Stale but Stable: Staleness-Adaptive Trust Regions for Stabilizing Asynchronous Reinforcement Learning

Junyao Yang[1,2,*], **Yucheng Shi**[1,†], **Zongxia Li**[1,3], **Zhongzhi Li**[1,4], **Ruhan Wang**[1,5], **Xiangxin Zhou**[1], **Kishan Panaganti**[1], **Haitao Mi**[1], **Leowei Liang**[1]

[1]Tencent Hy LLM Frontier [2]National University of Singapore [3]University of Maryland, College Park [4]University of Georgia [5]Indiana University

[†]Project Lead, [*]Corresponding Author

junyaoyang@u.nus.edu, yuchengshi@global.tencent.com

Asynchronous reinforcement learning improves throughput by decoupling rollout generation from optimization, but the resulting staleness is an inevitable byproduct, compounded jointly by policy lag, engine delays, and mixture-of-experts routing. From a trust-region perspective, this mismatch is critical: in the finite-horizon improvement bound, training-inference divergence governs the approximation error, whereas PPO clipping only gates sampled outward updates and therefore acts as a sampled surrogate rather than a full-policy constraint. As a result, the high-staleness update can remain weakly controlled in exactly the asynchronous regime where stale rollouts matter most.

We introduce the **S**taleness-**A**daptive **T**rust Region (**SAT**), which uses the detached sampled log-ratio as a practical staleness proxy, identifies the high-mismatch tail within each batch through **Staleness-based kernel function scaling**, and contracts only the sign-selected endpoint of the nominal PPO interval using **Effective contraction factors**. This design preserves the baseline behavior on ordinary tokens, while making the update more conservative exactly on newly intercepted outward bands. We prove local interval containment and pointwise pessimism relative to PPO, making explicit how the adaptive rule reshapes update geometry under heterogeneous asynchronous staleness.

We evaluate SAT in a fully decoupled asynchronous reinforcement learning setup built on Qwen3-30B-A3B-Base, leveraging SGLang as the inference engine and Megatron as the training pipeline. In this setting, SAT-GSPO w/ R3 attains the best observed AIME24 avg@8, reaching 35.83 at lag 1 and 34.79 at lag 8, while SAT-GSPO reaches 34.17 at lag 1. Adaptive clipping and routing replay act as complementary stabilizers, with the former targeting the mismatch tail and the latter mitigating routing inconsistency. More broadly, the results indicate that aligning the clip interval with observed staleness heterogeneity is an effective way to stabilize the reported asynchronous regime.

Project Page: https://jyyang26.github.io/stable_async_analysis

Date: July 21, 2026

1 Introduction

Asynchronous reinforcement learning serves as a core technique for training large foundation models, accelerating training speeds and maximizing GPU utilization. However, this asynchronous paradigm simultaneously suffers from training-inference mismatch and high staleness, both of which can significantly degrade the resulting model performance. Shown in Figure 1, for the batch-wise asynchronous pipeline in which rollout and optimization run on decoupled resources and weights are broadcast every several trainer steps [12, 20, 21]. We call this interval step *lag*, and the starting observation is that lag is only a coarse systems label. What

the loss actually sees is the staleness between the behavior policy that generated token and the target policy that optimizes it, summarized by the log importance ratio between target and behavior policy, and in modern deployments this mismatch mixes policy-update lag with implementation effects from engine kernels, low-precision numerics, and mixture-of-experts routing. Its distribution is heterogeneous and long-tailed even within a single rollout window. We call this failure mode **Staleness-associated Asynchronous RL instability**: fully illustrated in Section 2.2, as the staleness grows, with training-inference divergence inflates and fixed-clip asynchronous updates become increasingly brittle, ultimately threatening late-stage training stability.

In the finite-horizon policy-improvement bound, the approximation penalty is a cumulative total-variation divergence, and state-wise target-behavior policy divergence is exactly a behavior expectation of the absolute target-behavior policy ratio deviation; a hard ratio bound on *all* actions would therefore control the penalty, with detailed derivation shown in Section 3. PPO clipping imposes no such bound: it is a sampled-objective device that suppresses outward gradients beyond a nominal boundary, preserves pull-back updates, and inspects nothing beyond the one sampled action. Within that surrogate, a fixed radius treats every token as equally reliable with exactly the wrong prior under heterogeneous asynchronous mismatch, where a uniformly small radius starves learning on the bulk of tokens while a uniformly large one grants the high-staleness tail the same outward allowance as reliable data. Yet the clip radius is precisely the handle that decides where outward gradients stop, motivating the central question of the paper: **Can the clip radius itself adapt to the staleness so that asynchronous RL becomes more stable exactly on the high-staleness scenerio?**

From the perspective of the finite-horizon LLM policy-improvement bound, we arrive at a simple design principle: to suppress future training–inference divergence growth, the clip radius should contract precisely on the observed high-staleness tail rather than remain fixed for every token. We therefore propose **S**taleness-**A**daptive **T**rust Region (**SAT**), which adapt the clip contraction to the observed staleness. SAT first treats the detached sampled log-ratio as a practical per-token staleness proxy, identifies the high-staleness tail of each batch through a self-calibrating quantile, and maps unusually large observed staleness to a tighter outward allowance through **Staleness-based kernel function scaling** while leaving ordinary tokens and pull-back updates at the baseline behavior. SAT then contracts only the sign-selected endpoint of PPO's nominal clip interval on those tokens through **Effective contraction factors**, thereby cutting off outward high-staleness updates earlier on the side that would locally enlarge the divergence penalty.

Across extensive asynchronous experiments under identical setting, we find that SAT achieves better performance in high-staleness scenarios, and that the top-performing SAT-GSPO w/ R3 variant also maintains comparatively low observed staleness. The two mechanisms are complementary in the reported runs: routing replay governs the level of the logged mismatch, while SAT changes where divergence gradients stop. This paper makes three contributions:

- **Problem formalization.** We formalize batch-wise asynchronous RL, separate the configured lag from the realized sampled mismatch $d_{b,t}$ and its policy-update and implementation components, document the mismatch-inflation and evaluation-collapse signatures that emerge as lag grows, and make precise, via the finite-horizon policy-improvement bound, what a hard all-action ratio envelope would control and what the sampled clipped objective actually optimizes.
- **Method and analysis.** We introduce SAT, a direction-aware, staleness-adaptive contraction of PPO's nominal clip interval, and characterize its update geometry exactly: the adaptive interval is contained in PPO's, the surrogate is pointwise pessimistic, and the two objectives differ only on one sign-selected outward band per gated token.
- **Asynchronous evaluation.** We run controlled experiments at configured lags 1 and 8, showing that SAT achieves better performance in high-staleness scenarios and that the top-performing SAT variant also maintains comparatively low observed staleness. We position SAT within the asymmetric-clipping family, where it differs from PPO's fixed radius and DPPO's behavior-probability-weighted boundary by setting the active boundary from the observed staleness of the current batch.

2 Async RL Preliminaries and Staleness Symptoms

2.1 Asynchronous RL and realized mismatch

A prompt x is answered by a response $y_b = (y_{b,1}, \ldots, y_{b,T_b})$ sampled autoregressively; the state at position t is $s_{b,t} = (x, y_{b,1}, \ldots, y_{b,t-1})$. Rewards are sparse and sequence-level, $R(y) \in [-\xi, \xi]$, and $\hat{A}_{b,t}$ denotes the advantage estimate, taking the GRPO group as an example. The *behavior policy* $\mu_{b,t}$ is the distribution the rollout engine actually sampled token (b, t) from; the *train policy* $\pi^{(j)}$ is the policy at the current trainer update j. The token-level importance ratio and per-token log-ratio are:

$$r_{b,t} = \frac{\pi^{(j)}(y_{b,t} \mid s_{b,t})}{\mu_{b,t}(y_{b,t} \mid s_{b,t})}, \qquad d_{b,t} \triangleq \log r_{b,t}, \tag{1}$$

and we write $\mathcal{T}_{\text{act}}$ for the active (non-padding) token positions of one batch. The per-state total variation is:

$$D_{\text{TV}}(\mu \,\|\, \pi)[s] = \tfrac{1}{2} \sum_{a} |\mu(a \mid s) - \pi(a \mid s)|.$$

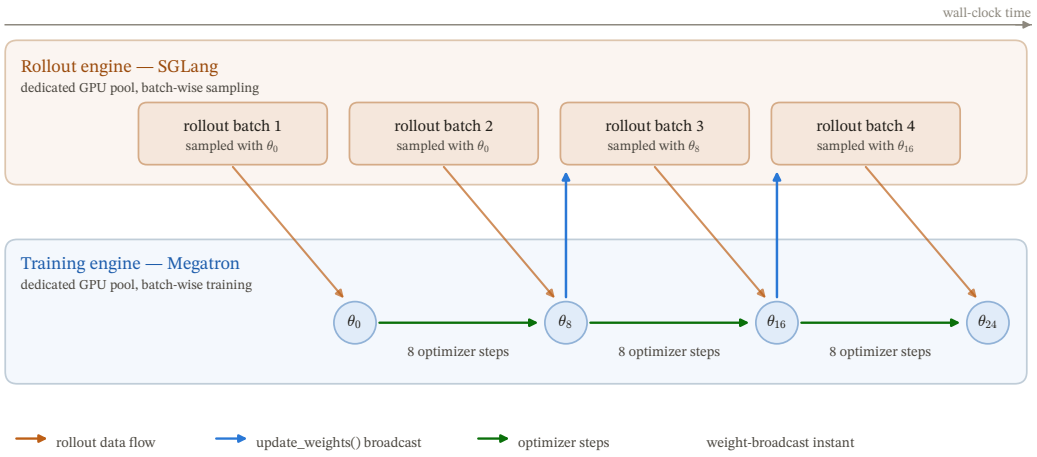


Figure 1 Batch-wise asynchronous RL. Rollout and optimization run on decoupled resources with periodic weight broadcast, so one in-flight batch is typically trained at a configured lag of n policy versions. This is the setting behind $\mu_{b,t}$, $\pi^{(j)}$, and $d_{b,t}$. With full notation table shown in Appendix A, synchronous RL and fully Asynchronous RL workflows can be found in Appendix B.1 and Appendix B.2, respectively.

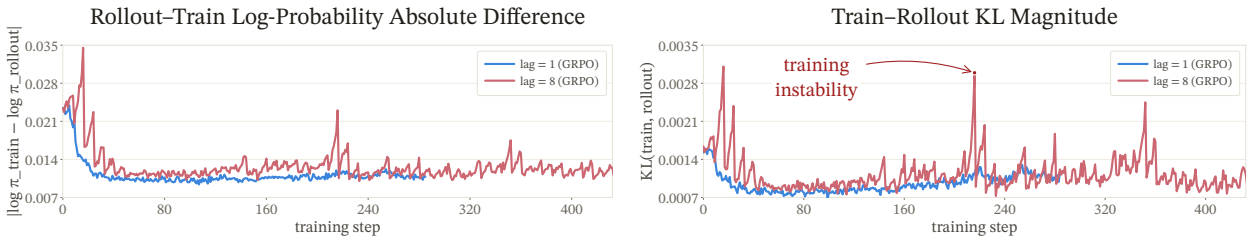
Configured lag versus realized mismatch in batch-wise asynchronous RL. Our experiments use the batch-wise asynchronous regime of Figure 1: weights are broadcast every n trainer steps, and we call n the *configured lag*. Write $d_{b,t} = \log r_{b,t}$ for the observed sampled mismatch, let $\pi^{(\ell)}$ be the trainer-side reference log-probability at the rollout checkpoint, and define $\Delta^{\pi}_{b,t} = \log \pi^{(j)} - \log \pi^{(\ell)}$ and $\Delta^{\text{impl}}_{b,t} = \log \pi^{(\ell)} - \log \mu_{b,t}$ at $(y_{b,t}, s_{b,t})$. Then:

$$d_{b,t} = \log r_{b,t} = \Delta^{\pi}_{b,t} + \Delta^{\text{impl}}_{b,t}, \tag{2}$$

where $\Delta^{\pi}_{b,t}$ is the policy-update mismatch accumulated over the n optimizer steps of the window and $\Delta^{\text{impl}}_{b,t}$ is the implementation mismatch between the trainer-side reference and the actual rollout engine. The first term $\Delta^{\pi}_{b,t}$ is the object PPO's ratio machinery is designed to correct. $\Delta^{\text{impl}}_{b,t}$ can remain nonzero even at identical weights because the two engines differ in kernels, numerics, and MoE routing. Engine, router, and hardware effects are useful *diagnostic hypotheses* inside $\Delta^{\text{impl}}_{b,t}$, but without additional intermediate distributions they do not form a unique additive decomposition; Appendix C discusses them in that spirit. The operational consequence is the one stated in the introduction: the configured lag does not determine the realized distribution of $d_{b,t}$, so any mechanism keyed directly to n addresses only part of the problem.

2.2 Staleness-associated training instability

The blog storyline and the experiments point to the same background fact: once the batch-wise async lag becomes large enough, the main issue is no longer only a looser lower bound but an observable loss of training stability. In the controlled comparisons below, all runs share optimizer, data, reward, and model; only the configured lag differs. We therefore read these figures as background evidence for what async RL must explain, not as a theorem about a unique causal threshold.



(a) Training–inference mismatch $\bar{d}_\pi$. **(b)** Train–rollout KL.

Figure 2 Background instability diagnostics under a variance method at configured lag 1 and 8. This figure tracks two batch-averaged mismatch diagnostics of a variance method across training under two configured lag settings, with the two curves in each panel distinguishing the mild-lag and high-lag runs under an otherwise identical setup. The panels also mark where the high-lag run first leaves its low-mismatch baseline band, together with the numerical range covered by each axis.

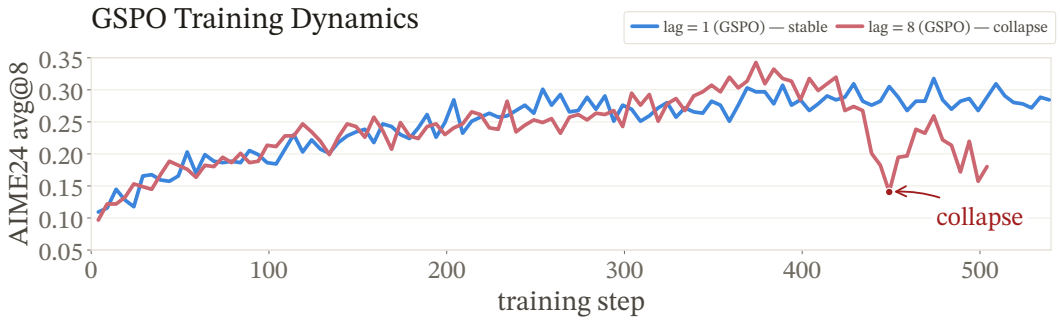


Figure 3 Background evaluation collapse under a variance method at configured lag 1 and 8. This figure plots the evaluation benchmark score of a variance method across training under two configured lag settings, where the collapse step of the high-lag run is explicitly annotated with a point and label. The panel is used to align the timing of evaluation collapse with the mismatch signals of Figure 2.

Two instability symptoms recur under high-lag asynchronous training. Figure 2 and Figure 3 shows how batch-wise asynchronous training deteriorate as the lag grows, from which three observations stand out.

- **Mismatch inflation under increased staleness.** Mismatch $\bar{d}_\pi$ and its KL counterpart stay in a narrow band at lag 1 but develop heavy-tailed spikes at lag 8, where $\bar{d}_\pi$ exceeds 0.03 and the KL diagnostic exceeds 2.8×10^{-3}, indicating a widening of the ratio distribution rather than a shift of its center.
- **Training collapse due to increase of staleness.** The lag-1 run tracks the evaluation-benchmark band near 0.30, whereas the lag-8 run peaks near 0.34 before falling to about 0.14 by step 449.

3 LLM Policy Improvement and Ratio Control in Async RL

3.1 Finite-horizon LLM policy-improvement surrogate

Consider an undiscounted, finite-horizon generation problem with $|R(y)| \le \xi$, behavior policy μ, and target policy π, and define $\mathcal{J}(\pi) = \mathbb{E}_{y\sim\pi}[R(y)]$. We assume trajectory support compatibility: at every μ-reachable

state used below, $\pi(\cdot \mid s) \ll \mu(\cdot \mid s)$. When necessary, an end-of-sequence action and padding make the horizon fixed. Following the Kakade–Langford line of analysis [4, 9], define the linear surrogate:

$$L'_\mu(\pi) \;=\; \mathbb{E}_{y\sim\mu}\left[R(y)\sum_{t=1}^{T}\Big(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\Big)\right]. \tag{3}$$

A representative finite-horizon lower bound [8] is:

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \;\geq\; L'_\mu(\pi) \;- 4\xi\,\mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{T} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t)\,\|\,\pi(\cdot \mid s_t)\big)\right]. \tag{4}$$

The penalty is an expected *cumulative* token-level divergence, not a length-normalized average. Equation (4) is the policy-improvement statement relevant for the rest of the paper: smaller statewise divergence makes the approximation term less pessimistic, but it does not by itself imply that an arbitrary smaller update is automatically better. Non-decrease still depends on the surrogate gain dominating the divergence penalty.

In a genuinely asynchronous system the behavior conditional can vary with the realized scheduler and policy version. Equation (4) should then be read conditional on that schedule, or after augmenting the state with the relevant scheduler/version information so that the trajectory behavior factorizes through the stated conditionals.

3.2 Ratio envelopes and the sampled clip in async RL

The relation between ratio control and Section 3.1's policy-improvement surrogate is exact but conditional on support compatibility.

Lemma 3.1 (**Ratio deviation as a statewise divergence identity**). *Fix a state s and suppose $\pi(\cdot \mid s) \ll \mu(\cdot \mid s)$. Then, with $r_s(a) = \pi(a \mid s)/\mu(a \mid s)$,*

$$D_{\mathrm{TV}}\big(\mu(\cdot \mid s)\,\|\,\pi(\cdot \mid s)\big) \;=\; \frac{1}{2}\,\mathbb{E}_{a\sim\mu(\cdot|s)}\big|r_s(a) - 1\big|. \tag{5}$$

Proof. $\frac{1}{2}\sum_a |\pi(a \mid s) - \mu(a \mid s)| = \frac{1}{2}\sum_a \mu(a \mid s)\,|r_s(a) - 1|$; absolute continuity rules out unaccounted target mass on actions with $\mu(a \mid s) = 0$.

Full-vocabulary softmax sampling satisfies the support condition in exact arithmetic—our experiments disable nucleus truncation for precisely this reason (Section 5.1). Top-k or top-p truncation can violate it and requires separate treatment (Appendix E).

Equation (5) is the step that connects a hard ratio envelope to the policy-improvement bound of Equation (4). If, at a fixed state, every action satisfies a hard ratio bound, then the divergence term in Equation (4) is correspondingly controlled:

Proposition 3.2 (**Ideal all-action ratio region**). Under the conditions of Lemma 3.1, if:

$$|r_s(a) - 1| \;\leq\; \varepsilon \qquad \text{for } \mu\text{-almost every action } a, \tag{6}$$

then $D_{\mathrm{TV}}(\mu\,\|\,\pi)[s] \leq \varepsilon/2$. More generally, if $1 - \varepsilon_{\mathrm{low}}(a) \leq r_s(a) \leq 1 + \varepsilon_{\mathrm{high}}(a)$ for every action, then:

$$D_{\mathrm{TV}}(\mu\,\|\,\pi)[s] \;\leq\; \frac{1}{2}\,\mathbb{E}_{a\sim\mu}\max\big\{\varepsilon_{\mathrm{low}}(a),\,\varepsilon_{\mathrm{high}}(a)\big\}, \tag{7}$$

which for constant asymmetric radii is at most $\frac{1}{2}\max\{\varepsilon_{\mathrm{low}}, \varepsilon_{\mathrm{high}}\}$.

Proof. Each assumption gives the corresponding pointwise bound on $|r_s(a) - 1|$; substitution into Equation (5) proves each claim.

This proposition describes an *ideal all-action feasible set*. The quantifier is essential: a bound on the one sampled action says nothing about the rest of the vocabulary distribution. That gap is exactly why sampled clipping in async RL should be read as a surrogate motivated by the hard-envelope argument above, not as the hard envelope itself.

4 SAT: From Heterogeneous Mismatch to an Adaptive Clip

4.1 From Policy Improvement to a Staleness-Adaptive Radius

Section 3 ends one step short of a method: shown in Lemma 3.1, statewise divergence controls the finite-horizon approximation term, that divergence is exactly a behavior expectation of absolute ratio deviation, and a hard all-action ratio envelope would therefore control the penalty (Proposition 3.2)—but nothing yet says what the *sampled* objective should do when the envelope is unavailable. To make the target of that chain explicit, we name once the two LLM-regime divergence quantities:

$$\begin{aligned} D_{\mathrm{TV}}^{\max}(\mu\|\pi) &\triangleq \max_{s_t} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t)\|\pi(\cdot \mid s_t)\big), \\ D_{\mathrm{TV}}(\mu,\pi) &\triangleq \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{|y|} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t)\|\pi(\cdot \mid s_t)\big)\right]. \end{aligned} \tag{8}$$

In finite-horizon LLM setting with horizon T, $\gamma = 1$, and $\xi = \max_y |R(y)|$, the improvement gap admits both:

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \geq L'_\mu(\pi) - 2\xi\, T(T-1)\left(D_{\mathrm{TV}}^{\max}(\mu\|\pi)\right)^2, \tag{9}$$

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \geq L'_\mu(\pi) - 4\xi\, D_{\mathrm{TV}}(\mu,\pi). \tag{10}$$

The first is the finite-horizon analogue of a max-divergence trust-region bound; the second restates Equation (4) in the named notation and is the cumulative token-level form that a token-wise sampled objective can actually influence on long responses. What PPO optimizes, however, is neither bound but the sampled clipped surrogate:

$$\mathcal{S}_{b,t}^{\mathrm{PPO}} = \min\Big(r_{b,t}\hat{A}_{b,t},\ \bar{r}_{b,t}^{\mathrm{PPO}}\hat{A}_{b,t}\Big), \qquad \bar{r}_{b,t}^{\mathrm{PPO}} = \mathrm{clip}\big(r_{b,t},\, 1-\varepsilon_{\mathrm{low}},\, 1+\varepsilon_{\mathrm{high}}\big), \tag{11}$$

whose only guardrail is the fixed pair $(\varepsilon_{\mathrm{low}}, \varepsilon_{\mathrm{high}})$: the clipped branch suppresses outward gradients beyond the nominal boundary, preserves pull-back updates, and inspects nothing beyond the one sampled action.

Why a fixed radius is an imperfect default. A fixed ε treats every sampled token as equally reliable. In the batch-wise asynchronous pipeline of Section 2, however, the realized mismatch $d_{b,t}$ mixes policy-update and implementation terms (Equation (2)) and is heterogeneous within a single rollout window: most tokens remain close to the behavior policy, while a small tail is affected by policy lag, training–inference engine differences, router choices, or numerical effects. A uniformly tiny radius therefore suppresses useful learning on the bulk of the data, while a uniformly large radius grants the high-staleness tail the same outward allowance as reliable tokens. Read against Equations (9)–(10), both defaults miss the point: the divergence penalty is driven by where the mismatch is large, and no single global radius can be simultaneously permissive on the bulk and conservative on the tail.

A sampled proxy, not a divergence estimate. The observable that exposes this tail is the sampled log-ratio $d_{b,t} = \log r_{b,t}$, which is already computed in off-policy training. It must not, however, be interpreted as D_{TV}: the sampled action's contribution to the D_{TV} sum carries a behavior-probability weight,

$$\tfrac{1}{2}\,\mu(a \mid s)\,\big|r(a \mid s) - 1\big| \;=\; \tfrac{1}{2}\,\big|\pi(a \mid s) - \mu(a \mid s)\big|, \tag{12}$$

so a rare action can have an enormous ratio while moving little probability mass. As detailed in Section 6, its use as a per-token risk score is an empirical design choice, and probability-mass-weighted alternatives are natural follow-ups.

Design of SAT. These observations fix what an adaptive rule may claim and how it should act. It cannot impose a hard all-action trust region, so it must operate at the same surrogate level as Equation (11). Specifically, SAT contracts PPO's nominal interval only on tokens whose observed mismatch is unusually large *for the current batch*, and only on the side that moves the sampled ratio farther from one—the direction that locally enlarges the divergence penalty. Consequently, it leaves ordinary tokens and pull-back updates untouched, reducing exactly to the baseline objective when disabled.

4.2 Staleness-Adaptive Trust Region

SAT targets a simple but useful goal, which reshapes the sampled PPO surrogate so that tokens with unusually large observed mismatch receive a tighter outward allowance, while ordinary tokens and pull-back updates retain the baseline behavior. The whole method is defined by the sequence of formulas below.

Detached sampled token-level staleness. SAT begins from the sampled mismatch signal already available in asynchronous training and treats the detached token-level log-ratio as a practical staleness proxy for the current sample:

$$d_{b,t} = \log r_{b,t}, \qquad d_{b,t}^{\mathrm{sg}} \triangleq \mathrm{sg}[d_{b,t}], \tag{13}$$

where sg denotes stop-gradient. This detached score is not itself optimized; it only decides how conservative the clip interval should be on the current sample.

Staleness-based kernel function scaling. Rather than compare $|d_{b,t}|$ with a fixed global threshold, SAT calibrates the high-staleness tail relative to the current batch through the detached empirical quantile:

$$\widehat{F}_{|d|}(x) = \frac{1}{|\mathcal{T}_{\mathrm{act}}|} \sum_{(b,t)\in\mathcal{T}_{\mathrm{act}}} \mathbf{1}\{|d_{b,t}| \le x\}, \qquad q = \mathrm{sg}\Big[\inf\{x \ge 0 : \widehat{F}_{|d|}(x) \ge \alpha\}\Big]. \tag{14}$$

The role of q is purely referential: it marks where the current batch begins to enter its mismatch tail, where α denotes the target quantile threshold, which set to 0.90 in our implementation. Because q is recomputed from the same active tokens being optimized, the method adapts to rescaling of the mismatch profile without introducing another hand-tuned absolute cutoff. SAT then maps the sign-separated mismatch magnitudes through a monotone kernel:

$$\psi(u;q) = \frac{1}{1+(u/q)^2}, \qquad u_{+,b,t} = [d_{b,t}^{\mathrm{sg}}]_+, \quad u_{-,b,t} = [-d_{b,t}^{\mathrm{sg}}]_+, \tag{15}$$

where $\psi(0;q) = 1$, $\psi(q;q) = \frac{1}{2}$, and $\partial\psi/\partial u \le 0$. The sign split is the key structural choice: when the sampled ratio is already above 1, the adaptive rule should only tighten the upper side of the interval; when it is below 1, it should only tighten the lower side. Otherwise the method would also suppress pull-back moves that already head back toward the behavior policy.

Effective contraction factors. The contraction is applied only on the high-staleness tail:

$$c_{\pm,b,t} = \begin{cases} \psi(u_{\pm,b,t};q), & q > 0 \text{ and } |d_{b,t}^{\mathrm{sg}}| > q, \\ 1, & \text{otherwise}, \end{cases} \tag{16}$$

which yields the adaptive radii:

$$\tilde{\varepsilon}_{\mathrm{low},b,t} = \varepsilon_{\mathrm{low}}\, c_{-,b,t}, \qquad \tilde{\varepsilon}_{\mathrm{high},b,t} = \varepsilon_{\mathrm{high}}\, c_{+,b,t}. \tag{17}$$

Since $0 < c_{\pm,b,t} \le 1$, these radii never expand the original PPO interval. Ungated tokens recover the baseline radii exactly, while gated positive-mismatch tokens only shrink the upper side and gated negative-mismatch tokens only shrink the lower side.

- **SAT core mechanism.** SAT is conservative only where the sampled mismatch is unusually large and only on the side that would move the sampled ratio farther away from one. The explicit $q > 0$ branch

avoids division by zero when most mismatch values vanish, and under the inverse-CDF convention the strict gate $|d| > q$ selects at most the top decile of active positions.

SAT sampled objective. SAT then obtains the final sampled surrogate by dropping the adaptive radii into the original clipped objective:

$$\bar{r}_{b,t}^{\mathrm{SAT}} = \mathrm{clip}\big(r_{b,t},\, 1-\tilde{\varepsilon}_{\mathrm{low},b,t},\, 1+\tilde{\varepsilon}_{\mathrm{high},b,t}\big), \tag{18}$$

$$\mathcal{S}_{b,t}^{\mathrm{SAT}} = \min\big(r_{b,t}\hat{A}_{b,t},\ \bar{r}_{b,t}^{\mathrm{SAT}}\hat{A}_{b,t}\big). \tag{19}$$

This substitution shows that SAT is not a new objective family but a drop-in replacement for PPO clipping: it replaces $\bar{r}_{b,t}^{\mathrm{PPO}}$ by $\bar{r}_{b,t}^{\mathrm{SAT}}$ inside the same sampled surrogate. When $c_{\pm,b,t} = 1$, the underlying base objective is recovered exactly. Because the sign-selected endpoint moves inward, the adaptive interval remains contained in PPO's nominal interval, while the untouched side matches PPO. Pull-back branches therefore stay unchanged, and only the outward band between the adaptive and nominal boundaries is newly clipped. The extra cost is one quantile plus a few elementwise tensor operations per batch.

Sequence-level objectives. Equations (13)–(19) describe the token-ratio implementation. For the reported GSPO extension, the scalar in the clipped objective and in the SAT score is instead the length-normalized sequence ratio:

$$\rho_b^{\mathrm{seq}} = \exp\Big(\frac{1}{T_b}\sum_{t=1}^{T_b}\log r_{b,t}^{\mathrm{tok}}\Big), \qquad r_{b,t}^{\mathrm{tok}} = \frac{\pi^{(j)}(y_{b,t} \mid s_{b,t})}{\mu_{b,t}(y_{b,t} \mid s_{b,t})}, \tag{20}$$

broadcast to the response's active token positions before the same active-token quantile and objective aggregation are applied. The empirical quantile therefore length-weights sequences, and every token of one response receives the same sequence-ratio score. The interval and scalar-surrogate claims below apply verbatim after this substitution; the token-level D_{TV} interpretation does not, because token log-ratios can cancel inside ρ_b^{seq} (Section 6).

4.3 Update Geometry of SAT

The construction of Section 4.2 admits an exact local characterization, at the same surrogate level as Section 3.2: SAT does not impose a hard all-action trust region; its entire effect on the sampled objective is confined to one sign-selected outward band per gated token. The following proposition collects the interval geometry, the pointwise ordering, and the exact locus of change relative to Equation (11).

Proposition 4.1 (**Update geometry of SAT**). Fix a sampled token with ratio r, advantage $\hat{A}$, and contraction factors $c_{\pm,b,t}$ from Equation (16), and define $\mathcal{I}^{\mathrm{PPO}} = [1-\varepsilon_{\mathrm{low}},\, 1+\varepsilon_{\mathrm{high}}]$ and $\mathcal{I}_{b,t}^{\mathrm{SAT}} = [1-\varepsilon_{\mathrm{low}}c_{-,b,t},\, 1+\varepsilon_{\mathrm{high}}c_{+,b,t}]$ for positive base radii. Then: (i) *Containment:* $\mathcal{I}_{b,t}^{\mathrm{SAT}} \subseteq \mathcal{I}^{\mathrm{PPO}}$; the intervals coincide when the gate is off, and when it fires, the endpoint selected by $\mathrm{sign}(d_{b,t})$ moves strictly inward while the opposite endpoint is unchanged. (ii) *Pointwise pessimism:* $\mathcal{S}^{\mathrm{SAT}}(r,\hat{A}) \le \mathcal{S}^{\mathrm{PPO}}(r,\hat{A})$. (iii) *Exact locus of change:* holding the stopped clip limits fixed and away from the clipping kinks, the derivatives with respect to r differ exactly on the newly clipped outward bands

$$\{\hat{A}>0,\ 1+\varepsilon_{\mathrm{high}}c_{+,b,t} < r < 1+\varepsilon_{\mathrm{high}}\} \ \vee\ \{\hat{A}<0,\ 1-\varepsilon_{\mathrm{low}} < r < 1-\varepsilon_{\mathrm{low}}c_{-,b,t}\}, \tag{21}$$

where SAT has zero derivative and PPO retains its outward derivative $\hat{A}$. Pull-back branches are unchanged, and ratios already outward-clipped by PPO have zero derivative under both objectives.

Proof. (i) Equation (16) gives $0 < c_{\pm,b,t} \le 1$, so multiplying a positive base radius by $c_{\pm,b,t}$ can only move the corresponding boundary toward 1 or leave it fixed; strictness on the gated side follows from $\psi(u;q) < 1$ at $u > q > 0$. (ii)–(iii) For $\hat{A} > 0$, compare $\mathcal{S}^{\mathrm{PPO}} = \hat{A}\min\{r, 1+\varepsilon_{\mathrm{high}}\}$ with $\mathcal{S}^{\mathrm{SAT}} = \hat{A}\min\{r, 1+\varepsilon_{\mathrm{high}}c_{+,b,t}\}$: since $c_{+,b,t} \le 1$, the adaptive boundary is no larger, which gives the ordering, and the two functions and their derivatives differ exactly on $1+\varepsilon_{\mathrm{high}}c_{+,b,t} < r < 1+\varepsilon_{\mathrm{high}}$. The case $\hat{A} < 0$ is symmetric with max and the lower boundary, confining the difference to $1-\varepsilon_{\mathrm{low}} < r < 1-\varepsilon_{\mathrm{low}}c_{-,b,t}$, and $\hat{A} = 0$ is immediate.

Remark: local mechanism and its empirical signature. The mechanism of SAT operates locally at the surrogate level by modulating the gradient contributions of individual positions within each batch. Rather than shifting or lowering the average logged mismatch band, its predicted empirical signature is tail suppression, which is characterized by fewer runaway outward updates and the absence of late-stage collapse. By selectively adjusting these extreme deviations, SAT ensures that the optimization curves safely track their base recipes without undergoing catastrophic policy degradation. The comprehensive theoretical foundation of this mechanism, including all formal proofs and empirical monitoring quantities, is fully detailed in Appendix G.

5 Experiments

5.1 Setup

Pipeline and model. All experiments run on the slime batch-wise asynchronous RL framework [21] with SGLang [20] as the rollout engine and Megatron [12] as the trainer, on fully decoupled GPU pools (Figure 1). The backbone is Qwen3-30B-A3B-Base [14], an MoE model chosen deliberately: it activates the policy-update *and* implementation mismatch terms of Equation (2), including MoE routing. The configured lag is controlled at $n \in \{1, 8\}$ trainer steps per weight broadcast, so the experimental comparisons in the main text are always within this batch-wise async regime rather than across synchronization regimes.

Training data, batching, and sampling. Training prompts total 30,712 by combining DAPO-Math-17k [16] and Dolci-RL-Zero-Math-7B [7] and shuffled rollout. Each iteration consumes 256 prompts with 16 samples each, yielding 4096 responses for a GRPO group size of 16 across 544 total rollout iterations. Rollouts are sampled at temperature 0.95, top-p of 1.0, and top-k of -1 to retain full vocabulary support as required by Lemma 3.1 under a 32k-token response limit. The reward is based on the rule-based verifier. And clip radii are symmetric at 0.2 with a constant learning rate of 10^{-6}. SAT uses the hill kernel with exponent 2 with reference quantile 0.90. Appendix F lists full hyper-parameters, hardware, and parallelism, while metrics and statistical scope are deferred to Appendix F.

5.2 Main-result

Baselines. We evaluate combinations of **GRPO** [11] and **GSPO** [19] across **vanilla**, **R3** [5], **TIS** [18], and **combined stabilizer** configurations. This grid is augmented by four SAT variants and **DPPO** [8], specifically the variant using D_{TV} to denote total variation divergence. Appendix E unifies these methods within one design space. All configurations share identical optimization, data, and rewards, varying only by stabilizer and configured lag. Appendix F tabulates all reproducible hyperparameters.

SAT achieves better performance in high staleness scenerio. Table 1 shows that **SAT-GSPO w/ R3 ranks first at both lag settings, reaching** 35.83 **at lag** 1 **and** 34.79 **at lag** 8. The lead persists as the configured lag increases, indicating that the advantage is not confined to the milder asynchronous regime. Relative to GRPO and GSPO, the gains are 4.58 and 3.58 points at lag 1, and 4.62 and 3.33 points at lag 8; relative to DPPO, the margins are 1.87 and 2.08 points. Overall, the combination of SAT, GSPO, and R3 is the strongest configuration in the reported grid, including the more demanding lag-8 setting.

SAT variant with top performance also maintains comparatively low observed staleness. The $\bar{d}_\pi \downarrow$ columns show that sampled mismatch rises with configured lag for all methods, but the better-performing non-collapsed configurations occupy a lower mismatch band than the plain baselines. In particular, **SAT-GSPO w/ R3 records** $\bar{d}_\pi = 0.0056$ **and** 0.0076**, compared with** 0.0097 **and** 0.0109 **for GSPO**. Furthermore, we observe that under the vanilla configuration, both GRPO and GSPO exhibit training collapse respectively at 429 and 424 steps in high-staleness regimes. This highlights the severe risk of accelerating throughput, demonstrating its critical threat to training stability within reinforcement learning frameworks.

Table 1 Main evaluation results. This table presents the main performance eval results and the last-epoch sampled mismatch dpi, where parenthesized values mark the best checkpoint steps. The table also notes the deep red entries that later experience collapse, with **bold** and underline mark the best and runner-up within each metric column.

Method	lag = 1	lag = 8	lag = 1	lag = 8
-	AIME24 ↑		$\bar{d}_\pi$ ↓	
Qwen3-30B-A3B-Base	9.38	9.38	—	—
GRPO [11]	31.25 (429)	30.17 (429 collapse)	0.0103	0.0110
GSPO [19]	32.25 (474)	31.46 (424 collapse)	0.0097	0.0109
GRPO w/ R3	31.83 (314)	29.58 (214)	0.0058	0.0218
GSPO w/ R3	34.00 (379)	<u>33.13</u> (214)	0.0091	0.0158
GRPO w/ TIS	31.62 (254)	31.46 (374)	0.0108	0.0169
GSPO w/ TIS	32.88 (319)	31.21 (279)	0.0151	0.0208
GRPO w/ TIS w/ R3	31.79 (324)	31.79 (239)	**0.0048**	0.0081
GSPO w/ TIS w/ R3	31.88 (329)	32.92 (359)	<u>0.0054</u>	**0.0068**
DPPO [8]	33.96 (324)	32.71 (289)	0.0108	0.0118
SAT-GRPO	32.71 (404)	32.08 (299)	0.0101	0.0108
SAT-GSPO	<u>34.17</u> (254)	32.71 (359)	0.0095	0.0107
SAT-GRPO w/ R3	33.75 (284)	<u>33.13</u> (289)	0.0056	0.0079
SAT-GSPO w/ R3	**35.83** (379)	**34.79** (359)	0.0056	<u>0.0076</u>

5.3 Training–inference mismatch trajectories over training

Routing replay stabilizes the log prob difference while SAT thrives at lower values. Figure 4 tracks the mismatch diagnostic across variance methods behind Table 1. The integration of routing routing replay (R3) is essential for stabilization: methods utilizing R3 restrict the log prob difference to a tight fluctuation around 0.007. Conversely, without R3, the metric deviates noticeably, drifting to around 0.011. Second, in the later stages of GRPO training, omitting R3 results in a sudden, sharp growth in the log prob difference. Moreover, **SAT achieves better performance while maintaining a lower log prob difference** without training collapse, proving its robustness in managing training dynamics without relying on excessive policy divergence.

5.4 Asymmetric clipping in SAT

The benchmark table and the mismatch trajectories suggest that the stabilizers differ not only in strength but also in update geometry. A useful organizing principle is the asymmetric clipping family, which includes PPO, DPPO, SPO, DRPO, and SAT. For a fixed sampled token with ratio r_t and advantage $\hat{A}_t$, the unclipped surrogate has derivative $\hat{A}_t$ with respect to r_t, so the sign of $\hat{A}_t$ determines the locally preferred direction. Relative to $r_t = 1$, the sampled update is locally diverging when $\operatorname{sign}(\hat{A}_t(r_t - 1)) > 0$ and locally converging when $\operatorname{sign}(\hat{A}_t(r_t - 1)) < 0$. PPO therefore keeps the gate:

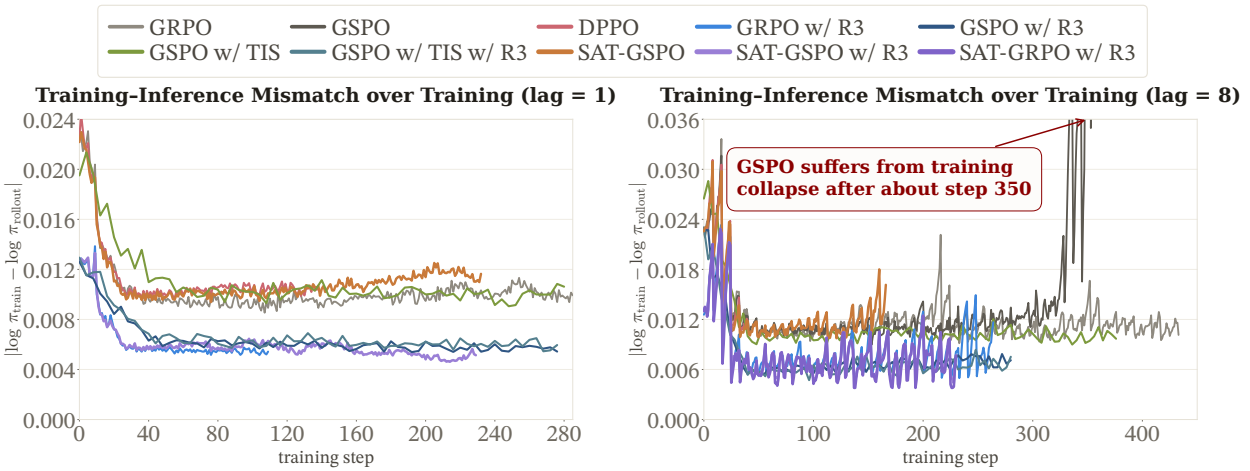
$$M_t^{\mathrm{PPO}} = \mathbf{1}\Big\{\operatorname{sign}(\hat{A}_t(r_t - 1)) \le 0 \ \vee \ |r_t - 1| \le \varepsilon\Big\}, \tag{22}$$

which clips only those updates that would move the sampled ratio farther away from one and preserves the updates that move it back toward the behavior reference.

DPPO retains the directional logic but updates the discriminator. The sampled total variation distance D_t^{TV} bounds the ratio deviation $|r_t - 1|$, logically defining the induced gate M_t^{DPPO}:

$$\begin{aligned} D_t^{\mathrm{TV}} &= \left|\pi^{(j)}(y_t \mid s_t) - \mu(y_t \mid s_t)\right| = \mu(y_t \mid s_t)\,|r_t - 1|, \\ |r_t - 1| &\le \frac{\delta}{\mu(y_t \mid s_t)}, \\ M_t^{\mathrm{DPPO}} &= \mathbf{1}\Big\{\operatorname{sign}(\hat{A}_t(r_t - 1)) \le 0 \ \vee \ |r_t - 1| \le \delta/\mu(y_t \mid s_t)\Big\}. \end{aligned} \tag{23}$$

This gating mechanism imposes a more stringent constraint on high-probability head tokens, while granting a



(a) configured lag = 1 **(b)** configured lag = 8

Figure 4 Training–inference mismatch $\bar{d}_\pi$ over training at configured lag 1 and 8. This figure tracks the batch-averaged sampled mismatch of variance methods across training, with a shared legend distinguishing configurations and separate panels contrasting the mild-lag setting with the high-lag setting.

more permissive update allowance to rare, low-probability tokens residing in the long tail of the distribution.

The results in Table 1 and Figure 5 support this view. DPPO reaches 33.96 at lag 1 and 32.71 at lag 8, outperforming the GRPO baseline at both lags and avoiding the later collapse seen in GRPO and GSPO. This shows that an asymmetric gate with a token-adaptive D_{TV} threshold can already improve stability in the reported async regime, although DPPO still trails SAT-GSPO at lag 1 and SAT-GSPO w/ R3 at both lags.

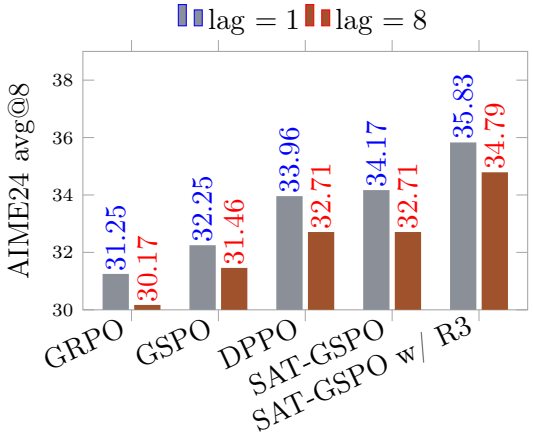


Figure 5 AIME24 avg@8 for GRPO, GSPO, DPPO, and SAT variants.

Together with Figure 4, the comparison also shows that asymmetry alone is not sufficient: the rule used to set the effective boundary matters. DPPO and SAT both preserve pull-back updates and intercept outward ones, but SAT chooses the active boundary from the observed mismatch tail of the current batch. Equations (18) and (19), together with Propositions 4.1 and 4.1, formalize this point: SAT maintains the asymmetric attribute while cutting off only the newly intercepted outward band. A broader method-by-method comparison is deferred to Appendix E and Appendices E.1–E.6.

6 Limitations and Open Questions

The current scope is intentionally narrow, and the main limitations are:

- **Surrogate-level control.** SAT contracts a sampled nominal interval rather than the realized policy itself. The learned policy can still move outside that interval, unobserved vocabulary actions remain unconstrained, and the proxy $|\log r|$ should not be identified with either D_{TV} or version age because it omits the probability-mass weight in Equation (12) and mixes policy lag with implementation mismatch.
- **Relative adaptation.** Because the reference quantile q is recomputed for every batch, the method responds to the current mismatch tail rather than to an absolute scale. It therefore does not imply monotone shrinkage with configured lag once PPO's original boundary is already active.

7 Conclusion

The trust-region interpretation of SAT is narrow but clear. D_{TV} controls a finite-horizon approximation term, and D_{TV} is exactly a behavior expectation of absolute ratio deviation, so a hard all-action ratio bound would control D_{TV}. PPO does not impose such a bound and instead clips an advantage-dependent sampled surrogate. SAT operates at that same surrogate level by using a detached mismatch proxy to contract the sign-selected nominal radius on the high-staleness tail of each batch. Its effect is therefore local but precise: it cuts off newly intercepted outward updates earlier while preserving pull-back gradients, and it reduces exactly to the underlying recipe when disabled. On the reported grid, SAT-GSPO w/ R3 attains the best-observed AIME24 avg@8 at both configured lags, reaches 35.83 at lag 1 and 34.79 at lag 8, and avoids the late collapse seen in several fixed-clip lag-8 baselines. Together with the interval-containment and pointwise-pessimism results, these findings give a concrete account of what an adaptive ε can contribute in asynchronous RL.

References

[1] Chris Dong, Aman Goel, and Dian Yu. Staleness in fully asynchronous RL. https://appliedcompute.com/research/staleness-in-fully-async-rl, 2026. Applied Compute Research.

[2] Jiale Guo, Yuting Sun, Zhen Huang, Zewen Wang, Zhuoyi Wen, Zhen Zhang, Jin Zhou, and Stanley Kok. KPop: Taming training-inference mismatch in reinforcement learning with adaptive masking regions. https://ringtech.notion.site/kpop, 2026.

[3] Zhenyu Hou, Yilin Li, Jie Tang, and Yuxiao Dong. Single-rollout asynchronous optimization for agentic reinforcement learning. *arXiv preprint arXiv:2607.07508*, 2026.

[4] Sham Kakade and John Langford. Approximately optimal approximate reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML)*, 2002.

[5] Wenhan Ma, Hailin Zhang, Liang Zhao, Yujiao Song, Yiyuan Wang, Zhifang Sui, and Fuli Luo. Stabilizing MoE reinforcement learning by aligning training and inference routers. *arXiv preprint arXiv:2510.11370*, 2025.

[6] Microsoft AI Team. MAI-Thinking-1: Building a hill-climbing machine. https://microsoft.ai/pdf/mai-thinking-1.pdf, 2025.

[7] Team Olmo, Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, Jake Poznanski, Kyle Lo, Luca Soldaini, Matt Jordan, Mayee Chen, Michael Noukhovitch, Nathan Lambert, Pete Walsh, Pradeep Dasigi, Robert Berry, Saumya Malik, Saurabh Shah, Scott Geng, Shane Arora, Shashank Gupta, Taira Anderson, Teng Xiao, Tyler Murray, Tyler Romero, Victoria Graf, Akari Asai, Akshita Bhagia, Alexander Wettig, Alisa Liu, Aman Rangapur, Chloe Anastasiades, Costa Huang, Dustin Schwenk, Harsh Trivedi, Ian Magnusson, Jaron Lochner, Jiacheng Liu, Lester James V. Miranda, Maarten Sap, Malia Morgan, Michael Schmitz, Michal Guerquin, Michael Wilson, Regan Huff, Ronan Le Bras, Rui Xin, Rulin Shao, Sam Skjonsberg, Shannon Zejiang Shen, Shuyue Stella Li, Tucker Wilde, Valentina Pyatkin, Will Merrill, Yapei Chang, Yuling Gu, Zhiyuan Zeng, Ashish Sabharwal, Luke Zettlemoyer, Pang Wei Koh, Ali Farhadi, Noah A. Smith, and Hannaneh Hajishirzi. Olmo 3, 2025. URL https://arxiv.org/abs/2512.13961.

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

[9] John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. Trust region policy optimization. In *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, 2015. arXiv:1502.05477.

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

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

[12] Mohammad Shoeybi, Mostofa Patwary, Raul Puri, Patrick LeGresley, Jared Casper, and Bryan Catanzaro. Megatron-LM: Training multi-billion parameter language models using model parallelism. *arXiv preprint arXiv:1909.08053*, 2019.

[13] Tencent HY. Stabilizing RLVR via token-level gradient diagnosis and layerwise clipping (SeqClip / GradLoc). https://hy.tencent.com/research/100015, 2025. Code: https://github.com/Tencent-Hunyuan/GradLoc.

[14] An Yang et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

[15] Jiacai Yao, Xinyi Zhou, Penghui Qi, Wee Sun Lee, Liu Bo, and Tianyu Pang. Rethinking the divergence regularization in LLM reinforcement learning. *arXiv preprint arXiv:2606.09821*, 2026.

[16] 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.

[17] Xiaoyu Zhao, Yang Liu, Kang Xu, Jiale Guo, Zewen Wang, Yuting Sun, Xiangyu Kong, Qian Cao, Lin Jiang, Zhuoyi Wen, Zhen Zhang, and Jin Zhou. Small leak can sink a great ship: Boost RL training on MoE with IcePop. https://ringtech.notion.site/icepop, 2025.

[18] Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Haiyang Jiang, Yuqiong Liu, Haoyan Lin, Chenxuan Wu, Fei Hu, An Yang, Jingren Zhou, and Junyang Lin. Stabilizing reinforcement learning with LLMs: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025.

[19] 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. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[20] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. SGLang: Efficient execution of structured language model programs. *arXiv preprint arXiv:2312.07104*, 2023.

[21] Zilin Zhu, Chengxing Xie, Xin Lv, and slime Contributors. slime: An LLM post-training framework for RL scaling. https://github.com/THUDM/slime, 2025.

A Notation

Table 2 Notation used throughout the paper.

Symbol	Meaning
$x,\ y_b,\ s_{b,t}$	Prompt; response $y_b = (y_{b,1}, \ldots, y_{b,T_b})$; state $s_{b,t} = (x, y_{b,1}, \ldots, y_{b,t-1})$.
$b,\ t,\ j$	Response index; token index; absolute trainer-update index.
$T_b,\ \mathcal{T}_{\text{act}}$	Length of response b; active (non-padding) token positions of the microbatch.
$\mu_{b,t}$	Behavior policy, namely the distribution the rollout engine actually sampled token (b,t) from.
$\pi^{(j)}$	Current trainer policy at update j.
$\pi^{(\ell)}$	Trainer-side reference log-probability at the rollout checkpoint.
$n,\ N_{b,t}$	Configured pipeline lag (trainer steps per weight broadcast); realized per-token lag.
$r_{b,t},\ d_{b,t}$	Sampled-token importance ratio and its log-ratio, $d_{b,t} = \log r_{b,t}$.
ρ_b^{seq}	GSPO length-normalized sequence ratio in Equation (20).
$\hat{A}_{b,t},\ R(y),\ \xi$	Advantage estimate; sequence reward; reward bound with $\lvert R(y)\rvert \le \xi$.
D_{TV}	Per-state total variation $\frac{1}{2}\sum_a \lvert\mu(a \mid s) - \pi(a \mid s)\rvert$.
$L'_\mu(\pi)$	Finite-horizon linear surrogate in Equation (3).
$\varepsilon_{\text{low}},\ \varepsilon_{\text{high}}$	Base clip radii; $\tilde{\varepsilon}_{\text{low/high},b,t}$ are the SAT radii in Equation (17).
$\psi(u;q),\ q,\ c_{\pm,b,t}$	Hill kernel; detached microbatch quantile reference; gated contraction factors.
$\bar{d}_\pi$	Sampled training–inference mismatch $\mathbb{E}_{(b,t)\in\mathcal{T}_{\text{act}}}\lvert\log r_{b,t}^{\text{tok}}\rvert$ from Equation (52).

B Synchronization Regimes of Asynchronous RL

The main text now keeps only the batch-wise asynchronous pipeline in Figure 1, because all reported experiments are conducted in that regime. This appendix records the two endpoint cases referenced there—Appendix B.1 for synchronous RL and Appendix B.2 for fully asynchronous RL. They clarify why the configured lag is only a coarse systems label and why the realized mismatch can vary substantially even when the nominal training recipe stays fixed.

B.1 Synchronous RL

In synchronous RL, rollout and optimization alternate on one shared GPU pool and every batch is trained by the same parameter vector that produced it:

$$j = \ell, \qquad \theta_{\text{roll}} = \theta_{\text{train}} = \theta^{(\ell)}, \qquad N_{b,t} = 0, \qquad \mu_{b,t} = \pi^{(\ell)}, \qquad r_{b,t} = 1 \text{ a.s.} \tag{24}$$

At this pre-update behavior point, the sampled ratio carries no off-policy correction, and under the usual on-policy assumptions the surrogate gradient coincides with the corresponding policy-gradient estimator. This regime therefore removes staleness as a learning issue, but it pays a clear systems cost: generation, optimization, and weight broadcast are serialized, so the two engines never overlap.

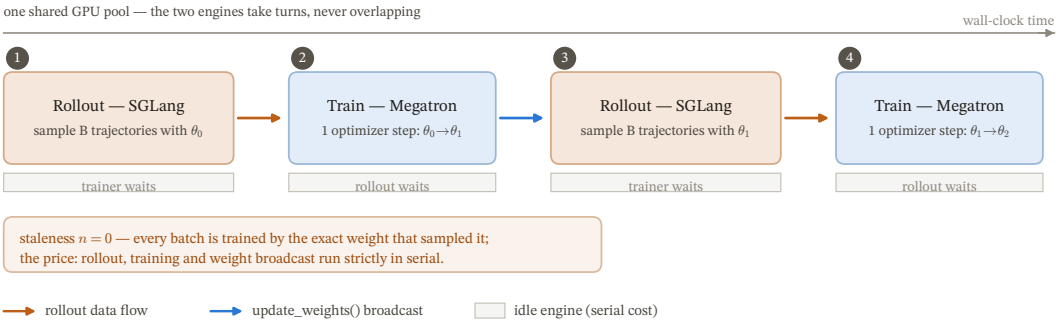


Figure 6 Synchronous RL. Rollout and training share one GPU pool and alternate in time, so each batch is consumed by the same weight version that generated it.

B.2 Fully asynchronous RL

Fully asynchronous RL removes the fixed batchwise handoff. Rollouts stream continuously into a queue, the trainer pulls data whenever the queue has enough completed trajectories, and new weights can be hot-swapped into the rollout engine while a response is still being decoded. A single trajectory can therefore span several policy versions before it is even enqueued, and then wait under still newer versions before it is trained. A convenient decomposition writes the realized lag as:

$$N_{b,t} = N_{\mathrm{PQS},b,t} + N_{\mathrm{IQS},b,t}, \tag{25}$$

where $N_{\mathrm{PQS},b,t}$ counts the versions that elapse while the trajectory is being generated and $N_{\mathrm{IQS},b,t}$ counts the versions that elapse while the completed trajectory waits in the queue. Following Dong et al. [1], the first term grows with the response-length tail of the workload and the second depends on queue occupancy and the rollout-to-trainer throughput ratio.

This regime is important conceptually because it makes the central quantity of the paper random at the token level. Even if the system exposes one nominal queue depth or one nominal update cadence, the relevant quantity for optimization is the realized sampled mismatch $d_{b,t} = \log r_{b,t}$, not a single preconfigured scalar. This is precisely the setting in which a batch-adaptive mechanism such as SAT becomes more natural than a rule keyed to a fixed lag alone.

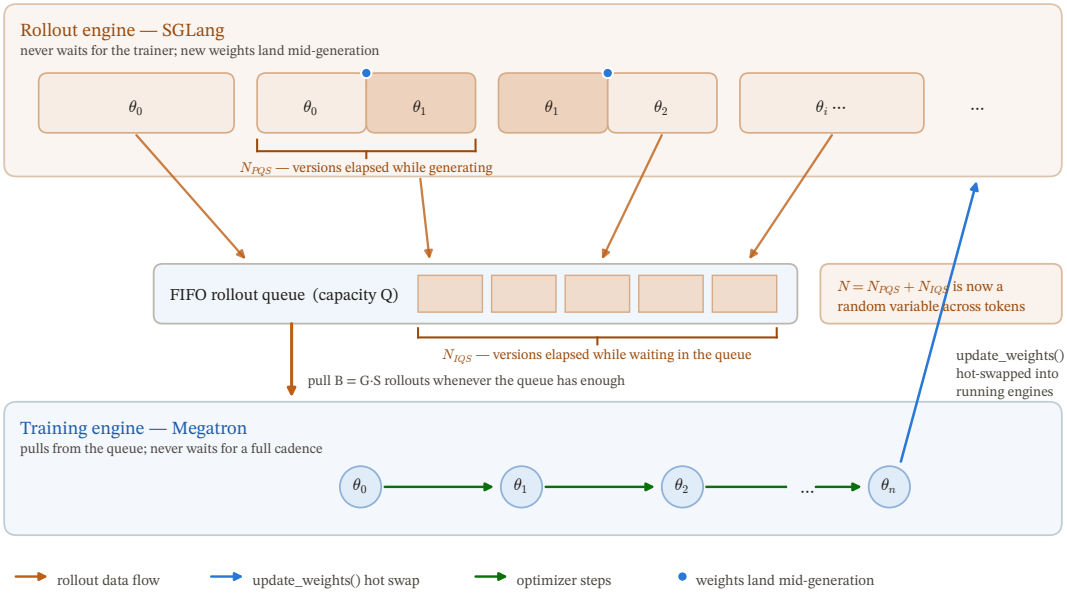


Figure 7 Fully asynchronous RL. Rollouts stream through a queue, weights can change during generation, and the realized lag varies across tokens.

C Mismatch Sources as Diagnostic Hypotheses

Equation (2) in the main text gives an exact two-term split of the sampled log-ratio. Write $\Delta^{\pi}_{b,t} = \log \pi^{(j)} - \log \pi^{(\ell)}$ for policy-update mismatch and $\Delta^{\mathrm{impl}}_{b,t} = \log \pi^{(\ell)} - \log \mu_{b,t}$ for implementation mismatch, both evaluated at $(y_{b,t}, s_{b,t})$. Then:

$$d_{b,t} = \log r_{b,t} = \Delta^{\pi}_{b,t} + \Delta^{\mathrm{impl}}_{b,t}. \tag{26}$$

The term $\Delta^{\pi}_{b,t}$ records policy-update mismatch inside the asynchronous window. The term $\Delta^{\mathrm{impl}}_{b,t}$ records implementation mismatch between the trainer-side reference and the actual rollout engine. The purpose of the present section is not to produce a unique additive decomposition of all sources of mismatch, because

such a decomposition would require additional intermediate distributions that the experiments do not log. Instead, the section records the hypotheses that are operationally useful when interpreting the measured $d_{b,t}$.

Policy updates are the most direct source. If a rollout is produced under $\theta^{(\ell)}$ and consumed after several optimizer steps, then a first-order expansion along the optimization path gives:

$$\log \pi^{(j)}(y_{b,t} \mid s_{b,t}) - \log \pi^{(\ell)}(y_{b,t} \mid s_{b,t}) \;\approx\; \sum_{k=\ell}^{j-1} \eta \left\langle \nabla_\theta \log \pi(y_{b,t} \mid s_{b,t}),\, g^{(k)} \right\rangle, \tag{27}$$

with learning rate η and update directions $g^{(k)}$. This term is exactly the object targeted by PPO-style importance reweighting. It need not be centered at zero, because consecutive updates tend to move in reward-improving directions rather than cancel each other.

Implementation mismatch remains even when the nominal weights are identical. Rollout and training may differ in fused kernels, attention kernels, KV-cache layout, numerical precision, or mixture-of-experts routing. The router case is particularly delicate because the Top-K mask is a discontinuous function of the router logits. Small numerical differences can activate a different expert set and change the forward pass abruptly. Hardware differences can amplify the same effect: the same model and code path can produce train–rollout KL values that differ by orders of magnitude across GPU generations. The paper therefore treats these mechanisms as diagnostic hypotheses that explain observed mismatch and motivate stabilizers such as R3, rather than as terms of an exact decomposition used in the proof.

D Derivation of the Finite-Horizon Improvement Bound

This section collects the full derivation behind the finite-horizon trust-region discussion in Section 2. We keep the notation of the main paper. A sequence $y = (y_1, \ldots, y_T)$ is generated autoregressively from prompt x, the behavior policy is μ, the target policy is π, rewards satisfy $|R(y)| \le \xi$, and the finite-horizon objective is $\mathcal{J}(\pi) = \mathbb{E}_{y\sim\pi}[R(y)]$. The surrogate used in the main text is:

$$L'_\mu(\pi) = \mathbb{E}_{y\sim\mu}\left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\right)\right]. \tag{28}$$

D.1 An exact performance-difference identity

Lemma D.1 (Exact finite-horizon identity)**.** *For any two sequence policies μ and π with compatible support on the μ-reachable states,*

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) = L'_\mu(\pi) - \Delta(\mu, \pi), \tag{29}$$

where

$$L'_\mu(\pi) = \mathbb{E}_{y\sim\mu}\left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\right)\right], \tag{30}$$

$$\Delta(\mu, \pi) = \mathbb{E}_{y\sim\mu}\left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\right)\left(1 - \prod_{j=t+1}^{T} \frac{\pi(y_j \mid s_j)}{\mu(y_j \mid s_j)}\right)\right]. \tag{31}$$

Proof. Start from the definition of the return difference:

$$\begin{aligned} \mathcal{J}(\pi) - \mathcal{J}(\mu) &= \mathbb{E}_{y\sim\pi}[R(y)] - \mathbb{E}_{y\sim\mu}[R(y)] \\ &= \sum_y \big(\pi(y \mid x) - \mu(y \mid x)\big) R(y). \end{aligned}$$

The sequence-probability difference admits the telescoping expansion:

$$\pi(y \mid x) - \mu(y \mid x) = \sum_{t=1}^{T} \left(\prod_{k=1}^{t-1} \mu(y_k \mid s_k) \right) \left(\pi(y_t \mid s_t) - \mu(y_t \mid s_t) \right) \left(\prod_{j=t+1}^{T} \pi(y_j \mid s_j) \right).$$

Substituting this identity and factoring out $\mu(y \mid x)$ gives:

$$\begin{aligned} \mathcal{J}(\pi) - \mathcal{J}(\mu) &= \sum_{y} \mu(y \mid x) R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right) \left(\prod_{j=t+1}^{T} \frac{\pi(y_j \mid s_j)}{\mu(y_j \mid s_j)} \right) \\ &= \mathbb{E}_{y \sim \mu} \left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right) \left(\prod_{j=t+1}^{T} \frac{\pi(y_j \mid s_j)}{\mu(y_j \mid s_j)} \right) \right]. \end{aligned}$$

Adding and subtracting the term with the future-product replaced by 1 yields:

$$\begin{aligned} \mathcal{J}(\pi) - \mathcal{J}(\mu) = \mathbb{E}_{y \sim \mu} &\left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right) \right] \\ &- \mathbb{E}_{y \sim \mu} \left[R(y) \sum_{t=1}^{T} \left(\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right) \left(1 - \prod_{j=t+1}^{T} \frac{\pi(y_j \mid s_j)}{\mu(y_j \mid s_j)} \right) \right], \end{aligned}$$

which is exactly Equations (30) and (31). □

D.2 A sequence-level total-variation bound

Lemma D.2 (Sequence D_{TV} is bounded by cumulative one-step D_{TV})**.** *Let $\mu_T(\cdot \mid s_1)$ and $\pi_T(\cdot \mid s_1)$ denote the induced sequence distributions over responses of length T. Then*

$$D_{\mathrm{TV}}\big(\mu_T(\cdot \mid s_1) \,\|\, \pi_T(\cdot \mid s_1)\big) \le \sum_{t=1}^{T} \mathbb{E}_{s_t \sim \mu} \Big[D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t) \,\|\, \pi(\cdot \mid s_t)\big) \Big]. \tag{32}$$

Proof. Write $P(y) = \mu_T(y \mid s_1)$ and $Q(y) = \pi_T(y \mid s_1)$. Then

$$2D_{\mathrm{TV}}(P \| Q) = \sum_{y} |P(y) - Q(y)|.$$

Using the product-difference identity:

$$a_1 \cdots a_T - b_1 \cdots b_T = \sum_{t=1}^{T} \left(\prod_{k=1}^{t-1} a_k \right) (a_t - b_t) \left(\prod_{j=t+1}^{T} b_j \right)$$

and the triangle inequality,

$$2D_{\mathrm{TV}}(P \| Q) \le \sum_{t=1}^{T} \sum_{y} \left(\prod_{k=1}^{t-1} \mu(y_k \mid s_k) \right) \left| \mu(y_t \mid s_t) - \pi(y_t \mid s_t) \right| \left(\prod_{j=t+1}^{T} \pi(y_j \mid s_j) \right).$$

For each fixed t, summing over $y_{t+1}, \ldots, y_T$ collapses the trailing product to 1, which leaves:

$$2D_{\mathrm{TV}}(P\|Q) \le \sum_{t=1}^{T} \sum_{y_1,\ldots,y_{t-1}} \left(\prod_{k=1}^{t-1} \mu(y_k \mid s_k) \right) \sum_{y_t} \left| \mu(y_t \mid s_t) - \pi(y_t \mid s_t) \right|.$$

The inner sum is $2D_{\mathrm{TV}}(\mu(\cdot \mid s_t)\|\pi(\cdot \mid s_t))$, while the outer sum is the expectation over s_t induced by μ. Dividing by 2 proves the claim. □

D.3 Max-divergence and average-divergence forms

Proposition D.3 (Max-divergence and average-divergence bounds)**.** Under the assumptions above,

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \ge L'_\mu(\pi) - 2\xi T(T-1)\left[D_{\mathrm{TV}}^{\max}(\mu,\pi)\right]^2, \tag{33}$$

where $D_{\mathrm{TV}}^{\max}(\mu,\pi) = \sup_s D_{\mathrm{TV}}(\mu(\cdot \mid s)\|\pi(\cdot \mid s))$, and also

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \ge L'_\mu(\pi) - 4\xi\, \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{T} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t) \,\|\, \pi(\cdot \mid s_t)\big) \right]. \tag{34}$$

Equation (34) is the form reported in the main text as Equation (4).

Proof of the max-divergence form. By Lemma D.1,

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) = L'_\mu(\pi) - \Delta(\mu,\pi).$$

To bound $\Delta(\mu,\pi)$, first use $|R(y)| \le \xi$:

$$\Delta(\mu,\pi) \le \xi \sum_{t=1}^{T} \mathbb{E}_{y_{\le t}\sim\mu}\left[\left| \frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right| \mathbb{E}_{y_{>t}\sim\mu(\cdot|s_{t+1})} \left| 1 - \frac{\pi(y_{>t} \mid s_{t+1})}{\mu(y_{>t} \mid s_{t+1})} \right| \right]. \tag{35}$$

The inner expectation is exactly twice the D_{TV} divergence between the future-trajectory distributions:

$$\mathbb{E}_{y_{>t}\sim\mu(\cdot|s_{t+1})} \left| 1 - \frac{\pi(y_{>t} \mid s_{t+1})}{\mu(y_{>t} \mid s_{t+1})} \right| = 2D_{\mathrm{TV}}\big(\mu_{>t}(\cdot \mid s_{t+1}) \,\|\, \pi_{>t}(\cdot \mid s_{t+1})\big).$$

Applying Lemma D.2 to this future-horizon D_{TV} and then upper-bounding each term by $D_{\mathrm{TV}}^{\max}(\mu,\pi)$ gives

$$D_{\mathrm{TV}}\big(\mu_{>t}(\cdot \mid s_{t+1}) \,\|\, \pi_{>t}(\cdot \mid s_{t+1})\big) \le (T-t) D_{\mathrm{TV}}^{\max}(\mu,\pi).$$

Substituting into Equation (35) yields

$$\begin{aligned}
\Delta(\mu,\pi) &\le 2\xi D_{\mathrm{TV}}^{\max}(\mu,\pi) \sum_{t=1}^{T} (T-t) \mathbb{E}_{s_t\sim\mu}\left[\sum_{y_t} \mu(y_t \mid s_t) \left| \frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1 \right| \right] \\
&= 2\xi D_{\mathrm{TV}}^{\max}(\mu,\pi) \sum_{t=1}^{T} (T-t) \mathbb{E}_{s_t\sim\mu}\Big[2D_{\mathrm{TV}}(\mu(\cdot \mid s_t)\|\pi(\cdot \mid s_t)) \Big] \\
&\le 4\xi \left[D_{\mathrm{TV}}^{\max}(\mu,\pi)\right]^2 \sum_{t=1}^{T} (T-t) \\
&= 2\xi T(T-1)\left[D_{\mathrm{TV}}^{\max}(\mu,\pi)\right]^2.
\end{aligned}$$

Substituting this bound into the exact identity proves Equation (33). □

Proof of the average-divergence form. Return to Equation (35). Instead of upper-bounding the future-trajectory D_{TV} by $(T-t)D_{\mathrm{TV}}^{\max}$, use the trivial bound $D_{\mathrm{TV}} \le 1$. This gives:

$$\begin{aligned}
\Delta(\mu,\pi) &\le 2\xi \sum_{t=1}^{T} \mathbb{E}_{y_{\le t}\sim\mu}\left[\left|\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\right|\right] \\
&= 2\xi \sum_{t=1}^{T} \mathbb{E}_{s_t\sim\mu}\left[\sum_{y_t} \mu(y_t \mid s_t)\left|\frac{\pi(y_t \mid s_t)}{\mu(y_t \mid s_t)} - 1\right|\right] \\
&= 2\xi \sum_{t=1}^{T} \mathbb{E}_{s_t\sim\mu}\Big[2D_{\mathrm{TV}}(\mu(\cdot \mid s_t)\|\pi(\cdot \mid s_t))\Big] \\
&= 4\xi\, \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{T} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t)\,\|\,\pi(\cdot \mid s_t)\big)\right].
\end{aligned}$$

Substituting this into Equation (29) proves Equation (34). □

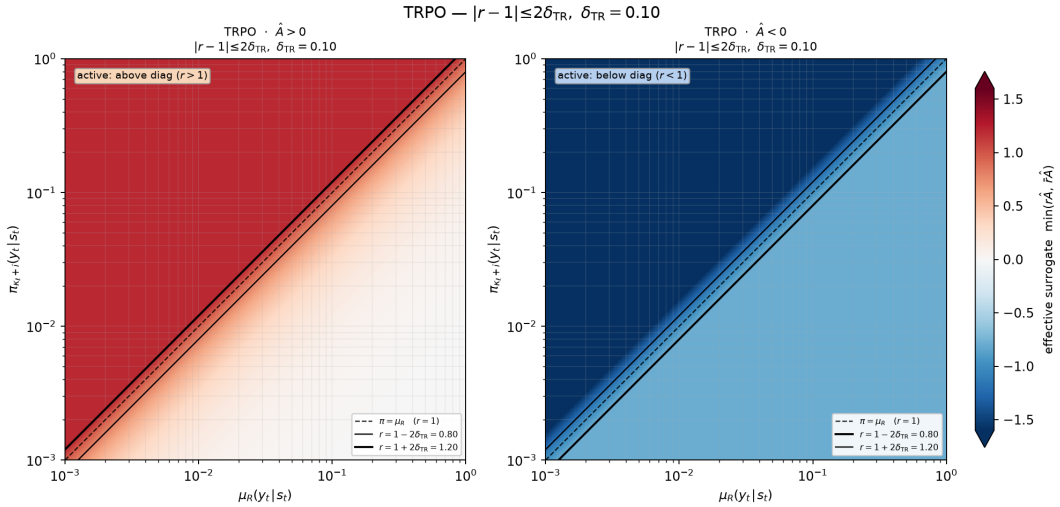
The two forms can be combined into the immediate corollary:

$$\mathcal{J}(\pi) - \mathcal{J}(\mu) \ge L'_\mu(\pi) - \min\left(2\xi T(T-1)\big[D_{\mathrm{TV}}^{\max}(\mu,\pi)\big]^2, 4\xi\, \mathbb{E}_{y\sim\mu}\left[\sum_{t=1}^{T} D_{\mathrm{TV}}\big(\mu(\cdot \mid s_t)\,\|\,\pi(\cdot \mid s_t)\big)\right]\right). \quad (36)$$

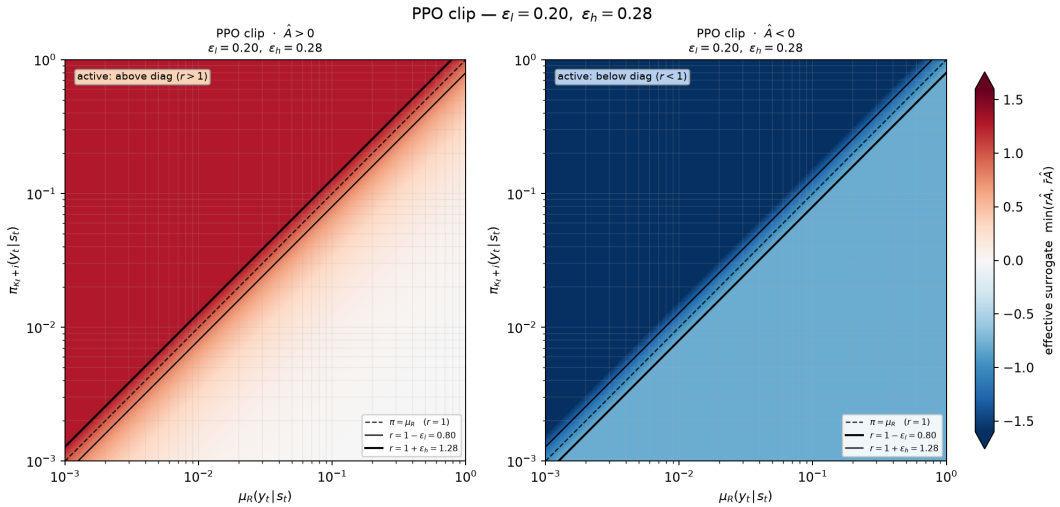
The max-divergence form is sharper for very small updates, whereas the average-divergence form avoids the quadratic dependence on the horizon and is more useful for long LLM responses.

E How Does Each Stability Approach Work in Async RL?

The main paper introduces SAT as one way to adapt the sampled trust-region surrogate to the observed mismatch distribution. The larger asynchronous-RL literature can be read through the same lens. Each method chooses one mathematical object built from the observed log-ratio or from closely related quantities, such as the granularity of the ratio, the denominator, a mask, a threshold, or a clip radius, and then modifies that object. Table 3 summarizes the design space, and the subsections below expand the methods in the same order as the companion HTML analysis. Each subsection keeps the defining equations and, when available, the matched empirical comparison in one place.



(a) TRPO

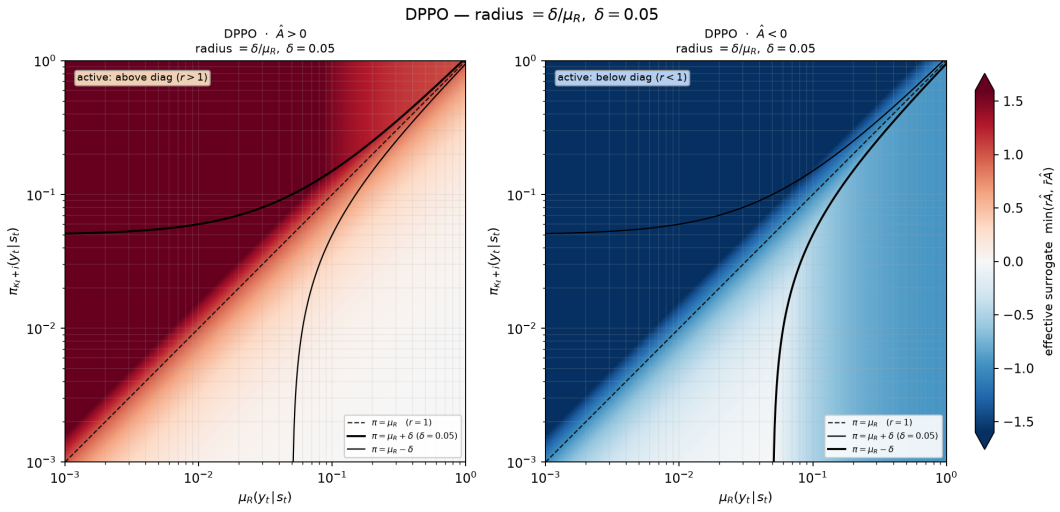


(b) PPO

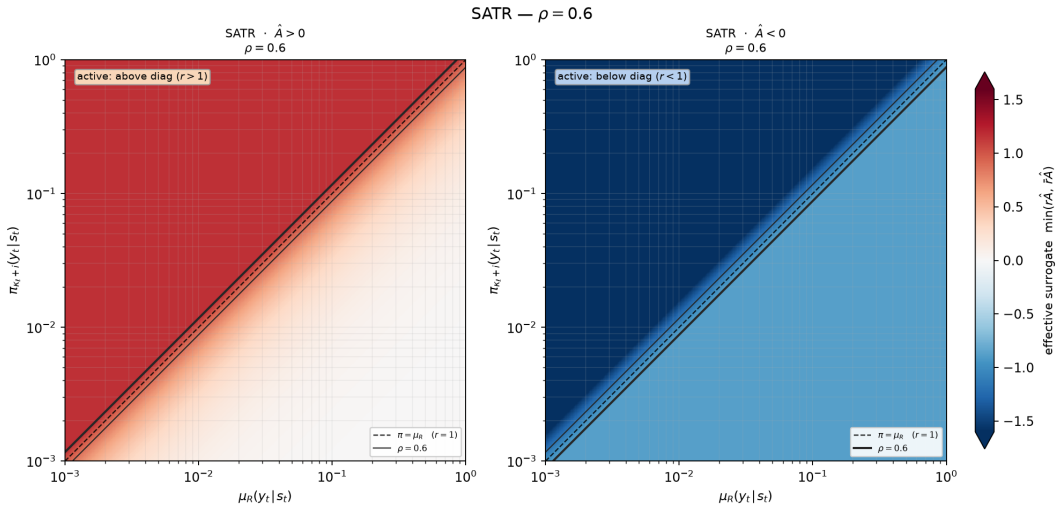
Figure 8 Asymmetric-clipping family, part I: TRPO and PPO.

Table 3 Async-RL stabilizers in one design space. The table records which mathematical object each method modifies and whether the gate depends on the update direction $\operatorname{sign}(\hat{A}_t(r_t - 1))$.

Method	Unit	Discriminator	Constraint	Weight $w_{b,t}$	Direction
PPO [10]	token	$\lvert r_{b,t} - 1\rvert$	$\lvert r_{b,t} - 1\rvert \le \varepsilon$	$\{0, 1\}$	asymmetric
TRPO [9]	state	$D_{\mathrm{TV}}[s_t]$	$D_{\mathrm{TV}}[s_t] \le \delta_{\mathrm{TR}}$	—	asymmetric
DPPO [8]	token	D_t^{TV}	$D_t^{\mathrm{TV}} \le \delta$	$\{0, 1\}$	asymmetric
DRPO [15]	token	D_t^{TV}	soft quadratic penalty	$[1 - \frac{1}{\delta}, 1 + \frac{1}{\delta}]$	asymmetric
TIS [18]	token	$r_{b,t}$	one-sided cap	$\min(r_{b,t}, C)$	one-sided
IcePop [3, 17]	token	train/inference ratio	symmetric range gate	$\{0, 1\}$ or in-range weight	symmetric
KPop [2]	token	Bernoulli KL	bidirectional threshold	$\{0, 1\}$	symmetric
GSPO [19]	sequence	$\lvert \rho_b^{\mathrm{seq}} - 1\rvert$	sequence clip	$\{0, 1\}$	asymmetric
SeqClip [13]	sequence	ρ_b^{seq}	$\rho_b^{\mathrm{seq}} \in [\alpha_{\mathrm{seq}}, \beta_{\mathrm{seq}}]$	$\{0, 1\}$	symmetric
R3 [5]	forward pass	routing mask	replay rollout Top-K mask	—	—
SAT (ours)	token	$\lvert d_{b,t}\rvert$ and $\lvert r_{b,t} - 1\rvert$	adaptive clip interval	$\{0, 1\}$	asymmetric, adaptive



(a) DPPO



(b) SAT

Figure 9 Asymmetric-clipping family, part II: DPPO and SAT. In this case, the contraction factor of SAT is set to 0.6, which is staleness-adaptive in implementation.

E.1 GSPO and SeqClip

GSPO moves importance weighting from the token to the response by defining the length-normalized sequence ratio:

$$\rho_b^{\text{seq}} = \left(\frac{\pi^{(j)}(y_b \mid x)}{\mu(y_b \mid x)}\right)^{1/T_b} = \exp\left(\frac{1}{T_b}\sum_{t=1}^{T_b} d_{b,t}\right), \tag{37}$$

and clipping at the response level:

$$\mathcal{S}_{\text{GSPO}} = \mathbb{E}_{\{y_b\}\sim\mu}\left[\frac{1}{G}\sum_{b=1}^{G}\min\Big(\rho_b^{\text{seq}}\hat{A}_b, \text{clip}(\rho_b^{\text{seq}}, 1-\varepsilon, 1+\varepsilon)\hat{A}_b\Big)\right]. \tag{38}$$

Length normalization removes the automatic exponential dependence on response length that the raw product of token ratios would induce, although token-level deviations can still reinforce or cancel. SeqClip adds a hard response-level gate:

$$\mathcal{S}_{\text{SeqClip}} = \mathbb{E}_{\{y_b\}\sim\mu}\left[\frac{1}{G}\sum_{b=1}^{G}\mathbf{1}\{\alpha_{\text{seq}} \le \rho_b^{\text{seq}} \le \beta_{\text{seq}}\}\min\Big(\rho_b^{\text{seq}}\hat{A}_b, \text{clip}(\rho_b^{\text{seq}}, 1-\varepsilon, 1+\varepsilon)\hat{A}_b\Big)\right]. \tag{39}$$

so responses outside a narrow interval are discarded altogether. Both methods smooth isolated token fluctuations by construction, but neither removes routing or engine mismatch, which is why the empirical combination with R3 remains relevant.

Empirical comparison. Equations (37) and (39) define the sequence-level ratio and the hard response-level gate. Table 4 summarizes the matched GSPO comparison under the shared setup of Appendix F. GSPO stays above GRPO at both configured lags, with 32.25 against 31.25 at lag 1 and 31.46 against 30.17 at lag 8. Both lag-8 runs later collapse in the logged window. The comparison therefore supports a difference in weighting granularity, but by itself it does not isolate which mismatch source drives the later instability. No standalone SeqClip run is reported in the current study.

Table 4 Reported best AIME24 pass@1 for the GSPO comparison.

Method	lag 1	lag 8
GRPO	31.25	30.17‡
GSPO	32.25	31.46‡

E.2 Rollout Routing Replay

R3 addresses mismatch from discrete MoE routing. Even at nominally identical weights, SGLang and Megatron can choose different Top-K experts for the same token, and asynchronous updates add another source of variation. R3 replays the rollout-time Top-K mask while keeping the current trainer logits in the softmax denominator,

$$g_{\text{replay},e}^{(j)} = \frac{I_{\text{infer},e}^{(\ell)}\,\exp\big(z_{\text{train},e}^{(j)}\big)}{\sum_{e'} I_{\text{infer},e'}^{(\ell)}\,\exp\big(z_{\text{train},e'}^{(j)}\big)}, \qquad \mathbf{h}_{\text{replay}}^{(j)} = \sum_{e=1}^{M} g_{\text{replay},e}^{(j)}\,\mathcal{E}_e^{(j)}(\mathbf{h}_{\text{train}}). \tag{40}$$

This aligns the discrete expert set with rollout while still allowing gradients to flow through the current router. It does not reproduce the entire rollout forward pass, since hidden states, logits, expert weights, and downstream layers may still differ, but it removes one important mismatch source at the routing branch.

Empirical comparison. Equation (40) defines the replayed routing gate. Table 5 summarizes the matched R3 comparison. The strongest gain appears for the GSPO pairing, which reaches 34.00 at lag 1 and 33.13 at lag 8. In the reported summary, the lag-8 GSPO w/ R3 run is not marked as a later-collapse setting. GRPO w/ R3 improves over GRPO at lag 1, but remains below GSPO w/ R3 at both lags. These numbers are consistent

with routing replay reducing MoE routing mismatch on the reported stack, but they do not establish a unique causal path.

Table 5 Reported best AIME24 pass@1 for the R3 comparison.

Method	lag 1	lag 8
GRPO	31.25	30.17‡
GSPO	32.25	31.46‡
GRPO w/ R3	31.83	29.58
GSPO w/ R3	34.00	33.13

E.3 TIS, IcePop, and KPop

TIS, IcePop, and KPop act on sampled tokens rather than whole responses. TIS is the mildest intervention: it caps only the overshoot side of the importance weight:

$$w_{b,t}^{\mathrm{TIS}} = \min\bigl(r_{b,t}, C\bigr), \qquad \mathcal{S}_{\mathrm{TIS}} = \mathbb{E}\left[\sum_{b,t} \mathrm{sg}[w_{b,t}^{\mathrm{TIS}}]\hat{A}_{b,t} \log \pi^{(j)}(y_{b,t} \mid s_{b,t})\right]. \tag{41}$$

It changes gradient magnitude but does not alter the clip interval. IcePop instead applies a two-sided gate to the train-engine versus inference-engine ratio:

$$k_{b,t} = \frac{\tilde{\mu}_{b,t}(y_{b,t} \mid s_{b,t})}{\mu_{b,t}(y_{b,t} \mid s_{b,t})}, \qquad M_{b,t}^{\mathrm{IcePop}} = k_{b,t}\,\mathbf{1}\{\alpha \le k_{b,t} \le \beta\}, \tag{42}$$

which is often simplified to a binary in-range mask. KPop replaces raw ratios with a Bernoulli-KL test:

$$D_{\mathrm{KL}}^{\mathrm{Bern}}(p\|q) = p\log\frac{p}{q} + (1-p)\log\frac{1-p}{1-q}, \qquad M_{b,t}^{\mathrm{KPop}} = \mathbf{1}\bigl\{D_{\mathrm{KL}}^{\mathrm{Bern}}(p_{b,t}^{\pi}\|p_{b,t}^{\mu}) \le \phi\bigr\}\mathbf{1}\bigl\{D_{\mathrm{KL}}^{\mathrm{Bern}}(p_{b,t}^{\mu}\|p_{b,t}^{\pi}) \le \phi\bigr\}. \tag{43}$$

where $p_{b,t}^{\pi}$ and $p_{b,t}^{\mu}$ are the trainer- and rollout-side probabilities of the sampled token. IcePop and KPop are symmetric in direction: they may discard tokens whose local gradient would move the sampled ratio back toward the behavior reference.

Empirical comparison. Equations (41)–(43) summarize the token-level mechanisms. The current study reports matched runs for TIS and its compositions with R3, but not standalone AIME24 grids for IcePop or KPop. Table 6 shows that the TIS family remains in a relatively narrow band from 31.21 to 32.92 across the two configured lags, and none of these runs is marked as a later-collapse setting. The strongest lag-8 value in this family is 32.92 for GSPO w/ TIS w/ R3.

Table 6 Reported best AIME24 pass@1 for the TIS comparison.

Method	lag 1	lag 8
GRPO w/ TIS	31.62	31.46
GSPO w/ TIS	32.88	31.21
GRPO w/ TIS w/ R3	31.79	31.79
GSPO w/ TIS w/ R3	31.88	32.92

E.4 Asymmetric clipping

The methods above are easiest to interpret against the asymmetric gate used by PPO and inherited by DPPO, SPO, DRPO, and SAT. For a fixed sampled token with ratio r_t and advantage $\hat{A}_t$, the unclipped surrogate has derivative $\hat{A}_t$ with respect to r_t, so the sign of $\hat{A}_t$ determines the locally preferred direction. Relative to $r_t = 1$, the sampled update is locally diverging when $\mathrm{sign}(\hat{A}_t(r_t - 1)) > 0$ and locally converging when

$\mathrm{sign}(\hat{A}_t(r_t-1)) < 0$. Away from the kinks, PPO therefore keeps the gate:

$$M_t^{\mathrm{PPO}} = \mathbf{1}\Big\{\mathrm{sign}(\hat{A}_t(r_t-1)) \le 0 \ \vee \ |r_t-1| \le \varepsilon\Big\}, \tag{44}$$

which clips only outward moves beyond the nominal interval and always preserves pull-back updates.

DPPO keeps the same directional logic but changes the discriminator. Under the sampled D_{TV} discriminator:

$$D_t^{\mathrm{TV}} = \big|\pi^{(j)}(y_t \mid s_t) - \mu(y_t \mid s_t)\big| = \mu(y_t \mid s_t)\,|r_t - 1|, \tag{45}$$

so $D_t^{\mathrm{TV}} \le \delta$ is equivalent to:

$$|r_t - 1| \le \frac{\delta}{\mu(y_t \mid s_t)}. \tag{46}$$

The induced mask is:

$$M_t^{\mathrm{DPPO}} = \mathbf{1}\Big\{\mathrm{sign}(\hat{A}_t(r_t-1)) \le 0 \ \vee \ |r_t-1| \le \delta/\mu(y_t \mid s_t)\Big\}, \tag{47}$$

which is loose on long-tail tokens and tight on head tokens. SPO and DRPO replace hard clipping with quadratic penalties. SPO uses:

$$\mathcal{S}_{\mathrm{SPO}}(\pi) = \mathbb{E}\left[\sum_t \left(r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\varepsilon}(r_t-1)^2\right)\right], \tag{48}$$

whose stationary point lies at $r_t^\star = 1 + \mathrm{sign}(\hat{A}_t)\varepsilon$, exactly on PPO's boundary. DRPO makes the same replacement in DPPO's sampled D_{TV} geometry:

$$\mathcal{S}_{\mathrm{DRPO}}(\pi^{(j)}) = \mathbb{E}_{y\sim\mu}\left[\sum_t \left(r_t\hat{A}_t - \frac{|\hat{A}_t|}{2\delta}\,\mu(y_t \mid s_t)(r_t-1)^2\right)\right]. \tag{49}$$

Unlike SAT, however, their thresholds are fixed in advance rather than inferred from the current mismatch tail.

Empirical comparison. Equations (47) and (49) place DPPO and DRPO inside the asymmetric-clipping family. Table 7 summarizes the reported DPPO comparison. DPPO reaches 33.96 at lag 1 and 32.71 at lag 8. It stays above the GRPO baseline at both lags and outside the runs marked as later collapse. This is consistent with a sampled D_{TV} gate improving stability on the reported stack, but it does not explain the remaining gap to SAT-GSPO w/ R3 or even to SAT-GRPO w/ R3.

Table 7 Reported best AIME24 pass@1 for the DPPO comparison.

Method	lag 1	lag 8
GRPO	31.25	30.17‡
GSPO	32.25	31.46‡
DPPO	33.96	32.71

E.5 Top-p configuration

Support mismatch appears when rollout truncates the behavior distribution but training evaluates an untruncated target policy. If rollout uses top-p below 1, actions outside the nucleus have zero behavior probability, so absolute continuity $\pi^{(j)} \ll \mu_{b,t}$ can fail. One fix is to replay the rollout truncation mask inside the trainer softmax, as in MAI-Thinking-1 [6]. If $\mathcal{T}_{b,t} \subseteq \mathcal{V}$ denotes the rollout nucleus, training uses:

$$\tilde{\mathrm{logit}}_{b,t}(a) = \begin{cases} \mathrm{logit}_{b,t}(a), & a \in \mathcal{T}_{b,t}, \\ -\infty, & a \notin \mathcal{T}_{b,t}, \end{cases} \qquad \pi^{(j)}(a \mid s_{b,t}) = \mathrm{softmax}(\tilde{\mathrm{logit}}_{b,t})(a). \tag{50}$$

Our experiments use the simpler alternative of disabling truncation at rollout entirely by setting temperature 1.0, top-p to 1.0, and top-k to -1, so the rollout support matches the full vocabulary.

Empirical note. Equation (50) gives the replayed support mask when rollout truncation is enabled. The companion analysis does not report a separate AIME24 grid for top-p. In the present experiments we instead disable truncation at rollout with temperature 1.0, top-p set to 1.0, and top-k set to -1, so rollout support matches the full vocabulary. Under this configuration, the reported trainer-to-rollout KL remains on the order of 10^{-3}, which is comparable to the post-R3 regime on the same stack.

E.6 Using rollout log probabilities

A separate choice concerns the denominator of the sampled ratio. Standard asynchronous PPO/GRPO often uses the trainer-side recomputation $\tilde{\mu}_{b,t}$ as the reference log-probability. Replacing it with the rollout log-probability restores the actual behavior denominator:

$$\log \tilde{\mu}_{b,t}(y_{b,t} \mid s_{b,t}) \leftarrow \log \mu_{b,t}(y_{b,t} \mid s_{b,t}), \qquad r_{b,t}^{\text{rollout-lp}} = \exp\Big(\log \pi^{(j)}(y_{b,t} \mid s_{b,t}) - \log \mu_{b,t}(y_{b,t} \mid s_{b,t})\Big) = r_{b,t}. \tag{51}$$

Under the support condition, this is the denominator required by the importance-sampling identity. It may also increase the observed variance of sampled ratios, because implementation mismatch is no longer absorbed by trainer-side recomputation. The change therefore exposes mismatch more faithfully, but does not reduce policy-version lag by itself.

Empirical comparison. Equation (51) defines the denominator switch. Table 8 records the observed variance of the sampled log-ratio with and without rollout log probabilities. The switch exposes more implementation mismatch to the optimizer and increases the measured variance in the reported runs. This is a property of the estimator used in optimization rather than a change to the deterministic population bound itself.

Table 8 Observed variance of the sampled log-ratio when rollout log probabilities are used as the denominator.

Method	lag 1	lag 8
GRPO	2.6×10^{-4}	1.4×10^{-4}
GRPO w/ rollout log probabilities	3.2×10^{-4}	2.7×10^{-4}

F Experimental Details

The backbone is Qwen3-30B-A3B-Base, an MoE model with 48 layers, hidden size 2048, 128 experts with Top-8 routing, MoE-FFN width 768, a softmax router, no auxiliary routing loss, and RoPE base 10^6. The benchmark is AIME24 with 30 problems. Each evaluation uses 16 samples per problem, a 30k response cap, top-p set to 1, and is run every five training iterations. Each logged evaluation score averages four evaluation runs, and the base-model reference is 9.38.

All experiments run on the slime asynchronous RL framework with SGLang as the rollout engine and Megatron as the trainer. The configured lag is controlled at $n \in \{1, 8\}$, with a weight-broadcast interval of 8 for the lag-8 runs. All runs in a comparison share the optimizer, data, and reward. Table 1 reports both the best logged AIME24 checkpoint over the training window and the corresponding last-epoch sampled mismatch values. The symbol ‡ marks runs that later collapse and do not recover in the logged window, even though the best-so-far checkpoint is still reported.

Training data consists of the union of DAPO-Math-17k [16] and Dolci-RL-Zero-Math-7B [7], with 30,712 prompts in total. We use a chat template, rollout shuffling, and balanced sampling. The reward is the rule-based PrimeMath verifier with multi-pattern answer extraction, Hendrycks-MATH normalization, and SymPy equivalence checking. Optimization uses Adam with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and constant learning rate 10^{-6}, without warmup or decay. Each iteration consumes 256 prompts and 16 samples per prompt,

for 4096 responses in total, with one minor step per broadcast and 544 rollout iterations. Rollout and evaluation use a maximum response length of 32k tokens and dynamic batching with a 32k per-GPU token cap. Sampling is performed at temperature 1.0, top-p set to 1.0, and top-k set to -1, so the rollout distribution has full-vocabulary support. The base clip radii are $(\varepsilon_{\text{low}}, \varepsilon_{\text{high}}) = (0.2, 0.2)$. The KL regularizer is low-variance KL with coefficient 10^{-3}, and the entropy bonus is zero. For SAT, we use the Hill kernel with exponent 2 and reference quantile 0.90. Hardware consists of one node with eight GPUs, using GPU0–3 for training and GPU4–7 for rollout, with TP= 4, PP= 1, CP= 1, EP= 4, ETP= 1, sequence parallelism, bf16 arithmetic with fp32 gradient all-reduce and fp32 attention softmax, the FlashAttention backend, and SGLang memory fraction set to 0.85.

The per-method settings differ only in the ratio and KL granularity and in the clipping or gating rule. GRPO uses per-token ratios and per-token KL with the standard asymmetric PPO clip. GSPO keeps the same outer objective but replaces the token ratio and KL by a sequence-level ratio and a sequence-level mean KL that is all-gathered and broadcast to all active tokens of the response. DPPO multiplies the GRPO clip by a hard mask that zeros sampled tokens only when the local update is outward and the sampled D_{TV} discriminator exceeds its threshold; in the reported runs, this is the DPPO-D_{TV} variant and the threshold is 10^{-3}. When SAT is layered on top of any base recipe, it contracts the shared radii using the ratio scalar of that recipe, namely the token ratio for GRPO and the broadcast sequence ratio for GSPO, and it is bit-identical to the base loss when disabled. R3 and TIS are toggled independently and therefore compose with both fixed-clip and adaptive-clip objectives.

Metrics and statistical scope We report (i) the best-observed AIME24 pass@1 over the logged training window; AIME24 has 30 problems, each evaluation uses 16 samples per problem, runs every five training iterations, and each logged score averages four evaluation runs (base-model reference: 9.38); and (ii) the sampled training–inference mismatch:

$$\bar{d}_\pi \;=\; \mathbb{E}_{(b,t)\in\mathcal{T}_{\text{act}}} \big|\log r^{\text{tok}}_{b,t}\big|, \tag{52}$$

computed from the token ratio of Equation (20) for all methods including GSPO, reported in the lower panel of Table 1 and as paired per-step curves in Figure 4. Every configuration is a single training seed, so each cell is one run: the numbers are a snapshot rather than confidence-interval estimates, and where the text calls two $\bar{d}_\pi$ values "close" we mean within the trailing-25% within-run fluctuation, not a multi-seed test. Lower $\bar{d}_\pi$ means less observed sampled mismatch; it is not a full-D_{TV} measurement and not by itself a return ranking.

Table 9 Per-method reproducible hyper-parameters. All other settings are shared and summarized above.

Hyper-parameter	GRPO	GSPO	DPPO
advantage estimator	GRPO	GSPO	DPPO
ratio / KL granularity	per-token	per-sequence	per-token
$(\varepsilon_{\text{low}}, \varepsilon_{\text{high}})$	$(0.2, 0.2)$	$(0.2, 0.2)$	$(0.2, 0.2)$
KL-loss coefficient	10^{-3}	10^{-3}	10^{-3}
KL-loss type	low-variance KL	low-variance KL	low-variance KL
entropy coefficient	0	0	0
extra mechanism	–	sequence-level KL all-gather	D_{TV} hard mask $\delta = 10^{-3}$
needs rollout log-prob	no	no	yes

Each cell of Table 1 is a single run at a fixed seed, so the empirical scope is intentionally narrow. The $\bar{d}_\pi$ curves in Figure 4 are raw per-step logs at a four-step stride without smoothing. Steady-state values quoted in the caption average the trailing 25% of each run, whereas the lower $\bar{d}_\pi$ block of Table 1 reports last-epoch means, so the two summaries can differ slightly for the same configuration. At lag 1, the trailing-25% values are approximately 0.0102 for GRPO, 0.0103 for DPPO, 0.0101 for GSPO w/ TIS, 0.0104 for SAT-GSPO, 0.0061 for GSPO w/ TIS w/ R3, 0.0055 for GRPO w/ R3, 0.0057 for GSPO w/ R3, and 0.0058 for SAT-GSPO w/ R3. At lag 8, the corresponding values are approximately 0.0068 for GSPO w/ TIS w/ R3, 0.0073 for GSPO w/ R3, 0.0085 for SAT-GRPO w/ R3, 0.0097 for GRPO w/ R3, and 0.0108 for SAT-GSPO, while the remaining shown runs lie roughly in the 0.0103–0.0131 range and the GSPO baseline diverges beyond the

plotted axis.

G SAT Runtime Diagnostics

Table 10 Diagnostics emitted at every optimizer step by the adaptive rule in Equations (16)–(17). They describe the implemented gate on sampled tokens rather than full-vocabulary D_{TV}.

Quantity	Meaning
$\mathbb{E}_{(b,t)\in\mathcal{T}_{\mathrm{act}}}\left[\varepsilon_{\mathrm{low}}\, c_{-,b,t}\right]$	Mean effective lower radius on active sampled tokens.
$\mathbb{E}_{(b,t)\in\mathcal{T}_{\mathrm{act}}}\left[\varepsilon_{\mathrm{high}}\, c_{+,b,t}\right]$	Mean effective upper radius on active sampled tokens.
$\mathbb{E}_{(b,t)\in\mathcal{T}_{\mathrm{act}}}\left[\mathbf{1}\{q>0,\ \lvert d_{b,t}\rvert > q\}\right]$	Realized gate rate; ties can make it smaller than 10%.
$\min_{(b,t)\in\mathcal{T}_{\mathrm{act}}} c_{\pm,b,t}$	Strongest contraction applied to a sampled token in the microbatch.
q	Detached microbatch quantile reference from Equation (14).