Improved Policy Optimization for Mixture-of-Experts Models: Importance Sampling and Rewarding from an Expert-Centric Perspective

Yining Qian[1], **Jinpeng Li**[2*], **Fei Mi**[2], **Lifeng Shang**[2], **Xiang Zhang**[1*]
[1]Southeast University, [2] Huawei Technologies
{yiningqian, x.zhang}@seu.edu.cn
{lijingpeng21, mifei2, Shang.lifeng}@huawei.com

Abstract

Reinforcement learning (RL) has demonstrated considerable promise in enhancing large language models. However, its application to Mixture-of-Experts (MoE) architectures is frequently hindered by training instability, primarily stemming from token-level misalignment in expert assignments between current and behavior policies. Existing approaches often oscillate between overly coarse sequence-level importance sampling, which ignores token-specific discrepancies, and restrictive expert-selection constraints that suppress beneficial policy exploration. To bridge this gap, we propose Expert Relative Policy Optimization (ERPO), which introduces expert-level importance sampling. By grouping tokens according to their routing assignments, ERPO mitigates the high variance of token-level importance sampling while overcoming the token-agnostic limitations of sequence-level methods. Furthermore, ERPO leverages this expert-centric granularity to introduce an Expert-Selection Entropy Reward, which dynamically adjusts routing uncertainty based on task-specific feedback. This unique mechanism ensures a rigorous alignment between reward signals and policy updates—a capability inherently unattainable by traditional importance sampling methods. Experimental results demonstrate that ERPO significantly outperforms strong baselines across multiple reasoning tasks, highlighting the efficacy of tailoring RL objectives to the structural inductive biases of MoE models.

1 Introduction

Reinforcement learning algorithms such as PPO (Schulman et al., 2017) and GRPO (Shao et al., 2024) have shown considerable promise in enhancing large language models (LLMs), particularly for complex reasoning tasks (OpenAI, 2024; Shao et al., 2024; Cobbe et al., 2021; Jain et al., 2024). These methods approximate on-policy training by computing a token-level importance sampling (IS) ratio, i.e., the probability ratio between the current policy and the old policy. Despite their efficiency, this introduces significant hurdles for Mixture-of-Experts (MoE) architectures (Shazeer et al., 2017), which suffer from severe training instability during the RL training stage. This instability stems from a fundamental structural misalignment: the expert assignments of the router often diverge between the old and new policies for the same input, resulting in inconsistent computational graphs. Consequently, token-level IS ratios become unreliable and exhibit high variance, as they attempt to compare probability distributions derived from structurally distinct forward paths.

To mitigate these fluctuations and gradient misalignment, GSPO (Zheng et al., 2025) employs *sequence-level importance sampling*. By aggregating token-level likelihoods into a single sequence-level IS ratio, it reduces sensitivity to routing variations of individual tokens. However, this stability comes at the expense of representational granularity: GSPO applies a uniform weight to all tokens within a response, implicitly treating them as equally significant. This assumption overlooks the inherent functional diversity and varying contribution of individual tokens, thereby discarding fine-grained signals essential for nuanced optimization. Other strategies, such as routing replay (Zheng et al., 2025) and soft replay (Zhang et al., 2025a; Zhao et al., 2025), stabilize training by either forcing the new policy to reuse experts from the old policy or penalizing routing shifts. While effective, these methods introduce non-negligible memory and communication overhead (Ma et al., 2025) and, more fundamentally, risk suppressing the discovery of superior expert selection strategies.

Considering these challenges, we propose **Expert Relative Policy Optimization** (ERPO), which introduces two core innovations. First, we develop

* Corresponding authors: Jinpeng Li, Xiang Zhang.

Findings of the Association for Computational Linguistics: ACL 2026, pages 39029–39038

Expert-level Importance Sampling (ExpertIS). Instead of the coarse sequence-level IS or the high-variance token-level IS, ExpertIS groups tokens by their assigned experts during the forward pass and computes a distinct IS ratio for each cluster. This design is grounded in two key insights: (1) tokens in reasoning trajectories naturally fall into functionally distinct categories (Wang et al., 2025b,a; Zhang et al., 2025d; Lv et al., 2024); and (2) experts in MoE models specialize in processing semantically or functionally coherent clusters (Lv et al., 2026). ExpertIS enables the model to harness the stability of aggregated IS calculation while preserving the capacity to differentiate between distinct token functions, thus addressing the limitation of GSPO.

Building upon this framework, ERPO further introduces an Expert-Selection Entropy Reward to explicitly optimize the router's strategy. This further ensures the training stability of the MoE model while simultaneously enhancing its performance. The expert-selection entropy is computed from the router's output distribution, reflecting the uncertainty in expert assignment per token. During training, this entropy serves as a dynamic reward to guide the router's specialization: for high-reward responses, ERPO reduces entropy to reinforce confident expert selection; conversely, for low-reward responses, it increases entropy to encourage exploration. This expert-targeted optimization is a unique advantage of ERPO. ERPO ensures a granularity alignment between the reward signal and the policy update, allowing for more precise refinement of the MoE's internal routing logic.

We evaluate ERPO on state-of-the-art MoE models across challenging mathematical reasoning benchmarks. Experimental results demonstrate that ERPO consistently outperforms competitive baselines, achieving superior accuracy and significantly enhanced training stability. Our contributions are summarized as follows:

- We propose ERPO algorithm, a new RL algorithm tailored for MoE models that improves training stability and reasoning performance.
- We introduce expert-level IS, which mitigates the instability of token-level IS while preserving the ability to distinguish between functional token groups, overcoming the limitation of sequence-level IS methods.
- We design an expert-selection entropy reward, which optimizes router decisions through a fine-grained alignment between rewards and policy updates.

2 Related Works

2.1 Reinforcement Learning for LLMs

Reinforcement Learning has proven effective in post-training large language models (LLMs) to enhance their reasoning abilities since the emergence of DeepSeek R1 (DeepSeek-AI et al., 2025). R1 employs Group Relative Policy Optimization (GRPO) (Shao et al., 2024) for post-training, which eliminates the expensive critic model in Proximal Policy Optimization (Schulman et al., 2017), instead estimating relative advantages within each group. However, GRPO faces challenges such as entropy collapse and a mismatch between token-level IS ratio and sequence-level reward. To address theses issues, the community develops variants of GRPO for broader use. DAPO (Yu et al., 2025) introduces token-level policy gradient loss to balance contributions across responses of different lengths and clip higher to avoid entropy collapse. Building on different token entropy patterns in RL process, updating only a minority of tokens with high entropy (Wang et al., 2025b) achieves comparable performance with full gradient update. GTPO and GRPO-S (Tan et al., 2025) propose a dynamic entropy weighted reward design that precisely aligns with token-level and sequence-level policy optimization perspectively, ensuring more effective learning at both levels. GSPO (Zheng et al., 2025) calculates IS ratio based on sequence likelihood, which aligns well with the sequence-level advantage and makes the training more robust to token-level fluctuations, particularly in MoE models.

2.2 Stable RL for Mixture-of-Experts Models

Mixture-of-Experts (MoE) models (Shazeer et al., 2017; Dai et al., 2024; Yang et al., 2025; OpenAI et al., 2025; Lv et al., 2026) divide each dense FFN layer into multiple expert blocks and sparsely activate router-selected experts for each token.

MoE models exhibit well-documented training instability when optimized with GRPO or PPO algorithms. One of the core underlying reasons is that GRPO adopts a token-level IS ratio, tokens within the same sample may be processed by different experts under the updated policy model compared with the old policy model, leading to a train-

inference mismatch and an unstable IS ratio. One intuitive solution is Routing Replay, first proposed by (Zheng et al., 2025), which caches the activated experts during the inference stage and replays routing patterns when calculating the IS ratio. Routing replay only updates parameters of experts activated by the old policy. Similarly, RSPO (Zhang et al., 2025a), a softer variant of Routing Replay, mitigates training instability by adjusting the IS ratio with a penalty based on the routing shift ratio. IcePop (Zhao et al., 2025) adopts a gradient-centric perspective, applying a dual-clip mechanism to prevent excessively large gradient updates for tokens that exhibit sharp routing shifts. Notably, the aforementioned methods overlook the potential value of exploring alternative routing patterns. An alternative solution entails allowing routing patterns to evolve dynamically while mitigating the fluctuations caused by individual tokens. For instance, GSPO (Zheng et al., 2025) computes the IS ratio based on the more stable sequence-level likelihood, rather than token-level log-likelihood, leveraging the inherent language modeling capabilities of MoE models.

3 Preliminaries

Notation In this paper, we use π_θ to denote language models parameterized by θ. For a prompt x sampled from a query set $\mathcal{D}$, the model π_θ generates a response y which is verified by a reward function $r(x, y)$. The likelihood of y is given by $\pi_\theta(y|x) = \Pi_{t=1}^{|y|}\pi_\theta(y_t|x, y_{<t})$, where $|y|$ is the number of tokens in y.

3.1 Formulation of RL Algorithms

Group Relative Policy Optimization (GRPO) (Shao et al., 2024) builds on the traditional PPO (Schulman et al., 2017) by replacing an expensive value model with group relative advantage estimation and improves train efficiency. Specifically, given an input x sampled from $\mathcal{D}$, the old policy model $\pi_{\theta_{\text{old}}}$ generates a group of G responses $\{y_i\}_{i=1}^G$. GRPO calculates advantage within a group and optimizes the following objective:

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|x)} [\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\min\Big(w_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(w_{i,t}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\Big)] \tag{1}$$

where the IS ratio $w_{i,t}(\theta)$ and advantage $\hat{A}_{i,t}$ for the token at position t in response y_i is defined as:

$$w_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|x, y_{i,<t})}, \quad \hat{A}_{i,t} = \hat{A}_i = \frac{r(x, y_i) - \text{mean}(\{r(x, y_i)\}_{i=1}^G)}{\text{std}(\{r(x, y_i)\}_{i=1}^G)} \tag{2}$$

For efficiency in practical training, the batch size used by $\pi_{\theta_{\text{old}}}$ to generate responses during the inference stage is typically much larger than the mini-batch used to update π_θ during training (Sheng et al., 2025). From a mathematical perspective, the IS ratio $w_{i,t}$ estimates how likely a token would be sampled by π_θ, therefore $w_{i,t}(\theta)\hat{A}_{i,t}$ corresponds to the expected advantage under π_θ. In essence, the IS ratio re-weights each token's gradient so that the update approximates training on samples generated by the new policy.

Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025) calculates IS ratio based on sequence likelihood and optimizes the following objective:

$$\mathcal{J}_{GSPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^G\sim\pi_{\theta_{\text{old}}}(\cdot|x)} [\frac{1}{G}\sum_{i=1}^{G} \min\Big(s_i(\theta)\hat{A}_i, \text{clip}\left(s_i(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_i\Big)] \tag{3}$$

where the advantage $\hat{A}_i$ is the same as in Equation 2 and the sequence-level IS ratio $s_i(\theta)$ for y_i is defined as:

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|x)}{\pi_{\theta_{\text{old}}}(y_i|x)}\right)^{\frac{1}{|y_i|}} \tag{4}$$

The sequence-level IS ratio $s_i(\theta)$ estimates how likely y_i would be sampled from π_θ and weights token gradients within the sequence equally.

3.2 Formulation of MoE models

A Mixture-of-Experts (MoE) layer consists of a router R and a set of experts $\{\text{Expert}_j\}_{j=1}^N$. For each input token $\tau \in \mathbb{R}^d$, the router computes a gating distribution $P(\tau)$ over all experts, from which a sparse subset of expert indices $I(\tau)$ is selected via a top-K operation. The final output is then derived as a weighted aggregation of the selected experts'

outputs, formulated as follows:

$$\begin{aligned} P(\tau) &= \text{Softmax}(R(\tau)) \\ I(\tau) &= \text{argtopK}(P(\tau)) \\ \text{Output}(\tau) &= \sum_{j \in I(\tau)} \text{Expert}_j(\tau)\, P(\tau)[j] \end{aligned} \tag{5}$$

4 Method

A critical challenge of GRPO lies in that the token-level IS ratio $w_{i,t}(\theta)$ may fluctuate drastically, especially in MoE models. A single token τ in y, which is generated by the old policy model $\pi_{\theta_{\text{old}}}$, may be processed by a distinct set of experts under the updated policy model π_θ, leading to the final token log likelihoods deviating significantly from those of $\pi_{\theta_{\text{old}}}$. Furthermore, GRPO suffers from a structural mismatch between the token-level IS ratio and the sequence-level reward.

In this section, we propose Expert Relative Policy Optimization (ERPO). Our primary goal is to stabilize the IS ratio and align the granularity of the optimization objective with the reward signal, ultimately enhancing the MoE model's overall performance and training convergence. ERPO consists of two critical parts: more stable Expert-level Importance Sampling (ExpertIS, §4.1) and Expert-Selection Entropy Reward (§4.2) which aligns the granularity of ExpertIS with rewards while dynamically adjusting expert-selection during training.

4.1 Expert-level Importance Sampling (ExpertIS)

To mitigate the extreme token-level instability in MoE models, ERPO employs expert-level IS which shifts the focus from individual tokens to token clusters based on expert selection, given that experts in MoE models naturally specialize in processing tokens with similar latent characteristics (Lv et al., 2026, 2025a; Wu et al., 2025).

In response y_i, let $\mathcal{X}_j^i$ denote the cluster of tokens assigned to Expert_j under π_θ, and $\mathcal{E}^i$ denote indices of all activated experts. Therefore, for each token τ in y_i and $j \in I(\tau)$, we have $\tau \in \mathcal{X}_j^i$ and $j \in \mathcal{E}^i$. To obtain a expert-level log likelihood of the corresponding token cluster, we compute the mean log likelihoods over all tokens in $\mathcal{X}_j^i$:

$$T_j^i(\theta) = \frac{1}{|\mathcal{X}_j^i|} \left(\sum_{\tau \in \mathcal{X}_j^i} \log \pi_\theta(\tau | x, y_{i,<\text{ts}(\tau)}) \right) \tag{6}$$

where $\text{ts}(\tau)$ denotes the position of τ in y_i. When computing $T_j^i(\theta)$, the expert indices $I(\tau)$ are detached from the computational graph.

ERPO computes the IS ratio based on the log likelihood of token clusters $T_j^i(\theta)$. The ExpertIS ratio for tokens in $\mathcal{X}_j^i$ is then defined as:

$$\begin{aligned} e_{i,j}(\theta) &= \exp\left(T_j^i(\theta) - T_j^i(\theta_{\text{old}})\right) \\ &= \frac{\left(\prod_{\tau \in \mathcal{X}_j^i} \pi_\theta(\tau | x, y_{i,<\text{ts}(\tau)})\right)^{\frac{1}{|\mathcal{X}_j^i|}}}{\left(\prod_{\tau \in \mathcal{X}_{j_{\text{old}}}^i} \pi_{\theta_{\text{old}}}(\tau | x, y_{i,<\text{ts}(\tau)})\right)^{\frac{1}{|\mathcal{X}_{j_{\text{old}}}^i|}}} \end{aligned} \tag{7}$$

It is worth noting that ExpertIS is layer-agnostic: for the expert blocks in each FFN layer, each expert independently computes its own IS ratio, and the underlying computation procedure is identical across all layers and experts. Accordingly, we omit explicit layer notation in subsequent derivations and, in line with the preliminaries section, reuse the notation $|\mathcal{E}^i|$ to denote the total number of experts activated in y_i across all layers.

This mechanism offers two distinct advantages. First, in MoE models, the ExpertIS ratio is more stable than the token-level IS ratio employed in GRPO, as it is agnostic to token-specific sampling variability. Second, unlike GSPO which treats all tokens within a sequence equally, ERPO distinguishes between different token clusters based on their expert selection. By preserving these expert-specific distinctions, ERPO provides a more effective and granular optimization signal that fully utilizes the unique semantic characteristics captured by experts.

4.2 Expert-Selection Entropy Reward

A unique advantage of ExpertIS is that it enables reward exploration at the expert level. This is because the expert-level reward granularity can inherently align the reward signal with policy updates, a critical alignment that is absent in GRPO. To leverage this advantage, we propose an Expert-Selection Entropy Reward, which further enhances post-trained MoE models by explicitly optimizing the router's expert selection strategy, while preserving training stability.

We first introduce expert-selection entropy H, which is defined as:

$$H_{i,j} = \frac{1}{|\mathcal{X}_j^i|} \sum_{\tau \in \mathcal{X}_j^i} -P(\tau)[j] \cdot \log P(\tau)[j] \tag{8}$$

$H_{i,j}$ measures the overall certainty of expert selection across the cluster of tokens in $\mathcal{X}_j^i$, where a lower entropy indicates that these tokens are more decisively routed to Expert$_j$. Specifically, when a response y_i yields a positive task reward R_{task}, the routing pattern should increase its certainty in assigning the tokens in y_i to their corresponding experts. This consolidates the successful routing pattern and is reflected by a reduction in $H_{i,j}$. Conversely, if R_{task} is negative, the expert selection for tokens in y_i should be relaxed to encourage exploration of alternative routing paths, thereby leading to an intentional increase in $H_{i,j}$. Therefore, we define Expert-Selection Entropy Reward R_{ent} as follows:

$$\begin{aligned} \Delta H_{i,j} &= H_{i,j} - H_{i,j_{\text{old}}} \\ \sigma_i &= -\text{sgn}\left(R_{\text{task},i} - \text{mean}(\{R_{\text{task},i}\}_{i=1}^{G})\right) \\ R_{\text{ent},i,j} &= \sigma_i \cdot \Delta H_{i,j} \end{aligned} \tag{9}$$

where $\text{sgn}(\cdot)$ is the sign function.

The overall reward $R_{i,j}$ for tokens in $\mathcal{X}_j^i$ is then calculated as a weighted combination of the task reward and the expert-selection entropy reward:

$$R_{i,j} = (1-\alpha)R_{\text{task},i} + \alpha R_{\text{ent},i,j} \tag{10}$$

where $\alpha \in [0,1]$ is a hyperparameter controlling the strength of the entropy regularization.

4.3 Expert Relative Policy Optimization (ERPO)

By integrating ExpertIS and expert-selection entropy reward, the final optimization objective of ERPO is formulated as:

$$\begin{aligned} &\mathcal{J}_{ERPO}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|x)} \\ &[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{E}^i|}\sum_{j\in\mathcal{E}^i}\min(e_{i,j}(\theta)\hat{A}_{i,j}, \\ &\text{clip}\left(e_{i,j}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,j})] \end{aligned} \tag{11}$$

where $\hat{A}_{i,j} = \frac{R_{i,j} - \text{mean}(\{R_{i,j}\}_{i=1}^{G})}{\text{std}(\{R_{i,j}\}_{i=1}^{G})}$ is the advantage estimation for Expert$_j$.

The advantages of ERPO (the stability of ExpertIS and the expert-level reward signal) can be further elucidated by analyzing the gradient of the ERPO objective function. For brevity, we omit the gradient clipping mechanism and define $\mathbb{E}[\cdot]$ as the expectation over inputs x sampled from $\mathcal{D}$ and responses $\{y_i\}_{i=1}^{G}$ sampled from $\pi_{\theta_{\text{old}}}(\cdot|x)$):

$$\begin{aligned} &\nabla_\theta \mathcal{J}_{ERPO}(\theta) \\ &= \nabla_\theta \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{E}^i|}\sum_{j\in\mathcal{E}^i} e_{i,j}(\theta)\hat{A}_{i,j}\right] \\ &= \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{E}^i|}\sum_{j\in\mathcal{E}^i} e_{i,j}(\theta)\hat{A}_{i,j}\nabla_\theta \log e_{i,j}(\theta)\right] \\ &= \mathbb{E}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\mathcal{E}^i|}\sum_{j\in\mathcal{E}^i} e_{i,j}(\theta)\hat{A}_{i,j}\frac{1}{|\mathcal{X}_j^i|}\sum_{\tau\in\mathcal{X}_j^i} \\ &\qquad \nabla_\theta \log \pi_\theta(\tau|x, y_{i,<ts(\tau)})\Bigg] \end{aligned} \tag{12}$$

Unlike GSPO, which assigns a uniform weight to the gradient of the log likelihoods of every token within a response, ERPO implements a more granular credit assignment by differentiating tokens based on their expert selection. For any token τ in $\mathcal{X}_j^i$, its gradient contribution is scaled by: $e_{i,j}(\theta)\hat{A}_{i,j}\frac{1}{|\mathcal{X}_j^i|}$, which not only differentiates token contributions based on the specific advantage $\hat{A}_{i,j}$ but also ensures a more balanced gradient magnitude. The factor $\frac{1}{|\mathcal{X}_j^i|}$ prevents "heavy-load" experts from disproportionately dominating the parameter updates simply due to high throughput. This normalization effectively eliminates the bias introduced by varying expert loads, ensuring that each expert is treated as an equal functional unit whose update magnitude is determined by its performance quality rather than its token volume.

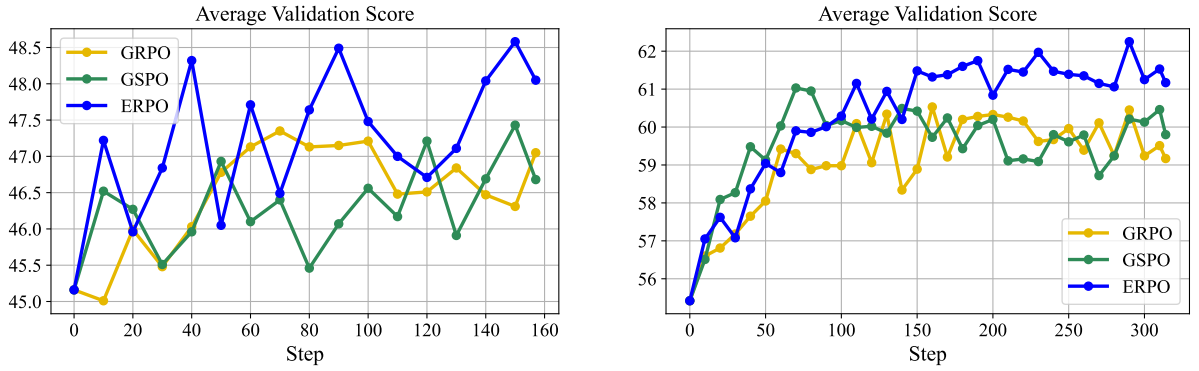
5 Experiments

5.1 Setup

Hyper-parameters We conduct experiments on GPT-OSS-20B with 4 experts activated out of 16 (OpenAI et al., 2025) and Qwen3-30B-A3B with 8 activated experts out of 128 (Yang et al., 2025). All experiments are carried out using the VeRL framework (Sheng et al., 2025). We compare our proposed method, ERPO, against strong baselines including GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025). For GPT-OSS-20B, we sample 4 responses per prompt and set the maximum generation length to 4096 under the medium reasoning effort. The training batch size is set to 256 with a mini-batch size of 64, corresponding to 4 updates per training batch. For Qwen3-30B-A3B, we sample 8 responses per prompt with a maxi-

Method	AIME24	AIME25	MATH500	Minerva	Olympiad	Avg.
GPT-OSS-20B						
Base	44.30	40.00	85.60	18.38	37.50	45.24
GRPO	50.41	42.49	86.40	19.12	38.54	47.35
GSPO	49.79	42.71	**87.20**	19.49	37.95	47.43
ERPO w/o R_{ent}	49.17	43.13	87.40	**20.59**	38.99	47.86
ERPO w. $R_{\text{ent-abs}}$	48.33	45.00	86.40	18.75	30.14	47.52
ERPO	**51.04**	**46.25**	87.00	19.49	**39.14**	**48.58**
Qwen3-30B-A3B						
Base	63.96	48.13	89.00	34.93	41.10	55.42
GRPO	72.50	61.67	90.02	34.19	44.07	60.53
GSPO	73.96	61.25	90.80	35.66	43.47	61.03
ERPO w/o R_{ent}	**76.04**	61.67	91.40	36.03	45.55	62.14
ERPO w. $R_{\text{ent-abs}}$	74.17	60.00	90.80	**36.76**	45.25	61.40
ERPO	73.96	**63.75**	**92.20**	35.66	**45.70**	**62.25**

Table 1: Main evaluation results. For each method, we report the highest average performance across five tasks. Results in **bold** denote the best performance on each task. Note that R_{ent} is a tailored expert-level reward; accordingly, ablation of R_{ent} or $R_{\text{ent-abs}}$ is not applicable to GSPO and GRPO.



(a) GPT-OSS-20B.

(b) Qwen3-30B-A3B.

Figure 1: Average validation score throughout training across different models.

mum generation length of 14336 under the thinking mode. The training batch size is set to 128 with a mini-batch size of 32, also resulting in 4 updates per training batch. For GRPO and ERPO, we set the clip ratios to $\epsilon_{\text{low}} = 0.2$ and adopt the *Clip Higher* strategy in DAPO (Yu et al., 2025) and set $\epsilon_{\text{high}} = 0.28$. For GSPO, we adopt the recommended settings from the GSPO paper, using $\epsilon_{\text{low}} = 3 \times 10^{-4}$ and $\epsilon_{\text{high}} = 4 \times 10^{-4}$. We set the reward weighting coefficient α in ERPO to 0.2.

Task and Datasets We use the DeepScaleR dataset (Luo et al., 2025) for training, which contains approximately 40,000 mathematical prompts curated from AIME and AMC problems prior to 2024. For evaluation, we adopt AIME24, AIME25, MATH500 (Hendrycks et al.), Minerva (Lewkowycz et al., 2022) and OlympiadBench (Huang et al., 2024) as benchmark datasets. All responses are evaluated using rule-based validation. For each problem in AIME, we sample 16 responses and report the Avg@16 metric. For MATH500, Minerva and OlympiadBench, we sample a single response per problem and compute the average score. During evaluation, we set the temperature to 1 and the top-p value to 0.7.

5.2 Main Results

Table 1 summarizes the performance of ERPO and competitive baselines across five comprehensive mathematical benchmarks. ERPO outperforms both GRPO and GSPO on mathematical tasks, achieving the highest overall average performance. The advantage of ERPO is particularly evident on the challenging AIME25 benchmark, where it makes substantial improvements on all baselines by over 3 points on the GPT-OSS-20B model and

by over 2 points on Qwen3-30B-A3B.

Figure 1 shows the average validation score throughout the training process. ERPO consistently exhibits the strongest performance across all benchmarks. Notably, while GRPO and GSPO tend to struggle with performance improvement or even experience degradation in the later stages of training, ERPO maintains a steady upward trend. This sustained improvement suggests that ERPO manages to stabilize RL training for MoEs and continuously enhance the model's capabilities throughout the entire training process.

5.3 Ablation Study

We conduct ablation studies on ERPO under two distinct settings to evaluate the contributions of its components:

- **ERPO w/o R_{ent}**: The reward coefficient α is set to 0, retaining only the ExpertIS $e_{i,j}(\theta)$ to assess the impact of removing the expert-selection entropy reward entirely.
- **ERPO w. $R_{\text{ent-abs}}$**: Eq. 9 is modified as:

$$R_{\text{ent-abs},i,j} = \sigma_i \cdot H_{i,j},$$

  which verifies the effectiveness of our relative entropy optimization.

Aside from these specific modifications, we maintain the same training recipe as used in the standard ERPO configuration. The average score across the five mathematical tasks are summarized in Table 1.

Removing the expert-selection entropy reward (w/o R_{ent}) results in a noticeable performance decline compared to the full ERPO, though it still outperforms the baselines. The performance gains over baselines suggest that the ExpertIS ratio alone is sufficient to substantially improve model performance. However, the performance drop without R_{ent} by ERPO is particularly significant for GPT-OSS-20B. This model exhibits relatively high sequence entropy during the initial training stages, signaling lower confidence during generation as the model explores alternative reasoning paths. These results demonstrate that R_{ent} manages to explicitly optimizes expert selection and thus effectively guides the model toward more accurate and confident reasoning paths throughout the training.

Using $R_{\text{ent-abs}}$ leads to a performance decline across both models compared to the full ERPO. This confirms that optimizing the absolute entropy under the current policy π_θ is insufficient for refining expert selection. The shortfall likely stems from the expert selection shift between the old and new policies; since the old policy generates the responses, ignoring its expert selection certainty may lead to suboptimal performance. Therefore, optimizing the relative change in expert-selection entropy—rather than simply penalizing absolute levels—provides a more effective signal for aligning and refining expert routing.

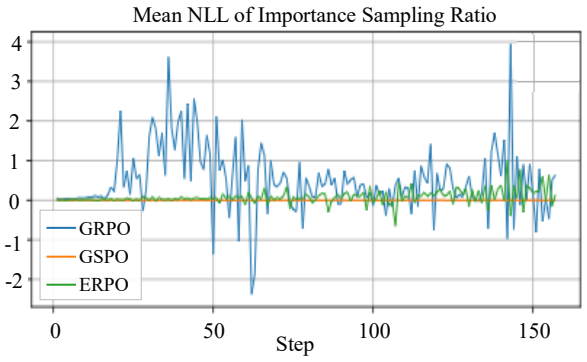


Figure 2: Mean negative log value of IS ratio during training.

6 Method Analysis

Beyond showing effectiveness via downstream performance, we present further evidence supporting some key claims.

More Stable Importance Sampling Figure 2 shows the mean negative log value of the IS ratio for each method. GRPO, with token-level IS, fluctuates drastically from -2 to 4, which leads to unstable training process. ERPO maintains a stable IS ratio during training, which further validates the rationality of ExpertIS. This stability allows ERPO to generate more reliable, lower-variance optimization signals compared to the high-variance updates of GRPO. Although GSPO also maintains a stable IS ratio, it does so at the sequence level, yielding only sparse and coarse-grained updates that fail to capture functional diversity across tokens and thus limit optimization efficiency. Overall, ERPO achieves a decent trade-off: it retains the fine-grained distinctions necessary for handling diverse token categories while providing the training stability essential for effective RL.

Motivation Validation ERPO is motivated by the observation that MoE models naturally categorize tokens into functionally coherent clusters via expert routing. To validate this foundation, we first

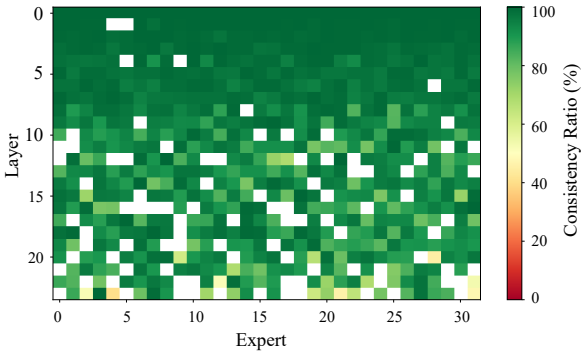


Figure 3: Most experts process a high proportion of identical tokens across policy updates, confirming that token-cluster-based importance sampling and entropy designs are mathematically well-founded. The heatmap displays the consistency ratio of tokens assigned to each expert from $\pi_{\theta_{\text{old}}}$ to π_θ. Clusters with fewer than 100 tokens are masked (white), as their small size relative to the total input 70k tokens renders them uninformative and biased.

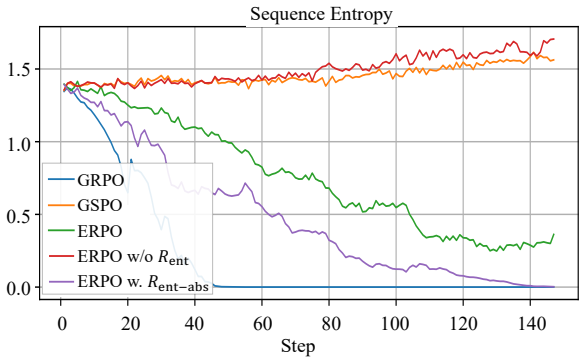


Figure 4: Sequence entropy during training.

demonstrate that token-expert assignments remain highly consistent throughout training. We quantify this consistency using a consistency ratio for each expert, defined as the token overlap between assignments from the old policy $\pi_{\theta_{\text{old}}}$ and the current policy π_θ. As shown in Figure 3, most experts process a high proportion of identical tokens across policy updates. Although some experts in deeper layers exhibit minor routing shifts—likely due to load-balancing dynamics—the overall consistency ratio remains consistently high. This stability confirms that expert-based token clusters provide a reliable and stable basis for computing IS ratios.

Furthermore, this consistency is a necessary precondition for the expert-selection entropy difference ΔH (Eq.9) to be a meaningful optimization signal. That is, the entropy terms $H_{i,j}$ and $H_{i,j_{\text{old}}}$ must be computed on largely the same set of tokens for their difference to validly indicate whether the new policy has learned an improved routing distribution. Without such assignment consistency, ΔH would be mathematically groundless, as it would compare entropies derived from incomparable token clusters.

It is crucial to note that the observed high consistency in token-expert assignments does not imply training stability is guaranteed. On the contrary, it is precisely the small proportion of tokens that do change their expert assignments during policy updates that introduces significant variance as shown in Figure 2. This phenomenon highlights the key fragility of MoE models under RL fine-tuning: even a limited degree of routing instability can disproportionately degrade training stability, underscoring the necessity of a method like ERPO, which is designed to mitigate this sensitivity at the expert level.

Sequence Entropy Analysis Figure 4 illustrates the evolution of output entropy during RL training, calculated as the average entropy of the model's final token distribution per sequence. The results delineate a clear spectrum of behaviors: ERPO without R_{ent} and GSPO fail to reduce high initial entropy, indicating insufficient policy refinement. Conversely, ERPO with $R_{\text{ent-abs}}$ and GRPO suffer from entropy collapse, leading to suboptimal outputs (Yu et al., 2025; Zhang et al., 2025b; Lv et al., 2025b; Zhang et al., 2025c). In contrast, the full ERPO framework avoids both extremes: it achieves an initial controlled entropy reduction, followed by a subsequent stable phase. Combined with the downstream performance results in Table 1, this balanced entropy pattern, neither collapsing to zero nor remaining excessively high, appears to be conducive to more effective and stable optimization.

7 Conclusion

In this paper, we propose ERPO, a novel RL algorithm for MoE models that introduces expert-level importance sampling and an expert-selection entropy reward. By optimizing at the expert granularity, ERPO strikes a critical balance between maintaining exploration and stabilizing the training process. We show that ERPO consistently surpasses strong baselines like GRPO and GSPO across various math benchmarks. This work demonstrates that leveraging expert-level structures is critical to advancing the reasoning capabilities of MoEs.

Limitations

During the forward passes of both the current policy and the old policy, ERPO accesses routing probabilities to categorize tokens into expert-specific clusters and calculate corresponding expert-selection entropy. While it avoids the heavy overhead of caching historical routing patterns as in Routing Replay that significantly inflates storage and communication costs, it still introduces additional computational overhead compared to GRPO. Nonetheless, compared to methods like Routing Replay, ERPO avoids caching old routing patterns and replaying in the new policy.

Ethical Considerations

The research presented in this paper focuses on improving stability and performance in reinforcement learning for Mixture-of-Experts models. All experiments were conducted using publicly available datasets for mathematical reasoning, which do not contain sensitive personal information or harmful content. As ERPO is a general-purpose optimization algorithm, we do not foresee any direct negative ethical impact or significant misuse risks arising from this work.

Acknowledgement

This work was supported by the National Key R&D Program of China (2023YFC3806004).

References

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Damai Dai, Chengqi Deng, Chenggang Zhao, R. X. Xu, Huazuo Gao, Deli Chen, Jiashi Li, Wangding Zeng, Xingkai Yu, Y. Wu, Zhenda Xie, Y. K. Li, Panpan Huang, Fuli Luo, Chong Ruan, Zhifang Sui, and Wenfeng Liang. 2024. Deepseekmoe: Towards ultimate expert specialization in mixture-of-experts language models. *Preprint*, arXiv:2401.06066.

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, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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. *Sort*, 2(4):0–6.

Zhen Huang, Zengzhi Wang, Shijie Xia, Xuefeng Li, Haoyang Zou, Ruijie Xu, Run-Ze Fan, Lyumanshan Ye, Ethan Chern, Yixin Ye, and 1 others. 2024. Olympicarena: Benchmarking multi-discipline cognitive reasoning for superintelligent ai. *Advances in Neural Information Processing Systems*, 37:19209–19253.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *Preprint*, arXiv:2403.07974.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models, 2022. *URL https://arxiv. org/abs/2206.14858*, 1.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. Notion Blog.

Ang Lv, Yuhan Chen, Kaiyi Zhang, Yulong Wang, Lifeng Liu, Ji-Rong Wen, Jian Xie, and Rui Yan. 2024. Interpreting key mechanisms of factual recall in transformer-based language models. *Preprint*, arXiv:2403.19521.

Ang Lv, Jin Ma, Yiyuan Ma, and Siyuan Qiao. 2026. Coupling experts and routers in mixture-of-experts via an auxiliary loss. In *The Fourteenth International Conference on Learning Representations*.

Ang Lv, Ruobing Xie, Yining Qian, Songhao Wu, Xingwu Sun, Zhanhui Kang, Di Wang, and Rui Yan. 2025a. Autonomy-of-experts models. In *Forty-second International Conference on Machine Learning*.

Ang Lv, Ruobing Xie, Xingwu Sun, Zhanhui Kang, and Rui Yan. 2025b. The climb carves wisdom deeper than the summit: On the noisy rewards in learning to reason. *Preprint*, arXiv:2505.22653.

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

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien

Bubeck, and 108 others. 2025. gpt-oss-120b & gpt-oss-20b model card. *Preprint*, arXiv:2508.10925.

OpenAI. 2024. Learning to reason with llms.

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

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

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *Preprint*, arXiv:1701.06538.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, page 1279–1297. ACM.

Hongze Tan, Jianfei Pan, Jinghao Lin, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. 2025. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *Preprint*, arXiv:2508.04349.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. 2025a. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *Preprint*, arXiv:2507.15778.

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. 2025b. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Songhao Wu, Ang Lv, Ruobing Xie, Xingwu Sun, Di Wang, Rui Yan, and Yankai Lin. 2025. Union-of-experts: Experts in mixture-of-experts are secretly routers.

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, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv: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, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Di Zhang, Xun Wu, Shaohan Huang, Yaru Hao, Li Dong, Zewen Chi, Zhifang Sui, and Furu Wei. 2025a. Towards stable and effective reinforcement learning for mixture-of-experts. *Preprint*, arXiv:2510.23027.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025b. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *Preprint*, arXiv:2507.02841.

Xiaoqing Zhang, Huabin Zheng, Ang Lv, Yuhan Liu, Zirui Song, Xiuying Chen, Rui Yan, and Flood Sung. 2025c. Divide-fuse-conquer: Eliciting "aha moments" in multi-scenario games. *Preprint*, arXiv:2505.16401.

Yanzhi Zhang, Zhaoxi Zhang, Haoxiang Guan, Yilin Cheng, Yitong Duan, Chen Wang, Yue Wang, Shuxin Zheng, and Jiyan He. 2025d. No free lunch: Rethinking internal feedback for llm reasoning. *Preprint*, arXiv:2506.17219.

Xin Zhao, Yongkang Liu, Kuan Xu, Jia Guo, Zihao Wang, Yan Sun, Xinyu Kong, Qianggang Cao, Liang Jiang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. 2025. Small leak can sink a great ship–boost rl training on moe with icepop!

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.