(p_t/q_t)| \ \lesssim\ \sqrt{2\delta\,(1-p)/p}. \tag{8}$$

The proof is a Taylor expansion (Appendix A.2). The tolerated log-ratio deviation grows as p decreases, which is the probability-adaptive behavior required by Lemma 1: a single δ gives wider tolerance where noise is large (low p) and tighter tolerance where it is small (high p). The tolerance $\sqrt{2\delta(1-p)/p}$ grows faster than the noise scale $\sigma(1-p)$, so binary KL is a sufficient acceptance condition that is uniformly more permissive for rare tokens rather than an exact noise-matching test. This margin removes the over-masking of §3.2, and δ sets the width of the acceptance region at every probability level at once.

5.3 SYMMETRIC FORMULATION FOR BIDIRECTIONAL DISCREPANCY

Token pairs (p_t, q_t) concentrate in a band around the diagonal with deviations in both directions (Figure 3), so the acceptance region should be centered on the diagonal. A one-sided binary KL constraint is not, because $D^B_{\mathrm{KL}}(p \,\|\, q)$ and $D^B_{\mathrm{KL}}(q \,\|\, p)$ differ:

Proposition 2 (Directional leakage of one-sided binary KL)**.** *For $q = p + \Delta$, $D^B_{\mathrm{KL}}(q \,\|\, p) - D^B_{\mathrm{KL}}(p \,\|\, q) = \frac{(1-2p)\,\Delta^3}{6p^2(1-p)^2} + O(\Delta^4)$. Hence for $p < \frac{1}{2}$ and $q > p$ there exist pairs with $D^B_{\mathrm{KL}}(p \,\|\, q) \le \delta < D^B_{\mathrm{KL}}(q \,\|\, p)$, which a one-sided rule admits and a symmetric rule rejects.*

Theorem 1 (Two-sided control under symmetric binary KL)**.** *If $D^B_{\mathrm{KL}}(p \,\|\, q) \le \delta$ and $D^B_{\mathrm{KL}}(q \,\|\, p) \le \delta$, then under the quadratic approximation $|p - q| \le \sqrt{2\delta \cdot \min\big(p(1-p),\, q(1-q)\big)}$.*

Proofs are in Appendix A. A forward-only mask lets through tokens where the inference engine assigns much higher probability than the training engine; these carry extreme IS ratios and noisy gradients. KPop accepts only the intersection of the two one-sided regions, a band whose width scales as $\sqrt{p(1-p)}$ around the diagonal. The quadratic approximation is used only for interpretation; the mask evaluates exact binary KL, and the exact and approximate boundaries nearly coincide for $\delta \le 0.75$ (Appendix B.7). Since the two directions differ only at third order in Δ, symmetric and one-sided masks agree on most small-discrepancy tokens and differ on the large-discrepancy tail, which is where MoE routing flips live (§6.4).

5.4 Why Retaining Rare Tokens Helps

Rare tokens have large gradients, so one might expect that masking them aggressively, as IcePop does, is what stabilizes training. Our analysis says otherwise, for three reasons.

Gradient size and unreliability are different quantities. By Lemma 1, the engine noise on $\log(p_t/q_t)$ has scale $\sigma\|\nabla_{\mathbf{z}} \log p_{y_t}\|$. A large ratio on a rare token is therefore what a small, benign perturbation produces, not evidence of corruption. A constant ratio threshold is effectively a threshold on the gradient norm: it preferentially deletes the tokens whose gradients $\mathbf{e}_y - \mathbf{p}$ point toward currently unlikely continuations, which is the exploration signal RLVR needs (Yu et al., 2025).

What KPop keeps is still bounded. Every retained token satisfies $|p_t - q_t| \le \sqrt{2\delta\, \min(p_t(1-p_t), q_t(1-q_t))}$ (Theorem 1): the two engines agree on the probability mass of the event within a margin that shrinks with p_t. Tokens whose mismatch is large relative to their probability scale, including routing-flip outliers, are still rejected at every probability level (Figure 3).

Magnitude is controlled elsewhere. A rare token enters a batch only with frequency q_t, and its update passes through the same drift clip on u_t/p_t as every other token. Empirically, retaining these tokens does not cause gradient spikes: the gradient norm of symmetric KPop stays flat throughout training (Appendix Figure 16(b); Appendix Figure 12), while KPop keeps about twice as many exactly matching tokens and improves AIME 2025 accuracy over IcePop on Ring-mini (§6.3).

5.5 Relation to Update Trust Regions (DPPO)

DPPO (Qi et al., 2026) also uses a binary KL, but as a trust region on the *policy update*. With $d_t = D^B_{\mathrm{KL}}(p_t \,\|\, u_t)$, it keeps a token's gradient according to the advantage sign,

$$m_t^{\mathrm{DPPO}} = \begin{cases} \mathbb{1}\{d_t \le \kappa_+ \vee u_t \le p_t\}, & A_t > 0, \\ \mathbb{1}\{d_t \le \kappa_- \vee u_t \ge p_t\}, & A_t < 0, \end{cases} \quad g_t^{\mathrm{DPPO}} = m_t^{\mathrm{DPPO}}\, \mathrm{sg}\Big[\min\Big(\tfrac{u_t}{p_t}, c\Big)\Big]\, A_t \nabla_\theta \log u_t, \tag{9}$$

i.e. it blocks updates that have already moved far from $\pi_{\theta_{\mathrm{old}}}$ in the advantage-improving direction (sg is stop-gradient and c a ratio cap).

Proposition 3 (DPPO is blind to engine mismatch)**.** *m_t^{DPPO} and g_t^{DPPO} depend on (p_t, u_t, A_t) only. In particular, at the first update on each rollout batch ($u_t = p_t$, so $d_t = 0$), every token passes, whatever q_t is.*

DPPO could instead be anchored at the inference engine, replacing p_t by q_t in Eq. (9), so that its divergence sees the mismatch. This does not recover a mismatch filter:

Proposition 4 (Rollout-anchored DPPO leaks by advantage sign)**.** *With anchor q_t, at $u_t = p_t$ the DPPO gate becomes $\mathbb{1}\{D^B_{\mathrm{KL}}(q_t \,\|\, p_t) \le \kappa_\pm\} \vee \mathbb{1}\{\mathrm{sign}(A_t)\,(p_t - q_t) \le 0\}$. Every token with $p_t \le q_t$ passes when $A_t > 0$, and every token with $p_t \ge q_t$ passes when $A_t < 0$, however large the mismatch. In the second case the token receives weight $\min(p_t/q_t, c)$, which can reach c.*

Both statements follow directly from Eq. (9). Anchoring DPPO at q_t also makes it spend its trust-region budget on engine noise rather than on policy drift. KPop tests both directions independently of A_t, so it rejects the tokens that Propositions 3 and 4 admit. Conversely, KPop does not limit policy

drift. The two controls are complementary and compose into a single estimator,

$$g_t^{\text{comb}} = \mathcal{M}_{\text{KPop}}(t)\, m_t^{\text{DPPO}}\, \text{sg}\Big[\tfrac{p_t}{q_t}\min\Big(\tfrac{u_t}{p_t}, c\Big)\Big]\, A_t\, \nabla_\theta \log u_t, \tag{10}$$

which keeps DPPO's drift control while removing the mismatched tokens it cannot see. Appendix Table 2 summarizes the distinction.

6 Experiments

We study a dense model where mismatch is mild (§6.2) and MoE models where routing discrepancy is severe (§6.3), and ablate symmetry and δ (§6.4).

6.1 Setup

Models and data. We use the dense Qwen3-4B-Instruct-2507 (Yang et al., 2025), a low-discrepancy control, and two public MoE models with top-K routing: Ring-mini-2.0 and Ring-flash-2.0 (100B total, 6.1B active parameters).[2] Qwen3-4B and Ring-mini are trained on DAPO-Math (Yu et al., 2025) with binary reward. Ring-flash is trained on a mixture of mathematics, code, logic, and STEM RLVR data.

Evaluation. We report Pass@1 averaged over 32 samples (Avg@32) on AIME 2024, AIME 2025, and HMMT 2025 every 25 steps. For Ring-flash we also report LiveCodeBench v6 (code generation) and ARC-AGI-2 (abstract reasoning).[3] Tables report the best checkpoint.

Methods. All methods build on GRPO (Eq. 1); on the dense model we additionally compare with DPPO (§6.2). Baselines: TIS (truncated per-token IS) and Seq-TIS (sequence-level) (Zheng et al., 2025) with cap $C{=}3$; Seq-MIS (sequence-level masked IS) and Geo-MIS (geometric-mean MIS) (Li et al., 2025); and IcePop ($\alpha{=}0.5$, $\beta{=}5.0$). KPop selects δ from $\{0.1, 0.5, 0.75, 2.0\}$ (§6.4).

Infrastructure. SGLang (Zheng et al., 2024) for inference and Megatron-LM (Shoeybi et al., 2019) for training on 32 H200 GPUs. Learning rate 10^{-6}, 8 responses per prompt, batch size 16, asymmetric clipping $\epsilon_+{=}0.28$ (Yu et al., 2025). All methods share the training seed (42).

6.2 Dense Model

On Qwen3-4B-Instruct, $p_t \approx q_t$ for most tokens. Among mismatch corrections, KPop ($\delta{=}0.75$) achieves the best accuracy on all three benchmarks (Appendix Figure 11). Because the discrepancy mask rarely fires here, the gain comes from not filtering exploratory low-probability tokens unnecessarily. TIS, Seq-TIS, Seq-MIS, and IcePop track vanilla GRPO closely, since their corrections have little effect when the ratio is near one (TIS later collapses after step 1500, Appendix C.3); Geo-MIS collapses around step 800, indicating that its geometric-mean threshold is too aggressive even when per-token discrepancy is small.

Comparison with DPPO. We also train DPPO (Qi et al., 2026) in this setup with the same data and budget, using the official binary-KL implementation and defaults ($\kappa_\pm{=}0.05$, ratio cap $c{=}10^4$). DPPO reaches higher best-checkpoint accuracy than KPop: 70.2/61.4/38.3 vs. 64.0/58.0/27.0 on AIME24/AIME25/HMMT25. On this model the comparison mainly changes the *update rule*, not the mismatch filter: KPop keeps GRPO's PPO ratio clip and changes only which tokens enter it, whereas DPPO replaces that clip with a binary-KL trust region on policy drift, the component it was designed to improve. Conversely, DPPO applies no inference–training correction (Proposition 3), which is the component KPop provides and which matters when the engines disagree (§6.3). The two act on different terms and compose (Eq. 10): with small mismatch, DPPO's update rule is the stronger choice, and the KPop mask adds to it at $O(1)$ cost.

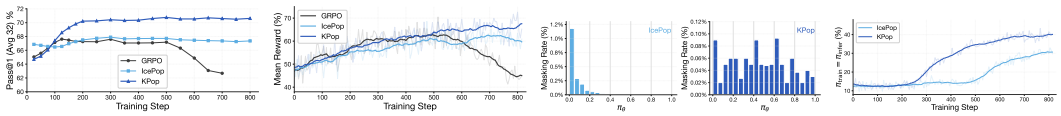


Figure 4: Ring-mini (MoE). From left: AIME 2025 accuracy, smoothed training reward, masking rate by token-probability bin, and fraction of tokens with exact train/inference agreement. GRPO becomes unstable after step 400; IcePop is stable but plateaus and concentrates masking on rare tokens; KPop masks evenly across bins, keeps about $2\times$ more exactly matching tokens, and reaches the highest accuracy.

Table 1: Ring-flash-2.0 (100B total, 6.1B active MoE), best checkpoint (%).

Method	AIME25	HMMT25	LiveCodeBench v6	ARC-AGI-2
IcePop	87.97	75.72	73.95	5.69
KPop, one-sided, δ=0.75	88.85	78.69	74.67	8.53
KPop, symmetric, δ=0.75	88.75	78.58	75.33	**9.88**
KPop, symmetric, δ=2.0	**90.16**	**79.42**	**75.50**	6.44

6.3 MoE Models

Ring-mini. KPop (δ=0.5) reaches about 71% AIME 2025 accuracy by step 200 and stays above 69% (Figure 4). GRPO becomes unstable after step 400, while IcePop stabilizes training but plateaus at 65–67%. The masking statistics match the analysis: IcePop masks over 1.2% of tokens with $p_t < 0.05$ and almost none with $p_t > 0.4$, whereas KPop spreads masking across probability levels and retains about twice as many tokens on which the two engines agree exactly. KPop thus removes corrupted tokens selectively rather than pruning rare ones indiscriminately.

Ring-flash (100B). At larger scale and across domains, symmetric KPop with δ=0.75 outperforms IcePop on all four benchmarks, including code (+1.4) and abstract reasoning (+4.2), and δ=2.0 gives the best mathematics and code scores (Table 1). Both symmetric runs remain stable while filtering more than 30% of tokens.

6.4 Ablations: Symmetry and δ

What the symmetry ablation supports. On the dense model (Appendix Figure 16), one-sided variants are competitive at some stages. This is expected: by Proposition 2 the two KL directions differ only at third order in $|p_t - q_t|$, so when mismatch is small both masks accept nearly the same tokens. Its clear effect is on stability: one-sided variants show larger, spikier gradient norms (Appendix Figure 16(b)), since the tokens they alone accept have extreme one-directional ratios. Where mismatch is heavier, symmetry also pays off in accuracy: on the Ring-flash MoE, symmetric δ=0.75 beats the one-sided variant by 0.7 on LiveCodeBench and 1.4 on ARC-AGI-2 while matching it on mathematics (Table 1).

Effect of δ. On the dense model, tightening from δ=2.0 to 0.75 improves accuracy, and the δ=2.0 curve declines after about 750 steps (Appendix Figure 16(c)). Theorem 1 suggests why: with δ=2.0 the admitted gap $\sqrt{2\delta p(1-p)}$ reaches 1 at p=0.5, so the mask barely filters intermediate- and high-probability tokens, and the mismatch it admits can accumulate over long training. On Ring-flash, however, δ=2.0 is best on three of four benchmarks, so the best δ depends on the model's discrepancy scale.

Choosing δ. δ bounds engine discrepancy in a probability-normalized unit, so it transfers better than a ratio window: δ=0.75 works on both the dense 4B model and the 100B MoE, and Ring-mini selects 0.5. In practice we recommend recording the symmetric binary KL of all tokens in a short pilot run,

[2] https://huggingface.co/inclusionAI/Ring-mini-2.0, https://huggingface.co/inclusionAI/Ring-flash-2.0

[3] https://livecodebench.github.io, https://arcprize.org/arc-agi/2

setting δ near the knee of its distribution, and validating a small neighboring grid. Smaller δ filters more strictly; larger δ retains more tokens.

7 Related Work

RL for LLMs and decoupled systems. RLVR with GRPO (Shao et al., 2024; Guo et al., 2025) avoids the value network of PPO (Schulman et al., 2017; Ouyang et al., 2022). Production systems generate with vLLM (Kwon et al., 2023) or SGLang (Zheng et al., 2024) and train with Megatron-LM (Shoeybi et al., 2019; Yan et al., 2026); the resulting inference–training mismatch is a recognized cause of RL collapse (Liu et al., 2025).

IS corrections and MoE routing. V-trace (Espeholt et al., 2018) and Retrace (Munos et al., 2016) truncate IS ratios for staleness. For engine mismatch, TIS (Zheng et al., 2025) caps $\pi_{\theta_{\text{old}}}(y_t)/\mu_{\theta_{\text{old}}}(y_t)$, and Seq-MIS and Geo-MIS (Li et al., 2025) mask at the sequence level; all use ratio cuts that ignore the probability-dependent noise of §4. MoE models (Fedus et al., 2022; Jiang et al., 2024) amplify mismatch through routing. R3 (Ma et al., 2025) replays inference-time routing in the training engine, which removes routing flips at extra communication cost but not other numerical discrepancy; token-level masks such as KPop are complementary.

Trust regions and binary KL. KL trust regions originate in TRPO (Schulman et al., 2015). DPPO (Qi et al., 2026) replaces PPO's ratio clip with a divergence constraint, including a sampled-token binary KL, between the old and current training policies; TRM (Li et al., 2025) derives sequence-level trust-region bounds. KPop does not claim binary KL itself as new. Its contribution is a symmetric, advantage-independent binary KL between the *training and inference engines*, with a tolerance derived from the softmax noise structure, a proof that symmetry is needed, and a demonstration that it complements update trust regions (§5.5).

8 Conclusion and Limitations

Constant-ratio masks such as IcePop stabilize decoupled RLVR but over-mask rare tokens, because softmax normalization makes engine noise on the log-ratio proportional to each token's gradient norm. KPop replaces the ratio with a symmetric binary KL whose tolerance adapts to token probability, at $O(1)$ cost per token. It filters engine mismatch, an error that update trust regions such as DPPO do not see, and composes with them. Across a dense model and two MoE models up to 100B parameters, KPop retains more exploratory signal, trains stably, and improves accuracy.

Limitations. (i) Each configuration is trained once with a shared seed; Avg@32 reduces evaluation variance but does not measure run-to-run variance. (ii) We show which tokens each mask removes but do not run a stepwise matched-masking-rate comparison, which needs an adaptive IcePop threshold and full retraining. (iii) The best δ is model-dependent (0.5–2.0); we give a selection procedure, not a universal value. (iv) The first-order analysis of §4 does not model routing flips; the mask itself is exact. (v) We train DPPO only on the dense model; MoE comparisons and the composed estimator (Eq. 10) are future work. (vi) Binary KL ignores redistribution among unsampled tokens, which does not affect the sampled-token weight (Appendix B.6).

Reproducibility Statement

Section 6.1 lists models, data, evaluation protocol, baselines with their hyperparameters, and training infrastructure. Algorithm 1 gives the KPop loss, and Eqs. (9)–(10) formalize DPPO and the composed estimator. All proofs are in Appendix A, and the empirical verification protocol for Lemma 1 is in Appendix B.3. All models and training data except the Ring-flash RLVR mixture are public. We will release code upon acceptance.

Ethics Statement

This work improves the numerical stability of reinforcement-learning fine-tuning for language models. It involves no human subjects or personal data, and we do not foresee risks beyond those of LLM training in general.

AI USE STATEMENT

We used generative AI tools to assist with manuscript writing and polishing, literature search, and research execution, including assistance with experiment code, analysis, and figure drafting. Verification included code review and tests, checks of reported numbers against training and evaluation logs, and symbolic checks of the derivations. The authors take responsibility for the final manuscript, methods, and reported results.

REFERENCES

Lasse Espeholt, Hubert Soyer, Remi Munos, et al. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *ICML*, 2018.

William Fedus, Barret Zoph, and Noam Shazeer. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *JMLR*, 23(1):5232–5270, 2022.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948*, 2025.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, et al. Mixtral of experts. *arXiv:2401.04088*, 2024.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, et al. Efficient memory management for large language model serving with PagedAttention. In *SOSP*, 2023.

Yingru Li, Jiacai Liu, Jiawei Xu, et al. Trust region masking for long-horizon LLM reinforcement learning. *arXiv:2512.23075*, 2025.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training-inference mismatch. `https://richardli.xyz/rl-collapse`, September 2025.

Wenhan Ma, Hailin Zhang, Liang Zhao, Yifan Song, Yudong Wang, Zhifang Sui, and Fuli Luo. Stabilizing MoE reinforcement learning by aligning training and inference routers. *arXiv:2510.11370*, 2025.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. Safe and efficient off-policy reinforcement learning. In *NeurIPS*, 2016.

Long Ouyang, Jeff Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, 2022.

Penghui Qi, Xiangxin Zhou, Zichen Liu, et al. Rethinking the trust region in LLM reinforcement learning. *arXiv:2602.04879*, 2026.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. In *ICML*, 2015.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300*, 2024.

Mohammad Shoeybi, Mostofa Patwary, Raul Puri, et al. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv:1909.08053*, 2019.

Zijie Yan et al. Scalable training of mixture-of-experts models with Megatron Core. *arXiv:2603.07685*, 2026.

An Yang et al. Qwen3 technical report. *arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv:2503.14476*, 2025.

Xin Zhao, Yongkang Liu, Kuan Xu, et al. Small leak can sink a great ship — boost RL training on MoE with IcePop! `https://ringtech.notion.site/icepop`, 2025.

Chujie Zheng, Kai Dang, Bowen Yu, et al. Stabilizing reinforcement learning with LLMs: Formulation and practices. *arXiv:2512.01374*, 2025.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, et al. SGLang: Efficient execution of structured language model programs. *arXiv:2312.07104*, 2024.

A Proofs

A.1 Proof of Lemma 1 (Heteroscedastic Token Divergence)

Proof. By Eq. (4), to first order $\log(p_y/q_y) \approx -\xi_y + \sum_j p_j \xi_j = -\langle \mathbf{e}_y - \mathbf{p}, \boldsymbol{\xi} \rangle$. For i.i.d. zero-mean ξ_j with variance σ^2, the variance of a linear form $\langle \mathbf{a}, \boldsymbol{\xi} \rangle$ is $\sigma^2 \|\mathbf{a}\|^2$, so $\mathrm{Var}[\log(p_y/q_y) \mid \mathbf{p}] = \sigma^2 \|\mathbf{e}_y - \mathbf{p}\|^2 = \sigma^2\big((1-p)^2 + \sum_{j \neq y} p_j^2\big)$, which is Eq. (5). The approximation $\mathrm{Std} \approx \sigma(1-p)$ follows when $\mathcal{R} \ll (1-p)^2$. Finally, $\nabla_{\mathbf{z}} \log p_y = \mathbf{e}_y - \mathbf{p}$ by differentiating the log-softmax. □

A.2 Proof of Proposition 1 (Local Log-Ratio Form)

Proof. Write $q = p\, e^{-\ell}$. Then $q - p = -p\ell + \frac{p}{2}\ell^2 + O(p|\ell|^3)$, so $(q-p)^2 = p^2\ell^2 + O(p^2|\ell|^3)$. The quadratic term of the binary KL is $D^B_{\mathrm{KL}}(p \,\|\, q) = (q-p)^2/[2p(1-p)] + O(|q-p|^3)$ (Appendix B.5). Substituting gives $D^B_{\mathrm{KL}}(p \,\|\, q) = \frac{p}{2(1-p)}\ell^2 + O(|\ell|^3)$, and rearranging the leading term yields Eq. (8). □

A.3 Proof of Proposition 2 (Directional Leakage)

Proof. Let $f(q) = D^B_{\mathrm{KL}}(q \,\|\, p) - D^B_{\mathrm{KL}}(p \,\|\, q)$ for fixed p. Expanding both divergences around $q = p$ (Appendix B.5), the quadratic terms coincide and $f(p+\Delta) = \frac{1-2p}{6p^2(1-p)^2}\Delta^3 + O(\Delta^4)$. For $p < \frac{1}{2}$ and $\Delta > 0$ small, the leading term is positive, so $D^B_{\mathrm{KL}}(q \,\|\, p) > D^B_{\mathrm{KL}}(p \,\|\, q)$. By continuity, any δ in the non-empty interval $[D^B_{\mathrm{KL}}(p \,\|\, q)\,, D^B_{\mathrm{KL}}(q \,\|\, p))$ satisfies $D^B_{\mathrm{KL}}(p \,\|\, q) \leq \delta < D^B_{\mathrm{KL}}(q \,\|\, p)$. □

A.4 Proof of Theorem 1 (Two-Sided Control)

Proof. The quadratic approximation of $D^B_{\mathrm{KL}}(p \,\|\, q) \leq \delta$ gives $|p-q| \leq \sqrt{2\delta\, p(1-p)}$, and that of $D^B_{\mathrm{KL}}(q \,\|\, p) \leq \delta$ gives $|p-q| \leq \sqrt{2\delta\, q(1-q)}$. Both hold under the symmetric mask, which yields the stated bound. □

A.5 Proofs of Propositions 3 and 4 (DPPO)

Proof. Proposition 3: every quantity in Eq. (9) (d_t, the comparisons $u_t \lessgtr p_t$, A_t, and the weight $\min(u_t/p_t, c)$) is a function of (p_t, u_t, A_t). At $u_t = p_t$, $d_t = D^B_{\mathrm{KL}}(p_t \,\|\, p_t) = 0 \leq \kappa_{\pm}$, so $m_t^{\mathrm{DPPO}} = 1$. Proposition 4: replacing the anchor p_t by q_t and setting $u_t = p_t$ turns d_t into $D^B_{\mathrm{KL}}(q_t \,\|\, p_t)$ and the escape conditions into $p_t \leq q_t$ (for $A_t > 0$) and $p_t \geq q_t$ (for $A_t < 0$), i.e. $\mathrm{sign}(A_t)(p_t - q_t) \leq 0$. The escape condition does not involve the size of the mismatch, and the weight becomes $\min(p_t/q_t, c)$. □

B Additional Technical Details

B.1 DPPO and KPop at a Glance

Table 2: DPPO and KPop use binary KL on different probability pairs, for different purposes.

	DPPO	KPop
Probabilities compared	$\pi_{\theta_{\text{old}}}$ vs. π_θ (both training engine)	$\pi_{\theta_{\text{old}}}$ (training) vs. $\mu_{\theta_{\text{old}}}$ (inference)
Error controlled	policy-update drift	inference–training engine mismatch
Depends on advantage sign	yes (one-sided per sign)	no (two-sided for every token)
At $u_t = p_t$	every token passes	large p_t–q_t mismatch still rejected
Role	optimization trust region	rollout reliability filter

B.2 Higher-Order Expansion and Routing Flips

Let $J \sim \mathbf{p}$ index the vocabulary and $\bar{\xi} = \mathbb{E}_J[\xi_J]$. The exact identity in Eq. (4) has the cumulant expansion

$$\log \frac{p_y}{q_y} = -\xi_y + \log \mathbb{E}_J\left[e^{\xi_J}\right] = -\xi_y + \bar{\xi} + \frac{1}{2}\sum_j p_j(\xi_j - \bar{\xi})^2 + \frac{1}{6}\sum_j p_j(\xi_j - \bar{\xi})^3 + O(\|\boldsymbol{\xi}\|^4). \quad (11)$$

The first-order term gives Lemma 1; the higher-order terms depend on the whole perturbation vector and are not observable from (p_t, q_t). When routing is unchanged, the perturbation is small and the first-order prediction matches the measured noise (R^2=0.89 on MoE tokens, Appendix B.3). When routing flips, the perturbation is large and the expansion loses accuracy; measured noise for these tokens is higher and largely independent of p. This affects only the explanation, not the algorithm: KPop evaluates the exact binary KL in both directions for every token, so routing flips introduce no approximation error into the mask, and they are exactly the large-discrepancy tokens it rejects.

B.3 Empirical Verification Details

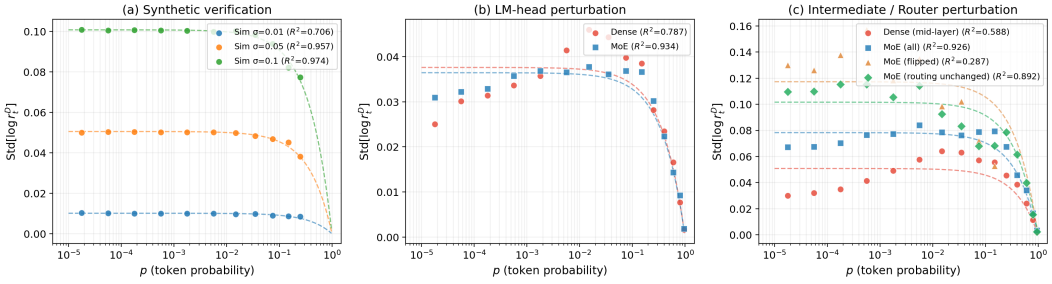


Figure 5: Empirical verification of Lemma 1. $\mathrm{Std}[\log(\pi_{\theta_{\text{old}}}/\mu_{\theta_{\text{old}}})]$ vs. token probability p (log scale). Dashed lines show the prediction $\hat{\sigma}(1-p)$. **(a)** Synthetic perturbations. **(b)** LM-head perturbation on dense and MoE models. **(c)** Deeper perturbation with MoE routing decomposition.

We verify the predicted scaling $\mathrm{Std}[\log(\pi_{\theta_{\text{old}}}/\mu_{\theta_{\text{old}}})] \propto (1-p)$ under three progressively realistic settings on Qwen3-8B (dense) and Qwen3-30B-A3B (MoE). For each setting, we bin tokens by probability p, compute the empirical standard deviation within each bin, fit $\hat{\sigma}(1-p)$ by least squares, and report the coefficient of determination R^2.

(a) Synthetic. Gaussian hidden-state noise at $\sigma \in \{0.01, 0.05, 0.1\}$ applied to the final hidden states before the LM head. The empirical standard deviation decreases monotonically with p and fits $\hat{\sigma}(1-p)$ with $R^2 \in [0.71, 0.97]$.

(b) LM-head. Perturbing the final hidden states of real models and reprojecting through the LM head gives R^2=0.79 (dense) and 0.93 (MoE), so the $(1-p)$ dependence survives the correlations introduced by W_{LM}.

(c) Intermediate layers. Perturbations at intermediate layers degrade the dense fit (R^2=0.59) because of nonlinear amplification in later layers. For MoE, tokens with unchanged routing follow the prediction closely (R^2=0.89), while routing-flipped tokens have a higher, largely probability-independent noise floor (R^2=0.29). The aggregate fit remains strong (R^2=0.93) because unchanged-routing tokens dominate.

B.4 Error Bound for the Combined-Ratio Implementation

Our implementation clips the combined ratio u_t/q_t (Algorithm 1) instead of applying p_t/q_t and clipping u_t/p_t separately as in Eq. (7).

Lemma 2. *Under symmetric KPop, the difference between the combined-clip and decomposed token losses satisfies* $|L_t^{\text{comb}} - L_t^{\text{dec}}| \le |A_t|(1+\epsilon_+)\sqrt{2\delta(1-p_t)/p_t}$ *to leading order.*

B.5 Taylor Expansion Details

For $q = p + \Delta$,

$$D_{\text{KL}}^B(p \,\|\, q) = \frac{\Delta^2}{2p(1-p)} + \frac{(2p-1)\,\Delta^3}{3p^2(1-p)^2} + O(\Delta^4),$$
$$D_{\text{KL}}^B(q \,\|\, p) = \frac{\Delta^2}{2p(1-p)} + \frac{(2p-1)\,\Delta^3}{6p^2(1-p)^2} + O(\Delta^4).$$

The quadratic term gives the heteroscedastic band of Theorem 1; the difference of the cubic terms gives Proposition 2.

B.6 Binary KL vs. Full Categorical KL

Let $\mathbf{p}_{-y}$ and $\mathbf{q}_{-y}$ be the two engines' distributions over the vocabulary with token y removed and renormalized. The chain rule for KL gives the exact decomposition

$$D_{\text{KL}}(\mathbf{p} \,\|\, \mathbf{q}) = D_{\text{KL}}^B(p_y \,\|\, q_y) + (1-p_y)\, D_{\text{KL}}(\mathbf{p}_{-y} \,\|\, \mathbf{q}_{-y})\,. \tag{12}$$

The first term measures the discrepancy of the sampled event; the second measures only how probability is redistributed among tokens that were not sampled. The policy-gradient contribution of a sampled token is weighted by the system ratio p_y/q_y, which the first term determines exactly and the second does not affect: if $p_y = q_y$, the importance weight is exactly one, however the remaining mass is redistributed. Binary KL is therefore not a surrogate for full KL; it is the part of the full KL that the sampled-token importance weight depends on. It also needs only $O(1)$ inference-side information per token, whereas full KL requires the inference engine to store full-vocabulary distributions.

Table 3: What symmetric binary KL and full categorical KL control.

	Symmetric binary KL	Full categorical KL
Detects sampled-token probability mismatch	yes	yes
Detects redistribution among unsampled tokens	no	yes
Determines the sampled-token weight p_y/q_y	yes	no
Inference-side storage per token	$O(1)$	$O(V)$
Can reject a token when only unsampled probabilities moved	no	yes

A full-KL filter would therefore answer a different question (whether the two engines' whole next-token distributions agree), and could be added on top of KPop; we leave evaluating such a global filter to future work.

B.7 Accuracy of the Quadratic Approximation

Theorem 1 characterizes the KPop acceptance region under a quadratic approximation. Since the mask evaluates the exact binary KL (Definition 2), the approximation affects only the interpretive bound, not the algorithm. Figure 6 compares the exact boundary, obtained by solving $\max(D_{\text{KL}}^B(p \,\|\, q)\,, D_{\text{KL}}^B(q \,\|\, p)) = \delta$ numerically, against $|p-q| \le \sqrt{2\delta\, p(1-p)}$ for $\delta \in$

$\{0.5, 0.75, 2.0\}$. For $\delta \leq 0.75$ the two boundaries nearly coincide. At $\delta = 2.0$ the exact region is strictly wider, so the theorem gives a conservative bound and never overstates the filtering strength of KPop.

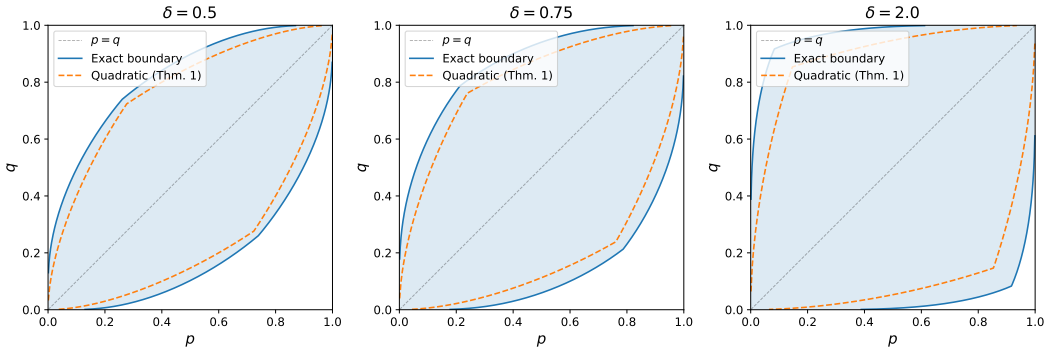


Figure 6: Exact symmetric binary KL acceptance boundary (solid) vs. the quadratic approximation of Theorem 1 (dashed). The shaded region shows pairs (p, q) accepted by the exact mask.

C Additional Experimental Results

C.1 IcePop, TIS, and GRPO on Ring-mini

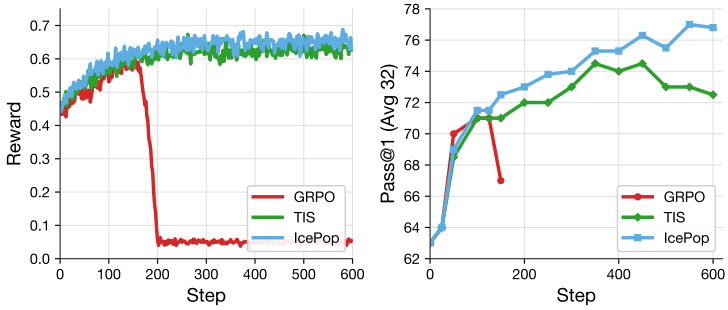


Figure 7: Training dynamics of IcePop, TIS, and vanilla GRPO on Ring-mini (MoE). **(Left)** Training reward. GRPO collapses around step 170 (gradient explosion leads to NaN loss) while IcePop and TIS keep improving. **(Right)** AIME 2025 accuracy (Avg@32). GRPO terminates early; IcePop reaches the best final accuracy.

C.2 Token Masking Across Training Steps

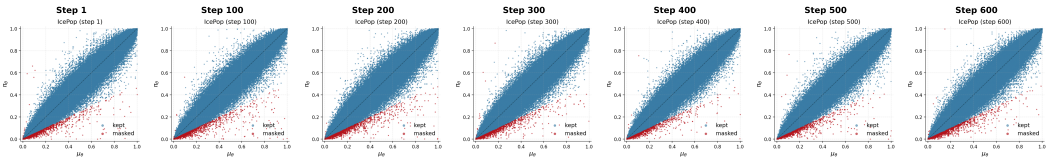


Figure 8: IcePop masking decisions across training steps. Red points (masked) remain concentrated in the low-probability region throughout training, confirming persistent over-masking of rare tokens.

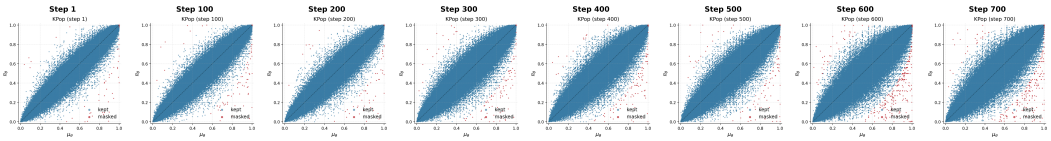


Figure 9: KPop ($\delta = 0.5$) masking decisions across training steps. Masked tokens distribute uniformly along the acceptance boundary at all probability levels. As training progresses, the point cloud widens slightly due to policy drift, and KPop masks the outliers adaptively.

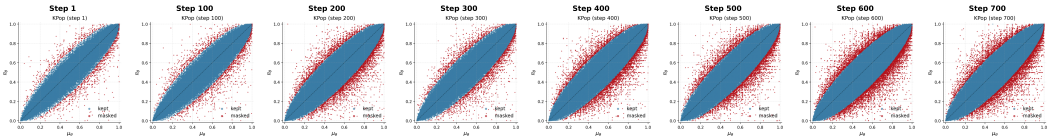
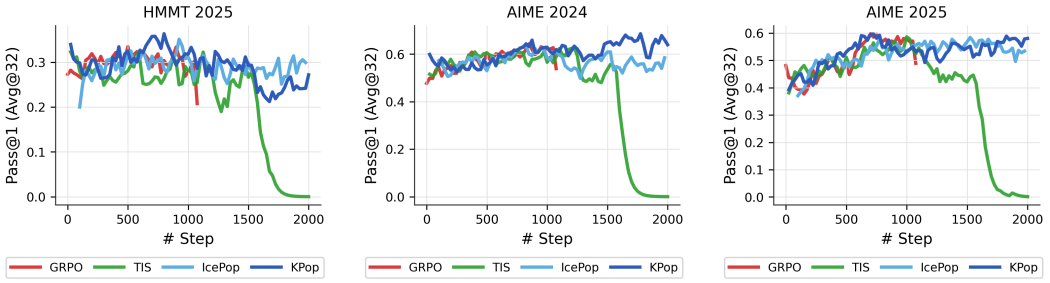


Figure 10: KPop ($\delta = 0.1$) masking decisions across training steps. A tighter threshold masks more tokens overall, with the boundary contracting toward the diagonal.

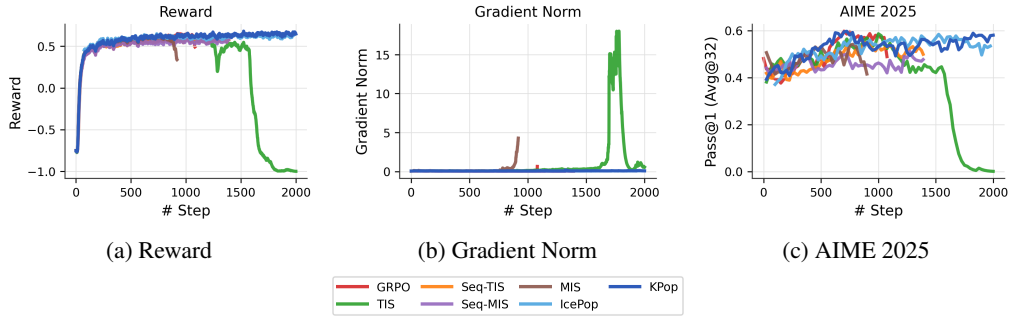
C.3 Dense Model Training Metrics



(a) HMMT 2025 (b) AIME 2024 (c) AIME 2025

Figure 11: Avg@32 on Qwen3-4B-Instruct (dense), EMA-smoothed (α=0.6). KPop (δ=0.75) is best among the methods shown on all three benchmarks; Geo-MIS collapses around step 800. Reward, gradient norm, and further baselines are in Appendix C.3.

Figure 12 compares KPop against all baselines on Qwen3-4B-Instruct, showing reward, gradient norm, and AIME 2025 accuracy. KPop achieves the highest reward among these methods and maintains stable gradients throughout training, while TIS collapses after step 1500. Figure 13 shows response length and clip ratio; Figure 14 confirms that the probability mismatch between engines remains small on this dense model.



(a) Reward (b) Gradient Norm (c) AIME 2025

Figure 12: KPop vs. all baselines on Qwen3-4B-Instruct (dense): reward, gradient norm, and AIME 2025 evaluation.

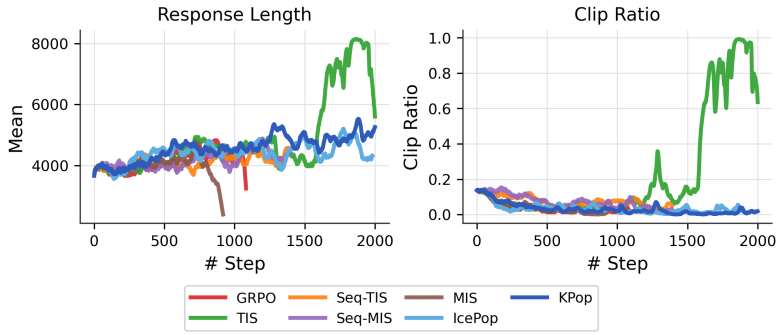


Figure 13: Response length and clip ratio for Qwen3-4B-Instruct.

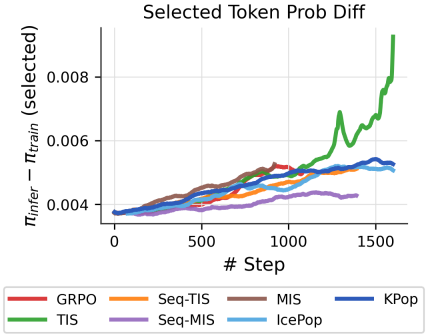
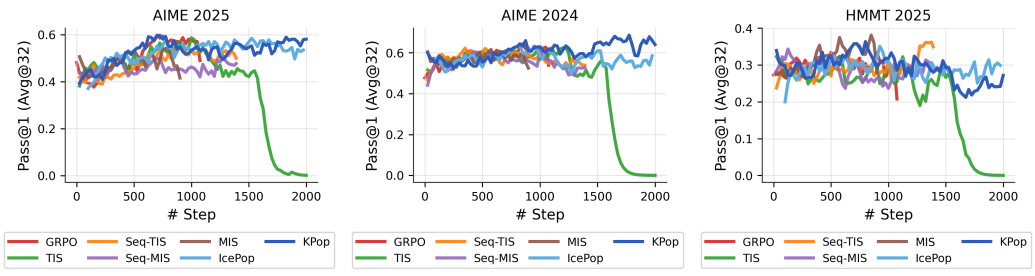


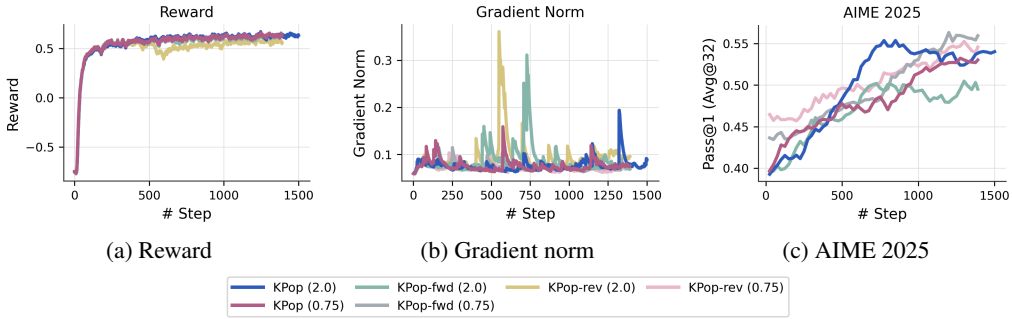
Figure 14: Selected probability difference between training and inference engines for Qwen3-4B-Instruct.



(a) AIME 2025 (b) AIME 2024 (c) HMMT 2025

Figure 15: Per-benchmark evaluation with all baselines on Qwen3-4B-Instruct. KPop has the best accuracy among these methods on all benchmarks; TIS collapses after step 1500.

C.4 Ablation Training Metrics



(a) Reward (b) Gradient norm (c) AIME 2025

Figure 16: Ablation on Qwen3-4B-Instruct: one-sided vs. symmetric KPop, $\delta \in \{0.75, 2.0\}$.

We provide additional training metrics for the ablation study on Qwen3-4B-Instruct. Figure 17 shows response length and clip ratio across KPop variants. Figure 18 shows the probability difference between engines. Figure 19 provides per-benchmark evaluation curves comparing symmetric KPop (δ=2.0) with one-sided variants and with δ=0.75.

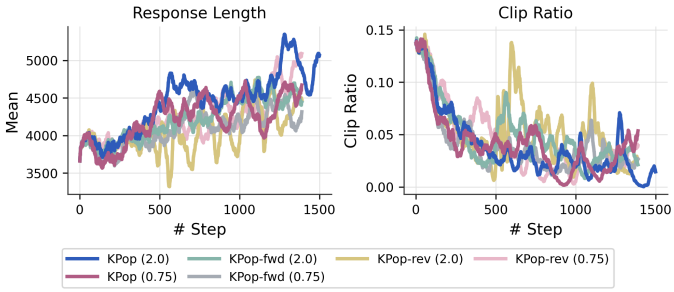


Figure 17: Response length and clip ratio for Qwen3-4B-Instruct (ablation studies).

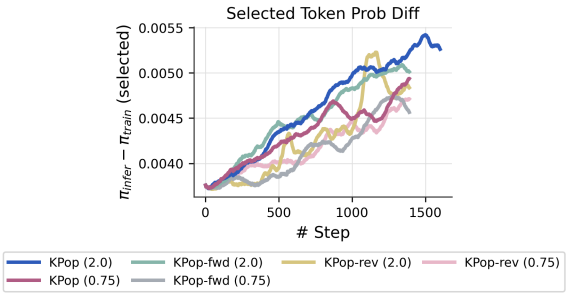
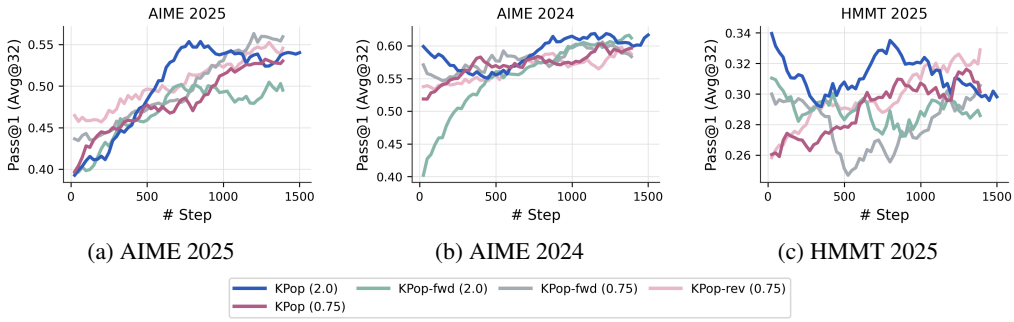


Figure 18: Selected probability difference between training and inference engines for Qwen3-4B-Instruct (ablation studies).

D Algorithm and Notation



(a) AIME 2025 (b) AIME 2024 (c) HMMT 2025

Figure 19: Ablation on Qwen3-4B-Instruct. Symmetric KPop (δ=2.0) vs. forward-only and reverse-only variants, and δ=0.75.

Algorithm 1 KPop policy loss (combined-ratio implementation) (per token t)

Require: $\log \pi_{\theta_{\text{old}}}, \log \pi_\theta, \log \mu_{\theta_{\text{old}}}, A_t, \delta, \epsilon_-, \epsilon_+$

$r_t \leftarrow \exp(\log \pi_\theta - \log \mu_{\theta_{\text{old}}})$

$d_{\text{fwd}} \leftarrow D^B_{\text{KL}}(\pi_{\theta_{\text{old}}} \,\|\, \mu_{\theta_{\text{old}}})$; $d_{\text{rev}} \leftarrow D^B_{\text{KL}}(\mu_{\theta_{\text{old}}} \,\|\, \pi_{\theta_{\text{old}}})$

$m_{\text{disc}} \leftarrow \mathbb{1}[d_{\text{fwd}} \le \delta] \cdot \mathbb{1}[d_{\text{rev}} \le \delta]$

$\ell_t \leftarrow m_{\text{disc}} \cdot \min(A_t r_t,\ A_t \,\text{clip}(r_t, 1-\epsilon_-, 1+\epsilon_+))$

Table 4: Notation summary.

Symbol	Meaning
$q_t = \mu_{\theta_{\text{old}}}(y_t)$	Inference engine probability (behavioral policy)
$p_t = \pi_{\theta_{\text{old}}}(y_t)$	Training engine probability before gradient update
$u_t = \pi_\theta(y_t)$	Training engine probability during the gradient update
$\pi_{\theta_{\text{old}}}(y_t)/\mu_{\theta_{\text{old}}}(y_t)$	Discrepancy ratio (system noise)
$\pi_\theta(y_t)/\pi_{\theta_{\text{old}}}(y_t)$	Staleness ratio (policy drift)
A_t	Sequence-level advantage applied per token ($A_t = \hat{A}_i$ for all t in sequence i)
δ	Binary KL threshold for discrepancy masking
α, β	IcePop ratio bounds
$\kappa_\pm, c$	DPPO binary-KL thresholds and ratio cap
$\boldsymbol{\xi}$	Logit perturbation between engines (§4.1)