EEPO: EXPLORATION-ENHANCED POLICY OPTIMIZATION VIA SAMPLE-THEN-FORGET

Anonymous authors
Paper under double-blind review

ABSTRACT

Balancing exploration and exploitation remains a central challenge in reinforcement learning with verifiable rewards (RLVR) for large language models (LLMs). Current RLVR methods often overemphasize exploitation, leading to entropy collapse, reduced exploratory capacity, and ultimately limited performance gains. Although techniques that add randomness increase policy stochasticity, they frequently fail to escape dominant behavioral modes. The resulting sample-and-reward dynamics amplify these modes, eroding exploration and leading to entropy collapse. We introduce ***E**xploration-**E**nhanced **P**olicy **O**ptimization* (EEPO), a novel framework that promotes exploration through two-stage rollouts with adaptive unlearning. In the first stage, the model generates half of the trajectories; it then undergoes a lightweight, temporary unlearning step to suppress these sampled responses, forcing the second stage to explore different regions of the output space. This *sample-then-forget* mechanism actively steers the policy away from dominant modes and encourages mode-seeking exploration. Across five reasoning benchmarks, EEPO consistently outperforms baselines, achieving average gains of 24.3% on Qwen2.5-3B, 33.0% on Llama3.2-3B-Instruct, and 10.4% on Qwen3-8B-Base.

1 INTRODUCTION

The emergence of OpenAI's o1 (OpenAI) and DeepSeek-R1 (DeepSeek-AI et al., 2025) marks a significant advance in LLM reasoning capabilities, particularly for challenging tasks such as mathematics (Cobbe et al., 2021; Hendrycks et al., 2021b) and programming (Chen et al., 2021; Codeforces, 2025). A key driver of this progress is reinforcement learning with verifiable rewards (RLVR). Despite its success, RLVR remains challenged by the classic exploration–exploitation dilemma (Sutton & Barto, 2018). Specifically, policies tend to over-emphasize exploitation of high-reward trajectories, leading to entropy collapse and reduced exploratory capacity (Yu et al., 2025; Cui et al., 2025). This not only causes premature performance saturation but also prevents the discovery of diverse reasoning strategies essential for robust generalization.

A growing body of work has attempted to mitigate this issue, but most approaches increase exploration in an indiscriminate manner. Common strategies such as increasing the softmax temperature or adding entropy regularization (Hou et al., 2025) operate by flattening the distribution indiscriminately. While this raises stochasticity, it still fails to shift probability mass away from dominant trajectories, often causing instability or degraded performance when applied strongly. More recent efforts take a closer view at entropy collapse: DAPO (Yu et al., 2025) alleviates it by adjusting clipping ranges to give low-probability actions more headroom, and (Cui et al., 2025) analyze how high-probability updates drive entropy decay. Although these refinements provide meaningful gains, they largely remain indiscriminate—boosting randomness rather than suppressing dominant behaviors—and thus struggle to avoid premature convergence toward a narrow set of trajectories.

To address this gap, we propose Exploration-Enhanced Policy Optimization (EEPO), a new RLVR framework that promotes exploration by equipping the rollout process with a *sample-then-forget* mechanism. EEPO employs two-stage rollouts: the model first generates trajectories, then performs a lightweight, temporary unlearning step that suppresses the modes just explored. This encourages subsequent rollouts to deviate from dominant behaviors and uncover alternatives trajectories, effectively steering the policy toward other promising regions of the output space rather than getting stuck in a single dominant mode. Notably, this mechanism is applied only during the rollout phase,

leaving the main policy update unchanged. This decoupling allows the rollout model to broaden the trajectory space without modifying the actor's policy optimization, while the enriched trajectories, in turn, provide better supervision for exploitation during policy learning.

Concretely, EEPO modifies the GRPO rollout by decomposing one-shot group sampling into three steps. First, Stage 1 samples half of the trajectories; second, an unlearning operation is applied to the rollout model to suppress the just-sampled modes; third, Stage 2 samples the remaining half from the updated model. Sampling in Stages 1 and 2 mirrors GRPO; the key change is the intervening unlearning step. For the exploration setting, we make three design choices: (1) to impose stronger penalties on dominant regions, we replace the standard negative log-likelihood with a complementary loss that penalizes high-probability tokens more than low-probability ones; (2) to trigger intervention at the onset of mode collapse, we introduce an entropy-conditioned gating mechanism that activates unlearning only when exploration deteriorates (i.e., low entropy); and (3) to keep the intervention lightweight and temporary, we apply a single-step gradient update to the GRPO rollout model—synchronized from the actor in each iteration and used solely for sampling—thereby decoupling unlearning from policy optimization and confining its effect to the rollout phase.

To validate our approach, we evaluate EEPO on five challenging mathematical reasoning benchmarks using three distinct LLMs. The benchmarks include Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and three competition-level datasets: AMC 2023, AIME 2024, and AIME 2025. EEPO consistently outperforms the baselines, yielding average relative improvements over GRPO of 24.3% on Qwen2.5-3B, 33.0% on Llama3.2-3B-Instruct, and 10.4% on Qwen3-8B-Base. Furthermore, our analyses show that EEPO achieves superior performance through more effective exploration while maintaining comparable training time to standard GRPO.

2 PRELIMINARIES

We begin by reviewing RLVR and its prevalent implementation, the GRPO algorithm, which has been widely adopted for training large-scale reasoning models. We then analyze its limitations related to insufficient exploration and revisit existing solutions attempted to mitigate this issues.

2.1 RL FOR TRAINING LARGE-SCALE REASONING MODELS

Reinforcement Learning with Verifiable Rewards (RLVR) . The success of RLVR relies on reliable reward signals (DeepSeek-AI et al., 2025), typically provided by a rule-based reward model that delivers precise feedback for tasks in mathematical, coding, and logical reasoning domains. Consider a mathematical dataset $\mathcal{D} := \{(q, a)\}$, where q denotes a question and a denotes its corresponding ground-truth answer. The reward depends solely on the correctness of the final prediction $\hat{a}$ compared to a, without enforcing constraints on the reasoning process:

$$r(\hat{a}, a) = \mathbb{1}[\hat{a} \equiv a]. \tag{1}$$

The RLVR objective is often implemented using the large-scale policy optimization method GRPO (DeepSeek-AI et al., 2025). Compared to proximal policy optimization (PPO; Schulman et al., 2017), GRPO improves computational efficiency by eliminating the need for a separate value function.

Group Relative Policy Optimization (GRPO) . Given a question q and a set of responses, i.e., reasoning paths, $O = \{o_1, o_2, \ldots, o_G\}$ sampled from the old policy model π_{old}, GRPO directly computes advantages to optimize the policy model π using the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{\sum_{i=1}^{G} |o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \min \left[r_{i,t}(\theta) \hat{A}_i, \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right) \hat{A}_i \right] - \beta \mathbb{D}_{\text{KL}}[\pi_\theta \parallel \pi_{\text{ref}}]. \tag{2}$$

Here, π_{ref} denotes a reference model used to constrain policy updates via a KL divergence penalty. The score $\hat{A}_i$ represents the normalized advantage of response o_i, computed as $\hat{A}_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}$, where $\{r_1, \ldots, r_G\}$ denotes the rewards corresponding to the sampled responses in the group O.

The importance weight $r_{i,t}(\theta)$ denotes the probability ratio between current and old policies:

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})} \tag{3}$$

This importance sampling ratio is crucial for obtaining *unbiased* gradient estimates when responses are sampled from π_{old} rather than the current policy π_θ.

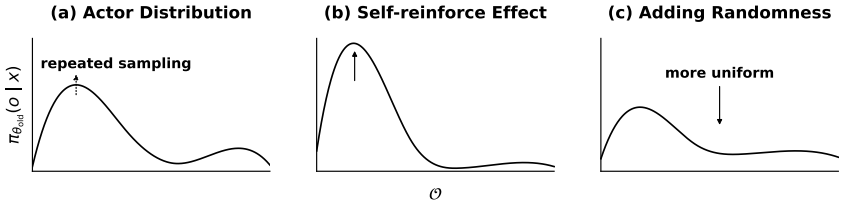

Figure 2: Illustration of exploration challenges in GRPO. (a) Policy distribution showing imbalanced modes with a dominant peak. (b) Self-reinforcement effect where the dominant mode becomes increasingly concentrated through positive feedback. (c) Effect of adding randomness (e.g., entropy regularization) which flattens the distribution but maintains the relative dominance of modes.

2.2 Revisiting the Insufficient Exploration Problem

We examine the exploration problem through entropy metrics and performance changes on test and OOD benchmarks to characterize the issue and its implications. Figure 1 presents our analysis of GRPO's behavior during training on the MATH dataset. We observe two interconnected phenomena:

(1) Rapid entropy collapse: Despite incorporating substantial entropy regularization ($\lambda = 1 \times 10^{-3}$)[1], the policy entropy decreases precipitously within the first few training steps, indicating rapid convergence to deterministic behaviors. This collapse stems from GRPO's inherently exploitative objective function (Equation 2), which prioritizes reward maximization over exploration.

(2) Deteriorating generalization: As entropy collapses, we observe a divergent trend: while MATH test accuracy continues to improve, performance on OOD benchmarks such as AMC 23 plateaus early. This suggests that reduced exploration causes the model to overfit to the training distribution rather than learn robust reasoning strategies that generalize.

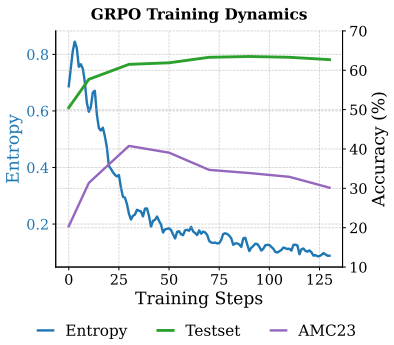

Figure 1: GRPO training dynamics: rapid entropy collapse accompanies rising Testset and decline on AMC23.

To explain entropy collapse, we hypothesize that when entropy begins to decline, the policy has developed partial but uncertain knowledge about the problem. This manifests in the response distribution of the policy as *multiple modes*—multiple plausible reasoning traces may exist for a given question. Importantly, these modes are *imbalanced*: a dominant mode receives disproportionately more probability mass than others, as illustrated in Figure 2(a). Once responses are predominantly sampled from this dominant mode and receive positive feedback, the policy reinforces it further, amplifying its probability while suppressing alternative responses. This *self-reinforcing dynamic* creates a feedback loop that inhibits exploration and ultimately leads to entropy collapse, as shown in Figure 2(b). This process is particularly problematic: once the policy finds a dominant mode that is correct, it prevents the discovery of alternative, potentially superior reasoning strategies, leading to local optima and overfitting to the training distribution. The theoretical analysis of the intuition is provided in Appendix F.1, which shows that RL updates are intrinsically self-reinforcing / mode-seeking.

Current approaches to enhance exploration primarily increase randomness in the policy optimization or sampling process, such as strengthening entropy term or raising sampling temperature. These methods essentially flatten the policy distribution to make it more uniform, as depicted in Figure 2(c). However, they fail to fundamentally break the self-reinforcing loop: the dominant mode remains most likely to be sampled even after flattening. This observation motivates our central question: *How can we enable the policy to explore plausible behaviors beyond the dominant mode?*

3 Method

We present EEPO, a novel approach that enhances the exploration of GRPO through strategic trajectory unlearning. We first provide an overview of our method, then detail its implementation.

[1]This value is significantly larger than the 1×10^{-4} suggested by SimpleRL (Zeng et al., 2025).

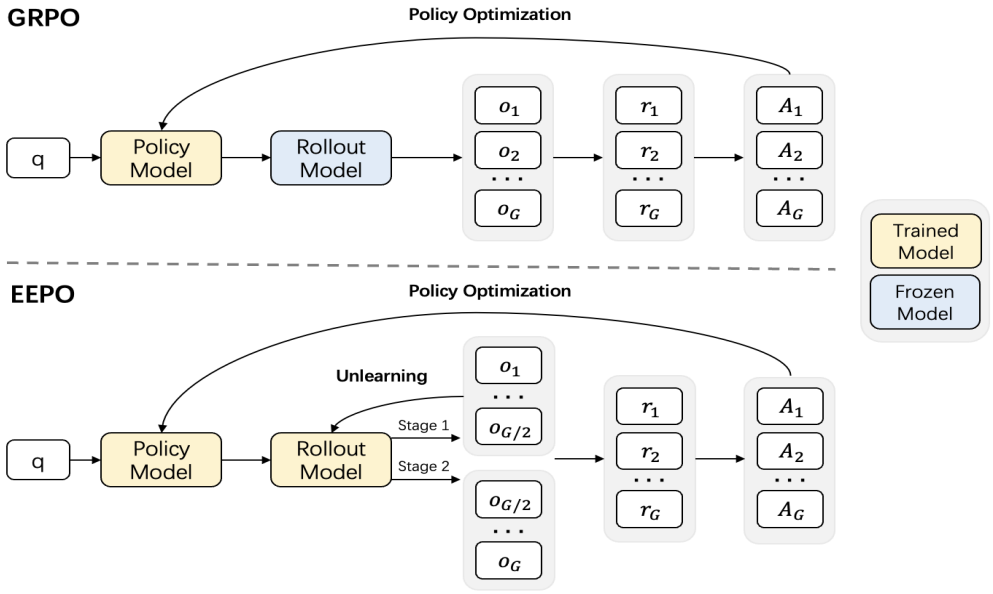

Figure 3: Comparison of GRPO and EEPO rollout processes. GRPO samples all trajectories simultaneously from a fixed rollout model, while EEPO introduces an unlearning step on the rollout model between two sampling stages to promote exploration of diverse modes.

3.1 Exploration-Enhanced Policy Optimization

To address the self-reinforcing dynamics that lead to entropy collapse, we propose a framework that promotes exploration by modifying the rollout process, as shown in Figure 3. The key idea is to prevent the rollout model from repeatedly sampling from dominant modes by *unlearning* previously sampled responses during rollout generation.

Figure 3 illustrates the key difference between GRPO and EEPO. In GRPO, the rollout model π_{rollout} (corresponding to π_{old} in Equation 2) samples all responses $O = \{o_1, o_2, \ldots, o_G\}$ simultaneously. These responses are then used to compute rewards and advantages for policy optimization. While EEPO introduces a *sample-then-forget* mechanism that modifies this process, instead of sampling all G trajectories at once, it divides the rollout into two stages separated by an unlearning step:

- *Stage 1 sampling:* Sample $G/2$ trajectories $\{o_1, o_2, \ldots, o_{G/2}\}$ from π_{rollout}.
- *Unlearning:* Update π_{rollout} to forget the sampled trajectories.
- *Stage 2 sampling:* Sample the remaining trajectories $\{o_{G/2+1}, \ldots, o_G\}$ from the updated model.

After collecting all G trajectories across both stages, we compute their rewards and apply the standard GRPO objective (Equation 2) to update the policy model. Importantly, the denominator in Equation 3 uses the rollout model's probabilities, ensuring unbiased gradient estimates. Following standard GRPO practice, the rollout model is synchronized with the actor model at the beginning of each iteration, so the unlearning effect is *temporary* and does not affect the policy model.

This approach *decouples* policy optimization from exploration. While the policy model π_θ focuses on reward maximization through standard policy optimization, the rollout model actively explores alternative trajectory spaces by suppressing previously visited regions. As shown in Figure 4, the unlearning step explicitly encourages Stage 2 to sample from previously underexplored regions, effectively breaking the self-reinforcing loop that causes entropy collapse.

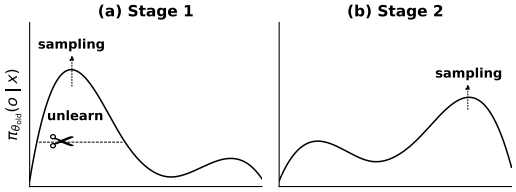

Figure 4: Unlearning suppresses the dominant mode and enables exploration of alternative modes that would otherwise be hard to reach.

3.2 Adaptive Unlearning for Dominant Mode Suppression

Our goal is to temporarily suppress dominant modes in π_{rollout} when entropy begins to collapse, while preserving non-dominant, informative modes. An effective unlearning mechanism for this setting should: (a) activate at the onset of mode collapse, (b) penalize dominant regions more than others, and (c) remain lightweight. We realize these desiderata with three simple designs.

Complementary Loss to Suppress Dominant Modes The unlearning strength should increase with token probability: strong in dominant regions with high probability mass and weak elsewhere. However, minimizing the standard *negative log-likelihood* (NLL) does not meet this requirement.

$$\mathcal{L}_{\text{NLL}} = -\log \pi_{\text{rollout}}(o_{k,t} \mid q, o_{k,<t}), \tag{4}$$

since it penalizes low-probability predictions more than high-probability ones (where the loss approaches 0). We instead use a complementary loss that reverses this emphasis:

$$\mathcal{L}_{\text{comp}} = -\log\big(1 - \pi_{\text{rollout}}(o_{k,t} \mid q, o_{k,<t})\big), \tag{5}$$

which imposes stronger penalties on dominant (high-probability) tokens and weaker penalties on small-probability modes.

To ensure numerical stability when $\pi_{\text{rollout}}(o_{k,t} \mid q, o_{k,<t}) \to 1$, we clip the probability before applying the loss:

$$p_{\text{clip}} = \min(\pi_{\text{rollout}}(o_{k,t} \mid q, o_{k,<t}),\, 1-\epsilon)\,, \tag{6}$$

where $\epsilon > 0$ is a small constant that prevents $1 - p_{\text{clip}}$ from approaching zero. The stabilized loss is:

$$\mathcal{L}_{\text{comp}} = -\log\big(1 - p_{\text{clip}}\big). \tag{7}$$

Entropy-Conditioned Activation We activate unlearning only when exploration deteriorates, as indicated by low entropy; when entropy is high, no intervention is applied. We implement this via an entropy-based indicator:

$$\mathbb{I}_t = \mathbb{I}\Big[\overline{\mathcal{H}}_t^{(m)} < \alpha\Big]\,, \tag{8}$$

where $\alpha > 0$ is a threshold and $\overline{\mathcal{H}}_t^{(m)}$ is the m-step moving average of the actor (or rollout) token entropy at step t:

$$\overline{\mathcal{H}}_t^{(m)} = \frac{1}{m}\sum_{j=0}^{m-1} \mathcal{H}_{t-j}. \tag{9}$$

Here $\mathcal{H}_t$ denotes the token-level entropy at step t. A short horizon (e.g., $m = 3$) promptly detects low-entropy phases. The indicator multiplicatively gates the complementary loss in Eq. 7, yielding the entropy-conditioned loss:

$$\mathcal{L}_{\text{unlearn}} \;=\; \mathbb{I}_t \cdot \Big[-\log\big(1 - p_{\text{clip}}\big)\Big], \tag{10}$$

Lightweight Unlearning via Single-Step Gradient Update we apply a single-step update to optimize the unlearning objective and confine its effect to the rollout model within each iteration. Let $o_k = (o_{k,1}, \ldots, o_{k,T_k})$ denote the k-th trajectory in the stage-1 rollout set $O_1 = \{o_1, o_2, \ldots, o_{G/2}\}$. The entropy-conditioned unlearning loss over O_1 is:

$$\mathcal{L}(O_1) = \frac{1}{|O_1|}\sum_{o_k \in O_1} \frac{1}{T_k}\sum_{t=1}^{T_k} \mathbb{I}_t \Big[\log\big(1 - p_{\text{clip}}(o_{k,t})\big)\Big]. \tag{11}$$

where p_{clip} denotes the clipped probability and $\mathbb{I}_t$ is the entropy-based activation indicator. We then perform a single gradient ascend step without momentum to unlearn these trajectories:

$$\theta' \leftarrow \theta' + \eta\, \nabla_{\theta'} \mathcal{L}(\theta')\,, \tag{12}$$

where θ' parameterizes the rollout model, which is synchronized from the policy model (parameterized by θ), $\theta' \leftarrow \theta$, as in GRPO's implementation (see Figure 3). Consequently, the unlearning effect is temporary—confined to the rollout model within the current iteration, without accumulation—and does not alter the policy parameters or optimization.

Algorithm 1 summarizes the EEPO procedure. It follows GRPO's structure but incorporates adaptive unlearning between the two rollout stages. After sampling the first $G/2$ trajectories (Stage 1), we check if policy entropy falls below threshold α. If so, we perform a single gradient step to unlearn

Algorithm 1: EEPO — Exploration-Enhanced Policy Optimization

Initialize: actor θ^0; learning rates η_{GRPO}, η; group size G; iteration K; entropy threshold α

for $k = 0$ to $K - 1$ **do**

  Sample $q \sim \mathcal{D}$; set $\theta' \leftarrow \theta^k$ // sample query and synchronize rollout from actor

  Sample $\{o_i\}_{i=1}^{G/2} \sim \pi_{\theta'}(\cdot \mid q)$ // Stage 1: sample $G/2$ trajectories

  if $\overline{\mathcal{H}}^{(m)}(\pi_{\theta'}) < \alpha$ **then** // single-step adaptive unlearning

    $\theta' \leftarrow \theta' - \eta \nabla_{\theta'} \mathcal{L}_{\text{unlearn}}(\{o_i\}_{i=1}^{G/2})$

  end if

  Sample $\{o_i\}_{i=G/2+1}^{G} \sim \pi_{\theta'}(\cdot \mid q)$ // Stage 2: sample remaining trajectories

  Form $O \leftarrow \{o_i\}_{i=1}^{G}$ and compute advantages $\{A(o)\}_{o \in O}$

  $\theta^{k+1} \leftarrow \theta^k + \eta_{\text{GRPO}} \nabla_\theta J_{\text{GRPO}}(\theta^k; O, r)$ // update actor with GRPO

end for

these trajectories using the complementary loss, temporarily modifying only the rollout model. We then sample the remaining $G/2$ trajectories (Stage 2) from the potentially modified rollout model. Finally, we update the policy with GRPO's objective on all G trajectories. Note that in Eq. 3, the denominator is computed using the rollout model $\pi_{\theta'}$ that generated each trajectory.

The theoretical analysis of EEPO's effect is provided in Appendix F.2 and Appendix F.3. It shows that complementary unlearning can be characterized as a *mode-favoring mass transport process* that directly counteracts the self-reinforcement or mode-seeking effect of the RL update.

Appendix G presents the *convergence analysis* of EEPO's policy update, demonstrating that EEPO converges to a stationary point at a rate of $\mathcal{O}(1/\sqrt{T})$.

4 Experiment

4.1 Experimental Setup

Datasets. We train on the MATH dataset (Hendrycks et al., 2021a) using 8.5K hard problems (difficulty levels 3-5) following SimpleRL (Zeng et al., 2025). We evaluate on five mathematical reasoning benchmarks: Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), AMC 2023, AIME 2024. For the stronger Qwen3-8B-Base, we additionally include AIME 2025.

Models. We experiment with four LLMs: Qwen2.5-3B (Yang et al., 2024), Llama-3.2-3B-Instruct (Team, 2024), Qwen2.5-7B-Instruct (Yang et al., 2024) and Qwen3-14B-Base (Yang et al., 2025).

Training Details. We employ a binary reward (+1 for correct answer, 0 otherwise) without format constraints. All models are trained using VERL (Sheng et al., 2024) with GRPO for 2 epochs, using batch size 128, learning rate 5×10^{-7}, and 8 rollouts per question. For EEPO, we set entropy threshold $\alpha = 0.3$ and unlearning rate $\eta = 3 \times 10^{-3}$.

Further details of the experimental setup are provided in Appendix B. Experiments on Qwen3-14B-Base are provided in Appendix C.

4.2 Baselines

We compare EEPO against GRPO and several variants that are explicitly designed to enhance exploration.

Base/Instruct Model. The base model, or its instruction-tuned variant without any additional reasoning-specific training, serves as a performance lower bound.

GRPO. Standard GRPO applied to the base or instruction-tuned model using default training settings.

Increased Entropy Regularization. This variant enhances exploration by increasing the entropy weight in the training objective, encouraging the policy to generate more diverse outputs. It represents a common approach where stronger entropy regularization is used to promote exploration.

Method	Minerva Math	Olympiad Bench	AMC 23	AIME 24	Average
Qwen2.5-3B	11.8	7.9	20.0	0.0	9.9
GRPO	22.4	27.9	30.3	3.3	21.0
- Higher Temp.	25.0	25.2	32.5	3.3	21.5
- Increased Ent.	25.0	29.6	37.5	3.3	23.9
- Clip High.	22.1	26.1	40.0	3.3	22.9
- More rollouts.	21.7	26.8	37.5	6.7	23.2
DAPO	22.8	27.5	35.0	6.7	23.0
EEPO	23.5	29.3	45.0	6.7	26.1 **(+24.3)**

Table 1: Performance of EEPO compared to baseline methods on Qwen2.5-3B across four math benchmarks. Baseline results report the best performance across different hyperparameter settings (refer to Fig. 5). Average relative performance improvements (%) over GRPO are highlighted in blue.

Higher Sampling Temperature. This variant applies a higher sampling temperature during the actor's decoding process to promote exploration and reduce output determinism. Temperature-based softmax exploration (also known as Boltzmann exploration) is a widely used method to implement the ε-greedy algorithm in stochastic policies. As the temperature $t \to 0$, the policy becomes nearly greedy; as $t \to \infty$, the action distribution approaches uniform, effectively increasing exploration.

Clip Higher. This variant incorporates the "clip higher" heuristic from DAPO, which encourages the selection of rare or low-probability tokens during training. It is one of the most widely used exploration-enhancing baselines in modern RLVR pipelines.

Increased Number of Rollouts. This baseline increases the number of rollouts per training step to expand the trajectory space and encourage broader exploration. It is designed to evaluate whether EEPO with 8 rollouts can match or outperform GRPO with a larger number of rollouts (default: 16).

4.3 Experimental Results

Overall results across three LLMs. To validate the effectiveness of our method across different models and scales, we compare EEPO with baselines on three model families—Qwen2.5-3B, Llama3.2-3B-Instruct, and Qwen3-8B-Base. Tables 1–3 report the results. EEPO consistently outperforms GRPO and all exploration-enhanced GRPO variants across models and scales. Relative to standard GRPO, EEPO improves average accuracy by 24.3% on Qwen2.5-3B (21.0% $\to$ 26.1%), 33.0% on Llama3.2-3B-Instruct (17.6% $\to$ 23.4%), and 10.4% on Qwen3-8B-Base (34.7% $\to$ 38.3%). This pattern indicates that EEPO's sample-then-forget mechanism yields targeted exploration that scales from 3B to 8B parameters and transfers across base and instruction-tuned policies, providing a robust and model-agnostic improvement for mathematical reasoning under RLVR.

Method	Minerva Math	Olympiad Bench	AMC 23	AIME 24	Average
Llama3.2-3B-Instruct	14.3	12.1	20.0	10.0	14.1
GRPO	19.5	17.5	20.0	13.3	17.6
- Higher Temp.	20.6	19.1	22.5	10.0	18.1
- Increased Ent.	20.2	18.1	30.0	10.0	19.6
- Clip High.	19.1	17.3	25.0	16.7	19.5
- More rollouts.	19.1	17.2	22.5	16.7	18.9
DAPO	18.8	18.1	25.0	13.3	18.8
EEPO	20.6	18.1	35.0	20.0	23.4 **(+33.0)**

Table 2: Performance on Llama3.2-3B-Instruct.

Comparison with baselines. We compare EEPO to four exploration strategies, each evaluated at its best hyperparameter setting (Figure 5). Despite careful tuning, all baselines fail to match EEPO's performance. While these strategies can outperform GRPO, gains are modest and require brittle tuning. Temperature-based exploration exhibits a clear exploration–exploitation trade-off: performance peaks around 1.2 but degrades sharply at higher values (1.5). We also observe substantially longer training time at the best temperatures (1.2) due to the much longer reasoning paths caused by inefficient exploration (Figure 7). Clip-higher and entropy regularization likewise swing between under- and over-exploration and lag behind EEPO across all models. Increasing the number of rollouts provides benefits but plateaus quickly while computational cost also grows substantially (Figure 7). In contrast, EEPO achieves larger gains by enabling targeted exploration within the rollout process.

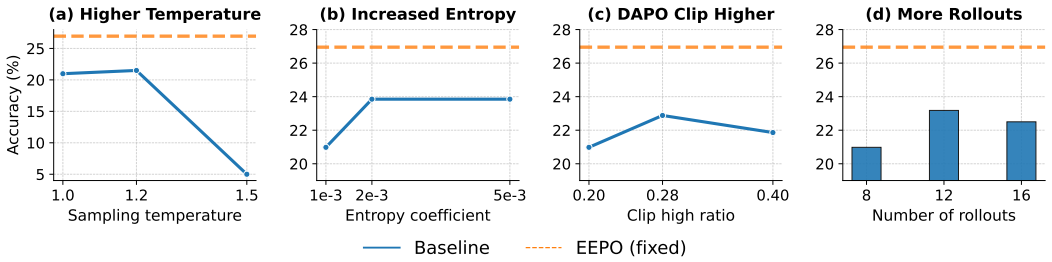

Figure 5: Impact of hyperparameter choices on baselines performance using Qwen2.5-3B. Each subplot shows the average accuracy across four math benchmarks as a function of (a) temperature, (b) entropy coefficient, (c) clip higher ratio, and (d) number of rollouts. The orange dashed line represents the EEPO with fixed hyperparameters.

Generalization to benchmarks. To assess generalization, we evaluate EEPO against baselines on five diverse math reasoning benchmarks, as shown in Tables 1–3. Our method achieves consistent improvements over GRPO across all benchmarks. Performance continues to improve on harder and distribution-shifted splits where baselines plateau. On a competition-level benchmark with Qwen2.5-3B, EEPO reaches 45.0% compared to 30.3% for GRPO. These gains stem from EEPO's sustained exploration and superior entropy maintenance (Figures 6), which prevent the entropy collapse that leads to overfitting on the training distribution and degraded generalization (Figure 1).

Method	Minerva Math	Olympiad Bench	AMC 23	AIME 24	AIME 25	Average
Qwen3-8B-Base	33.1	36.0	52.5	10	13.3	29.0
GRPO	41.2	45.5	50.0	20.0	16.6	34.7
- Higher Temp.	40.1	44.3	55.0	16.7	20.0	35.22
- Increased Ent.	40.4	42.8	60.0	16.7	20.0	35.9
- Clip High.	40.1	41.6	55.0	16.7	10.0	32.7
- More rollouts.	40.8	44.0	57.5	16.7	16.7	35.1
DAPO	40.1	43.1	62.5	13.3	16.7	35.1
EEPO	41.5	44.3	62.5	20.0	23.3	38.3 **(+10.4)**

Table 3: Performance on Qwen3-8B-Base.

5 Analysis

Effectiveness of EEPO: Exploration Enhancement and Quality Preservation. To understand the effectiveness of EEPO, we compare its training dynamics with GRPO, as shown in Figure 6.

The entropy dynamics in Figure 6(a) reveal how sample-then-forget changes exploration behavior. While GRPO exhibits continuous entropy collapse indicating that responses samples increasingly concentrate on high-probability modes, EEPO maintains consistently higher entropy throughout training. Notably, EEPO's Stage 2 achieves higher entropy than Stage 1, suggesting that temporary

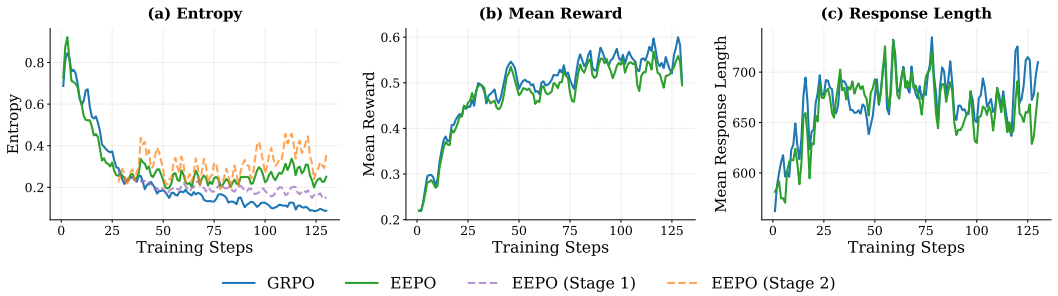

Figure 6: Training dynamics comparison between EEPO and GRPO. (a) Entropy evolution shows EEPO maintains higher exploration ability throughout training, with Stage 2 exhibiting increased entropy compared to Stage 1, demonstrating effective exploration enhancement. In contrast, GRPO exhibits monotonic entropy decay. (b) Mean training rewards remain comparable between the two methods, reflecting similar exploitation capability. (c) Response length distributions show similar patterns, indicating preserved generation quality.

response suppression successfully forces the model to explore low-density regions that the original actor rarely visits. This entropy gap demonstrates that our mechanism effectively prevents mode collapse by strategically sampling from diverse regions of the probability distribution.

Despite this enhanced exploration, generation quality remains preserved. Figure 6(b-c) shows that both mean rewards and response lengths of EEPO remain stable and comparable to GRPO. These results validate our hypothesis: temporarily suppressing sampled responses can enhance exploration by steering the actor away from high-probability regions toward other plausible alternatives, while preserving the generation capabilities necessary for effective training.

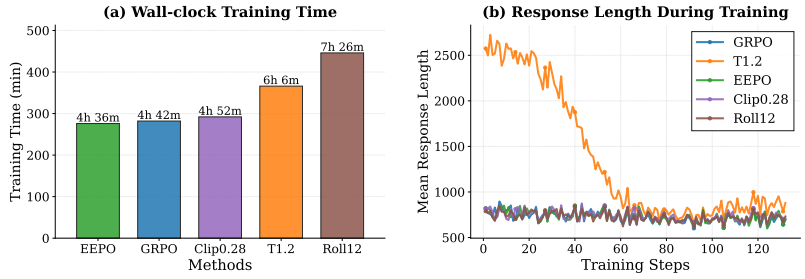

Figure 7: Training efficiency comparison on Qwen3-8B-Base. (a) Wall-clock training time for EEPO and baseline methods. (b) Mean response length during training for each method. EEPO achieves the fastest training time while maintaining stable response lengths.

Training Efficiency. We evaluate the computational efficiency of EEPO and baseline methods on Qwen3-8B-Base using B200 GPUs. As shown in Figure 7(a), EEPO achieves comparable training time to standard GRPO. This is primarily due to a slight reduction in the mean response length under EEPO (Fig. 6c), which modestly lowers the cost of generating trajectories and offsets the additional computation introduced by unlearning. Among baseline configurations, higher sampling temperatures significantly slow training by approximately 30%, as these methods generate substantially longer responses throughout training (Figure 7(b)). Additional rollouts incur the highest computational cost due to increased trajectory sampling, while adjusting the clipping ratio has minimal impact on efficiency. These results demonstrate that EEPO achieves superior performance through effective exploration while preserving the training efficiency of the original GRPO algorithm.

Details of the hyperparameter analysis and ablation are provided in Appendix D.

6 Related Work

Reinforcement learning with verifiable rewards. RLVR (Shao et al., 2024; DeepSeek-AI et al., 2025; Team et al., 2025) has recently attracted growing interest for its ability to incentivize reasoning

in LLMs using rule-based verifiable rewards. Notably, DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrates that RLVR can elicit emergent reasoning behaviors through extended chain-of-thought outputs, achieving strong performance on reasoning-intensive tasks. Despite these advances, RLVR faces challenges in exploration, often leading to early convergence and performance plateaus.

Exploration in RL. Policy gradient methods rely on policy stochasticity for exploration, but policies tend to rapidly collapse into deterministic behavior due to the exploitative nature of objectives. Common remedies *increase policy randomness* through ϵ-greedy policies (Sutton & Barto, 2018), temperature adjustment (Hou et al., 2025; Chen et al., 2025a), or entropy regularization (Hou et al., 2025). Recent work shows exploration is driven by high-entropy tokens (Wang et al., 2025), while Chen et al. (2025b) propose Pass@k rewards to encourage broader search. However, these methods remain inefficient as they ignore the action space structure. We propose a *strategic exploration strategy* that explicitly discourages revisiting previously sampled trajectories during rollout, encouraging sequential exploration of different modes.

Machine Unlearning for LLMs Machine unlearning for LLMs studies removing the influence of specific data (e.g., sensitive or copyrighted content) without retraining models from scratch (Liu et al., 2024). Typical motivations include privacy compliance and mitigating bias or harmful behaviors. Common approaches involve weight editing (Mitchell et al., 2022) or gradient-based optimization (Jang et al., 2023) to forget targeted data, and inference-time strategies such as prompt manipulation. However, prior work primarily focuses on knowledge erasure, whereas EEPO repurposes and tailors unlearning for RL exploration: during rollout generation, we temporarily unlearn previously sampled trajectories to prevent the rollout model from repeatedly sampling from dominant modes.

We provide an extended discussion of related work in Appendix A.

7 Conclusion

We introduced EEPO, an exploration-enhanced policy optimization framework that augments the rollout process with a sample-then-forget mechanism. By temporarily suppressing recently sampled trajectories during rollouts, EEPO encourages exploration of alternative modes in the output distribution that would otherwise remain underexplored. Our method transforms indiscriminate stochasticity into strategic exploration, breaking the self-reinforcing loop that causes insufficient exploration and entropy collapse. Extensive experiments across three models and five mathematical reasoning benchmarks demonstrate that EEPO consistently outperforms existing methods while maintaining comparable training efficiency. These results establish EEPO as a practical and effective approach for addressing the exploration-exploitation trade-off in RLVR.

Ethics Statement

All authors have read and adhered to the ICLR Code of Ethics. Our study relies solely on publicly available datasets and models, as detailed in Appendix B. No private or personally identifiable information was used. The work aims to advance the scientific understanding of PO methods while upholding principles of transparency, fairness, and responsible research.

Reproducibility Statement

The codebase will be made publicly available upon acceptance. All base models and PO benchmarks used in this work are publicly accessible. All experiments were conducted using NVIDIA A100 80GB GPUs and B200 184G GPUs with Python 3.12 and PyTorch 2.7.

References

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile

Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Zhipeng Chen, Yingqian Min, Beichen Zhang, Jie Chen, Jinhao Jiang, Daixuan Cheng, Wayne Xin Zhao, Zheng Liu, Xu Miao, Yang Lu, Lei Fang, Zhongyuan Wang, and Ji-Rong Wen. An empirical study on eliciting and improving r1-like reasoning models, 2025a. URL https://arxiv.org/abs/2503.04548.

Zhipeng Chen, Xiaobo Qin, Youbin Wu, Yue Ling, Qinghao Ye, Wayne Xin Zhao, and Guang Shi. Pass@k training for adaptively balancing exploration and exploitation of large reasoning models, 2025b. URL https://arxiv.org/abs/2508.10751.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Codeforces. Codeforces - competitive programming platform, 2025. URL https://codeforces.com/. Accessed: 2025-03-18.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars, 2025. URL https://arxiv.org/abs/2503.01307.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng 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. *arXiv preprint arXiv:2402.14008*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021a.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021b.

Zhenyu Hou, Xin Lv, Rui Lu, Jiajie Zhang, Yujiang Li, Zijun Yao, Juanzi Li, Jie Tang, and Yuxiao Dong. T1: Advancing language model reasoning through reinforcement learning and inference scaling, 2025. URL https://arxiv.org/abs/2501.11651.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. Knowledge unlearning for mitigating privacy risks in language models. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14389–14408, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.805. URL https://aclanthology.org/2023.acl-long.805/.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Sijia Liu, Yuanshun Yao, Jinghan Jia, Stephen Casper, Nathalie Baracaldo, Peter Hase, Yuguang Yao, Chris Yuhao Liu, Xiaojun Xu, Hang Li, Kush R. Varshney, Mohit Bansal, Sanmi Koyejo, and Yang Liu. Rethinking machine unlearning for large language models, 2024. URL https://arxiv.org/abs/2402.08787.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. Memory-based model editing at scale, 2022. URL https://arxiv.org/abs/2206.06520.

OpenAI. Learning to reason with llms. urlhttps://openai.com/index/learning-to-reason-with-llms/. Accessed: 15 March 2025.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct Preference Optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

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

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv:2409.19256*, 2024.

Richard S Sutton and Andrew G Barto. *Reinforcement Learning: An Introduction*. MIT press, 2018.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi K1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Llama Team. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning, 2025. URL https://arxiv.org/abs/2506.01939.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2. 5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

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

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild, 2025. URL https://arxiv.org/abs/2503.18892.

A Related Work

Reinforcement learning with verifiable rewards. Reinforcement learning has shown considerable promise in improving the capabilities of language models, particularly through reinforcement learning from human feedback (RLHF) (Ouyang et al., 2022; Bai et al., 2022; Rafailov et al., 2023), which aligns model outputs with human preferences. Building on this foundation, reinforcement learning with verifiable rewards (RLVR) (Shao et al., 2024; DeepSeek-AI et al., 2025; Team et al., 2025) has recently attracted growing interest for its ability to incentivize reasoning in LLMs using rule-based, automatically verifiable reward signals. Notably, DeepSeek-R1 (DeepSeek-AI et al., 2025) demonstrates that RLVR can elicit emergent reasoning behaviors (Gandhi et al., 2025) such as summarization, backward reasoning, verification, and self-reflection, often manifested through long chain-of-thought (CoT) outputs. This leads to strong performance across a wide range of reasoning-intensive tasks, such as mathematics, programming, and other problem-solving domains. The SimpleRL framework further explores how extended reasoning chains emerge under various RL training regimes. Despite these advances, RLVR still faces notable challenges in performance and stability. For example, limited exploration capabilities often lead to early convergence, resulting in performance plateaus that hinder further progress.

Exploration in RL. Policy gradient methods typically rely on randomization in the policy to encourage exploration, based on the intuition that a stochastic policy enables the agent to visit a diverse set of actions and states. However, the inherent stochasticity of the policy is insufficient, as policies tend to rapidly collapse into deterministic behavior—commonly referred to as "entropy collapse"—due to the exploitative nature of the objective function. To mitigate this issue, common remedies *increase policy randomness* by using an ϵ-greedy policy (Sutton & Barto, 2018), adjusting the softmax temperature (Hou et al., 2025; Chen et al., 2025a), or incorporating an entropy term into the objective to promote uncertainty (Hou et al., 2025). Wang et al. (2025) further show that exploration is disproportionately driven by a minority of high-entropy tokens. In parallel, Chen et al. (2025b) propose to replace the standard Pass@1 reward with Pass@k, thereby relaxing correctness constraints and encouraging the policy to maintain broader search behavior. Although these methods have shown utility, they remain inefficient as they fail to consider the structure of the action space. In contrast, we propose a *strategic exploration strategy* that explicitly discourages revisiting previously sampled trajectories by reducing their likelihood during the rollout process. This encourages the agent to sequentially explore different modes of the action distribution at a given state, thereby visiting a more diverse set of actions. Importantly, these methods are orthogonal to ours and can be combined with our approach to further enhance exploration in RLVR.

Machine Unlearning for LLMs Machine unlearning for LLMs studies how to remove the influence of specific data (e.g., sensitive or copyrighted content) without retraining models from scratch (Liu et al., 2024). Typical motivations include privacy compliance and mitigating bias or harmful behaviors. Common approaches involve weight editing (Mitchell et al., 2022) or gradient-based optimization (Jang et al., 2023) to forget targeted data, as well as inference-time strategies such as prompt manipulation. However, prior work primarily focuses on knowledge erasure, whereas EEPO repurposes and refines unlearning for RL exploration: during rollout generation, we temporarily unlearn previously sampled responses to prevent the rollout model from repeatedly sampling dominant modes.

B Detailed Experimental Setup

Datasets. We use the MATH dataset (Hendrycks et al., 2021a) for RL training. Following the setup of SimpleRL (Zeng et al., 2025), we train on the hard data, which contains 8.5K problems with difficulty levels ranging from 3 to 5. For evaluation, we adopt five challenging mathematical reasoning benchmarks: Minerva Math (Lewkowycz et al., 2022), OlympiadBench (He et al., 2024), and three recent competition-level datasets—AMC 2023, AIME 2024, and AIME 2025. For smaller models (Qwen2.5-3B and LLaMA-3.2-3B-Instruct), evaluation is conducted on the first four benchmarks. For the stronger Qwen3-8B-Base model, we additionally include AIME 2025.

Models. To demonstrate the generality of our approach, we experiment with three LLMs from different model families and scales.

- Qwen2.5-3B (Yang et al., 2024): a base model from the Qwen2.5 series, with stronger pretraining and support for long-context inputs.
- Llama-3.2-3B-Instruct (Team, 2024): an instruction-following model based on Meta's Llama architecture, included to evaluate cross-family generalization.
- Qwen3-8B-Base (Yang et al., 2025): a larger base model from the Qwen3 family, used to assess performance at a larger scale.

Reward Function. We employ a binary reward based on answer correctness: +1 for a correct final answer and 0 otherwise. We exclude format-based rewards, which can constrain exploration and degrade performance (Zeng et al., 2025), particularly when training base models.

Implementation Details. All models are trained using the VERL framework (Sheng et al., 2024), employing the GRPO algorithm. We use a batch size of 128, a mini-batch size of 64, a learning rate of 5×10^{-7}, and 8 rollouts, training for 2 epochs. The KL loss and entropy loss coefficient are set to 1×10^{-4} and 1×10^{-5}, respectively. The maximum response length varies by model: up to 4K tokens for Qwen2.5-3B, and up to 6K tokens for both LLaMA-3.2-3B-Instruct and Qwen3-8B-Base. During evaluation, we use greedy decoding to compute pass@1 accuracy. All experiments are conducted on compute clusters equipped with NVIDIA A100 GPUs (80GB) and B200 GPUs.

C Experiments on Large-scale Models

To assess how EEPO scales with model size, we extend our experiments from 3B and 8B models to a larger 14B model, Qwen3-14B-Base. The results are summarized in Table 4.

Table 4: Results on Qwen3-14B-Base across five reasoning benchmarks.

Method	Benchmark					Avg.
	Minerva Math	OlympiadBench	AMC23	AIME24	AIME25	
GRPO	36.8	48.6	67.5	23.3	26.7	40.6
EEPO	39.3	50.1	67.5	36.7	30.0	44.7

As shown in Table 4, EEPO continues to provide consistent improvements over GRPO on Qwen3-14B-Base, particularly on the more challenging benchmarks (e.g., AIME24 and AIME25). This suggests that EEPO scales well with model size and remains effective in the 3B–14B range.

D Ablation on Hyperparameters

We study the effect of two key hyperparameters in EEPO: (i) the entropy threshold α that controls when unlearning is activated, and (ii) the unlearning learning rate η that controls the step size.

D.1 Entropy Threshold α

The entropy threshold α determines when the policy entropy is sufficiently low that additional exploration should be encouraged. In practice, we select α by inspecting the training curves (cf. Fig. 2), where we observe that (1) the generalization performance begins to degrade when the entropy enters roughly the $[0.2, 0.4]$ range, with a tipping point around 0.3, and (2) before this range, entropy decays rapidly, whereas afterward the decay becomes much flatter, indicating that the policy has already become highly concentrated.

To quantify the effect of this choice, we conduct an ablation over $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.4\}$, where $\alpha = 0.0$ corresponds to GRPO (no intervention). As shown in Table 5, EEPO consistently improves over GRPO across a reasonably wide range of α, with the best performance achieved at $\alpha = 0.3$.

Table 5: Ablation on the entropy threshold α.

α	0.0	0.1	0.2	0.3	0.4
Avg. acc.	21.0	25.2	24.8	26.1	25.4

D.2 Unlearning Learning Rate η

The unlearning learning rate η controls the step size of the complementary unlearning update. In practice, we choose η to be as large as possible while keeping the unlearning process stable.

We perform an ablation over $\eta \in \{0, 1 \times 10^{-4}, 1 \times 10^{-3}, 3 \times 10^{-3}, 1 \times 10^{-2}\}$, where $\eta = 0$ reduces to GRPO (no unlearning). The results are summarized in Table 6. Performance improves steadily as η increases up to 3×10^{-3}, while an overly large rate (10^{-2}) makes the unlearning step unstable and degrades performance, which is consistent with intuition.

Table 6: Ablation on the unlearning learning rate η.

η	0	1e−4	1e−3	3e−3	1e−2
Avg. acc.	21.0	23.3	24.4	26.1	22.5

E Additional comparison with GRPO variants

To make the gain of EEPO more directly and comparable, we also provide the following fair comparisons, where EEPO is s implemented on GRPO and its variants.

Method	**Avg. acc.**
GRPO	21.0
EEPO	26.1
GRPO + Increased Entropy	23.9
EEPO + Increased Entropy	27.9
GRPO + Clip High	22.9
EEPO + Clip High	26.6

Table 7: Average accuracy of GRPO variants and their EEPO-enhanced counterparts. EEPO provides gains of 3.7–5.1 absolute accuracy points over already strong exploration-enhanced baselines.

Table 7 reports the average accuracy of GRPO and its variants, together with their EEPO-enhanced counterparts. EEPO consistently yields an absolute improvement of about 3.7–5.1 points over the corresponding exploration-enhanced GRPO methods.

F Self-Reinforcement Effect and How EEPO Counters It

We provide a theoretical analysis to support the intuitions in Figure 2 and the design of EEPO.

We work in a standard neural network setting with a feature extractor $\phi : \mathcal{Q} \to \mathbb{R}^{d\times 1}$ and a linear softmax head parameterized by $W \in \mathbb{R}^{d\times V}$. For a given query q, the logits and probabilities are

$$z^t = (W^t)^\top \phi(q), \quad p^t = \mathrm{Softmax}(z^t), \tag{13}$$

where V is the number of candidates (e.g., different reasoning modes) and p_i^t denotes the probability of candidate i at step t.

F.1 Self-Reinforcement of RL Updates

We consider an RL objective where $\boldsymbol{r} \in \{0,1\}^V$ is a reward vector with $r_i = 1$ for positive candidates and $r_i = 0$ otherwise. For this fixed q, the expected reward is

$$J(\boldsymbol{w}^t) \triangleq \sum_{i=1}^{V} r_i \, p_i^t. \tag{14}$$

We optimize J by gradient ascent on the model parameters. Assuming the feature extractor $\phi(q)$ is fixed, it suffices to analyze the dynamics of the logits z^t. Using the Softmax Jacobian $\frac{\partial p_i}{\partial z_k} = p_i(\mathbb{I}[i = k] - p_k)$, we obtain

$$\frac{\partial J}{\partial z_k} = \sum_{i=1}^{V} r_i \, \frac{\partial p_i}{\partial z_k} = \sum_{i=1}^{V} r_i \, p_i(\mathbb{I}[i = k] - p_k) = p_k\Big(r_k - \sum_{i=1}^{V} r_i p_i\Big). \tag{15}$$

Denote the average reward (i.e., the total probability mass on positive candidates) as

$$\bar{r}^t \triangleq \sum_{i=1}^{V} r_i p_i^t. \tag{16}$$

Then Eq. 15 becomes

$$\frac{\partial J}{\partial z_k} = p_k^t(r_k - \bar{r}^t). \tag{17}$$

A gradient-ascent step with learning rate $\eta > 0$ and fixed $\phi(q)$ corresponds to

$$z^{t+1} = z^t + \eta \|\phi(q)\|_2^2 \, \nabla_z J = z^t + \eta' \, g^t, \tag{18}$$

where $\eta' = \eta\|\phi(q)\|_2^2$ and $g_k^t = \partial J / \partial z_k$. In coordinates,

$$z_k^{t+1} = \begin{cases} z_k^t + \eta' p_k^t(1 - \bar{r}^t), & \text{if } r_k = 1, \\ z_k^t - \eta' p_k^t \bar{r}^t, & \text{if } r_k = 0. \end{cases} \tag{19}$$

Thus all positive candidates ($r_k = 1$) receive a positive logit update proportional to p_k^t, and all negatives ($r_k = 0$) receive a negative update. Below we focus on *relative* changes among positive candidates, since the decay of negative candidates is straightforward.

Let $\mathcal{P} = \{i : r_i = 1\}$ denote the set of positive candidates. We have the following lemma.

Lemma 1 (Self-reinforcing RL updates)**.** *Consider two positive candidates (modes) $i, j \in \mathcal{P}$ with probabilities p_i^t and p_j^t at step t. Let p^{t+1} be obtained by applying Eq. 19 and then re-normalizing with Softmax. Then*

$$\frac{p_i^{t+1}}{p_j^{t+1}} = \exp\big(\eta'(1 - \bar{r}^t)(p_i^t - p_j^t)\big) \, \frac{p_i^t}{p_j^t}. \tag{20}$$

In particular, if $p_i^t > p_j^t$ and $\bar{r}^t < 1$, then

$$\frac{p_i^{t+1}}{p_j^{t+1}} > \frac{p_i^t}{p_j^t}, \tag{21}$$

i.e., the more probable positive candidate i becomes strictly more dominant relative to j after one gradient-ascent step.

Proof. For $i, j \in \mathcal{P}$, we have from Eq. 19 that

$$z_i^{t+1} = z_i^t + \eta' p_i^t(1 - \bar{r}^t), \quad z_j^{t+1} = z_j^t + \eta' p_j^t(1 - \bar{r}^t). \tag{22}$$

Hence the logit difference evolves as

$$z_i^{t+1} - z_j^{t+1} = (z_i^t - z_j^t) + \eta'(1 - \bar{r}^t)(p_i^t - p_j^t). \tag{23}$$

Using $p_k = e^{z_k} / \sum_u e^{z_u}$, we obtain

$$\frac{p_i^{t+1}}{p_j^{t+1}} = \exp\big(z_i^{t+1} - z_j^{t+1}\big) = \exp\big(\eta'(1-\bar{r}^t)(p_i^t - p_j^t)\big) \, \exp\big(z_i^t - z_j^t\big) = \exp\big(\eta'(1-\bar{r}^t)(p_i^t - p_j^t)\big) \, \frac{p_i^t}{p_j^t}. \tag{24}$$

If $p_i^t > p_j^t$ and $\bar{r}^t < 1$, then the exponent is strictly positive, so the ratio increases. □

Interpretation. Lemma 1 formalizes the "rich get richer" behavior among positive candidates: the larger p_i^t is, the stronger the multiplicative factor in Eq. 20. Thus any small imbalance between correct candidates is amplified: the dominant positive candidate $i^\star = \arg\max_{i\in\mathcal{P}} p_i^t$ acquires a positive drift in its log-odds against every other positive candidate, leading to the self-reinforcing, mode-seeking dynamics illustrated in Fig. 2(a–b).

F.2 Unlearning as an Anti-Self-Reinforcement Operation

We now analyze the unlearning step in EEPO. In our implementation this step is instantiated via a simple complementary loss, which we show induces an opposite, *anti-self-reinforcing* effect: it explicitly suppresses the sampled candidate and shifts probability mass toward alternative candidates.

For a given index y, consider the complementary loss

$$\mathcal{L}_{\text{comp}}(p^t, y) = -\log\big(1 - p_y^t\big), \tag{25}$$

which heavily penalizes large p_y^t. Using the chain rule and Softmax derivatives, we obtain

$$\frac{\partial \mathcal{L}_{\text{comp}}}{\partial z_k} = \frac{\partial \mathcal{L}_{\text{comp}}}{\partial p_y}\frac{\partial p_y}{\partial z_k} = \frac{1}{1-p_y^t}\, p_y^t(\mathbb{I}[k=y] - p_k^t), \tag{26}$$

so

$$\frac{\partial \mathcal{L}_{\text{comp}}}{\partial z_k} = \begin{cases} p_y^t, & \text{if } k = y, \\ -\dfrac{p_y^t p_k^t}{1-p_y^t}, & \text{if } k \neq y. \end{cases} \tag{27}$$

A gradient-descent step on $\mathcal{L}_{\text{comp}}$ with learning rate $\eta > 0$ and fixed $\phi(q)$ leads to

$$z^{t+1} = z^t - \eta\|\phi(q)\|_2^2\, \nabla_z \mathcal{L}_{\text{comp}} = z^t - \eta'\, h^t, \tag{28}$$

where $\eta' = \eta\|\phi(q)\|_2^2$ and $h_k^t = \partial \mathcal{L}_{\text{comp}}/\partial z_k$. In coordinates,

$$z_k^{t+1} = \begin{cases} z_k^t - \eta' p_y^t, & \text{if } k = y, \\ z_k^t + \eta' \dfrac{p_y^t p_k^t}{1-p_y^t}, & \text{if } k \neq y. \end{cases} \tag{29}$$

Thus the complementary loss decreases the logit of the selected candidate y and increases all other logits.

Lemma 2 (Global anti-self-reinforcement)**.** *Let p^{t+1} be obtained by applying Eq. 29 and re-normalizing with Softmax. Then:*

(i) $p_y^{t+1} < p_y^t$;

(ii) for any $k \neq y$, we have

$$\frac{p_k^{t+1}}{p_y^{t+1}} > \frac{p_k^t}{p_y^t}. \tag{30}$$

Proof. Let $N^t = e^{z_y^t}$ and $A^t = \sum_{j\neq y} e^{z_j^t}$, so $p_y^t = N^t/(A^t + N^t)$. From Eq. 29,

$$N^{t+1} = e^{z_y^{t+1}} = e^{z_y^t - \eta' p_y^t} = N^t e^{-\eta' p_y^t} < N^t, \tag{31}$$

and

$$A^{t+1} = \sum_{j\neq y} e^{z_j^{t+1}} = \sum_{j\neq y} e^{z_j^t + \eta' \frac{p_y^t p_j^t}{1-p_y^t}} > \sum_{j\neq y} e^{z_j^t} = A^t. \tag{32}$$

Since $p_y = N/(A+N)$ is increasing in N and decreasing in A, we obtain $p_y^{t+1} < p_y^t$, proving (i).

For (ii), for any $k \neq y$,

$$z_k^{t+1} - z_y^{t+1} = (z_k^t - z_y^t) + \eta'\Big(\frac{p_y^t p_k^t}{1-p_y^t} + p_y^t\Big), \tag{33}$$

where the increment is strictly positive since $p_y^t > 0$ and $p_k^t \geq 0$. Thus

$$\frac{p_k^{t+1}}{p_y^{t+1}} = \exp\big(z_k^{t+1} - z_y^{t+1}\big) = \exp\Big(\eta'\Big(\frac{p_y^t p_k^t}{1-p_y^t} + p_y^t\Big)\Big) \exp\big(z_k^t - z_y^t\big) > \frac{p_k^t}{p_y^t}. \tag{34}$$

□

That is, one unlearning step always decreases the probability of the sampled mode y and strictly increases the ratio p_k/p_y for every alternative k.

Lemma 3 (Local anti-self-reinforcement)**.** *Consider two candidates i and j with probabilities p_i^t and p_j^t at step t, and suppose $p_i^t > p_j^t$. Apply one gradient-descent step on the sum of complementary losses $\mathcal{L}_{\text{comp}}(p^t, i) + \mathcal{L}_{\text{comp}}(p^t, j)$ with update rule 29, and let p^{t+1} denote the resulting distribution after re-normalizing with Softmax. Then*

$$\frac{p_i^{t+1}}{p_j^{t+1}} < \frac{p_i^t}{p_j^t}, \tag{35}$$

i.e., when both i and j are unlearned once with the complementary loss and i is initially more probable than j, the probability ratio of i relative to j strictly decreases after one unlearning step.

Proof. Because Softmax preserves log-ratios, $\frac{p_i}{p_j} = \exp(z_i - z_j)$ holds at every step. Thus it suffices to study the change of the logit difference $\Delta(z_i - z_j)$.

Using Eq. 29, the contribution of unlearning $y = i$ is

$$\Delta z_i^{(i)} = -\eta' p_i^t, \quad \Delta z_j^{(i)} = \eta' \frac{p_i^t p_j^t}{1-p_i^t}, \tag{36}$$

and the contribution of unlearning $y = j$ is

$$\Delta z_i^{(j)} = \eta' \frac{p_j^t p_i^t}{1-p_j^t}, \quad \Delta z_j^{(j)} = -\eta' p_j^t. \tag{37}$$

Summing the two effects, the total logit updates are

$$\Delta z_i = -\eta' p_i^t + \eta' \frac{p_j^t p_i^t}{1-p_j^t}, \tag{38}$$

$$\Delta z_j = \eta' \frac{p_i^t p_j^t}{1-p_i^t} - \eta' p_j^t, \tag{39}$$

so the change in the logit difference is

$$\Delta(z_i - z_j) = \Delta z_i - \Delta z_j = \eta'\Big(-p_i^t + p_j^t + \frac{p_i^t p_j^t}{1-p_j^t} - \frac{p_i^t p_j^t}{1-p_i^t}\Big). \tag{40}$$

A direct algebraic simplification yields

$$\Delta(z_i - z_j) = -\eta' \frac{(p_i^t - p_j^t)\left(2p_i^t p_j^t - p_i^t - p_j^t + 1\right)}{(1-p_i^t)(1-p_j^t)}. \tag{41}$$

For probabilities $p_i^t, p_j^t \in (0,1)$, the denominator $(1-p_i^t)(1-p_j^t)$ is positive, and the factor $2p_i^t p_j^t - p_i^t - p_j^t + 1 = (1-p_i^t)(1-p_j^t) + p_i^t p_j^t$ is also strictly positive. If $p_i^t > p_j^t$, then $(p_i^t - p_j^t) > 0$, so the overall expression is strictly negative:

$$\Delta(z_i - z_j) < 0. \tag{42}$$

Therefore

$$\frac{p_i^{t+1}}{p_j^{t+1}} = \exp\big(z_i^{t+1} - z_j^{t+1}\big) = \exp\big(\Delta(z_i - z_j)\big)\, \frac{p_i^t}{p_j^t} < \frac{p_i^t}{p_j^t}, \tag{43}$$

which proves the claim. □

Interpretation. Lemmas 2 and 3 show that complementary unlearning acts as a negative feedback on the sampled modes: each time a mode is sampled and unlearned, its probability is pushed down, and its advantage over other modes is reduced. This is exactly the opposite of the rich-get-richer effect in Lemma 1, and already suggests an anti-collapse behavior.

F.3 Where does the unlearned probability mass go?

Lemmas 2 and 3 show that complementary unlearning decreases the probability of the selected mode y and increases the log-odds of every other mode relative to y. However, they do not yet specify *where* the probability mass removed from y goes. In particular, we would like to understand whether this mass is redistributed preferentially toward already plausible modes or spread uniformly across the tail.

To answer this question, we analyze the *gradient flow* induced by the complementary loss. We again fix a query q and suppress its dependence in the notation. Let $p(\tau)$ denote the time-dependent distribution over candidates and $z(\tau)$ the corresponding logits. We consider the continuous-time limit of a gradient-descent dynamics on $\mathcal{L}_{\text{comp}}(p(\tau), y)$:

$$\frac{dz_k}{d\tau} = -\frac{\partial \mathcal{L}_{\text{comp}}}{\partial z_k}, \qquad p(\tau) = \mathrm{Softmax}(z(\tau)). \tag{44}$$

Using Eq. equation 27, and absorbing the positive factor $\|\phi(q)\|_2^2$ into the time scaling τ, we obtain the logit flow for a fixed index y:

$$\frac{dz_k}{d\tau} = \begin{cases} -p_y, & k = y, \\ \dfrac{p_y p_k}{1 - p_y}, & k \neq y, \end{cases} \tag{45}$$

where $p_k = p_k(\tau)$ and $p_y = p_y(\tau)$.

Since $p = \mathrm{Softmax}(z)$, differentiating $p_k = \exp(z_k)/\sum_u \exp(z_u)$ yields the standard relation

$$\frac{dp_k}{d\tau} = p_k\Big(\frac{dz_k}{d\tau} - \sum_u p_u \frac{dz_u}{d\tau}\Big). \tag{46}$$

Let $S_1 \triangleq \sum_{u \neq y} p_u = 1 - p_y$ and $S_2 \triangleq \sum_{u \neq y} p_u^2$. Using Eq. equation 45, we compute

$$\begin{aligned} \sum_u p_u \frac{dz_u}{d\tau} &= p_y \frac{dz_y}{d\tau} + \sum_{u \neq y} p_u \frac{dz_u}{d\tau} \\ &= p_y(-p_y) + \sum_{u \neq y} p_u \frac{p_y p_u}{1 - p_y} \\ &= -p_y^2 + \frac{p_y}{S_1} S_2. \end{aligned} \tag{47}$$

Exact probability flow. Substituting Eq. equation 45 and Eq. equation 47 into Eq. equation 46 gives closed-form expressions for the probability dynamics.

For the selected mode y,

$$\frac{dp_y}{d\tau} = p_y\Big(-p_y - \Big[-p_y^2 + \frac{p_y}{S_1} S_2\Big]\Big) = -p_y^2\Big(1 - p_y + \frac{S_2}{S_1}\Big) < 0, \tag{48}$$

so the probability of y always decreases, as expected.

For any $k \neq y$, we obtain

$$\begin{aligned} \frac{dp_k}{d\tau} &= p_k\Big(\frac{p_y p_k}{1 - p_y} - \Big[-p_y^2 + \frac{p_y}{S_1} S_2\Big]\Big) \\ &= p_k p_y\Big(\frac{p_k}{1 - p_y} + p_y - \frac{S_2}{S_1}\Big). \end{aligned} \tag{49}$$

It is convenient to introduce the *fractional growth rate*

$$\gamma_k \triangleq \frac{1}{p_k}\frac{dp_k}{d\tau}, \qquad k \neq y. \tag{50}$$

From Eq. equation 49 we have

$$\gamma_k = p_y\Big(\frac{p_k}{1-p_y} + p_y - \frac{S_2}{S_1}\Big). \tag{51}$$

Lemma 4 (Mass prefers higher-probability modes)**.** *Fix y and consider the gradient flow equation 45–equation 46. For any two distinct candidates $i, j \neq y$, their fractional growth rates satisfy*

$$\gamma_i - \gamma_j = \frac{p_y}{1-p_y}(p_i - p_j). \tag{52}$$

In particular, if $p_i > p_j$, then $\gamma_i > \gamma_j$.

Proof. Recall that for $k \neq y$ we defined the fractional growth rate

$$\gamma_k = \frac{1}{p_k}\frac{dp_k}{d\tau} = p_y\Big(\frac{p_k}{1-p_y} + p_y - \frac{S_2}{S_1}\Big), \tag{53}$$

where $S_1 = \sum_{u\neq y} p_u = 1 - p_y$ and $S_2 = \sum_{u\neq y} p_u^2$. Taking the difference for any $i, j \neq y$ gives

$$\gamma_i - \gamma_j = p_y\Big(\frac{p_i}{1-p_y} + p_y - \frac{S_2}{S_1}\Big) - p_y\Big(\frac{p_j}{1-p_y} + p_y - \frac{S_2}{S_1}\Big) \tag{54}$$

$$= p_y\frac{p_i - p_j}{1-p_y}. \tag{55}$$

Since $p_y > 0$ and $1 - p_y > 0$, the sign of $\gamma_i - \gamma_j$ is the same as the sign of $p_i - p_j$, proving the claim. □

Lemma 4 shows that, among all non-selected modes $k \neq y$, the unlearning flow *systematically favors those with larger current probability*: probability mass removed from y is reallocated so that modes with higher p_k always have a higher instantaneous growth rate than those with lower p_k. Thus the unlearned mass is not spread uniformly across the tail, but preferentially flows into already promising regions of the distribution.

Corollary 1 (The top alternative always gains probability)**.** *Let $i^\star \in \arg\max_{k\neq y} p_k$ be any most probable candidate among the non-selected modes. Under the same gradient flow, we have*

$$\frac{dp_{i^\star}}{d\tau} > 0 \quad \textit{whenever } p_y > 0. \tag{56}$$

Proof. For $i^\star$, Eq. equation 49 gives

$$\frac{dp_{i^\star}}{d\tau} = p_{i^\star}p_y\Big(\frac{p_{i^\star}}{1-p_y} + p_y - \frac{S_2}{S_1}\Big). \tag{57}$$

Since $p_{i^\star}$ is the largest element among $\{p_k : k \neq y\}$, we have

$$S_2 = \sum_{k\neq y} p_k^2 \leq p_{i^\star}\sum_{k\neq y} p_k = p_{i^\star}S_1, \tag{58}$$

so $S_2/S_1 \leq p_{i^\star}$. Therefore

$$\frac{p_{i^\star}}{1-p_y} + p_y - \frac{S_2}{S_1} \geq \frac{p_{i^\star}}{1-p_y} + p_y - p_{i^\star} \tag{59}$$

$$= p_y\Big(\frac{p_{i^\star}}{1-p_y} + 1\Big) > 0. \tag{60}$$

Since $p_{i^\star} > 0$ and $p_y > 0$, the whole expression $p_{i^\star}p_y(\cdot)$ is strictly positive, which implies $\frac{dp_{i^\star}}{d\tau} > 0$. □

Interpretation. Corollary 1 guarantees that, whenever a mode y is unlearned, at least one alternative mode—the most probable one among $\{k \neq y\}$—must receive a net gain in probability. Combined with Lemma 4, this shows that the probability mass removed from y is redistributed in a *mode-favoring* way: higher-probability alternatives grow faster than lower-probability ones, so mass is preferentially pushed toward already plausible modes rather than spread uniformly over low-probability regions.

Putting Lemma 1, Lemma 2, Lemma 4, and Corollary 1 together, we obtain a concrete picture behind Fig. 2: standard RLVR updates are intrinsically self-reinforcing and mode-seeking, while EEPO's complementary unlearning step implements a *mode-favoring mass transport* that repeatedly siphons probability mass out of the currently dominant sampled mode and reallocates it toward other high-probability modes, especially the strongest alternative. This theoretical behavior matches the empirical entropy and diversity trends observed in Fig. 4.

G Convergence of EEPO's Policy Update

EEPO modifies only the *rollout generation* process, while the policy π_θ is always updated by an importance-weighted GRPO objective that aggregates all collected trajectories. This can be regarded as mixing on-policy and slightly off-policy samples and correcting the distribution mismatch with importance sampling. Below we show that this policy update converges to a stationary point at the usual $O(1/\sqrt{T})$ rate, where T is the number of outer iterations.

Policy objective and update. Let

$$J(\boldsymbol{\theta}) \triangleq \frac{1}{n} \sum_{i=1}^{n} J_i(\boldsymbol{\theta}) \tag{61}$$

denote the GRPO-style objective over n trajectories, where each J_i encodes the clipped, normalized advantage term for one question–trajectory pair (consistent with Eq. 2). For a trajectory τ_i we write the corresponding policy-gradient term as

$$\nabla J_i(\boldsymbol{\theta}) \;=\; \sum_{t=1}^{T_i} \nabla_{\boldsymbol{\theta}} \log \pi_{\boldsymbol{\theta}}(a_{i,t} \mid s_{i,t})\, \hat{A}_i, \tag{62}$$

where $\hat{A}_i$ is the normalized advantage attached to τ_i.

At outer iteration $t = 0, 1, \ldots, T-1$, EEPO collects a mini-batch $\mathcal{B}_t$ of trajectories using its two-stage rollout procedure. Let $\pi_{\text{roll}}^{(t)}$ denote the rollout distribution that actually generates $\mathcal{B}_t$. For each $\tau_i \in \mathcal{B}_t$ we define the trajectory-level importance weight

$$w_i(\boldsymbol{\theta}^t) \triangleq \frac{\pi_{\boldsymbol{\theta}^t}(\tau_i)}{\pi_{\text{roll}}^{(t)}(\tau_i)}. \tag{63}$$

In implementation these weights are clipped to improve numerical stability; for clarity of exposition we write $w_i(\boldsymbol{\theta}^t)$ for the (clipped) value used in the update.

The policy update in EEPO can then be written as

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t + \eta_t\, g_t, \qquad g_t \triangleq \frac{1}{|\mathcal{B}_t|} \sum_{\tau_i \in \mathcal{B}_t} w_i(\boldsymbol{\theta}^t)\, \nabla J_i(\boldsymbol{\theta}^t), \tag{64}$$

where $\eta_t > 0$ is the learning rate at iteration t. When $\pi_{\text{roll}}^{(t)} = \pi_{\boldsymbol{\theta}^t}$ and $w_i \equiv 1$, Eq. 64 reduces to standard on-policy GRPO; EEPO corresponds to the case where $\pi_{\text{roll}}^{(t)}$ is temporarily perturbed by the unlearning step, and the weights w_i compensate for this perturbation.

Convergence guarantee. We now state a non-convex convergence result for the update rule 64. We adopt common assumptions from stochastic non-convex optimization: J is L-smooth (Lipschitz continuous gradient), the per-sample gradients ∇J_i are uniformly bounded by a constant $\sigma > 0$, and the (clipped) importance weights are uniformly bounded by a constant $w_{\max} > 0$. Moreover, since each $\tau_i \in \mathcal{B}_t$ is drawn from the same rollout distribution $\pi_{\text{roll}}^{(t)}$ that appears in the denominator of w_i, the importance-weighted mini-batch gradient g_t is an unbiased estimator of $\nabla J(\boldsymbol{\theta}^t)$ in the ideal (unclipped) case; clipping only changes the constants but not the rate.

Theorem G.1 (Convergence of EEPO policy update)**.** *Assume that J is L-smooth, that $\|\nabla J_i(\boldsymbol{\theta})\| \le \sigma$ for all i and $\boldsymbol{\theta}$, and that the importance weights used in Eq. 64 satisfy $|w_i(\boldsymbol{\theta})| \le w_{\max}$ for all $i, \boldsymbol{\theta}$. Let $\{\boldsymbol{\theta}^t\}_{t=0}^T$ be generated by Eq. 64 with step sizes $\eta_t = \eta = c/\sqrt{T}$, where*

$$c = \sqrt{\frac{2\big(J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^0)\big)}{L\, w_{\max}^2 \sigma^2}}, \tag{65}$$

and $\boldsymbol{\theta}^\star$ is any maximizer of J. Then

$$\min_{0\le t\le T-1} \mathbb{E}\big[\|\nabla J(\boldsymbol{\theta}^t)\|^2\big] \;\le\; w_{\max}\,\sigma \sqrt{\frac{2L\big(J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^0)\big)}{T}}. \tag{66}$$

In particular,

$$\min_{0\le t\le T-1} \mathbb{E}\big[\|\nabla J(\boldsymbol{\theta}^t)\|^2\big] = O\Big(\frac{1}{\sqrt{T}}\Big), \tag{67}$$

so the EEPO policy update converges to a stationary point of J at the standard non-convex rate.

Proof. By L-smoothness of J we have

$$\begin{aligned} J(\boldsymbol{\theta}^{t+1}) &\ge J(\boldsymbol{\theta}^t) + \big\langle \nabla J(\boldsymbol{\theta}^t), \boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t \big\rangle - \frac{L}{2}\big\|\boldsymbol{\theta}^{t+1} - \boldsymbol{\theta}^t\big\|^2 \\ &= J(\boldsymbol{\theta}^t) + \eta_t \big\langle \nabla J(\boldsymbol{\theta}^t), g_t \big\rangle - \frac{L\eta_t^2}{2}\|g_t\|^2. \end{aligned} \tag{68}$$

Taking expectations over the mini-batch sampling at iteration t, we obtain

$$\mathbb{E}\big[J(\boldsymbol{\theta}^{t+1})\big] \ge \mathbb{E}\big[J(\boldsymbol{\theta}^t)\big] + \eta_t\, \mathbb{E}\big[\|\nabla J(\boldsymbol{\theta}^t)\|^2\big] - \frac{L\eta_t^2}{2}\, \mathbb{E}\big[\|g_t\|^2\big]. \tag{69}$$

By the boundedness assumptions on ∇J_i and w_i,

$$\|g_t\| \le \frac{1}{|\mathcal{B}_t|} \sum_{\tau_i \in \mathcal{B}_t} |w_i(\boldsymbol{\theta}^t)|\, \|\nabla J_i(\boldsymbol{\theta}^t)\| \le w_{\max}\,\sigma, \tag{70}$$

so $\mathbb{E}[\|g_t\|^2] \le w_{\max}^2 \sigma^2$ and thus

$$\mathbb{E}\big[\|\nabla J(\boldsymbol{\theta}^t)\|^2\big] \le \frac{\mathbb{E}[J(\boldsymbol{\theta}^{t+1})] - \mathbb{E}[J(\boldsymbol{\theta}^t)]}{\eta_t} + \frac{L\eta_t w_{\max}^2 \sigma^2}{2}. \tag{71}$$

Summing over $t = 0, \dots, T-1$ and using $\eta_t = \eta = c/\sqrt{T}$, we obtain

$$\begin{aligned} \frac{1}{T}\sum_{t=0}^{T-1} \mathbb{E}\big[\|\nabla J(\boldsymbol{\theta}^t)\|^2\big] &\le \frac{J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^0)}{T\eta} + \frac{L\eta w_{\max}^2 \sigma^2}{2} \\ &= \frac{1}{\sqrt{T}}\left(\frac{J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^0)}{c} + \frac{Lc w_{\max}^2 \sigma^2}{2}\right). \end{aligned} \tag{72}$$

Choosing $c = \sqrt{2(J(\boldsymbol{\theta}^\star) - J(\boldsymbol{\theta}^0))/(L w_{\max}^2 \sigma^2)}$ minimizes the right-hand side and yields the desired bound. Finally, $\min_t x_t \le \frac{1}{T}\sum_t x_t$ for any nonnegative sequence, which gives the stated result. □

Remark. This result depends only on the fact that EEPO's two-stage rollouts are properly reweighted by the corresponding importance ratios $\pi_{\boldsymbol{\theta}}(\tau)/\pi_{\text{roll}}^{(t)}(\tau)$. The unlearning step changes the rollout distribution $\pi_{\text{roll}}^{(t)}$ (and hence the distribution of trajectories), but it does not change the form of the policy update in Eq. 64. Therefore, under the same mild assumptions as standard importance-weighted policy gradient, EEPO achieves the usual $O(1/\sqrt{T})$ convergence rate to a stationary point; the sample-then-forget mechanism affects *which* trajectories are seen, but not the optimization stability of the policy.

H THE USE OF LARGE LANGUAGE MODELS

In preparing this manuscript, we used a large language model (LLM) solely for polishing the writing style and improving the clarity of the manuscript. The LLM was not used for generating research ideas, designing experiments, conducting analyses, or deriving results. All scientific contributions, including the conceptualization, methodology, experiments, and conclusions, were developed entirely by the authors.