Is the Importance Ratio Necessary for Stable Reinforcement Learning in LLMs?

Shuibai Zhang[*†] University of Wisconsin–Madison
Junhyuck Kim[*] KRAFTON
Gyeongman Kim KRAFTON
Jaewoong Cho 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 can lead 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 relative to a standard implementation that recomputes old-policy likelihoods. 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., 2025b,a], 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

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

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

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. Moreover, 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: 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% relative to the standard recipe that recomputes old-policy likelihoods (Figure 3 and Table 1). Furthermore, LGPO remains stable in the fully on-policy setting where GRPO fails (Figure 4), even when a KL penalty is added to GRPO (Section 5.1). 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 in common LLM RL settings their stabilizing effect comes from the trust region implemented by clipping, while the correction term is dispensable across GRPO, Dr. GRPO, DAPO, and GSPO.
2. We identify a failure mode of ratio-based clipping in fully on-policy training, where the importance ratio is identically 1 and clipping never constrains updates.
3. We propose LGPO, which enforces a trust region via likelihood-based gating using only the current policy's likelihood, without old- or reference-policy log-probabilities. LGPO matches GRPO with less training time and remains stable in the fully on-policy setting where GRPO, with or without a small KL penalty, collapses.

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:

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

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:

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

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]; we evaluate a KL-regularized variant as a baseline in Section 5.1.

Following PPO, modern RL algorithms specialized for LLM post-training, including GRPO and its variants [Zheng et al., 2025a, Yu et al., 2025, Yue et al., 2025b, Chen et al., 2025], use the importance ratio to correct for distribution mismatch and implement clipping. 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):

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



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. Runs in this section are trained for 300 steps, sufficient to show the stability contrast.

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. The correction weight makes the surrogate an unbiased estimate of the on-policy objective when training on samples from $\pi_{\theta_{\text{old}}}$, so removing it biases the gradient (Appendix B). Our claim is thus empirical rather than statistical: the clipped surrogate is already biased, and we ask whether the additional bias matters for stability when N_{updates} is small, as in common LLM RL settings.

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.

We experiment with RL training of Qwen2.5-3B-Instruct on the Countdown task.[3] 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., 2025a]: removing the correction term alone is not harmful to training stability (see Appendix E.1 and Figure 9). The same holds for Dr. GRPO [Liu et al., 2025c], which does not divide advantages by the group standard deviation; there, removing clipping also did not destabilize training within 300 steps, which we attribute to its smaller per-sample updates (Appendix F.1). In contrast, a GPG-style objective [Chu et al., 2026] with neither importance ratio nor clipping collapses in our setup unless a sufficiently strong KL penalty is added (Appendix F.2). Training without any trust region can thus be stable in some configurations, but not uniformly so.

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, as we show next, ratio-based clipping fails to provide any constraint in the fully on-policy setting, motivating a trust region mechanism independent of the importance ratio.

[3] 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.

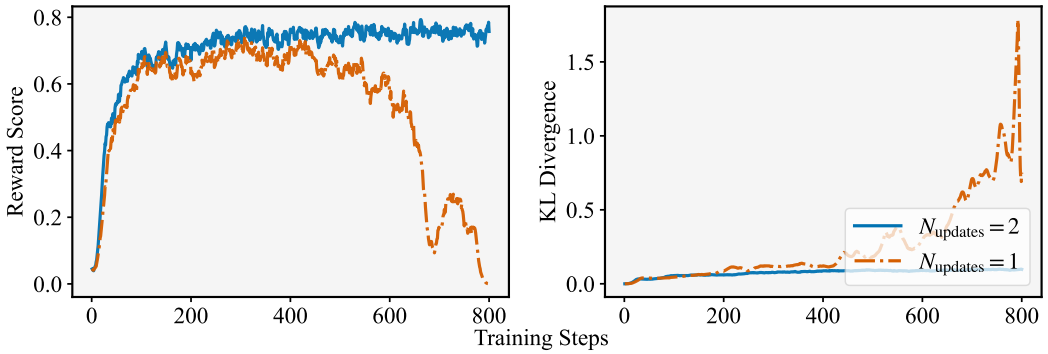


Figure 2: Performance comparison of GRPO with $N_{\text{updates}} = 1$ (fully on-policy) and $N_{\text{updates}} = 2$ on the Countdown task. N_{updates} denotes the number of gradient updates performed per batch of sampled data. We show the reward score (top) and the KL divergence from the reference policy (bottom). Without a functioning trust region constraint ($N_{\text{updates}} = 1$), training collapses and the policy diverges significantly from the reference model. Fully on-policy runs are trained for 800 steps because collapse in this regime emerges later in training.

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

Since the importance ratio is thus only required for the trust region constraint (Section 3.1), relying on it has two drawbacks: (1) the overhead of recomputing old-policy likelihoods with the training engine (Section 5.1), and (2) no constraint at all in the fully on-policy setting. This motivates a trust region mechanism that is both efficient and robust across training regimes.

4 Trust Region via Likelihood-Based Gating

We propose a simple alternative soft trust region constraint: likelihood-based gating, where the likelihood is the probability of the *current* policy generating the sampled data, so no old policy probabilities need to be computed or stored. To motivate this, we first 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 a single-outcome perturbation)**.** *Let $p \in \Delta^{K-1}$ be a categorical distribution over a finite outcome space $\{1, \dots, K\}$, with $p_i > 0$. Consider another distribution $q = p + \Delta p$, where $\sum_{i=1}^{K} \Delta p_i = 0$ and $q_i > 0$.*

Fix an outcome 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. Since the outcome space is arbitrary, the theorem applies both at the *token level*, with $p = \pi(\cdot \mid x, y_{<t})$ the next-token distribution and $q = \pi'(\cdot \mid x, y_{<t})$ that of an

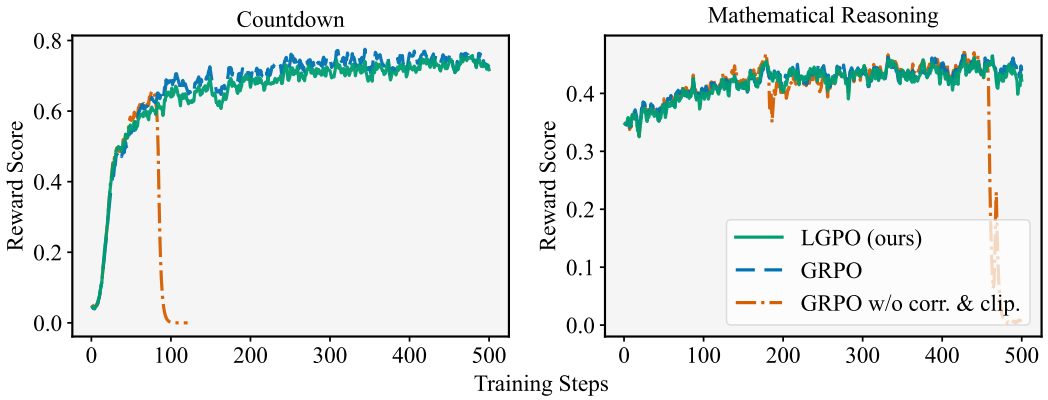


Figure 3: Training curves comparing LGPO, GRPO, and GRPO without importance sampling correction and clipping on Countdown (top) and mathematical reasoning (bottom), trained for 500 steps. LGPO maintains stability comparable to GRPO, indicating that likelihood-based gating serves as an effective alternative trust region constraint.

updated policy π', and at the *sequence level*, with $p = \pi(\cdot \mid x)$ the distribution over responses, which is finite because responses have bounded length over a finite vocabulary. 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. Three considerations determine the gating statistic: (i) Theorem 4.1 identifies extreme likelihood as a sensitivity signal; (ii) with verifiable rewards, the advantage is assigned per sequence, so we gate at the sequence level to align the trust region with the unit of learning [Zheng et al., 2025a]; and (iii) averaging the token log-likelihoods (the log of their geometric mean) normalizes for length, so a single threshold is comparable across responses. We thus define the *sequence-level average log-likelihood* under the current policy π_θ as

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

A sequence with high $\bar{\ell}_\theta(x, y)$ is dominated by tokens from near-saturated next-token distributions ($p_k \to 1$), the regime that the token-level instantiation of Theorem 4.1 identifies as most sensitive. We only gate high-likelihood sequences: for a sequence-level average, a low $\bar{\ell}_\theta(x, y)$ typically reflects many moderate-probability tokens, where the sensitivity $\delta^2/(2p_k(1 - p_k))$ is small, and such sequences are rarely observed (Appendix E.2); additionally gating them yields no benefit (Appendix F.5).

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.

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***

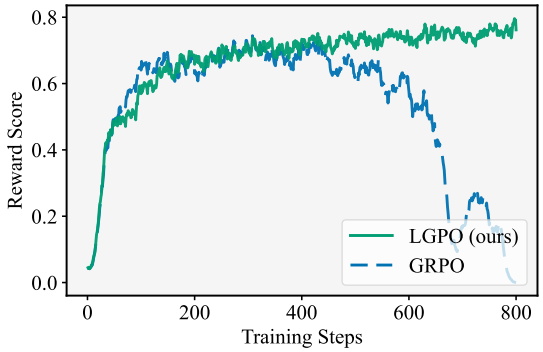


Figure 4: Fully on-policy ($N_{\text{updates}} = 1$) training dynamics on Countdown using Qwen2.5-3B-Instruct (800 steps). GRPO collapses due to the lack of an effective trust region constraint when the importance ratio is fixed at 1, whereas LGPO remains stable via likelihood-based gating.

5 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., 2024], 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].[4] 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. In practice, we recommend setting τ near the top 10–20% quantile of the sequence-level average likelihood of sampled responses (cf. Figure 10); the same τ is used unchanged across both models and both tasks (sensitivity analysis in Appendix F.6). We compare the LGPO objective against GRPO and GRPO *w/o correction & clipping* (introduced in Section 3.1). All methods are implemented in verl [Sheng et al., 2025] with identical hyperparameters (Appendix D.1); only the training horizon differs by purpose: 300 steps for the analyses in Section 3.1, 500 steps for the main comparisons, and 800 steps for fully on-policy runs, where collapse emerges later.

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 ∼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 ∼18% speedup. This is measured against the standard verl recipe, which recomputes old-policy log-probabilities with the training engine because those returned by the rollout engine (e.g., vLLM) differ numerically [Yao et al., 2025, vLLM and TorchTitan Teams, 2025]; removing this forward pass saves roughly 15% of per-step compute in our setting. The gap

[4] `https://huggingface.co/datasets/ReasoningTransferability/math_rl_48k`

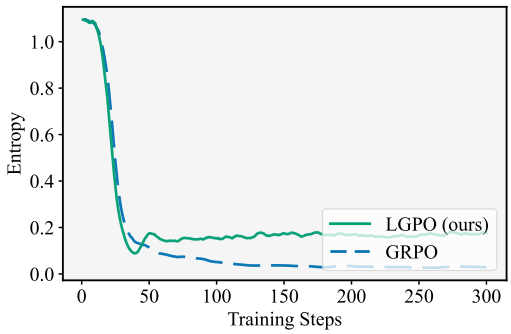


Figure 5: Generation entropy during training on Countdown. LGPO maintains higher entropy compared to GRPO, preventing premature convergence to a deterministic policy.

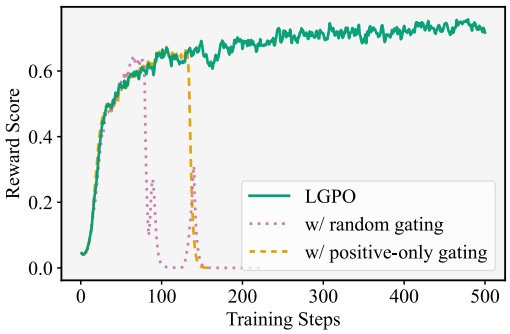


Figure 6: Countdown training curves comparing LGPO with random gating regardless of likelihood and gating only high-likelihood samples with positive advantage.

would be smaller against a baseline that reuses rollout log-probabilities and handles the mismatch (e.g., with truncated importance sampling [Yao et al., 2025]), whereas LGPO needs no old-policy log-probabilities from either engine.

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 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, LGPO remains stable (Figure 4): its gate depends solely on the current policy's likelihood, so it still filters out high-likelihood samples that induce large policy updates. Adding a KL penalty toward the reference policy (coefficient 0.001) does not prevent the collapse of fully on-policy GRPO, whose reward still crashes to zero around step 900, whereas LGPO trains stably without any reference log-probabilities (Appendix F.3); a KL penalty constrains the average deviation, but not the per-sample high-sensitivity updates of Theorem 4.1. The same pattern holds on mathematical reasoning with Qwen2.5-Math-1.5B (Appendix F.7).

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 11 in Appendix E.3). By gating out updates from already high-likelihood samples, LGPO prevents the policy from becoming overly deterministic, thereby sustaining exploration. This observation aligns with prior work showing that policy gradient updates on high-probability actions with positive advantages tend to reduce entropy [Cui et al., 2025]. The effect is reminiscent of maximum-entropy RL [Haarnoja et al., 2017, 2018], which explicitly rewards policy entropy; LGPO obtains a related effect implicitly, without an entropy bonus, by withholding updates that would further sharpen already near-deterministic responses.

Do high-likelihood samples really collapse training? Randomly discarding samples at LGPO's gating rate, regardless of their likelihood, leads to training collapse within 100 steps (Figure 6). This confirms that LGPO's stability arises from selectively suppressing high-likelihood updates, rather than simply reducing the number of active samples.

Do negative advantage high-likelihood samples also collapse training? Negative-advantage updates might seem safer than positive ones, as they push probabilities away from the boundary rather than increasing determinism. However, as shown in Figure 6, gating only positive-advantage high-likelihood samples while keeping negative-advantage ones still leads to collapse. Suppressing negative-advantage updates on high-likelihood samples is thus also crucial for stability. We hypothesize two complementary mechanisms. First, Theorem 4.1 is symmetric in the direction of the change,

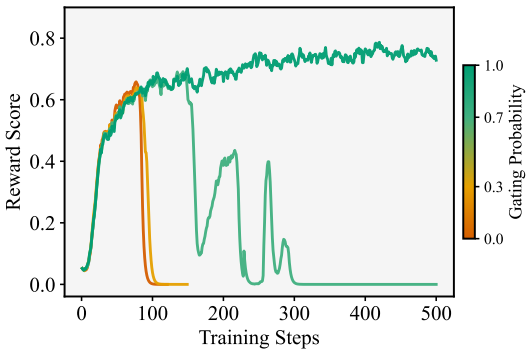


Figure 7: Effect of stochastic gating probability on training stability. Reducing the gating probability leads to earlier training collapse, indicating that strict gating is necessary for stability.

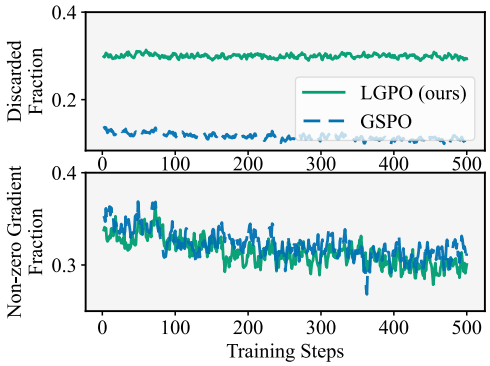


Figure 8: Although LGPO discards a larger fraction of total samples (top), the fraction of samples contributing non-zero gradient ($A \neq 0$ and retained) remains comparable to GSPO (bottom).

and although raw gradients at saturated tokens are small, Adam makes update magnitudes largely insensitive to gradient scale, so such probabilities can still move, at the highest KL cost per unit of movement. Second, a sequence-level negative advantage penalizes every token of a failed response, including locally correct tokens the policy is already confident in, which can erode established competencies, consistent with the false-penalty view of Wu et al. [2026].

Can we stochastically gate high-likelihood samples and find a sweet spot between stability and performance? We apply LGPO's gating mask only with a certain probability, to test whether keeping some signal from high-likelihood samples could balance stability and performance. As shown in Figure 7, 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 Figure 8 (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 8, bottom). With a token-weighted count, LGPO gates 24.7% of training tokens on Countdown (1.6% for GSPO), of which 48.9% have zero, 45.1% negative, and only 6.1% positive advantage (Appendix F.8). LGPO thus discards little signal from successful trajectories ($\sim$1.5% of all tokens), but suppresses a substantial share of negative-advantage updates, precisely the category whose retention causes collapse (Figure 6).

6 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., 2025a, Zhang et al., 2025a, Chen et al., 2025]. Others modify constraint mechanisms, such as using selective clipping for negative samples [Roux et al., 2025], adaptive gating [Gao et al., 2025], or replacing ratio-based clipping with a direct estimate of policy divergence [Qi et al., 2026], while GVPO alternatively reformulates the objective as a variance-matching regression [Zhang et al., 2025b]. 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 and Ding, 2025, Liu et al., 2025c], 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]. Wu et al. [2026] show that negative samples can be unfairly penalized in critic-free RLVR, which relates to our finding that negative-advantage high-likelihood samples destabilize training.

Closest to our analysis, concurrent work [Yao et al., 2026] shows that group-relative REINFORCE admits an off-policy interpretation and likewise argues that stability stems from regularizing the update rather than from the importance sampling correction. We complement it with controlled decompositions across four GRPO variants, the structural failure of ratio-based clipping in the fully on-policy regime, and LGPO, a ratio-free trust region that resolves this failure without old-policy likelihoods. GPG [Chu et al., 2026] trains without importance ratios or clipping, which in our setup is stable only with a strong KL penalty. For strongly off-policy or asynchronous training, TBA [Bartoldson et al., 2025] and OAPL [Ritter et al., 2026] also dispense with importance ratios, relying instead on KL regularization toward a behavior or reference policy whose log-probabilities must be computed; LGPO targets the fully and near on-policy regime and uses only the current policy's likelihood. Train–inference mismatch makes nominally on-policy training slightly off-policy [Yao et al., 2025]; it can be mitigated with truncated importance sampling [Yao et al., 2025] or bitwise-consistent kernels [vLLM and TorchTitan Teams, 2025], whereas LGPO's gate does not use old-policy log-probabilities.

7 Conclusion

We propose Likelihood-Gated Policy Optimization (LGPO), 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 computing old-policy likelihoods, and it remains robust in the fully on-policy setting where ratio-based clipping fails, offering a more efficient and stable framework for LLM RL training.

8 Limitations and Future Work

While LGPO demonstrates improved efficiency and stability, our study has limitations. First, due to computational constraints, our validation covers only 1.5B–3B models on verifiable reasoning tasks; the fully on-policy failure of clipping is structural and Theorem 4.1 is model-size agnostic, but transfer to much larger models remains to be verified. Second, LGPO is stable in all settings we tested (up to $N_{\text{updates}} = 16$; Appendix F.4), but as the role of the correction grows with off-policyness, we expect it to degrade at sufficiently large N_{updates}, where KL-regularized approaches such as TBA [Bartoldson et al., 2025] are more appropriate. Third, our speedup is measured against the standard recipe that recomputes old-policy log-probabilities, and our KL-regularized baseline uses a single coefficient (0.001). Future work includes finer-grained gating within a sequence, combining gating with ratio-based clipping when rollouts are reused (preliminary results in Appendix F.4 suggest they are complementary), and decoding-stage interventions that avoid generating samples that would be gated anyway.

Acknowledgments and Disclosure of Funding

This work was supported by KRAFTON and the University of Wisconsin–Madison, which provided funding and computational resources.

References

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

Brian Bartoldson, Siddarth Venkatraman, James Diffenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kailkhura. Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable LLM post-training. In *Advances in Neural Information Processing Systems*, 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.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. GPG: A simple and strong reinforcement learning baseline for model reasoning. In *International Conference on Learning Representations*, 2026.

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.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International Conference on Machine Learning*, pages 1352–1361. PMLR, 2017.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International Conference on Machine Learning*, pages 1861–1870. PMLR, 2018.

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)*, pages 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 *International Conference on Learning Representations*, volume 2024, pages 39578–39601, 2024.

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.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding R1-Zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025c.

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

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.

Daniel Ritter, Owen Oertell, Bradley Guo, Jonathan Chang, Kianté Brantley, and Wen Sun. LLMs can learn to reason via off-policy RL. *arXiv preprint arXiv:2602.19362*, 2026.

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

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

vLLM and TorchTitan Teams. No more train-inference mismatch: Bitwise consistent on-policy reinforcement learning with vLLM and TorchTitan. vLLM Blog, November 2025. URL `https://vllm.ai/blog/bitwise-consistent-train-inference`.

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.

Yihong Wu, Liheng Ma, Lingfeng Xiao, Muzhi Li, Xinyu Wang, Yingxue Zhang, and Jian-Yun Nie. Rethinking groups in critic-free RLVR. *arXiv preprint arXiv:2606.17250*, 2026.

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

Jian Xiong, Jingbo Zhou, Jingyong Ye, Qiang Huang, and Dejing Dou. Aapo: Enhancing the reasoning capabilities of llms with advantage margin. *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.

Chaorui Yao, Yanxi Chen, Yuchang Sun, Yushuo Chen, Wenhao Zhang, Xuchen Pan, Yaliang Li, and Bolin Ding. Group-relative REINFORCE is secretly an off-policy algorithm: Demystifying some myths about GRPO and its friends. In *International Conference on Learning Representations*, 2026.

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

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

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, 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, Yang Yue, 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, 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*, 2025a.

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*, 2025b.

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 and Bias 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). We also analyze the bias introduced by removing the importance sampling correction.

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

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

Equation (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,

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

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

Bias from removing importance weighting. The IS-weighted estimator is unbiased for the on-policy gradient at θ (assuming integrability), since $\mathbb{E}_{\pi_{\theta_{\text{old}}}}[r_t(\theta)\, A_t\, s_t(\theta)] = \mathbb{E}_{\pi_\theta}[A_t\, s_t(\theta)]$ at the token level. In contrast, the unweighted estimator that sets the correction weight to 1 targets the behavior-policy expectation and is generally biased. At the token level, the bias relative to the IS-corrected target is

$$\begin{aligned}\operatorname{Bias}_t(\theta) &:= \mathbb{E}_{\pi_{\theta_{\text{old}}}}[A_t\, s_t(\theta)] - \mathbb{E}_{\pi_{\theta_{\text{old}}}}[r_t(\theta)\, A_t\, s_t(\theta)] \\ &= \mathbb{E}_{y_t\sim\pi_{\theta_{\text{old}}}(\cdot|x,y_{<t})}\big[\big(1 - r_t(\theta)\big)\, A_t\, s_t(\theta)\big].\end{aligned} \tag{4}$$

Equation (4) shows that dropping importance weighting trades unbiasedness for potentially reduced variance: the bias vanishes in the fully on-policy setting ($r_t(\theta) \equiv 1$) and grows with the mismatch between π_θ and $\pi_{\theta_{\text{old}}}$, motivating a proximal/trust-region mechanism to keep rollouts from becoming stale across updates. Note that the clipped PPO/GRPO surrogate is itself already a biased estimator, since clipping discards gradient contributions; our experiments indicate that the additional bias from dropping the correction term is benign for stability and final performance when N_{updates} is small (Section 3.1).

C Proof of Theorem 4.1

Let $p = (p_1, \dots, p_K)$ denote a categorical distribution over a finite outcome space $\{1, \dots, K\}$, with $p_i > 0$ and $\sum_i p_i = 1$. Let $q = p + \Delta p$ be another distribution, where $\sum_i \Delta p_i = 0$ and $q_i > 0$. The argument below uses only these properties, so it applies to any finite outcome space. In particular, it covers the next-token distribution $p_i = \pi(i \mid x, y_{<t})$ over a vocabulary (token level) as well as the response distribution $\pi(\cdot \mid x)$ over sequences of bounded length from a finite vocabulary (sequence level), with q induced by an updated policy π' in either case. 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 and Implementation 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

D.1 Hyperparameters

Table 2 lists the training hyperparameters. All methods share identical hyperparameters within each setting; only the policy objective differs. Training horizons differ by experiment purpose: 300 steps for the analyses in Section 3.1, 500 steps for the main comparisons, and 800 steps for fully on-policy runs; the KL-regularized baseline (Appendix F.3) is run beyond this horizon.

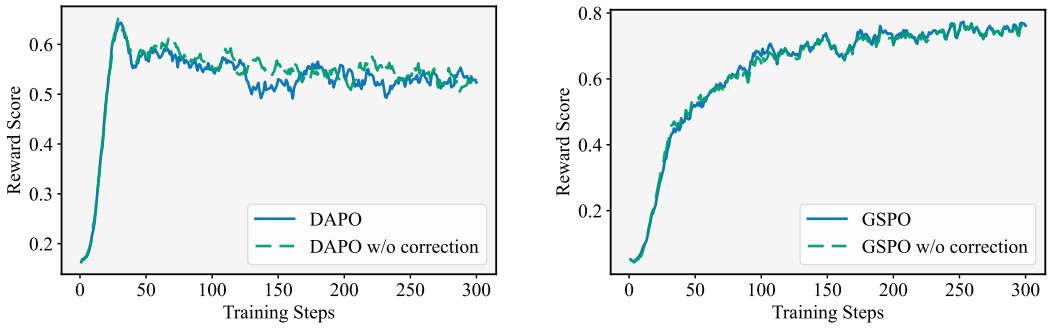
E Additional Experiments Results

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 9).

Table 2: Training hyperparameters.

Hyperparameter	Countdown	Mathematical reasoning
Models	Qwen2.5-3B-Instruct	Qwen2.5-3B-Instruct Qwen2.5-Math-1.5B
Framework	verl [Sheng et al., 2025]	
Optimizer	AdamW	
Learning rate	1×10^{-6}	
LR warmup steps	10	
Weight decay	0.1	
Prompt batch size	128	256
Minibatch size	64	64
N_{updates}	2	4
Minibatch size, fully on-policy ($N_{\text{updates}} = 1$)	128	–
Group size G	8	8
Advantage normalization	group mean and std. (GRPO)	
KL penalty	none (except KL baseline)	
LGPO threshold τ	-0.15	
Evaluation	–	average Pass@1 over 8 samples



(a) DAPO (b) GSPO

Figure 9: 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.

E.2 Likelihood Distribution of Sampled Responses

Figure 10 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.

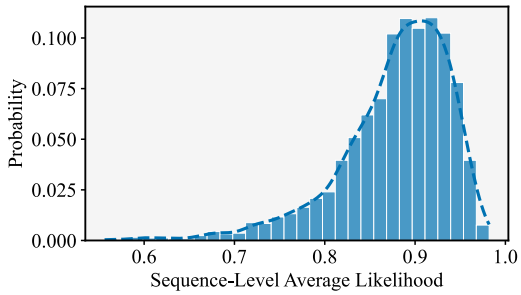


Figure 10: Distribution of sequence-level average likelihood $\exp\left(\bar{\ell}_\theta(x, y)\right)$ of responses sampled from Qwen2.5-3B-Instruct on MATH500.

E.3 Generation Entropy on Mathematical Reasoning Task

Figure 11 presents the generation entropy during training on the mathematical reasoning dataset. Similar to the results on the Countdown task, LGPO maintains higher entropy throughout training compared to GRPO, which exhibits a more rapid decline in entropy.

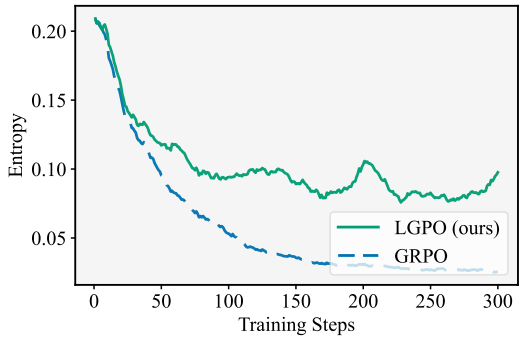


Figure 11: Generation entropy during training on the mathematical reasoning dataset. LGPO prevents rapid entropy collapse, similar to the trend observed in the Countdown task.

F Additional Analyses and Baselines

This section reports additional analyses and baselines. Unless otherwise noted, experiments use Qwen2.5-3B-Instruct on Countdown with the setup and hyperparameters of Section 5 and Appendix D.1.

F.1 Decomposing the Importance Ratio in Dr. GRPO

Dr. GRPO [Liu et al., 2025c] computes advantages without dividing by the group standard deviation, as is common in current GRPO implementations. We repeat the decomposition of Section 3.1 with Dr. GRPO-style advantages on Countdown, matching the setup of Figure 1 (300 steps). As shown in Table 3, removing the correction term alone does not harm stability, so together with GRPO, DAPO, and GSPO (Figure 9), the dispensability of the correction holds across four GRPO variants. Unlike for GRPO, removing clipping did not destabilize Dr. GRPO within this horizon. We attribute this to update magnitude: dividing by the group standard deviation inflates advantages on low-variance groups, so standard GRPO takes larger per-sample updates and depends more heavily on a trust region; the need for a trust region thus scales with update magnitude. Note that Dr. GRPO without clipping still computes importance ratios and the old-policy forward pass, so it provides none of LGPO's efficiency benefit. Whether it remains stable over longer horizons, on other tasks, or in the fully on-policy regime is left open.

Table 3: Final reward on Countdown after 300 steps when decomposing the importance ratio in Dr. GRPO.

Variant	Final reward
Dr. GRPO	0.74
Dr. GRPO *w/o correction*	0.71
Dr. GRPO *w/o clipping*	0.73

F.2 GPG-Style Objective without Importance Ratio and Clipping

GPG [Chu et al., 2026] reports that RL with verifiable rewards can be trained successfully without either an importance ratio or clipping. We reproduce a GPG-style objective under our controlled Countdown setup, with data, model, and hyperparameters identical to Figure 1. Without an importance ratio or clipping, it collapses within approximately 120–180 steps in both of our runs, consistent with

our *GRPO w/o corr. & clip.* results (Figure 3). It can be stabilized by adding a sufficiently strong KL loss, i.e., by reintroducing a trust-region-like mechanism. GPG's reported successes therefore show that training without clipping or an importance ratio can be stable under some configurations, while our results show that it is not uniformly stable across settings.

F.3 KL-Regularized GRPO in the Fully On-Policy Setting

A natural alternative for stabilizing fully on-policy training is a KL penalty toward the reference policy. We train fully on-policy GRPO ($N_{\text{updates}} = 1$) on Countdown with a KL penalty of coefficient 0.001, extending this run beyond the 800-step horizon of the other fully on-policy runs. The KL penalty does not prevent collapse: the reward crashes to zero around step 900, whereas LGPO trains smoothly throughout with a higher final reward, without computing any reference log-probabilities. We believe this is because a KL penalty constrains the average deviation from the reference policy, but does not suppress the per-sample high-sensitivity updates identified by Theorem 4.1, which LGPO's gating removes. We tested only this coefficient; a stronger KL penalty may delay or prevent collapse, as it does for the GPG-style objective (Appendix F.2), at the cost of restricting deviation from the initialization and of computing reference log-probabilities.

F.4 More Off-Policy Training

To assess robustness to more off-policy data, we train GRPO and LGPO on Countdown with $N_{\text{updates}} = 8$ and $N_{\text{updates}} = 16$ (Table 4). All runs train stably over the reported horizons, and LGPO reaches a higher peak and final reward than GRPO in both settings. At $N_{\text{updates}} = 16$, we additionally run a preliminary combination of the two mechanisms, applying the sequence-level likelihood gate on top of the clipped importance-ratio objective (*LGPO + clipping*). Without any tuning, the combination improves over clipping alone in both peak and final reward and moderates the entropy inflation that clipping alone shows at this reuse depth, indicating that the gate removes harmful updates that the ratio-based trust region does not filter. Gating alone is sufficient for both stability and the best performance at $N_{\text{updates}} = 16$, although this offers no guarantee in more strongly off-policy regimes: since the role of the importance sampling correction grows with off-policyness, we expect LGPO to degrade at sufficiently large N_{updates}, where KL-regularized approaches designed for strongly off-policy training [Bartoldson et al., 2025, Ritter et al., 2026] are more appropriate.

Table 4: Countdown training with more off-policy data. Final reward is reported at the last step (400 steps for $N_{\text{updates}} = 8$, 800 steps for $N_{\text{updates}} = 16$). All runs are stable.

Method	N_{updates}	Peak reward	Final reward	Final entropy
GRPO	8	0.74	0.72	–
LGPO	8	0.77	0.76	–
GRPO (clipping only)	16	0.68	0.66	2.21
LGPO (gating only)	16	0.78	0.76	0.80
LGPO + clipping	16	0.70	0.70	1.64

F.5 Two-Sided Likelihood Gating

Theorem 4.1 is two-sided, whereas LGPO gates only high-likelihood sequences. To verify directly that gating the low-likelihood side is unnecessary, rather than inferring it from the rarity of low-likelihood sequences alone (Figure 10), we additionally gate sequences whose average likelihood is below 0.6 (i.e., $\bar{\ell}_\theta(x, y) < \log 0.6$), the lower edge of the distribution in Figure 10, on top of the high-likelihood gate $\bar{\ell}_\theta(x, y) > \tau$. Two-sided gating trains stably with reward comparable to one-sided LGPO (0.71 vs. 0.72 at 500 steps), so gating the low-likelihood side provides no additional stability or performance benefit. The two sides are not symmetric for a sequence-level statistic: a high average log-likelihood forces most tokens of the sequence toward saturation ($p_k \to 1$), the sensitive regime of Theorem 4.1, whereas a low average log-likelihood instead reflects many moderate-probability tokens, for which $p_k(1 - p_k)$ is large and the sensitivity $\delta^2/(2p_k(1 - p_k))$ is small; tokens with probability truly near zero are rarely sampled in on-policy data in the first place.

F.6 Sensitivity to the Likelihood Threshold

We sweep the likelihood threshold τ on Countdown. Our default $\tau = -0.15$ is among the settings that give the most stable training. Looser thresholds (closer to 0), which gate only sequences with very high average likelihood, admit destabilizing high-likelihood samples and collapse early, whereas tighter thresholds (more negative) discard too much learning signal and plateau at a lower reward before destabilizing. The same $\tau = -0.15$ is used unchanged across both models (Qwen2.5-3B-Instruct and Qwen2.5-Math-1.5B) and both tasks. In practice, we recommend setting τ near the top 10–20% quantile of the sequence-level average likelihood of sampled responses (cf. Figure 10).

F.7 Fully On-Policy Training on Mathematical Reasoning

To check that the fully on-policy findings are not specific to one model and task, we also train Qwen2.5-Math-1.5B on the mathematical reasoning dataset with $N_{\text{updates}} = 1$. As on Countdown, fully on-policy GRPO degrades sharply late in training, while LGPO remains stable.

F.8 Composition of Gated Tokens

Table 5 breaks down, with a token-weighted count over the Countdown training run, which tokens LGPO gates. LGPO gates 24.7% of all training tokens, compared with 1.6% for GSPO's clipping. Zero-advantage tokens come from groups in which all sampled responses receive the same binary reward and therefore contribute no policy-gradient signal regardless of gating. Positive-advantage tokens gated by LGPO amount to only 1.5% of all training tokens, so LGPO discards little signal from successful trajectories; the largest non-zero category it removes is negative-advantage, high-likelihood samples, whose retention leads to collapse (Figure 6).

Table 5: Token-weighted composition of tokens gated by LGPO on Countdown, by advantage sign.

Advantage of gated tokens	Share of gated tokens	Share of all training tokens
Zero ($A = 0$)	48.9%	12.1%
Negative ($A < 0$)	45.1%	11.1%
Positive ($A > 0$)	6.1%	1.5%
Total gated by LGPO	100%	24.7%
Total clipped by GSPO (reference)	–	1.6%

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The three claims (the importance sampling correction is dispensable for stability while clipping is essential; ratio-based clipping fails in the fully on-policy setting; LGPO matches GRPO with $\sim$18% speedup relative to the standard recipe that recomputes old-policy likelihoods and remains stable where GRPO collapses) are supported by Sections 3.1, 3.2, and 5, and by Theorem 4.1.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 8 ("Limitations and Future Work") notes that our validation covers only 1.5B–3B models on verifiable reasoning tasks, that we have not characterized the exact N_{updates} boundary at which removing the importance sampling correction destabilizes training, that the speedup is measured against the standard recipe that recomputes old-policy log-probabilities, and that the KL-regularized baseline uses a single coefficient.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best

judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Theorem 4.1 states all assumptions ($p_i > 0$, $q_i > 0$, $\sum_i \Delta p_i = 0$, $-p_k < \delta < 1 - p_k$). The full proof is provided in Appendix C, factored into Lemmas C.1 and C.2.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 5 specifies the base models, evaluation benchmarks, training dataset (with URL), prompt and minibatch sizes, N_{updates}, group size $G = 8$, advantage normalization, and the LGPO likelihood threshold $\tau = -0.15$. The full training procedure is given in Algorithm 1 (Appendix D).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.

(c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [No]

   Justification: Code is not released alongside the paper; we plan to release the implementation publicly. All datasets used (math_rl_48k, MATH500, OlympiadBench, AIME24/25) and the base Qwen2.5 model checkpoints are publicly accessible.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 5 specifies the training data, evaluation protocol (average Pass@1 over 8 samples), prompt and minibatch sizes, N_{updates}, group size $G = 8$, advantage normalization, the LGPO likelihood threshold $\tau = -0.15$ (with a sensitivity analysis in Appendix F.6), and the training horizon of each experiment. Appendix D.1 lists the training hyperparameters, and Algorithm 1 provides the full training procedure.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: Each training curve is a single run due to the substantial compute cost of LLM RL post-training (multiple H100-GPU-hours per run). The qualitative effects we report (training collapse vs. stability across regimes, and the $\sim$18% wall-clock speedup) are large relative to typical run-to-run variance for these settings.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 5 reports the compute for the headline efficiency claim: 2$\times$ NVIDIA H100 GPUs for one epoch on the $\sim$7.5K-example math reasoning training set, taking 3.37 hours for GRPO and 2.85 hours for LGPO. All other experiments (Countdown, ablations, fully on-policy runs) used comparable 2$\times$ H100 hardware.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics.

   Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This is foundational research on RL post-training efficiency, not tied to any specific deployed application.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper poses no such risks; no new high-risk models or datasets are released.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Section 5 cites the original publications for all assets. Licenses: Qwen2.5-Math-1.5B and OlympiadBench are released under Apache-2.0; Qwen2.5-3B-Instruct is released under the Qwen Research License, which permits academic research use; MATH500 is derived from PRM800K, released under MIT; AIME24 and AIME25 problems are publicly distributed by the Mathematical Association of America for educational use; the math_rl_48k training dataset is publicly hosted by Huan et al. [2025] on HuggingFace. All assets are used within their stated terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification: The paper involves no crowdsourcing and no research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.

- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: The paper involves no research with human subjects.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.