Towards Stable and Effective Reinforcement Learning for Mixture-of-Experts

Anonymous ACL submission

Abstract

Reinforcement learning with verifiable rewards (RLVR) has emerged as a powerful paradigm for improving reasoning capabilities. However, training RLVR with Mixture-of-Experts (MoE) policies remains fragile and is often prone to reward collapse. We identify a MoE-specific source of instability, referred to as router shift (RS), where changes in expert routing across policy updates exacerbate off-policy mismatch. This effect leads to increasingly volatile importance-ratio signals and bursty clipping behavior, which consistently precede training collapse. Motivated by this diagnosis, we propose Router-Shift Policy Optimization (RSPO). RSPO computes a per-token router-shift ratio conditioned on the previously activated experts, applies stop-gradient and a lower-bound floor, and softly rescales importance ratios prior to clipping and aggregation. This design explicitly accounts for routing-induced distributional drift during off-policy optimization. We evaluate the effect of RSPO under two settings: a synthetic countdown task and real-world reasoning tasks on MATH and Code. Across both settings, RSPO achieves better performance and exhibits greater stability compared to recent MoE-based RLVR methods.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has become a central approach for post-training large language models (LLMs) in reasoning and code generation. By relying on deterministic, rule-based verifiers that provide sparse correctness signals, RLVR has been shown to elicit strong reasoning behaviors and achieve substantial gains on challenging tasks such as mathematical problem solving and program synthesis (OpenAI, 2024; Guo et al., 2025; Yang et al., 2025; Team et al., 2025a; Chen et al., 2025). In parallel, Mixture-of-Experts (MoE) architectures offer an efficient scaling mechanism by activating only a small subset of experts per token (Fedus et al., 2022), making them particularly attractive for large-scale RLVR training where computational efficiency is critical.

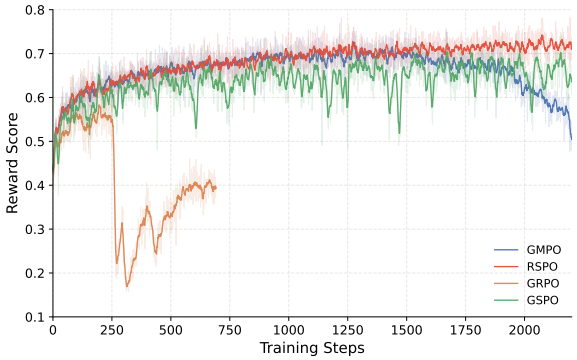

Figure 1: **Training instability on MoE.** Training reward versus step for GRPO and GRPO-style stabilizations (GSPO/GMPO/RSPO) on Qwen2.5-MoE under the Countdown RLVR setting. **Our RSPO achieves better performance while exhibiting stronger stability.**

Despite these advances, directly applying RLVR to MoE models remains brittle and often exhibits severe training instability (Zheng et al., 2025; Chen et al., 2025; Yang et al., 2025). As illustrated in Fig. 1, GRPO can suffer from abrupt reward collapse on MoE models. A key MoE-specific challenge is *router drift* (also referred to as router fluctuation): the activated experts and their routing probabilities for the *same* token may change substantially across policy updates (Dai et al., 2022; Zheng et al., 2025). Such routing changes can amplify off-policy mismatch and destabilize optimization. Moreover, RLVR commonly uses sequence-level rewards (binary correctness for an entire solution), while many practical implementations still apply token-level importance ratios and clipping, leading to additional variance and further compounding instability.

Existing stabilizations address this problem only partially. GSPO (Zheng et al., 2025) and GMPO (Zhao et al., 2025) reduce variance mis-

match by using sequence-level likelihood ratios or geometric-mean aggregation, which improves robustness to token-level outliers. However, these methods do not explicitly control the impact of *routing drift* on off-policy updates. A seemingly straightforward alternative is to constrain routing directly, e.g., freezing the router or replaying routing (Zheng et al., 2025) decisions across updates. In our experiments, these rigid strategies are unsatisfactory: freezing harms router adaptivity to the RL objective, while replay-based constraints limit router exploration and can degrade performance (see Sec. 5.4 and Appendix D).

In this work, we provide a concise diagnosis that links routing instability to optimization instability. Using lightweight training-time signals (without logging full token-level ratio distributions), we show that routing stability degrades over training and coincides with increasingly volatile off-policy mismatch signals and bursty clipping activity, which together increase the risk of reward collapse (Sec. 3). This diagnosis motivates a targeted intervention: to stabilize MoE off-policy RL, we should directly reduce the influence of tokens whose routing behavior drifts substantially across updates, *without* hard-freezing the router or fully replaying routing decisions.

Motivated by this, we propose `Router-Shift Policy Optimization` (`RSPO`), a router-aware modification to GRPO-style objectives. `RSPO` computes a per-token *router-shift ratio* from router scores on the *old activated experts* across MoE layers, applies a simple processing step (stop-gradient and lower-bound flooring), and multiplies the resulting trust weight into the importance ratio before the usual clipping and aggregation. This yields a *soft* adjustment mechanism: tokens with severe routing deviations contribute less to the policy update, mitigating routing-induced off-policy mismatch while preserving router adaptivity.

We evaluate `RSPO` under two complementary regimes. In a small-scale diagnostic setting (Qwen2.5-MoE on Countdown) (See Fig. 1), router-shift weighting consistently stabilizes GRPO and its variants when used as a plug-in component. In a large-scale benchmark setting (Qwen3-30B-A3B), `RSPO` (GMPO+RS) improves downstream Pass@1 on both **math** and **code** benchmarks and yields more stable training-time routing/optimization diagnostics compared to GRPO. Overall, our results highlight the importance of router-aware stabilization for MoE RLVR. Our main contributions are:

- **Diagnosis of MoE instability in off-policy RLVR.** We provide measurable evidence linking router drift to volatile off-policy mismatch signals and bursty clipping behavior that precede reward collapse.
- **Router-aware soft stabilization.** We propose `RSPO`, which computes a per-token router-shift ratio from old activated experts and uses it as a detached, floored trust weight to rescale importance ratios prior to clipping/aggregation, preserving router adaptivity.
- **Empirical validation at two scales.** We show that router-shift weighting acts as a plug-in stabilization module on Qwen2.5-MoE (Countdown) and that `RSPO` improves stability and final performance on Qwen3-30B-A3B across both math and code benchmarks.

2 Preliminaries

Group Relative Policy Optimization (GRPO). Given a query x, GRPO samples a group of G responses $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and computes group-relative advantages from a scalar reward $r(x, y_i)$. Let $\hat{A}_i$ denote the normalized group advantage (shared across tokens in y_i), and define the token-level importance ratio

$$w_{i,t}(\theta) \triangleq \frac{\pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid x, y_{i,<t})}. \tag{1}$$

GRPO optimizes a PPO-style clipped surrogate at the token level:

$$\ell_{i,t}^{\text{GRPO}}(\theta) \triangleq \min\Big(w_{i,t}(\theta)\hat{A}_i,\ \text{clip}\big(w_{i,t}(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\hat{A}_i\Big), \tag{2}$$

and averages it over tokens and group samples (full objective in Appendix E).

Group Sequence Policy Optimization (GSPO). GSPO addresses the mismatch between sequence-level rewards and token-level ratios by defining a *sequence-level* importance ratio via the geometric mean:

$$s_i(\theta) \triangleq \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log w_{i,t}(\theta)\right). \tag{3}$$

It then applies clipping at the sequence level:

$$\ell_i^{\text{GSPO}}(\theta) \triangleq \min\Big(s_i(\theta)\hat{A}_i,\ \text{clip}\big(s_i(\theta),\, 1-\epsilon,\, 1+\epsilon\big)\hat{A}_i\Big), \tag{4}$$

with the full expectation/averaging form given in Appendix E.

Geometric-Mean Policy Optimization (GMPO). GMPO also leverages geometric aggregation to reduce sensitivity to extreme token-wise ratios, but (unlike GSPO) keeps the token-level structure and typically performs token-wise clipping *before* geometric aggregation. We provide the complete formulation in Appendix E.

3 Diagnosing Instability in MoE Off-Policy RL

In this section, we characterize a failure mode frequently observed when applying off-policy RL (e.g., GRPO) to MoE language models: training becomes unstable and may collapse. Our goal is to provide measurable evidence linking *routing instability* (router drift between θ and θ_{old}) to increasingly *volatile* off-policy mismatch signals and more frequent activation of clipping mechanisms, which together contribute to optimization instability and eventual collapse.

3.1 Symptom: Training Instability and Reward Collapse

We start by illustrating the instability phenomenon on Qwen2.5 MoE trained with GRPO under the countdown task and rule-based reward protocol. Following GRPO, for each query x we sample G candidate responses $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$ and optimize the objective in Eq. (2) with ratio defined in Eq. (1).

We operationally define *collapse* as a sharp and sustained drop in validation score/reward accompanied by abnormally large KL / gradient norms. As shown in Fig. 1, GRPO can exhibit abrupt collapse on MoE models. GSPO mitigates collapse but often remains oscillatory, while GMPO may delay collapse yet can still fail in long runs.

This section establishes that instability is not an anecdotal artifact: it is a reproducible symptom in MoE off-policy RL and motivates a deeper diagnosis of its underlying cause.

3.2 Measuring Router Drift via a Router-Shift Ratio

We next introduce a lightweight statistic to quantify routing instability between the current policy $\theta = (\phi, \psi)$ and the old policy $\theta_{\text{old}} = (\phi_{\text{old}}, \psi_{\text{old}})$. Let $c_{i,t} = (x, o_{i,<t})$ denote the decoding context at token position t of response o_i. At each MoE layer $\ell \in \{1, \dots, L\}$, the router produces a distribution over experts, denoted by $r_{\phi}^{(\ell)}(e \mid c_{i,t})$.

Old activated experts. For each token (i, t) and layer ℓ, let $\{e_{i,t}^{(\ell,k)}\}_{k=1}^{K}$ be the top-K expert indices selected by the *old* router ϕ_{old} (i.e., the experts activated when computing the old-policy log-probabilities). We measure how much the current router changes its probability mass on these old activated experts.

Router-shift ratio. We first compute the layer-wise routing deviation

$$d_{i,t}^{(\ell)} \triangleq \frac{1}{K} \sum_{k=1}^{K} \Big| \log r_{\phi}^{(\ell)}\Big(e_{i,t}^{(\ell,k)} \mid c_{i,t}\Big) - \log r_{\phi_{\text{old}}}^{(\ell)}\Big(e_{i,t}^{(\ell,k)} \mid c_{i,t}\Big) \Big|. \tag{5}$$

and aggregate it across layers:

$$\Delta_{i,t} \triangleq \frac{1}{L} \sum_{\ell=1}^{L} d_{i,t}^{(\ell)}. \tag{6}$$

We then define the *router-shift ratio* as a bounded coefficient

$$\gamma_{i,t} \triangleq \exp(-\Delta_{i,t}) \in (0, 1], \tag{7}$$

where larger routing deviations yield smaller $\gamma_{i,t}$.

Logged severity statistics. In our large-scale GRPO runs, we log router-shift statistics as detached diagnostics. For numerical stability, we apply a floor $\gamma_{\min}$ (we use $\gamma_{\min} = 0.8$ in our implementation):

$$\tilde{\gamma}_{i,t} \triangleq \max(\gamma_{i,t}, \gamma_{\min}), \text{ClipFrac}_{\gamma_{\min}} \triangleq \Pr(\gamma_{i,t} < \gamma_{\min}). \tag{8}$$

Intuitively, $\text{ClipFrac}_{\gamma_{\min}}$ measures the fraction of tokens whose routing deviation is severe enough to fall below the threshold $\gamma_{\min}$. We use these statistics to track how routing instability evolves during training.

3.3 Router Drift Amplifies Off-Policy Mismatch and Triggers Clipping Instability

As shown in Fig. 2, GRPO training on Qwen3-30B-A3B for math reasoning exhibits a clear reward-collapse behavior at scale. We next analyze training-time stability signals to characterize how routing instability relates to off-policy optimization dynamics.

Routing-side instability. The top row of Fig. 5 tracks routing-severity signals derived from the router-shift ratio. Under GRPO, the router-shift ratio decreases while the router-shift clip fraction increases over training, indicating that routing deviations across policy updates become progressively more severe.

Optimization-side instability. The bottom row of Fig. 5 reports two lightweight optimization diagnostics: the importance-ratio signal (logged as `ppo_kl`) and the clipping activity `pg_clipfrac`. As routing drift accumulates, the importance-ratio signal becomes increasingly volatile and exhibits pronounced spikes, accompanied by more bursty clipping.

Together, these patterns suggest an instability cascade in which router drift amplifies off-policy mismatch and triggers frequent clipping, increasing the risk of training collapse.

Summary and Design Implications. We summarize our diagnosis as follows. First, MoE off-policy RL exhibits reproducible training instability and reward collapse (Fig. 1). Second, routing stability degrades over training, as reflected by a decreasing router-shift ratio and an increasing router-shift clip fraction (top row of Fig. 5). Third, this routing instability coincides with increasingly volatile off-policy mismatch signals and more frequent activation of clipping constraints (bottom row of Fig. 5), which together increase the risk of unstable optimization and eventual collapse.

These observations suggest that stabilizing MoE off-policy RL requires directly controlling the impact of routing drift on off-policy updates while preserving router adaptivity (i.e., without hard-freezing the router or fully replaying routing decisions). Motivated by this, in the next section we introduce a router-aware *soft* adjustment that uses the per-token router-shift ratio to down-weight the importance-ratio contribution of tokens with severe routing deviations.

4 Method

4.1 Overview

Sec. 3 shows that, in MoE off-policy RL, routing stability can degrade across policy updates and is accompanied by increasingly volatile off-policy mismatch signals and frequent clipping activations, which may culminate in reward collapse. Existing GRPO-style objectives (e.g., GSPO/GMPO) improve stability mainly through alternative ratio aggregation and clipping strategies, but they do not explicitly control the impact of *router drift* on the importance ratio.

We propose `Router-Shift Policy Optimization` (`RSPO`), a lightweight *router-aware* modification that can be plugged into GRPO and its variants. The key idea is to reuse the per-token router-shift ratio $\gamma_{i,t}$ defined in Sec. 3.2 as a trust signal, and multiply a processed version of it into the importance ratio before the base algorithm applies its clipping/aggregation steps.

4.2 Router-Shift Weight as a Plug-in Rescaling

Processed router-shift weight. Let $\gamma_{i,t} \in (0, 1]$ denote the router-shift ratio defined in Sec. 3.2. We apply two practical operations: (i) stop-gradient so it acts purely as a sample weight, and (ii) flooring to avoid vanishing contributions. Concretely,

$$\tilde{\gamma}_{i,t} \triangleq \mathrm{sg}\Big[\max(\gamma_{i,t},\, \gamma_{\min})\Big], \tag{9}$$

where $\gamma_{\min} \in (0, 1]$ is a hyperparameter and $\mathrm{sg}[\cdot]$ denotes stop-gradient.

Rescaling the importance ratio (before clipping). For any GRPO-style objective, define the token-level importance ratio

$$w_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid c_{i,t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid c_{i,t})},\ c_{i,t} = (x, o_{i,<t}). \tag{10}$$

`RSPO` replaces $w_{i,t}(\theta)$ with a router-aware adjusted ratio

$$\tilde{w}_{i,t}(\theta) \triangleq w_{i,t}(\theta) \cdot \tilde{\gamma}_{i,t}, \tag{11}$$

and feeds $\tilde{w}_{i,t}(\theta)$ into the *same* clipping/aggregation pipeline of the underlying base objective (GRPO/GSPO/GMPO). In other words, `RSPO` inserts a single rescaling step *right before* the base algorithm's clipping, leaving the rest unchanged.

Implementation note (log-space). Since ratios are computed in log space in our implementation, Eq. (11) is implemented stably as $\log \tilde{w}_{i,t} \leftarrow (\log \pi_\theta - \log \pi_{\theta_{\text{old}}}) + \log \tilde{\gamma}_{i,t}$ before exponentiation.

4.3 Instantiation Used in This Paper

In our main experiments, we instantiate `RSPO` on top of GMPO (denoted as **GMPO+RS**, or **RSPO**) since GMPO provides a strong and stable GRPO-style base objective for RLVR. In Sec. 5.3 we further show that the same router-shift rescaling can also be plugged into GRPO and GSPO, consistently improving training stability.

5 Experiments

5.1 Experimental Setup

Models and training regimes. We evaluate our method under two complementary regimes. **Small-scale diagnostic setting.** We conduct exploratory experiments and ablations on a Qwen2.5-MoE model pretrained on the Countdown task, primarily to stress-test training stability and isolate the effect of router-shift weighting. **Large-scale benchmark setting.** For final evaluation, we train Qwen3-30B-A3B on **math** and **code** tasks to assess downstream generalization at scale.

Baselines and hyperparameters. We compare against GRPO and two representative GRPO-style variants designed to improve stability: GSPO and GMPO. For GRPO we adopt the commonly used clipping range ϵ=0.2; GSPO/GMPO follow the recommended settings reported in their respective papers. All methods are trained under the same rollout budget (8 samples/step) to ensure fair comparison. For our method, we use a fixed router-shift floor $\gamma_{\min} = 0.8$ across both small and large settings and apply stop-gradient through the router-shift weight when used for optimization. We report mean results over 3 random seeds for training curves.

Training data and rule-based rewards. All settings use verifiable, rule-based rewards (RLVR). For the small-scale Countdown setting, training data is generated following the procedure of Qin et al. (2025). For large-scale **math** training, we use DeepScaleR (Luo et al., 2025). For large-scale **code** training, we combine multiple verifiable sources, including PrimeIntellect, LeetCode, TACO, and LiveCodeBench.

Evaluation protocol. For the small-scale setting, we monitor training progress by periodically evaluating on a held-out Countdown test set. For the large-scale setting, we evaluate both **math** and **code**. For math, we follow the Dr.GRPO protocol and report Pass@1 accuracy on five benchmarks: AIME24, AMC23, MATH500 (Hendrycks et al., 2021), Minerva (Lewkowycz et al., 2022), and OlympiadBench (Huang et al., 2024); AIME24 results are averaged over 32 runs. For code, we report Pass@1 on three benchmarks: MBPP, HumanEval, and LiveCodeBench. Unless otherwise stated, decoding is deterministic (temperature = 0.0) with one sample per input.

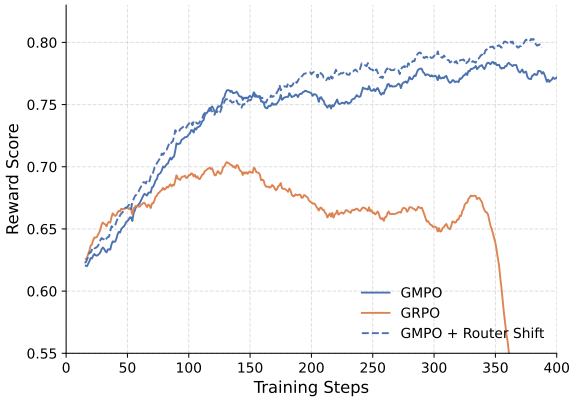

Figure 2: **Training reward dynamics on Qwen3-30B-A3B.** Training reward versus training step on the **math** RLVR setting. GRPO exhibits a clear reward collapse in the later stage of training, whereas GMPO remains more stable. RSPO (GMPO+RS) maintains stable training and achieves consistently higher reward.

Additional details are provided in Appendix A.1.

5.2 Main Results

Table 1 summarizes the final Pass@1 performance on Qwen3-30B-A3B after RL training. On **math** reasoning, GMPO+RS (RSPO) achieves the best average accuracy (77.1), improving over GMPO/GSPO (76.4) and GRPO (71.5). The gains are most apparent on challenging benchmarks such as Minerva and OlympiadBench, while remaining competitive on MATH500 where GMPO/GSPO are already strong. On **code**, RSPO yields consistent improvements across all three benchmarks, increasing the average from 82.5 (GMPO) to 85.2 and substantially outperforming GRPO (70.7). All reported results are averaged over three random seeds.

Figure 2 further illustrates training dynamics on Qwen3-30B-A3B. GRPO exhibits a clear reward collapse in the later stage of training, whereas GMPO is more stable but converges to a lower reward level. In contrast, RSPO (GMPO+RS) maintains stable training and achieves the highest reward trajectory, supporting our claim that router-aware weighting improves both stability and final performance at scale.

5.3 Ablations

Component contribution: router-shift weighting on top of GMPO. To isolate the effect of router-shift weighting in our final method, we compare GMPO with RSPO (GMPO+RS) under the same large-scale protocol. As shown in Table 1, adding

Table 1: **Main results on Qwen3-30B-A3B.** Pass@1 (%) on math and code benchmarks.

	Math						Code			
Method	**AIME24**	**AMC23**	**MATH500**	**Minerva**	**OlympiadBench**	**Avg.**	**LCB**	**MBPP**	**HumanEval**	**Avg.**
Base	80.4	90.0	90.7	47.7	62.0	74.2	52.9	86.4	83.5	74.3
GRPO (Shao et al., 2024)	77.0	82.5	91.8	48.2	58.1	71.5	41.2	81.4	89.6	70.7
GSPO (Zheng et al., 2025)	80.4	95.0	93.6	48.9	64.0	76.4	58.8	87.2	95.1	80.4
GMPO (Zhao et al., 2025)	80.1	92.5	94.2	49.3	65.9	76.4	64.7	87.2	95.7	82.5
GMPO+RS (RSPO)	80.1	95.0	94.2	50.7	65.8	**77.1**	70.5	88.2	97.0	**85.2**

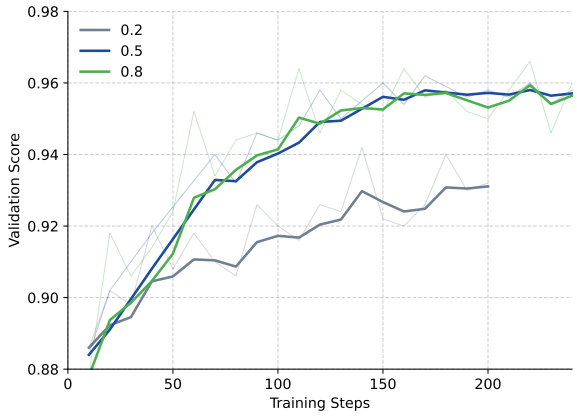

Figure 3: **Sensitivity to the router-shift floor $\gamma_{\min}$ (validation score).** Validation score versus training step on Qwen3-30B-A3B under the math RLVR setting. Curves correspond to $\gamma_{\min} \in \{0.2, 0.5, 0.8\}$, with all other hyperparameters fixed.

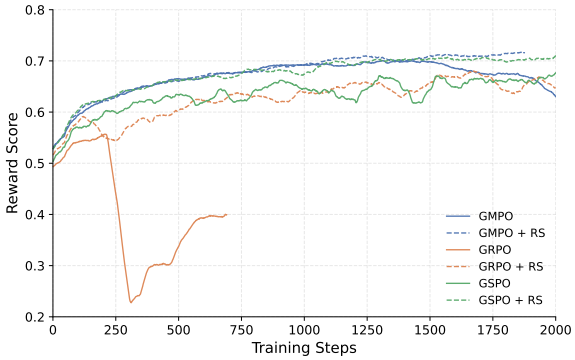

Figure 4: **Router-shift as a plug-in stabilization module (Qwen2.5-MoE, Countdown).** Training reward versus step for GRPO/GSPO/GMPO (solid) and their router-shift counterparts (dashed), using the same color per base algorithm.

router-shift weighting yields consistent improvements: on **math**, the average Pass@1 increases from 76.4 (GMPO) to 77.1; on **code**, it increases from 82.5 to 85.2. In addition to improved final accuracy, RSPO exhibits substantially more stable training dynamics than GRPO and reaches higher reward than GMPO (Fig. 2), supporting that router-shift weighting provides a complementary stabilization effect beyond geometric aggregation alone.

Sensitivity to the floor $\gamma_{\min}$. We sweep $\gamma_{\min} \in \{0.2, 0.5, 0.8\}$ on Qwen3-30B-A3B and evaluate training progress using the *validation score* tracked during RL training. Fig. 3 shows that $\gamma_{\min} = 0.5$ and 0.8 lead to comparable validation trajectories, while a too-small floor (e.g., 0.2) can over-suppress tokens with severe routing drift, weakening learning signals and degrading convergence. Unless otherwise stated, we use $\gamma_{\min} = 0.8$ as the default in all experiments.

Router-shift as a plug-in component. Router-shift weighting is a minimal modification that can be inserted into different GRPO-style objectives. On the Qwen2.5-MoE Countdown setting, we add the same router-shift weight (Sec. 4) to GRPO/GSPO/GMPO while keeping their original clipping/aggregation choices unchanged. Fig. 4 shows that router-shift consistently stabilizes training and improves the final reward/validation performance, with particularly strong benefits for GRPO which is most prone to collapse in MoE settings. This suggests that router-aware weighting is complementary to existing variance-control mechanisms and can serve as a general stabilization module.

Why stop-gradient on the router-shift weight? By default, we treat the router-shift weight as a detached sample weight. Allowing gradients to flow through the router-shift weight leads to rapid instability in our small-scale setting; for clarity, we report this ablation in Appendix B (Fig. 6). This motivates applying stop-gradient to the router-shift weight throughout the paper.

5.4 Alternative Router Stabilization Strategies

We additionally evaluate two intuitive strategies that directly constrain router dynamics: **router freezing** and **routing replay**. Freezing disables router updates entirely, assuming pretrained routing is already aligned with the RL objective. Rout-

ing replay caches routing decisions from the old policy and reuses them when evaluating the current policy, thereby eliminating routing drift.

In our experiments, these rigid strategies are not satisfactory: freezing the router limits adaptation to the RLVR objective, while routing replay restricts router exploration and incurs non-trivial memory/-communication overhead due to caching routing traces across layers and tokens. In contrast, RSPO achieves stable training without hard constraints by softly down-weighting tokens with severe routing deviations. We provide additional comparisons and implementation details for these alternatives in Appendix D.

5.5 Mechanism Diagnostics

To validate the mechanism identified in Sec. 3, we log lightweight training-time diagnostics that are available without recording full token-level ratio distributions. Specifically, we track (i) routing-severity signals from the router-shift ratio (Sec. 3.2), and (ii) optimization-side signals including the importance-ratio diagnostic (logged as `ppo_kl` in our code) and the PPO/GRPO clipping fraction `pg_clipfrac`. Across runs, we use the same threshold as in training, $\gamma_{\min} = 0.8$, when reporting router-shift clip fraction.

RSPO stabilizes both router-side and optimization-side signals. Fig. 5 contrasts GRPO and RSPO on Qwen3-30B-A3B. Under GRPO, routing stability degrades over training (router-shift ratio decreases and router-shift clip fraction rises), and the importance-ratio signal becomes increasingly volatile with bursty clipping activity. In contrast, RSPO maintains substantially more stable routing-severity statistics and reduces the volatility of both the importance-ratio diagnostic and `pg_clipfrac`, consistent with our hypothesis that softly down-weighting tokens with severe routing drift mitigates the instability cascade that can lead to reward collapse. We additionally report training entropy as an auxiliary indicator of policy collapse in Appendix Fig. 7.

5.6 Efficiency and Overhead

RSPO introduces additional overhead mainly from caching routing statistics needed to compute the router-shift weight (Sec. 3.2) and applying a per-token rescaling to the importance ratio. On Qwen3-30B-A3B, RSPO incurs a 20.8% reduction in training throughput compared to GMPO under the same training configuration, retaining 79.2% of GMPO throughput.

In terms of memory, RSPO caches old top-K routing information for each token and MoE layer. For Qwen3-30B-A3B with L=48, K=8, batch size 128, and response length 8192 (about 1.05M tokens per step), storing old top-K routing probabilities in FP16 requires approximately 0.75 GiB per device. Storing the corresponding expert indices requires an additional 0.75 GiB if stored as 16-bit integers (since there are 128 experts), yielding about 1.5 GiB extra memory in total. This overhead scales linearly with the per-device batch size and sequence length, and can be reduced by using compact index types and/or offloading cached indices to CPU memory.

6 Related Work

6.1 Reinforcement Learning for LLMs

DeepSeek-R1 (Guo et al., 2025) demonstrates the effectiveness of RLVR-style training for eliciting strong reasoning in LLMs. Its core optimization method, Group Relative Policy Optimization (GRPO) (Shao et al., 2024), simplifies PPO (Schulman et al., 2017) by removing the value model and using group-relative advantages. Following this line, several GRPO-style systems and analyses target efficiency and stability: DAPO (Yu et al., 2025) introduces dynamic sampling and larger clipping thresholds, while Dr. GRPO (Liu et al., 2025) studies and mitigates length bias by modifying GRPO's normalization terms. More broadly, recent works note that token-level importance ratios can induce high-variance updates under sequence-level rewards, motivating alternative aggregations. GMPO (Zhao et al., 2025) stabilizes optimization via geometric-mean aggregation, and GSPO (Zheng et al., 2025) similarly advocates sequence-level ratios and arrives at a closely related geometric-mean formulation. Notably, GSPO reports that geometric aggregation is particularly helpful when applying RL to Mixture-of-Experts (MoE) models (Zheng et al., 2025).

6.2 Stability in MoE Training

MoE models scale capacity through sparse expert activation but introduce challenges such as load imbalance and routing instability. Switch Transformer (Fedus et al., 2022) addresses these issues with an auxiliary load-balancing loss and capacity constraints, while Loss-Free Balancing (Wang

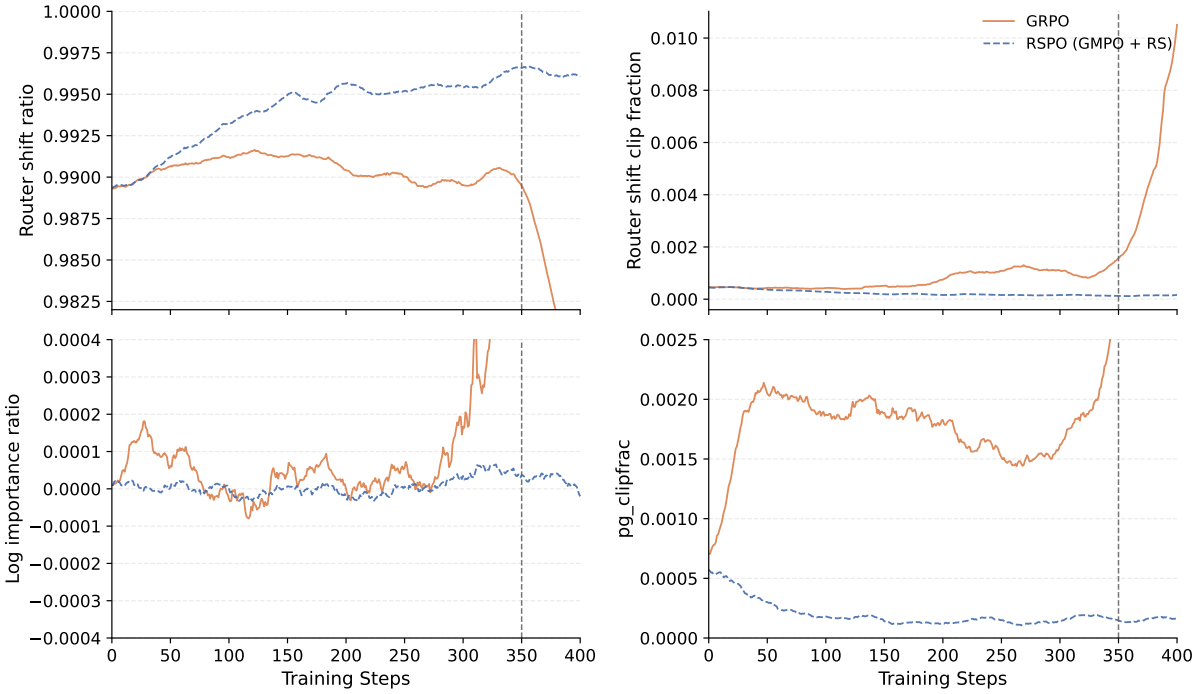

Figure 5: **Mechanism diagnostics on Qwen3-30B-A3B (math RLVR).** Router-side severity signals (top row) and optimization-side signals (bottom row) over training for GRPO vs RSPO. RSPO keeps routing deviations smaller and reduces volatility and bursty clipping in off-policy updates.

et al., 2024a) adjusts routing biases to improve expert utilization without auxiliary-loss interference. StableMoE (Dai et al., 2022) highlights routing fluctuation as a key source of instability and proposes stabilizing routing via distillation and freezing. Another line of work improves optimization under non-differentiable top-k routing through differentiable or straight-through relaxations (Wang et al., 2024b; Puigcerver et al., 2023; Zhou et al., 2022). MoE models can be especially brittle under RL fine-tuning due to sparse rewards and high-variance gradients, which further exacerbate routing fluctuations. For RL-specific stabilization, GSPO (Zheng et al., 2025) proposes reusing expert assignments and clipping sequence-level ratios to reduce off-policy variance, and Ring-lite (Team et al., 2025b) regularizes routing through constrained token-level routing budgets.

7 Conclusion

We studied a practical instability in off-policy RL training for Mixture-of-Experts language models. Our diagnosis indicates that router drift across policy updates co-occurs with volatile off-policy mismatch signals and bursty clipping activity, which can culminate in reward collapse. Motivated by this mechanism, we proposed **Router-Shift Policy Optimization (RSPO)**, a lightweight router-aware modification to GRPO-style objectives that computes a per-token router-shift ratio from old activated experts, applies stop-gradient and a lower-bound floor, and rescales importance ratios before clipping/aggregation. This soft adjustment preserves router adaptivity while reducing the impact of tokens with severe routing deviations.

Empirically, router-shift weighting acts as a plug-in stabilizer on Qwen2.5-MoE (Countdown), improving stability for GRPO/GSPO/GMPO, and RSPO (GMPO+RS) yields consistent gains on Qwen3-30B-A3B across both math and code benchmarks. More broadly, our results suggest that explicitly accounting for router dynamics is a key design principle for stable MoE post-training.

Limitations

Our study focuses on stabilizing off-policy RLVR for top-K routed MoE LLMs and has several limitations. First, most of our large-scale evaluations are conducted on the Qwen3-30B-A3B MoE architecture; while we also include small-scale diagnostics on Qwen2.5-MoE, additional evidence on other MoE routing designs (e.g., expert-choice routing) would strengthen generality. Second, we primarily consider verifiable, rule-based rewards for math

and code; the behavior under dense, learned, or preference-based rewards may differ. Third, our router-shift weight relies on cached old top-K routing statistics, which introduces non-zero memory/throughput overhead that may become more noticeable at longer sequence lengths or larger local batch sizes. Finally, we do not conduct a systematic study of interactions between RSPO and explicit MoE load-balancing objectives; we leave this to future work.

Ethical Considerations

All datasets used in this work are publicly available for research purposes, and all evaluated LLMs are accessible either as open-weight models or via public APIs. We do not anticipate major ethical concerns arising from this study.

References

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, and 1 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*.

Damai Dai, Li Dong, Shuming Ma, Bo Zheng, Zhifang Sui, Baobao Chang, and Furu Wei. 2022. Stablemoe: Stable routing strategy for mixture of experts. *arXiv preprint arXiv:2204.08396*.

William Fedus, Barret Zoph, and Noam Shazeer. 2022. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. *Journal of Machine Learning Research*, 23(120):1–39.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

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

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.

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. *Advances in neural information processing systems*, 35:3843–3857.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*.

OpenAI. 2024. Learning to reason with llms.

Joan Puigcerver, Carlos Riquelme, Basil Mustafa, and Neil Houlsby. 2023. From sparse to soft mixtures of experts. *arXiv preprint arXiv:2308.00951*.

Tian Qin, David Alvarez-Melis, Samy Jelassi, and Eran Malach. 2025. To backtrack or not to backtrack: When sequential search limits model reasoning. *arXiv preprint arXiv:2504.07052*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025a. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Ling Team, Bin Hu, Cai Chen, Deng Zhao, Ding Liu, Dingnan Jin, Feng Zhu, Hao Dai, Hongzhi Luan, Jia Guo, and 1 others. 2025b. Ring-lite: Scalable reasoning via c3po-stabilized reinforcement learning for llms. *arXiv preprint arXiv:2506.14731*.

Lean Wang, Huazuo Gao, Chenggang Zhao, Xu Sun, and Damai Dai. 2024a. Auxiliary-loss-free load balancing strategy for mixture-of-experts. *arXiv preprint arXiv:2408.15664*.

Ziteng Wang, Jun Zhu, and Jianfei Chen. 2024b. Remoe: Fully differentiable mixture-of-experts with relu routing. *arXiv preprint arXiv:2412.14711*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, and 1 others. 2025. Geometric-mean policy optimization. *arXiv preprint arXiv:2507.20673*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Yanqi Zhou, Tao Lei, Hanxiao Liu, Nan Du, Yanping Huang, Vincent Zhao, Andrew M Dai, Quoc V Le, James Laudon, and 1 others. 2022. Mixture-of-experts with expert choice routing. *Advances in Neural Information Processing Systems*, 35:7103–7114.

A Experimental Details

A.1 Models and Training Configurations

Small-scale setting (Qwen2.5-MoE, Countdown). The Qwen2.5-MoE model used in the small-scale diagnostic setting contains 12 Transformer layers. Each MoE layer has 8 experts and activates 1 expert per token (top-k=1). The model is pretrained on the Countdown task; the Countdown dataset is generated following the procedure described in Qin et al. (2025). For RL training in this setting, the maximum response length is 8K tokens.

Large-scale setting (Qwen3-30B-A3B, Math/-Code). The Qwen3-30B-A3B model contains 48 Transformer layers (L=48). Each MoE layer has 128 experts and activates 8 experts per token (K=8). For RL training in this setting, the maximum response length is 8K tokens.

Batch sizes and rollout group size. Across both settings, we use rollout group size G=8. For the small-scale setting, the global training batch size is 256 with mini-batch size 64. For the large-scale setting, the global training batch size is 128 with mini-batch size 64.

A.2 Algorithms and Hyperparameters

We use the same algorithmic choices across small and large settings unless otherwise stated. For GRPO, we use symmetric clipping with $\epsilon_{\text{low}} = \epsilon_{\text{high}} = 0.2$. For GMPO and GSPO, we follow the clipping ranges recommended in their original papers: GMPO uses $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (e^{-0.4}, e^{0.4})$, and GSPO uses $(\epsilon_{\text{low}}, \epsilon_{\text{high}}) = (3 \times 10^{-4}, 4 \times 10^{-4})$. For RSPO, we fix the router-shift floor to $\gamma_{\min} = 0.8$ in all experiments and apply stop-gradient through the router-shift weight when used for optimization (Sec. 4).

A.3 Rule-based Reward Verifiers

All tasks use verifiable, rule-based rewards (RLVR) with binary rewards in $\{0, 1\}$. For Countdown, rewards are computed by directly matching the model output to the target format/answer. For math reasoning, we verify final answers using a deterministic math verifier (`math_verify`). For code, we execute generated programs in a sandbox environment against unit tests; reward is 1 if all tests pass (and the program runs successfully), and 0 otherwise.

A.4 Evaluation Protocol Details

Math benchmarks. We follow the Dr.GRPO evaluation protocol and report Pass@1 accuracy. AIME24 results are averaged over 32 repeated evaluations. Decoding is deterministic with temperature $= 0$.

Code benchmarks. We report Pass@1 on MBPP, HumanEval, and LiveCodeBench. For LiveCodeBench, we evaluate using the v4–v5 benchmark suite. For training-time LiveCodeBench data, we use problems released prior to v5 to avoid leakage. Decoding uses temperature $= 0$ and maximum generation length of 32K tokens.

B Stop-Gradient on the Router-Shift Weight

As shown in Figure. 6 on the small Qwen2.5-MoE, backpropagating through γ triggers **early collapse** in both reward and validation curves, whereas the **stop-grad** setting yields smooth and stable optimization. Intuitively, since $\gamma = \exp(-|\Delta \log r|)$ aggregates layer-wise routing drift, letting $\partial \log \gamma / \partial \theta$ flow couples the router-shift penalty with the sequence-level geometric objective and clipping, thereby amplifying variance under non-smooth top-K routing.

C Training Entropy

We report training-time policy entropy as an auxiliary indicator of distribution collapse. A sharp drop in entropy suggests the policy becomes overly deterministic (mode collapse), which often co-occurs

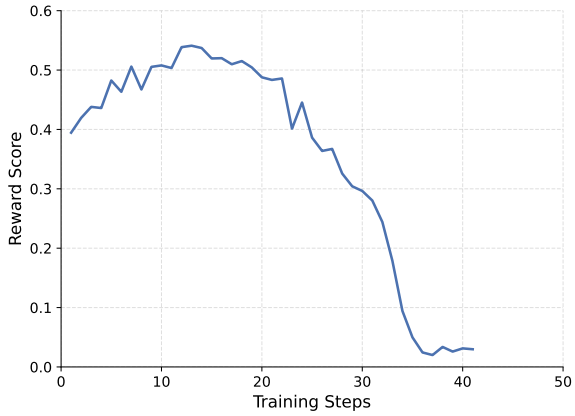

Figure 6: **Backpropagating through the router-shift weight leads to early collapse.** Reward/validation score versus step when gradients are allowed to flow through the router-shift weight. For readability, we plot only the unstable run; in contrast, the default detached setting remains stable throughout training (see main text).

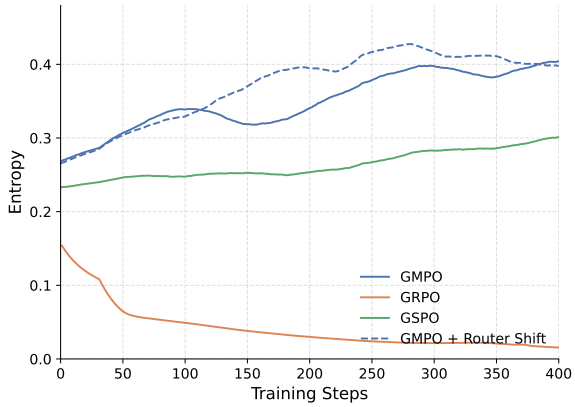

Figure 7: **Training entropy on Qwen3-30B-A3B (math RLVR).** Average token-level policy entropy during training for GRPO, GSPO, GMPO, and RSPO (GMPO+RS). GRPO rapidly collapses to near-zero entropy, while the other methods maintain higher entropy.

with unstable optimization. Fig. 7 shows that GRPO quickly exhibits a dramatic entropy decay, whereas GSPO/GMPO and RSPO maintain substantially higher entropy throughout training, consistent with improved stability.

D Additional Attempts on Router Stabilization

In addition to RSPO, we explored several heuristic strategies that aim to stabilize MoE RL training by *explicitly constraining* router dynamics. This appendix provides implementation details and empirical observations for these alternatives, which complement the brief discussion in Sec. 5.4.

All experiments in this section are conducted in the small-scale diagnostic setting (Qwen2.5-MoE on Countdown) under the same RLVR protocol as Sec. 5.1.

(i) Freezing the router. A straightforward approach is to freeze the router parameters throughout RL training, i.e., keeping ϕ fixed (no router updates) while updating the remaining parameters. This removes router drift by construction, but implicitly assumes that the pretrained routing is already well aligned with the RL objective. In practice, freezing reduces the model's ability to adapt expert allocation to the evolving policy updates and rewards, which can limit performance.

(ii) Routing replay with logit copying (logit replay). Motivated by the routing replay idea discussed in GSPO, we implement a variant that directly reuses the *old* router logits when evaluating the *current* policy during optimization. Concretely, during the update step, we replace the current router logits (or routing scores) with those cached from the old policy ϕ_{old}, so that both expert selection and expert weighting are fully aligned with the old router. A drawback is that the current router is no longer used to compute routing logits, so gradients cannot propagate to the router parameters, effectively preventing router learning.

(iii) Routing replay with expert-index reuse (index replay). As an alternative, we cache only the old top-K expert indices $\{e_{i,t}^{(\ell,k)}\}_{k=1}^{K}$ selected by ϕ_{old} and enforce the current policy to route to these stored indices during the update. Unlike logit replay, the current router still computes its own routing scores on the reused indices, but the discrete expert choices are constrained. This preserves the old routing support while allowing limited router gradients, at the cost of restricting router exploration and potentially introducing mismatch when the optimal routing changes.

Empirical results. Fig. 8 summarizes the training dynamics under these three strategies (router freezing, logit replay, and index replay). Overall, none of the rigid approaches consistently improves stability or final performance compared with our soft router-aware adjustment: freezing limits adaptation, while replay-based variants constrain router learning/exploration and may still exhibit unstable optimization behavior.

Practical considerations. Routing replay requires caching routing traces (logits or indices)

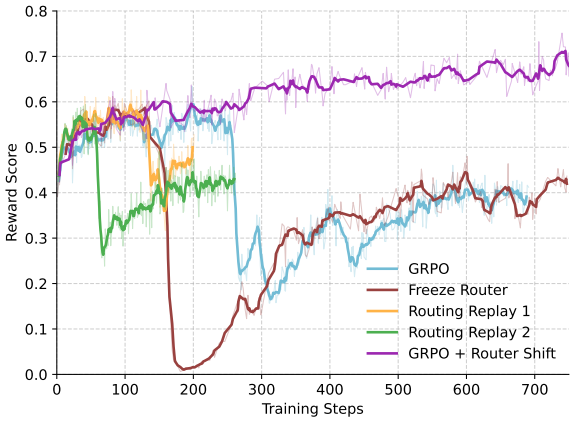

Figure 8: **Alternative router stabilization strategies on Qwen2.5-MoE (Countdown).** Training reward (or validation score) versus training step for (i) freezing the router, (ii) routing replay by copying old router logits (logit replay), and (iii) routing replay by reusing old expert indices (index replay).

across tokens and MoE layers, which can incur non-trivial memory and communication overhead in distributed training. In contrast, RSPO uses only lightweight statistics on old activated experts and applies a detached per-token weight, achieving stabilization without hard constraints.

E Full Objectives of GRPO-Style Baselines

GRPO. Given $x \sim \mathcal{D}$, sample $\{y_i\}_{i=1}^{G} \sim \pi_{\theta_{\text{old}}}(\cdot \mid x)$. The GRPO objective averages the token-level clipped surrogate:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{x,\{y_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\ell_{i,t}^{\text{GRPO}}(\theta)\right], \tag{12}$$

where $w_{i,t}(\theta)$ is defined in Eq. (1) and $\ell_{i,t}^{\text{GRPO}}(\theta)$ is defined in Eq. (2). The group-relative advantage $\hat{A}_i$ is computed by normalizing rewards within the group (mean/std over $\{r(x, y_i)\}_{i=1}^{G}$).

GSPO. GSPO computes the sequence-level ratio $s_i(\theta)$ (Eq. (3)) and applies sequence-level clipping:

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{x,\{y_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\ell_{i}^{\text{GSPO}}(\theta)\right], \tag{13}$$

where $\ell_{i}^{\text{GSPO}}(\theta)$ is defined in Eq. (4).

GMPO. GMPO uses geometric aggregation to form a robust sequence-level ratio, commonly by clipping token-wise ratios before aggregating:

$$\bar{s}_i(\theta) \triangleq \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log \text{clip}\big(w_{i,t}(\theta),\, \epsilon_1,\, \epsilon_2\big)\right), \tag{14}$$

and then optimizes a GRPO-style surrogate using $\bar{s}_i(\theta)$ and $\hat{A}_i$ (the exact clipping bounds follow each method's recommended settings).