IS THE IMPORTANCE RATIO NECESSARY FOR STABLE REINFORCEMENT LEARNING IN LLMS?

Shuibai Zhang[w*†], **Junhyuck Kim**[k*], **Gyeongman Kim**[k], **Jaewoong Cho**[k]
[w]University of Wisconsin–Madison, [k]KRAFTON

ABSTRACT

Reinforcement learning (RL) has become central to post-training large language models (LLMs). However, popular RL methods like GRPO incur non-negligible overhead by computing both old-policy and current-policy likelihoods to form importance sampling ratios. In this work, we propose Likelihood-Gated Policy Optimization (LGPO), which enforces a soft trust region constraint via likelihood-based gating, eliminating the need to compute old-policy likelihoods. Empirically, we show that removing the importance sampling correction term does not harm training stability, whereas removing the trust region mechanism leads to collapse. Moreover, ratio-based clipping can fail in fully on-policy training: the importance ratio stays at 1, so the ratio-based trust region constraint never activates. Under standard training settings where GRPO is stable, LGPO achieves comparable training stability and peak performance while reducing training time by $\sim$18% on average. In fully on-policy training, where GRPO fails, LGPO remains stable, enabling more efficient and robust LLM RL post-training across training regimes.

1 INTRODUCTION

Reinforcement learning (RL) plays a significant role in the post-training of modern large language models (LLMs) (Jaech et al., 2024; Guo et al., 2025). In particular, our focus is on RL with verifiable rewards, aiming to enhance the reasoning capabilities of LLMs (Zhang et al., 2025c). A central challenge in this field is achieving stable RL training while maintaining computational efficiency. To accelerate training in standard on-policy settings, it is common practice to perform multiple parameter updates per rollout batch. However, this introduces a distribution shift between the current policy and the data-generating (old) policy.

To address this deviation, importance sampling ratios, computed as the ratio of the current policy likelihood to the old policy likelihood, are typically used to correct the distribution mismatch and reduce gradient estimate bias. Complementary to this, trust region methods are employed to explicitly ensure stability by constraining the extent of policy updates. While TRPO (Schulman et al., 2015a) enforces an explicit KL divergence constraint, modern algorithms like PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) utilize importance ratio-based clipping to impose a soft trust region.

Despite being crucial for advancing LLM capabilities, RL incurs high computational cost, particularly for reasoning tasks that involve generating and learning from long sequences. Although recent works have sought to improve training and sample efficiency (Yu et al., 2025; Yue et al., 2025b; Chen et al., 2025; Liu et al., 2025a; Zheng et al., 2025c;b), they retain the reliance on importance sampling ratios to correct for distribution shift. This imposes a non-negligible overhead: for example, obtaining probabilities of 128 sequences of length 32K tokens for a 32B model requires approximately 4.06×10^{17} FLOPs. We challenge this standard practice, questioning whether the computational cost of importance sampling is strictly necessary for stability when policy drift is otherwise managed. In this work, we therefore ask:

Can we achieve more efficient LLM RL training without having to compute importance sampling ratios and maintain training stability?

In Section 3.1, we demonstrate that the importance sampling correction is not essential for stability, whereas the trust region mechanism is critical. Specifically, we find that removing the correction

[*]Equal contribution.
[†]Work done during an internship at KRAFTON.

term from GRPO does not destabilize training, provided clipping remains. Moreover, relying on the importance ratio for the trust region constraint has a fundamental limitation. As we show in Section 3.2, ratio-based clipping fails in the fully on-policy setting, where a single policy update is performed per rollout batch. In this regime, the importance ratio remains constant at 1, effectively removing the trust region constraint and leading to collapse.

Based on these observations, we propose Likelihood-Gated Policy Optimization (LGPO) in Section 4, along with a theoretical analysis motivating the use of likelihood-based gating. LGPO enforces a trust region via likelihood-based gating without computing importance ratios. Experiments on Qwen2.5 models (Qwen et al., 2025; Yang et al., 2024) across Countdown and mathematical reasoning tasks show that LGPO matches GRPO's stability and performance while reducing training time by $\sim$18% (Figure 3 and Table 1). Furthermore, LGPO remains stable in the fully on-policy setting where GRPO fails (Figure 4). We further perform case studies and ablations to better understand LGPO and the mechanism behind its stability (Section 5.2).

Overall, our contributions are summarized as follows:

1. We dissect the role of importance ratios, showing that the correction term is dispensable for stability in common LLM RL training settings, while the trust region constraint is crucial.
2. We identify a failure mode of ratio-based clipping in fully on-policy training, where it fails to constrain updates.
3. We propose LGPO, an efficient algorithm that enforces a trust region via likelihood-based gating without computing importance ratios, achieving comparable performance to GRPO with improved efficiency and robustness across training regimes, including the fully on-policy setting where GRPO fails.

2 BACKGROUND

A vanilla policy gradient algorithm in the context of RL training of LLMs aims to maximize the following objective:

$$\mathcal{L}^{\text{PG}}(\theta) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{T} A_t \log \pi_\theta(y_t|x, y_{<t})\right],$$

where x is a prompt sampled from the prompt distribution $\mathcal{D}$, and $y = (y_1, \ldots, y_T)$ is a sampled response of length T from the policy $\pi_\theta(\cdot \mid x)$. y_t denotes the token at position t and $y_{<t}$ its prefix. A_t is the advantage for token t.

REINFORCE (Williams, 1992) is a specific implementation of the vanilla policy gradient that uses a Monte Carlo estimator of the gradient, where the sampling is performed *fully on-policy*, which we define as the case where the data is always sampled from the current policy that is being optimized:

$$\nabla_\theta \mathcal{L}^{\text{PG}}(\theta) \approx \frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T_i} A_{i,t} \nabla_\theta \log \pi_\theta(y_{i,t}|x_i, y_{i,<t}).$$

Here N is the batch size, and $A_{i,t}$ is the advantage for the t-th token of the i-th sample.

Trust Region Policy Optimization (TRPO) (Schulman et al., 2015a) introduces the setting where multiple parameter updates using the samples generated from the "old policy" $\pi_{\theta_{\text{old}}}$ are possible, deviating from the fully on-policy setting. It is now common practice in LLM RL training to perform multiple parameter updates in a mini-batch manner to accelerate the training process. During these multiple parameter updates, a hard constraint on the KL divergence from the old policy is applied, preventing large policy changes, thus stabilizing the training. The gradient estimate is now a Monte Carlo estimate with *importance sampling ratio* correcting for the distribution mismatch between the old policy and the current policy. The TRPO objective is defined as:

$$\mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\sum_{t=1}^{T} r_t(\theta) A_t\right] \quad \text{s.t.} \quad \mathbb{E}\left[\sum_{t=1}^{T} \text{KL}\big(\pi_{\theta_{\text{old}}}(\cdot|x, y_{<t}) \,\|\, \pi_\theta(\cdot|x, y_{<t})\big)\right] \le \delta.$$

Here $r_t(\theta) = \frac{\pi_\theta(y_t|x,y_{<t})}{\pi_{\theta_{\text{old}}}(y_t|x,y_{<t})}$ is the importance sampling ratio and δ is a hyperparameter bounding the KL divergence. The expectation in the constraint is taken over $x \sim \mathcal{D}$ and $y \sim \pi_{\theta_{\text{old}}}(\cdot|x)$.

Proximal Policy Optimization (PPO) (Schulman et al., 2017) replaces the hard KL constraint of TRPO with the clipping operation shown in Eq. 1, which removes the incentive for moving the importance ratio outside of the interval $[1-\epsilon, 1+\epsilon]$, where ϵ is a hyperparameter. The PPO objective is defined as:

$$\mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_{\theta_{\text{old}}}(\cdot|x)}\left[\sum_{t=1}^{T}\min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right]. \tag{1}$$

This choice of clipping the objective based on the importance ratio can be viewed as serving two intertwined purposes: (1) discouraging updates that would push the policy far from the old policy, creating a "soft trust-region" effect, and (2) avoiding the use of samples with high importance ratios, thus reducing the variance of the gradient estimate.

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) is a variant of PPO that offers better efficiency and scalability. Instead of computing the advantage A_t using a learned value function via GAE (Schulman et al., 2015b), GRPO eliminates the value model by sampling a group of G responses $\{y_1, \ldots, y_G\}$ for each prompt x and computing advantages based on the relative rewards within the group. Specifically, for the i-th response in the group with reward R_i, the advantage is given by:

$$A_i = \frac{R_i - \mu_{\text{group}}}{\sigma_{\text{group}}},$$

where μ_{group} and σ_{group} are the mean and standard deviation of the rewards in the group. GRPO then optimizes the PPO objective (Eq. 1) using these group-relative advantages. Note that we do not incorporate a KL divergence penalty relative to the initial reference policy in this work, as training for complex reasoning tasks often requires significant deviation from the initialization (Yu et al., 2025).

Following PPO, modern RL algorithms specialized for LLM post-training, including GRPO and its variants (Zheng et al., 2025b; Yu et al., 2025; Yue et al., 2025b; Chen et al., 2025), use the importance ratio to correct for distribution mismatch and implement clipping. Please refer to Appendix F for a more detailed overview of related work. However, computing the importance ratio incurs additional memory and computational overhead at each sampling step. Given that in many practical LLM RL training settings the number of parameter updates per sampling step is small (e.g., 4), we ask: *is it possible to remove the importance ratio to achieve more efficient yet stable training?*

3 Are Importance Ratios Necessary for Stable Training?

3.1 Importance Sampling Correction is Dispensable

In PPO/GRPO, the importance ratio $r_t(\theta) = \frac{\pi_\theta(y_t|x,y_{<t})}{\pi_{\theta_{\text{old}}}(y_t|x,y_{<t})}$ plays two distinct roles. We can explicitly see this by examining the gradient of the surrogate objective for a single token at t (see derivation in Appendix A):

$$\nabla_\theta L_t(\theta) = \underbrace{\mathbf{1}_{\text{clip}}}_{\text{clipping}} \cdot A_t \cdot \underbrace{r_t(\theta)}_{\text{correction}} \cdot \nabla_\theta \log \pi_\theta(y_t|x, y_{<t}), \tag{2}$$

where $\mathbf{1}_{\text{clip}}$ is an indicator function that is 1 when the ratio is within the clipping range and 0 otherwise (see Appendix A for the full conditions). The term $r_t(\theta)$ in Equation 2 acts as a **correction** weight accounting for distribution shift, while $\mathbf{1}_{\text{clip}}$ implements **clipping**, the soft trust region constraint.

In this section, we examine the effect of removing each of these two roles one by one, and show that removing the correction role is not harmful to training stability in common LLM RL training settings, but removing the clipping role causes training instability.

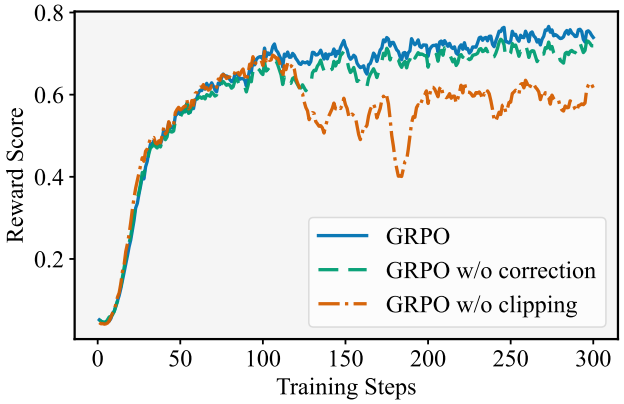


Figure 1: The reward score of the Countdown task during the RL training. We compare removing the importance sampling **correction** ($r_t(\theta)$ from Eq. 2) versus removing ratio-based **clipping** ($\mathbf{1}_{\text{clip}}$ from Eq. 2). Removing the correction has little effect, while removing clipping leads to significant instability.

We experiment with RL training of Qwen2.5-3B-Instruct on the Countdown task.[1] We use GRPO as our baseline algorithm. First, we remove the correction role by setting the gradient scaling term to 1, while preserving the ratio-based clipping mechanism (*w/o correction*). Second, we remove the clipping mechanism while retaining the importance sampling correction (*w/o clipping*). We denote the number of gradient updates performed per batch of sampled data as N_{updates} and use $N_{\text{updates}} = 2$ for this experiment.

We find that removing only the importance sampling correction from GRPO (*w/o correction*) has a negligible effect on training stability and only marginally worsens the scaling behavior, as illustrated in Figure 1. On the other hand, removing the clipping mechanism (*w/o clipping*) leads to significant instability. We observe similar findings for other GRPO-variants including DAPO (Yu et al., 2025) and GSPO (Zheng et al., 2025b): removing the correction term alone is not harmful to training stability (see Appendix E.1 and Figure 7).

These results indicate that the importance sampling correction is not strictly necessary, provided that a trust region constraint is maintained. Recall that importance ratio-based clipping serves two purposes: (1) enforcing a soft trust region, and (2) reducing variance from large importance ratios. With the correction term removed, the variance amplification caused by large importance ratios is eliminated, leaving clipping solely to enforce the trust region (see Appendix B for variance analysis). However, relying on the importance ratio for this purpose has a fundamental limitation. In the next section, we demonstrate that importance ratio-based clipping fails to provide any constraint in the fully on-policy setting, leading to training collapse. This failure further motivates our search for a trust region mechanism that is independent of the importance ratio.

3.2 Importance Ratio-Based Clipping Fails in Fully On-Policy Training

Recall that N_{updates} is the number of gradient updates performed per batch of sampled data. We define the *fully on-policy* setting as the case where $N_{\text{updates}} = 1$. In this setting, we sample rollouts using the current policy π_θ and perform exactly one gradient update using this data. Consequently, the behavior (old) policy $\pi_{\theta_{\text{old}}}$ is always identical to the current policy π_θ, and the importance ratio $r_t(\theta)$ remains fixed at 1. Since clipping is triggered only when the ratio deviates from 1, it is never activated in this setting, failing to serve as a trust region constraint.

We empirically observe that this leads to training collapse under the same experimental settings as in Section 3.1. As shown in Figure 2, fully on-policy GRPO, which effectively has no trust region

[1] Countdown is a verifiable numerical task where, given a set of numbers, the model must combine them using basic arithmetic operations to reach a target number.

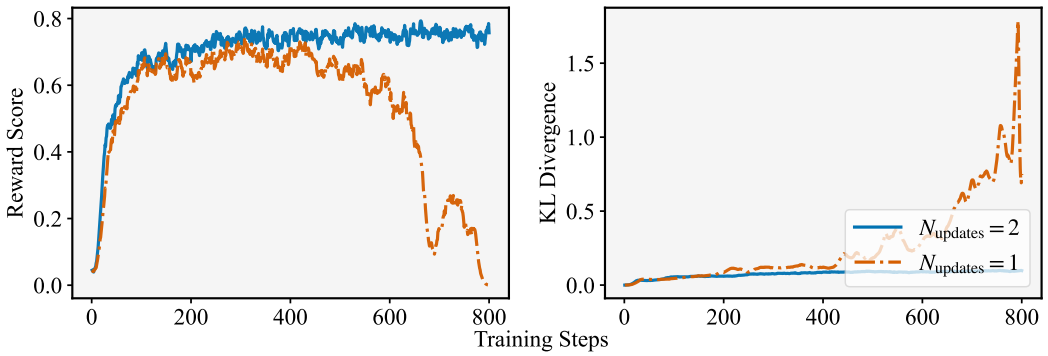


Figure 2: Performance comparison of GRPO with $N_{\text{updates}} = 1$ (fully on-policy) and $N_{\text{updates}} = 2$ on the Countdown task. N_{updates} denotes the number of gradient updates performed per batch of sampled data. We show the reward score (left) and the KL divergence from the reference policy (right). Without a functioning trust region constraint ($N_{\text{updates}} = 1$), training collapses and the policy diverges significantly from the reference model.

constraint, collapses after an initial phase of improvement. We also observe that the policy drifts far away from the initial reference policy.

In Section 3.1, we showed that the importance sampling correction is dispensable, meaning the importance ratio is only required for the trust region constraint. However, relying on the ratio for this purpose has two major drawbacks: (1) it incurs additional computational overhead to calculate policy likelihoods, and (2) as shown in this section, it fails to provide any constraint in the fully on-policy setting. These limitations motivate us to seek an alternative trust region mechanism that is both efficient (eliminating the need for importance ratios) and robust across training regimes.

4 Trust Region via Likelihood-Based Gating

We propose a simple yet effective alternative soft trust region constraint: likelihood-based gating. Here, we define the likelihood as the probability of the *current* policy generating the sampled data. We show that using solely the likelihood enables stable training without the need to compute or store old policy probabilities. To motivate this, we first theoretically analyze how the sample likelihood relates to the sensitivity of the policy update (measured by KL divergence) induced by the sample.

Theorem 4.1 (Minimal KL divergence under single-token perturbation)**.** *Fix a prompt x and a prefix $y_{<t}$. Let $p(\cdot) = \pi(\cdot \mid x, y_{<t}) \in \Delta^{K-1}$ denote the next-token distribution under some policy π, with $p_i > 0$. Consider another policy π' with next-token distribution $q(\cdot) = \pi'(\cdot \mid x, y_{<t}) = p(\cdot) + \Delta p(\cdot)$, where $\sum_{i=1}^{K} \Delta p_i = 0$ and $q_i > 0$.*

Fix a token k and enforce $\Delta p_k = \delta$ with $-p_k < \delta < 1 - p_k$. Then, as $|\delta| \to 0$,

$$\inf_{\substack{\Delta p_k = \delta \\ \sum_i \Delta p_i = 0}} \mathrm{KL}\big(p(\cdot) \,\|\, q(\cdot)\big) = \frac{\delta^2}{2\, p_k (1 - p_k)} + o(\delta^2).$$

In particular, for a fixed probability change $|\delta|$, the induced KL change grows as $p_k \to 0$ or $p_k \to 1$.

The proof is provided in Appendix C. In the context of LLM RL training, learning from a sample amounts to increasing or decreasing its likelihood under the policy. Theorem 4.1 implies that for a fixed magnitude of likelihood change, the induced policy change (measured by KL divergence) is largest when the sample likelihood is close to 0 or 1. This motivates our strategy of gating out samples with extreme likelihoods to mitigate these highly sensitive updates, thereby enforcing a soft trust region constraint.

Likelihood-Gated Policy Optimization (LGPO). Based on this motivation, we propose LGPO, which enforces the trust region constraint directly via likelihood-based gating. Given our focus

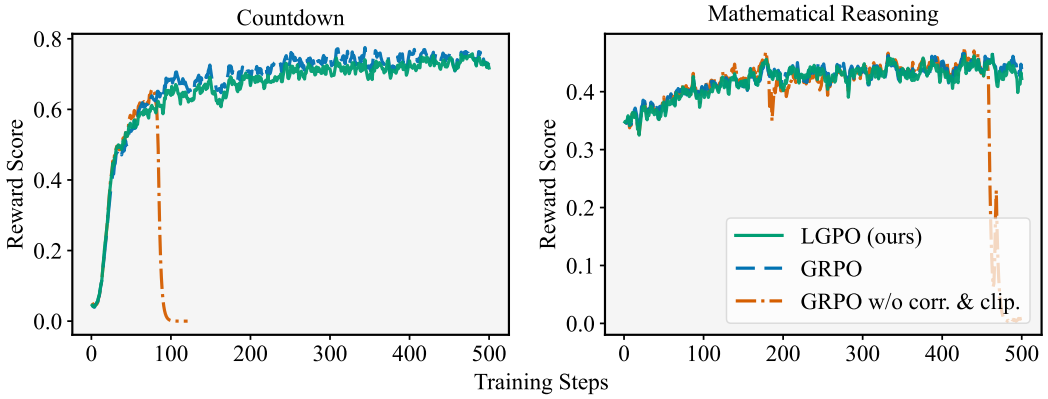


Figure 3: Training curves comparing LGPO, GRPO, and GRPO without importance sampling correction and clipping on Countdown (left) and mathematical reasoning (right). LGPO maintains stability comparable to GRPO, indicating that likelihood-based gating serves as an effective alternative trust region constraint.

on RL with verifiable rewards, where sequence-level outcome rewards are standard, we adopt the sequence-level average likelihood for likelihood-based gating. This ensures the trust region constraint is aligned with the learning signal (Zheng et al., 2025b). We define the *sequence-level average log-likelihood* under the current policy π_θ as

$$\bar{\ell}_\theta(x, y) := \frac{1}{T} \sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t}).$$

In practice, we only gate high-likelihood sequences, as low-likelihood sequences are rarely observed (see Appendix E.2).

We thus define the gating mask as

$$c_\theta(x, y) := \mathbf{1}(\bar{\ell}_\theta(x, y) \leq \tau),$$

where τ is a likelihood threshold in log-likelihood space. This yields the LGPO objective:

$$\mathcal{L}^{\text{LGPO}}(\theta) = \mathbb{E}_{x \sim \mathcal{D},\, y \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[c_\theta(x, y) \sum_{t=1}^{T} A_t \log \pi_\theta(y_t \mid x, y_{<t}) \right],$$

where $c_\theta(x, y)$ serves as a stop-gradient mask, effectively zeroing out the gradient contribution from samples that exceed the likelihood threshold. The full training procedure can be found in Algorithm 1 in Appendix D.

5 Experiments

We experiment with Qwen2.5-Math-1.5B and Qwen2.5-3B-Instruct. We first analyze training dynamics on the Countdown task, then extend to mathematical reasoning and report final performance on MATH500 (Lightman et al., 2023), OlympiadBench (He et al., 2024), and AIME24&25 (AIME, 2025) (average Pass@1 over 8 samples), along with training curves. For mathematical reasoning, we train on the high-quality dataset curated by Huan et al. (2025).[2] Unless otherwise specified, during RL training, we use a minibatch size of 64. For Countdown, we use a prompt batch size of 128 and perform two gradient updates per rollout batch ($N_{\text{updates}} = 2$). For mathematical reasoning, we use a prompt batch size of 256 and perform four gradient updates per rollout batch ($N_{\text{updates}} = 4$). We compute advantages using group relative normalization as in GRPO with a group size of $G = 8$. For LGPO, we set the likelihood threshold $\tau = -0.15$, which corresponds to $\exp(\tau) \approx 0.86$ average likelihood in probability space. We compare the LGPO objective against GRPO and GRPO *w/o correction & clipping* (introduced in Section 3.1).

[2] https://huggingface.co/datasets/ReasoningTransferability/math_rl_48k

Table 1: Final evaluation performance on mathematical reasoning benchmarks.

Benchmark	GRPO	LGPO
Qwen2.5-3B-Instruct		
MATH500	64.4	**65.2**
OlympiadBench	29.2	**30.7**
AIME24	**5.0**	4.2
AIME25	3.3	**3.7**
Avg.	*25.5*	***26.0***
Qwen2.5-Math-1.5B		
MATH500	73.0	**73.6**
OlympiadBench	**38.2**	35.0
AIME24	12.1	**17.9**
AIME25	8.8	**9.6**
Avg.	*33.0*	***34.0***

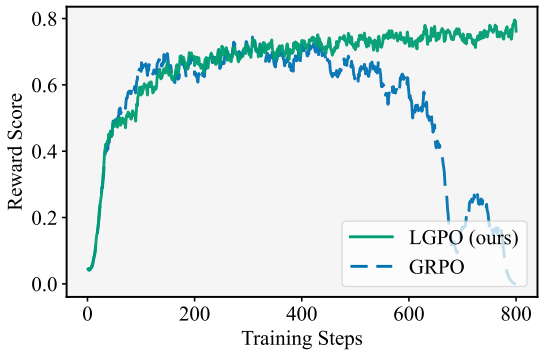


Figure 4: Fully on-policy ($N_{\text{updates}} = 1$) training dynamics on Countdown using Qwen2.5-3B-Instruct. GRPO collapses due to the lack of an effective trust region constraint when the importance ratio is fixed at 1, whereas LGPO remains stable via likelihood-based gating.

5.1 Main Results

Figure 3 shows that **LGPO achieves stable training without using importance ratios** on both Countdown and the mathematical reasoning training dataset. While removing importance sampling correction and clipping from GRPO leads to unstable training (*GRPO w/o corr. & clip.*), adding simple likelihood-based gating, yielding the LGPO objective, stabilizes training. We observe a negligible drop in peak reward score compared to GRPO. Table 1 reports evaluation scores after training for the same number of steps, demonstrating similar performance in learning mathematical reasoning capabilities that generalize to broader tasks.

While maintaining similar performance, **LGPO offers training speedups** by eliminating the need to compute old policy likelihoods. When training Qwen2.5-3B-Instruct on $\sim$7.5K training data using 2 NVIDIA H100 GPUs, GRPO takes 3.37 hours to train one epoch, while LGPO takes 2.85 hours, corresponding to a $\sim$18% speedup.

We also evaluate LGPO in the *fully on-policy* setting ($N_{\text{updates}} = 1$). For this experiment, we set the minibatch size to 128, matching the prompt batch size, to perform a single gradient update per rollout batch. As analyzed in Section 3.2 and shown in Figure 2, GRPO collapses in this setting because the importance ratio is constant at 1, falling back to a vanilla policy gradient without any trust region constraint. In contrast, Figure 4 shows that LGPO remains stable. This stability stems from LGPO's gating condition depending solely on the current policy's likelihood, independent of the importance ratio. Consequently, even when the importance ratio is constant at 1, LGPO effectively filters out high-likelihood samples that induce large policy updates, maintaining a robust trust region.

Having established LGPO's effectiveness and stability, we now turn to a deeper analysis of its underlying mechanisms.

5.2 Ablation Studies and Analysis

How does LGPO affect generation entropy? We observe that LGPO mitigates the rapid entropy collapse typically seen in GRPO during training on both the Countdown task (Figure 5) and the mathematical reasoning task (Figure 9 in Appendix E.3). By gating out updates from already high-likelihood samples, LGPO prevents the policy from becoming overly deterministic and sustains exploration, consistent with prior observations on entropy reduction (Cui et al., 2025).

Do high-likelihood samples really collapse training? We test a random gating baseline that discards samples at the same rate as LGPO, but randomly regardless of likelihood; as shown in Figure 6, it collapses within 100 steps. This confirms that LGPO's stability arises from selectively suppressing high-likelihood updates, rather than simply reducing the number of active samples.

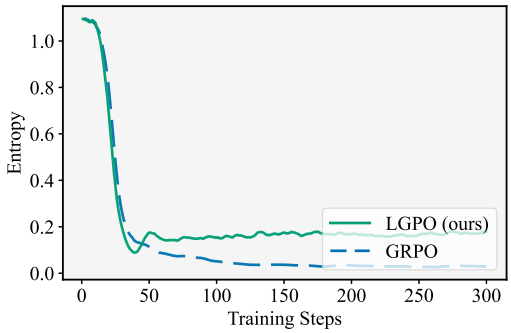


Figure 5: Generation entropy during training on Countdown. LGPO maintains higher entropy compared to GRPO, preventing premature convergence to a deterministic policy.

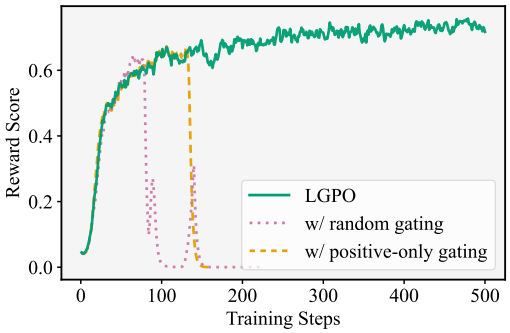


Figure 6: Countdown training curves comparing LGPO with random gating regardless of likelihood and gating only high-likelihood samples with positive advantage.

Do negative advantage high-likelihood samples also collapse training? We investigate whether it suffices to gate only high-likelihood samples with positive advantage (which induce positive gradient updates) while keeping those with negative advantage. Although Theorem 4.1 suggests that changing an extreme-likelihood probability in either direction induces a large KL change, negative-advantage updates might seem safer because they push probabilities away from the boundary, unlike positive updates that increase determinism. However, as shown in Figure 6, gating only positive-advantage high-likelihood samples still leads to collapse. This indicates that suppressing negative-advantage updates on high-likelihood samples is also crucial for stability, consistent with Theorem 4.1.

Can we stochastically gate high-likelihood samples and find a sweet spot between stability and performance? Since LGPO discards all learning signals from high-likelihood samples, one might wonder if stochastically gating only a fraction of them could balance stability and performance. We investigate this by applying LGPO's gating mask with a certain probability. As shown in Appendix E.3 (Figure 10), we find no such sweet spot. Allowing even a small fraction of high-likelihood samples to contribute leads to collapse, with lower gating probabilities causing earlier failure.

Which samples are being gated in LGPO? Are we losing useful training signals? To assess how much useful learning signal LGPO discards, we compare it to GSPO, a sequence-level clipping variant of GRPO, as a fair baseline that matches the sequence-level granularity, and track per-step discard statistics throughout training. Although Appendix E.3 (Figure 11, top) shows that LGPO discards a larger fraction of samples overall, we find that most of these additionally discarded samples have zero advantage ($A = 0$) and would contribute no policy gradient regardless. Consequently, the effective training signal that remains (measured by the fraction of retained samples with non-zero advantage, i.e., contributing non-zero gradient) stays comparable between LGPO and GSPO across training steps (Figure 11, bottom). This suggests that LGPO does not discard significantly more useful learning signal than standard importance ratio-based clipping methods.

6 Conclusion

We propose Likelihood-Gated Policy Optimization (LGPO), a novel algorithm that accelerates LLM RL training while ensuring stability. LGPO is motivated by the insight that importance sampling correction is dispensable for stability, and by theoretical analysis showing that extreme-likelihood samples induce large policy updates. By replacing ratio-based clipping with likelihood-based gating, LGPO enforces a trust region without the overhead of computing old-policy likelihoods. Furthermore, we identify a failure mode of ratio-based clipping in fully on-policy settings and demonstrate that LGPO remains robust where prior methods collapse. Overall, LGPO offers a more efficient and stable framework for LLM RL training. We discuss limitations and promising future directions in Appendix G.

REFERENCES

AIME. AIME Problems and Solutions. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions, 2025.

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, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

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

Yaru Hao, Li Dong, Xun Wu, Shaohan Huang, Zewen Chi, and Furu Wei. On-policy rl with optimal reward baseline. *arXiv preprint arXiv:2505.23585*, 2025.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3828–3850, 2024.

Maggie Huan, Yuetai Li, Tuney Zheng, Xiaoyu Xu, Seungone Kim, Minxin Du, Radha Poovendran, Graham Neubig, and Xiang Yue. Does math reasoning improve general llm capabilities? understanding transferability of llm reasoning. *arXiv preprint arXiv:2507.00432*, 2025.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2023.

Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. Cppo: Accelerating the training of group relative policy optimization-based reasoning models. *arXiv preprint arXiv:2503.22342*, 2025.

Liyuan Liu, Feng Yao, Dinghuai Zhang, Chengyu Dong, Jingbo Shang, and Jianfeng Gao. Flashrl: 8bit rollouts, full power rl, August 2025a. URL https://fengyao.notion.site/flash-rl.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*, 2025b.

Qwen, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Nicolas Le Roux, Marc G Bellemare, Jonathan Lebensold, Arnaud Bergeron, Joshua Greaves, Alex Fréchette, Carolyne Pelletier, Eric Thibodeau-Laufer, Sándor Toth, and Sam Work. Tapered off-policy reinforce: Stable and efficient reinforcement learning for llms. *arXiv preprint arXiv:2503.14286*, 2025.

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, 2015a.

John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, and Pieter Abbeel. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*, 2015b.

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, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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.

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

Changyi Xiao, Mengdi Zhang, and Yixin Cao. Bnpo: Beta normalization policy optimization. *arXiv preprint arXiv:2506.02864*, 2025.

Jian Xiong, Jingbo Zhou, Jingyong Ye, and Dejing Dou. Aapo: Enhance the reasoning capabilities of llms with advantage momentum. *arXiv preprint arXiv:2505.14264*, 2025.

Zhongwen Xu and Zihan Ding. Single-stream policy optimization. *arXiv preprint arXiv:2509.13232*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025a.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025b.

Han Zhang, Ruibin Zheng, Zexuan Yi, Zhuo Zhang, Hanyang Peng, Hui Wang, Zike Yuan, Cai Ke, Shiwei Chen, Jiacheng Yang, et al. Gepo: Group expectation policy optimization for stable heterogeneous reinforcement learning. *arXiv preprint arXiv:2508.17850*, 2025a.

Kaichen Zhang, Yuzhong Hong, Junwei Bao, Hongfei Jiang, Yang Song, Dingqian Hong, and Hui Xiong. Gvpo: Group variance policy optimization for large language model post-training. *arXiv preprint arXiv:2504.19599*, 2025b.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, et al. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*, 2025c.

Zizhuo Zhang, Jianing Zhu, Xinmu Ge, Zihua Zhao, Zhanke Zhou, Xuan Li, Xiao Feng, Jiangchao Yao, and Bo Han. Co-rewarding: Stable self-supervised rl for eliciting reasoning in large language models. *arXiv preprint arXiv:2508.00410*, 2025d.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. *arXiv preprint arXiv:2505.03335*, 2025a.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, et al. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*, 2025b.

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.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025b.

Haizhong Zheng, Yang Zhou, Brian R Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. Act only when it pays: Efficient reinforcement learning for llm reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*, 2025c.

A Gradient of the PPO Objective and the Role of Ratio-Based Clipping

In this section, we derive the gradient of the PPO/GRPO surrogate objective to explicitly show how the importance ratio $r_t(\theta)$ acts as the gating mechanism for gradient updates.

Recall the PPO objective from Eq. 1:

$$\mathcal{L}^{\text{PPO}}(\theta) = \mathbb{E}\left[\sum_{t=1}^{T} \min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)\right],$$

where $r_t(\theta) = \frac{\pi_\theta(y_t|x,y_{<t})}{\pi_{\theta_{\text{old}}}(y_t|x,y_{<t})}$.

We consider the contribution of a single token at position t to the gradient. Let $L_t(\theta) = \min(r_t(\theta)A_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon)A_t)$. We compute $\nabla_\theta L_t(\theta)$ by analyzing the two cases of the $\min$ operator.

Case 1: $A_t > 0$. The objective becomes $\min(r_t(\theta), \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon))A_t$. The clipping term is upper-bounded by $1+\epsilon$.

- If $r_t(\theta) \leq 1+\epsilon$, the active term is $r_t(\theta)A_t$. The gradient is $\nabla_\theta(r_t(\theta)A_t) = A_t\nabla_\theta r_t(\theta)$.
- If $r_t(\theta) > 1+\epsilon$, the active term is $(1+\epsilon)A_t$. Since this is constant with respect to θ (locally), the gradient is $\mathbf{0}$.

Case 2: $A_t < 0$. The objective becomes $\max(r_t(\theta), \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon))A_t$ (taking A_t out effectively flips min to max for the ratio comparison).

- If $r_t(\theta) \geq 1-\epsilon$, the active term is $r_t(\theta)A_t$. The gradient is $A_t\nabla_\theta r_t(\theta)$.
- If $r_t(\theta) < 1-\epsilon$, the active term is $(1-\epsilon)A_t$. The gradient is $\mathbf{0}$.

Gradient Expression. Using the identity $\nabla_\theta r_t(\theta) = r_t(\theta)\nabla_\theta \log \pi_\theta(y_t|x, y_{<t})$, we can combine these cases. The gradient is non-zero only when the importance ratio is within the "active" unclipped region. Defining the active indicator $\mathbf{1}_{\text{clip}}$:

$$\mathbf{1}_{\text{clip}} = \mathbf{1}\left[(A_t > 0 \wedge r_t(\theta) \leq 1+\epsilon) \vee (A_t < 0 \wedge r_t(\theta) \geq 1-\epsilon)\right].$$

The gradient for token t is:

$$\nabla_\theta L_t(\theta) = \mathbf{1}_{\text{clip}} \cdot A_t \cdot r_t(\theta) \cdot \nabla_\theta \log \pi_\theta(y_t|x, y_{<t}).$$

This derivation clarifies that the importance ratio $r_t(\theta)$ serves two roles as discussed in Section 3.1: (1) scaling the gradient magnitude via the multiplicative term $r_t(\theta)$, and (2) gating the updates (ratio-based clipping) by zeroing out the gradient via the condition in $\mathbf{1}_{\text{clip}}$. Specifically, if the ratio deviates too far from 1 (exceeds $1+\epsilon$ for positive advantage or drops below $1-\epsilon$ for negative advantage), the gradient is clipped to zero.

B IMPORTANCE SAMPLING VARIANCE ANALYSIS

This section explains (1) why large importance ratios can increase the variance of stochastic gradient estimates when used as multiplicative correction weights, and (2) why this specific source of variance disappears when the correction weight is removed (i.e., set to 1).

Setup and notation. Fix a prompt x and consider a response $y = (y_1, \ldots, y_T)$ generated by the old policy, $y \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. For each token position t, define the score function

$$s_t(\theta) = \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}),$$

and the token-level importance ratio

$$r_t(\theta) = \frac{\pi_\theta(y_t \mid x, y_{<t})}{\pi_{\theta_{\text{old}}}(y_t \mid x, y_{<t})}.$$

Let A_t denote the token-level advantage used in the surrogate objective. We assume absolute continuity: $\pi_{\theta_{\text{old}}}(y_t \mid x, y_{<t}) > 0$ whenever $\pi_\theta(y_t \mid x, y_{<t}) > 0$.

Surrogate objective and its gradient. Recall the PPO clipped surrogate objective in Eq. 1. Without clipping, the resulting surrogate objective for a fixed prompt x can be written as

$$\mathcal{L}_x(\theta) := \mathbb{E}_{y \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[\sum_{t=1}^{T} r_t(\theta)\, A_t \right].$$

Using $\nabla_\theta r_t(\theta) = r_t(\theta)\, \nabla_\theta \log \pi_\theta(y_t \mid x, y_{<t}) = r_t(\theta)\, s_t(\theta)$, the gradient is

$$\nabla_\theta \mathcal{L}_x(\theta) = \mathbb{E}_{y \sim \pi_{\theta_{\text{old}}}(\cdot|x)} \left[\sum_{t=1}^{T} r_t(\theta)\, A_t\, s_t(\theta) \right].$$

Large importance sampling correction can increase variance. To analyze variance, we consider the scalar projection of the gradient onto a unit vector u:

$$Z_t^{\text{corr}} := u^\top \big(r_t(\theta)\, A_t\, s_t(\theta) \big) = r_t(\theta)\, A_t \, \big(u^\top s_t(\theta) \big).$$

Then

$$\begin{aligned} \operatorname{Var}(Z_t^{\text{corr}}) &= \mathbb{E}\Big[\big(Z_t^{\text{corr}} \big)^2 \Big] - (\mathbb{E}[Z_t^{\text{corr}}])^2 \\ &= \mathbb{E}\Big[r_t(\theta)^2\, A_t^2 \, \big(u^\top s_t(\theta) \big)^2 \Big] - \big(\mathbb{E}\big[r_t(\theta)\, A_t \, \big(u^\top s_t(\theta) \big) \big] \big)^2 . \end{aligned} \tag{3}$$

Equation equation 3 makes the dependence on $r_t(\theta)^2$ explicit through the first term. If $r_t(\theta)$ is heavy-tailed or occasionally very large, the contribution $\mathbb{E}[r_t(\theta)^2\, A_t^2\, (u^\top s_t(\theta))^2]$ can become large, increasing variance and sensitivity of the gradient estimator. This is why clipping large $r_t(\theta)$ can reduce the variance contributed by the correction weights.

Variance analysis after removing correction. After removing the importance sampling correction by setting the correction weight to 1, define

$$Z_t^{\text{no-corr}} := u^\top \big(A_t\, s_t(\theta) \big) = A_t \, \big(u^\top s_t(\theta) \big).$$

Analogously,

$$\begin{aligned} \operatorname{Var}(Z_t^{\text{no-corr}}) &= \mathbb{E}\Big[\big(Z_t^{\text{no-corr}} \big)^2 \Big] - (\mathbb{E}[Z_t^{\text{no-corr}}])^2 \\ &= \mathbb{E}\Big[A_t^2 \, \big(u^\top s_t(\theta) \big)^2 \Big] - \big(\mathbb{E}\big[A_t \, \big(u^\top s_t(\theta) \big) \big] \big)^2 , \end{aligned}$$

which no longer contains $r_t(\theta)$. Thus, removing correction eliminates the specific variance amplification mechanism caused by large importance ratios.

C PROOF OF THEOREM 4.1

Fix a prompt x and a prefix $y_{<t}$, and let $p = (p_1, \dots, p_K)$ denote the categorical next-token distribution $p_i = \pi(i \mid x, y_{<t})$ with $p_i > 0$ and $\sum_i p_i = 1$. Let $q = p + \Delta p$ be another distribution induced by some policy π', where $\sum_i \Delta p_i = 0$ and $q_i > 0$. We write $\mathrm{KL}(p\|q) = \sum_i p_i \log \frac{p_i}{q_i}$.

Lemma C.1 (Quadratic expansion of KL near p)**.** *Let $p_{\min} := \min_i p_i > 0$ and assume $\|\Delta p\|_\infty \le \frac{1}{2} p_{\min}$. Then, as $\|\Delta p\|_2 \to 0$,*

$$\mathrm{KL}(p \,\|\, p + \Delta p) = \frac{1}{2} \sum_{i=1}^{K} \frac{(\Delta p_i)^2}{p_i} \;+\; o(\|\Delta p\|_2^2).$$

Proof. Write $x_i := \Delta p_i / p_i$, so $q_i = p_i(1 + x_i)$ and

$$\mathrm{KL}(p\|q) = \sum_i p_i \log \frac{p_i}{q_i} = -\sum_i p_i \log(1 + x_i).$$

Using $\log(1 + x) = x - \frac{1}{2}x^2 + R(x)$ with $R(x) = O(|x|^3)$ as $x \to 0$, we obtain

$$\mathrm{KL}(p\|q) = -\sum_i p_i x_i + \tfrac{1}{2} \sum_i p_i x_i^2 - \sum_i p_i R(x_i).$$

Because $\sum_i \Delta p_i = 0$, the linear term vanishes: $\sum_i p_i x_i = \sum_i \Delta p_i = 0$. Thus

$$\mathrm{KL}(p\|q) = \tfrac{1}{2} \sum_i \frac{(\Delta p_i)^2}{p_i} - \sum_i p_i R\Big(\tfrac{\Delta p_i}{p_i}\Big).$$

When $\|\Delta p\|_\infty \le \frac{1}{2} p_{\min}$ we have $|x_i| \le \frac{1}{2}$ and $|R(x_i)| \le C|x_i|^3$ for a universal C, hence

$$\Big| \sum_i p_i R\Big(\tfrac{\Delta p_i}{p_i}\Big) \Big| \le C \sum_i \frac{|\Delta p_i|^3}{p_i^2} \le \frac{C}{p_{\min}^2} \|\Delta p\|_\infty \sum_i (\Delta p_i)^2 = O(\|\Delta p\|_2^3) = o(\|\Delta p\|_2^2).$$

□

Lemma C.2 (Minimal quadratic cost for a single coordinate change)**.** *Fix k and a small non-zero change δ with $-p_k < \delta < 1 - p_k$. Among perturbations satisfying $\Delta p_k = \delta$ and $\sum_i \Delta p_i = 0$, the quadratic term in Lemma C.1 is minimized by*

$$\Delta p_j^* = -\,\delta\, \frac{p_j}{1 - p_k} \qquad (j \ne k),$$

and the minimal value equals

$$\min_{\substack{\Delta p_k = \delta \\ \sum_i \Delta p_i = 0}} \frac{1}{2} \sum_{i=1}^{K} \frac{(\Delta p_i)^2}{p_i} = \frac{\delta^2}{2\, p_k (1 - p_k)}.$$

Proof. The constraint $\sum_i \Delta p_i = 0$ implies $\sum_{j \ne k} \Delta p_j = -\delta$. Parameterize $\Delta p_j = -\delta r_j$ for $j \ne k$ with $\sum_{j \ne k} r_j = 1$. Then

$$\frac{1}{2} \sum_{i=1}^{K} \frac{(\Delta p_i)^2}{p_i} = \frac{1}{2} \left(\frac{\delta^2}{p_k} + \delta^2 \sum_{j \ne k} \frac{r_j^2}{p_j} \right).$$

Thus it suffices to minimize $\sum_{j \ne k} r_j^2 / p_j$ subject to $\sum_{j \ne k} r_j = 1$. Using Lagrange multipliers yields $r_j^* \propto p_j$, hence $r_j^* = \frac{p_j}{1 - p_k}$ and $\Delta p_j^* = -\delta \frac{p_j}{1 - p_k}$. Substituting back gives the stated minimum $\frac{\delta^2}{2 p_k (1 - p_k)}$. □

Proof of Theorem 4.1. By Lemma C.1,

$$\mathrm{KL}(p\|p+\Delta p) = \frac{1}{2}\sum_{i=1}^{K}\frac{(\Delta p_i)^2}{p_i} \;+\; o(\|\Delta p\|_2^2).$$

Under the constraints $\Delta p_k = \delta$ and $\sum_i \Delta p_i = 0$, Lemma C.2 gives

$$\min\; \frac{1}{2}\sum_{i=1}^{K}\frac{(\Delta p_i)^2}{p_i} = \frac{\delta^2}{2\,p_k(1-p_k)}.$$

Moreover, for the minimizing perturbation Δp^*, we have $\|\Delta p^*\|_2 = \Theta(|\delta|)$ for fixed p, so the remainder term is $o(\|\Delta p^*\|_2^2) = o(\delta^2)$. Combining these yields

$$\inf_{\substack{\Delta p_k=\delta \\ \sum_i \Delta p_i=0}} \mathrm{KL}(p\,\|\,p+\Delta p) = \frac{\delta^2}{2\,p_k(1-p_k)} + o(\delta^2),$$

as $|\delta| \to 0$, completing the proof. □

D Algorithmic Details

Algorithm 1 Likelihood-Gated Policy Optimization

Require: current policy π_θ, old (behavior) policy $\pi_{\theta_{\text{old}}}$, likelihood threshold τ, number of gradient updates per rollout batch N_{updates}

for each rollout batch **do**
  Roll out responses $y \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and compute advantages $\{A_t\}_{t=1}^{T}$ to form a rollout buffer $\mathcal{B} = \{(x, y, \{A_t\}_{t=1}^{T})\}$.
  Partition $\mathcal{B}$ into N_{updates} disjoint minibatches $\{\mathcal{B}_n\}_{n=1}^{N_{\text{updates}}}$.
  for $n = 1$ **to** N_{updates} **do**
    Initialize minibatch loss $L \leftarrow 0$.
    for each $(x, y, \{A_t\}_{t=1}^{T}) \in \mathcal{B}_n$ **do**
      $\bar{\ell}_\theta(x, y) \leftarrow \frac{1}{T}\sum_{t=1}^{T} \log \pi_\theta(y_t \mid x, y_{<t})$.
      $c_\theta(x, y) \leftarrow \mathbf{1}(\bar{\ell}_\theta(x, y) \le \tau)$ *(stop-gradient)*
      $L \leftarrow L - c_\theta(x, y)\sum_{t=1}^{T} A_t \log \pi_\theta(y_t \mid x, y_{<t})$.
    end for
    Update θ using $\nabla_\theta L$.
  end for
end for

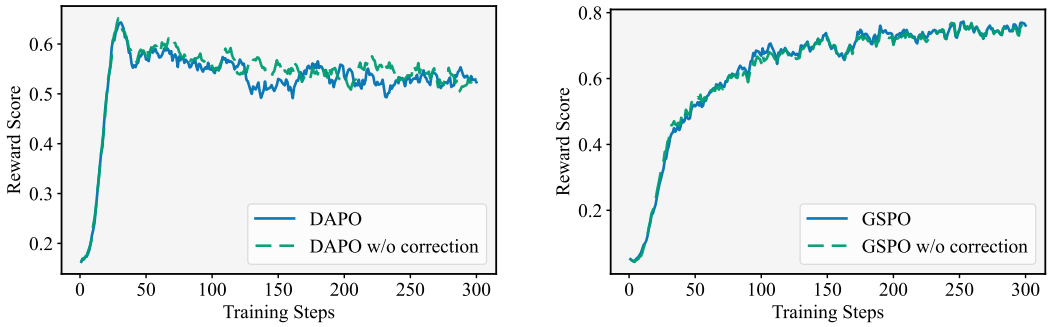
E Additional Experimental Results

E.1 Removing Importance Sampling Correction in DAPO and GSPO

In Section 3.1, we show that removing the importance sampling correction from GRPO has little effect on training stability. We further investigate the effect of removing the importance sampling correction in other GRPO-variants, specifically DAPO and GSPO. Similar to our findings with GRPO, removing the correction term alone does not destabilize training for these algorithms (Figure 7).

E.2 Likelihood Distribution of Sampled Responses

Figure 8 shows the distribution of the sequence-level average likelihood of responses sampled from Qwen2.5-3B-Instruct on the MATH500 dataset. Since low-likelihood sequences are rarely observed, LGPO only gates high-likelihood sequences to enforce the trust region constraint.



(a) DAPO (b) GSPO

Figure 7: The reward score of the Countdown task during the RL training. Removing the importance sampling correction has little effect on training stability for both DAPO and GSPO.

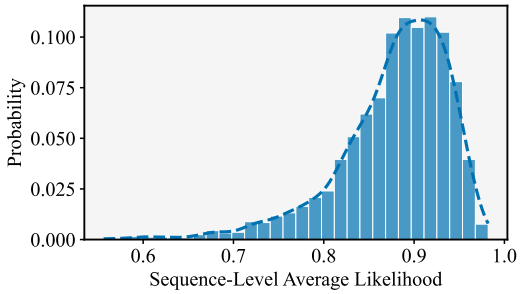


Figure 8: Distribution of sequence-level average likelihood $\exp\left(\bar{\ell}_\theta(x, y)\right)$ of responses sampled from Qwen2.5-3B-Instruct on MATH500.

E.3 Ablation Studies and Analysis: Additional Figures

This subsection provides additional figures supporting Section 5.2. Figure 9 reports generation entropy during mathematical reasoning training, showing that LGPO mitigates entropy collapse relative to GRPO. Figure 10 studies stochastic gating and shows that relaxing the gating probability leads to earlier collapse. Figure 11 analyzes per-step discard statistics and shows that, despite discarding more total samples, LGPO retains a comparable fraction of non-zero-gradient samples.

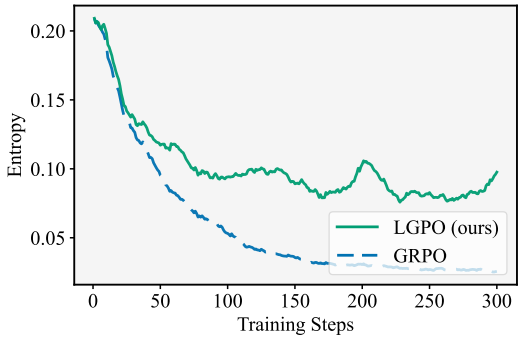


Figure 9: Generation entropy during training on the mathematical reasoning dataset. LGPO prevents rapid entropy collapse, similar to the trend observed in the Countdown task.

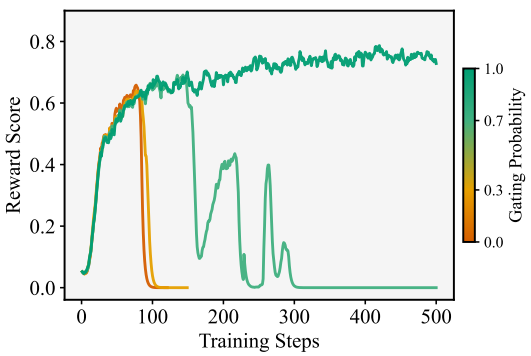


Figure 10: Effect of stochastic gating probability on training stability. Reducing the gating probability leads to earlier training collapse, indicating that strict gating is necessary for stability.

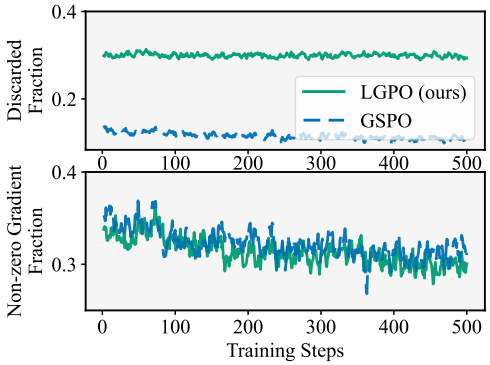


Figure 11: Although LGPO discards a larger fraction of total samples (top), the fraction of samples contributing non-zero gradient ($A \neq 0$ and retained) remains comparable to GSPO (bottom).

F Related Work

Reinforcement learning (RL) is increasingly applied to enhance LLM reasoning by adjusting the model's output distribution. Studies suggest that RL updates tend to concentrate probability mass on valid reasoning paths (Yue et al., 2025a; Wang et al., 2025). Additionally, research explores how prolonged training or self-play might extend the solution space beyond the base model's initial capabilities (Liu et al., 2025b; Zhao et al., 2025a), with recent analysis highlighting the role of entropy regulation in this process (Cui et al., 2025).

To address the inherent instability of RL, recent methods focus on refining optimization objectives and learning signals. One category targets importance sampling variance. Several approaches shift policy optimization from the token level to sequence-level or group-level expectations to reduce volatility (Zheng et al., 2025b; Zhang et al., 2025a; Chen et al., 2025). Others modify constraint mechanisms, such as using selective clipping for negative samples (Roux et al., 2025) or adaptive gating (Gao et al., 2025), while GVPO alternatively reformulates the objective as a variance-matching regression (Zhang et al., 2025b). Notably, SAPO (Gao et al., 2025) replaces hard clipping with soft gating but still operates on importance ratios; in contrast, LGPO eliminates the need for importance ratios entirely by gating on raw likelihood. The second category improves stability through robust learning signals. This includes enhancing advantage estimation via optimal baselines and global normalization (Hao et al., 2025; Xiong et al., 2025; Xu & Ding, 2025), as well as refining reward aggregation to handle noise (Xiao et al., 2025; Zhao et al., 2025b). To prevent reward hacking and collapse, recent works incorporate dynamic sample filtering (Lin et al., 2025; Yu et al., 2025) and cross-view invariance constraints (Zhang et al., 2025d).

G Limitations and Future Work

While LGPO demonstrates improved efficiency and stability, our study has limitations. First, due to computational constraints, we have not validated the method on large-scale models (e.g., >7B parameters). We note that concurrent work training a 30B MoE model (Zheng et al., 2025a) finds that importance sampling correction contributes to training stability at that scale, suggesting that our findings may be scale-dependent. Second, although we test up to $N_{\text{updates}} = 4$ updates per batch, reflecting common practice, we have not systematically identified the exact boundary where the lack of importance sampling correction might destabilize training. The necessity of this correction likely depends on a complex interplay of N_{updates}, learning rate, model size, and task difficulty, making it challenging to define a strict failure threshold. For future work, we envision improving sample efficiency through decoding-stage interventions. Since LGPO gates out high-likelihood samples, actively avoiding such samples during generation could prevent wasted computation, further enhancing efficiency. We leave the exploration of such active sampling strategies to future research.