FLASHREINFORCE: CRITIC-FREE SINGLE-ROLLOUT ASYNCHRONOUS RL FOR AGENTIC LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Asynchronous reinforcement learning suits long-horizon agents with irregular rollout times, but group-relative updates require repeated rollouts of each prompt. However, group-based sampling limits prompt coverage at a fixed rollout budget and introduces group synchronization barriers. One rollout per prompt avoids these constraints but requires learning from stale, variable-length trajectories without a sibling baseline. We present FLASHREINFORCE, a critic-free, one-pass framework built on One-Batch REINFORCE, a Sequence Trust Region, and Sample-Mean Optimization. These components preserve prompt coverage and signed feedback while controlling policy drift and length-amplified negative updates. In our experiments, FLASHREINFORCE remains stable through 6,000 updates on DeepSeek-R1-Distill-Qwen-1.5B at a policy lag of approximately four updates. On Qwen2.5-Math-1.5B, it reaches 38.0 mean accuracy across five benchmarks, outperforming the reported GRPO baseline by 1.7 percentage points with half the rollouts (256k versus 512k). Python-tool experiments extend to the 30B MoE model Qwen3-30B-A3B, with stable training at policy lag eight. On Qwen2.5-7B-Instruct, FLASHREINFORCE maintains tool use where GRPO stops making tool calls and achieves 98.3%/96.5% seen/unseen ALFWorld success.

1 INTRODUCTION

In long-horizon agent RL, tool calls, environment steps, and variable-length reasoning make rollout durations unpredictable (Noukhovitch et al., 2025; Fu et al., 2025; Wu et al., 2025). Asynchronous workers can submit each completed trajectory immediately, but methods such as GRPO estimate advantages from groups of sibling rollouts (Shao et al., 2024). However, group-based sampling limits prompt coverage at a fixed rollout budget, introduces group synchronization barriers, and is difficult to apply to online interactions that yield only a single trajectory. A budget of G rollouts could instead cover G distinct prompts, with each trajectory available for learning as soon as it finishes.

The difficulty is learning reliably from one rollout per prompt. We instead center rewards across prompts. With binary rewards, positive-only feedback gives failures no update; signed feedback allows them to contribute, but applies a coarse penalty to the whole trajectory. A failed trajectory can contain useful intermediate steps, yet its negative advantage penalizes every token (Sec. C). Length weighting then gives long failures extra influence. The challenge is to retain negative feedback without letting trajectory length amplify it.

Asynchrony adds a second challenge: completed trajectories may come from different policy snapshots, leaving both action-probability and history-distribution mismatch with the learner. Token importance sampling (IS) corrects actions at stored histories, but those histories still follow the behavior policy. Token correction therefore leaves a reason to consider drift over the whole trajectory. Our analysis relates the remaining surrogate error to accumulated policy drift, motivating sequence-level trust alongside IS.

FLASHREINFORCE combines three complementary components (Fig. 1). *One-Batch REINFORCE* provides signed feedback by centering rewards across independent prompts. The *Sequence Trust Region* complements token IS with trajectory-level drift screening. *Sample-Mean Optimization*

[1]Code and configurations will be released after the anonymous review period.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

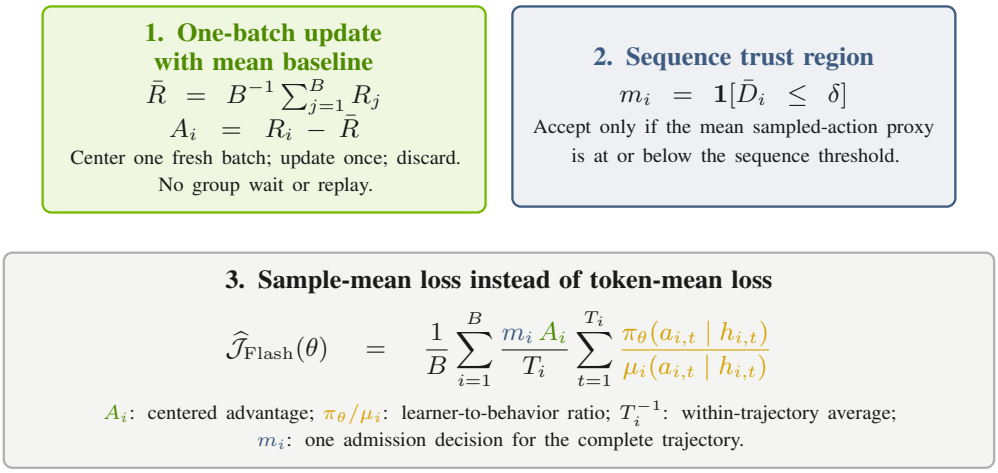


Figure 1: **FlashREINFORCE at a glance.** Each fresh batch supplies its baseline, receives one update, and is discarded. Sequence trust screens complete trajectories; sample mean limits length-amplified negative updates.

normalizes trajectory contributions to limit length-amplified negative feedback. Together, these components enable critic-free, one-pass asynchronous updates without waiting for sibling rollouts.

Experiments span reasoning, tool use, and interactive decision making. FLASHREINFORCE remains stable through 6,000 updates on DeepSeek-R1-Distill-Qwen-1.5B at policy lag ≈ 4. On Qwen2.5-Math-1.5B, its five-benchmark mean of 38.0 exceeds the reported GRPO baseline by 1.7 points with half the rollouts. Python-tool experiments extend to the 30B MoE Qwen3-30B-A3B, with stable performance through 1,450 updates at lag eight. On Qwen2.5-7B-Instruct, FLASHREINFORCE maintains tool use where GRPO stops making tool calls and reaches 98.3%/96.5% seen/unseen ALFWorld success. Ablations examine update freshness, signed feedback, trajectory weighting, and admission granularity.

Contributions. Our contributions are:

1. **One-Batch REINFORCE** for critic-free single-rollout learning across independent prompts.
2. **Sequence Trust Region** motivated by accumulated policy drift, alongside token IS.
3. **Sample-Mean Optimization** for limiting length-amplified negative feedback.
4. **Stable Asynchronous Training** through 6,000 updates at lag ≈ 4, with 30B MoE results at lag eight, reasoning and agent evaluations, and component ablations.

2 SETUP AND THE LOCAL OFF-POLICY OBJECTIVE

2.1 ASYNCHRONOUS ROLLOUT DATA

Rollout workers periodically receive policy snapshots, sample one trajectory per prompt, and submit trajectories independently as they finish. The learner batches the next B completions, which may span several behavior snapshots. Here, *fresh* means previously unused for an update, including trajectories from older snapshots. *Staleness* denotes behavior-policy age relative to the learner; *policy drift* denotes the distributional difference screened by the gate.

For prompt x_i, let τ_i contain policy-visible histories $h_{i,t}$, generated tokens $a_{i,t}$ for $t = 1, \ldots, T_i$, and a scalar terminal reward $R_i \in \mathbb{R}$. Let $\mu_i(a_{i,t} \mid h_{i,t})$ denote the behavior probability that actually generated token $a_{i,t}$; we store it with the trajectory. Recomputing an "old" probability later need not recover the sampling probability because the learner and inference engine may already differ (Zhang et al., 2026a; Zhong et al., 2026; Zhang et al., 2025). At learner parameters θ, define

$$\rho_{i,t}(\theta) = \exp(\log \pi_\theta(a_{i,t} \mid h_{i,t}) - \log \mu_i(a_{i,t} \mid h_{i,t})) \,. \tag{1}$$

2.2 LOCAL OFF-POLICY SURROGATE

Write $J(\pi) = \mathbb{E}_{\tau\sim\pi}[R(\tau)]$ for the return, d_μ^t for the distribution of the history h_t under a behavior policy μ, and $Q_t^\mu(h,a) = \mathbb{E}_\mu[R \mid h_t = h, a_t = a]$ for the behavior continuation value. Define $V_t^\mu(h) = \mathbb{E}_\mu[R \mid h_t = h]$ and $A_t^\mu(h,a) = Q_t^\mu(h,a) - V_t^\mu(h)$. Holding these behavior quantities fixed, define the local surrogate

$$\mathcal{L}_\mu(\pi) = J(\mu) + \sum_t \mathbb{E}_{h\sim d_\mu^t}\, \mathbb{E}_{a\sim\pi(\cdot|h)}\big[A_t^\mu(h,a)\big]. \tag{2}$$

The additive $J(\mu)$ makes $\mathcal{L}_\mu(\mu) = J(\mu)$. Histories and continuation values remain behavioral. Under common support, differentiating the inner expectation, replacing the continuation value with the sampled return, and changing measure from π to μ gives

$$\nabla_\theta \mathcal{L}_\mu(\pi_\theta) = \mathbb{E}_{\tau\sim\mu}\Big[\sum_t \rho_t(\theta)\, R(\tau)\, \nabla_\theta \log \pi_\theta(a_t \mid h_t)\Big], \tag{3}$$

Token IS in Eq. (3) exactly corrects the conditional action distribution at each stored history. The histories still follow d_μ^t, whereas $J(\pi_\theta)$ depends on $d_{\pi_\theta}^t$, leaving a history-distribution mismatch. For batches with multiple behavior snapshots, the identity applies separately to each trajectory's μ_i before averaging.

Full-sequence or prefix ratios can correct history occupancy, but multiplying ratios across a long trajectory can produce high variance. We retain the token ratio and characterize the remaining occupancy error through accumulated policy movement. For the analysis, use a fixed policy-token horizon T, with absorbing padding after termination. Let $\varepsilon_t = \sup_h D_{\mathrm{TV}}(\pi(\cdot \mid h), \mu(\cdot \mid h))$, $\kappa_t = \sup_h D_{\mathrm{KL}}(\mu(\cdot \mid h) \,\|\, \pi(\cdot \mid h))$, and $\bar\kappa = T^{-1}\sum_{t=1}^T \kappa_t$. If $|A_t^\mu(h,a)| \le A_{\max}$, then

$$|J(\pi) - \mathcal{L}_\mu(\pi)| \le 2A_{\max}\Big(\sum_t \varepsilon_t\Big)^2 \le A_{\max} T^2 \bar\kappa \tag{4}$$

(Prop. B.2). This follows the classical coupling argument reviewed by Li et al. (2026, Sec. 3.1). As in TRM, accumulated drift motivates sequence-level admission: deleting individual token-loss terms leaves the stored histories used by other terms unchanged. We implement this screen with a computable proxy.

3 FLASHREINFORCE

Building on this local off-policy view, FLASHREINFORCE combines fresh-batch reward centering, sequence-level trust, and trajectory-normalized updates.

3.1 ONE-BATCH REINFORCE WITH REWARD CENTERING

With one rollout per prompt, a sibling-rollout baseline is unavailable; with binary rewards, uncentered REINFORCE also gives failures no learning signal. FLASHREINFORCE therefore centers rewards over the next B completed trajectories to obtain signed feedback without a critic,

$$\bar R = \frac{1}{B}\sum_{j=1}^B R_j, \qquad A_i = R_i - \bar R. \tag{5}$$

For binary rewards with both outcomes present, $A_i < 0$ identifies failures; for general scalar rewards, it indicates below-batch-mean return. Each fresh batch receives one full-batch update and is discarded, avoiding additional drift from sequential updates on the same collected batch.

For on-policy leave-one-out centering, Hu (2026) shows that the benefit of a prompt-local baseline competes with estimation noise and reduced prompt coverage at a fixed rollout budget. Our 128×1 versus 32×4 comparison shows comparable learning over approximately 4,000 updates (Sec. A.1.3).

3.2 BEHAVIOR-CORRECTED SEQUENCE TRUST REGION

The accumulated-drift bound in Eq. (4) motivates screening complete trajectories, complementing token IS with sequence-level control. Seq-MIS uses sequence importance ratios for filtering (Liu et al., 2025a), while Trust Region Masking (TRM) also motivates sequence-level admission (Li et al., 2026).

Ratio thresholds measure relative change. As DPPO explains, they can over-restrict low-probability token updates that move little probability mass, yet admit larger mass shifts on high-probability tokens (Qi et al., 2026, Sec. 4.2). Following this divergence-based motivation, FLASHREINFORCE uses a KL proxy for trajectory admission while retaining token ratios for behavior correction.

For each sampled token, we collapse the behavior and learner distributions to the sampled action versus the remaining vocabulary mass, giving the Bernoulli KL proxy also studied in DPPO (Qi et al., 2026, Sec. 4.4),

$$d_{i,t} = p_{i,t} \log \frac{p_{i,t}}{q_{i,t}} + (1 - p_{i,t}) \log \frac{1 - p_{i,t}}{1 - q_{i,t}}, \tag{6}$$

where $p_{i,t} = \mu_i(a_{i,t} \mid h_{i,t})$ and $q_{i,t} = \pi_\theta(a_{i,t} \mid h_{i,t})$. This reward-independent proxy is a lower bound on categorical KL by the data-processing inequality and uses only the behavior probability already stored for IS, avoiding storage of full behavior distributions.

To compare drift across variable-length trajectories on a per-token scale, we use the mean proxy,

$$\bar{D}_i = \frac{1}{T_i} \sum_{t=1}^{T_i} d_{i,t}, \tag{7}$$

and define the stop-gradient sequence mask

$$m_i = \mathbf{1}[\bar{D}_i \le \delta]. \tag{8}$$

Rejection removes a trajectory's complete stored-history contribution when its proxy exceeds δ, which also implies mean full KL above δ (Sec. B). Section 4.3.3 evaluates admission granularity.

3.3 SAMPLE-MEAN OPTIMIZATION

Sequence trust selects trajectories; sample-mean reduction weights their contributions. Negative advantages penalize every token in a failed rollout, including useful steps. Token-mean reduction gives longer failures extra weight, amplifying this coarse feedback. FLASHREINFORCE instead averages within each trajectory before averaging across trajectories, retaining signed feedback while removing the explicit length multiplier:

$$\widehat{\mathcal{J}}_{\text{FlashREINFORCE}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \frac{m_i A_i}{T_i} \sum_{t=1}^{T_i} \frac{\pi_\theta(a_{i,t} \mid h_{i,t})}{\mu_i(a_{i,t} \mid h_{i,t})}, \tag{9}$$

Applying this normalization to the local gradient in Eq. (3), with A_i and m_i held fixed, gives the update direction

$$\widehat{g}_{\text{FlashREINFORCE}}(\theta) = \frac{1}{B} \sum_{i=1}^{B} \frac{m_i A_i}{T_i} \sum_{t=1}^{T_i} \rho_{i,t}(\theta) \nabla_\theta \log \pi_\theta(a_{i,t} \mid h_{i,t}). \tag{10}$$

An equivalent score-function formulation uses the detached-ratio loss (Sec. B),

$$\mathcal{L}(\theta) = -\frac{1}{B} \sum_{i=1}^{B} \frac{m_i A_i}{T_i} \sum_{t=1}^{T_i} \text{stopgrad}(\rho_{i,t}) \log \pi_\theta(a_{i,t} \mid h_{i,t}) \tag{11}$$

whose negative gradient is given by Eq. (10). The update uses no PPO-style ratio clipping, critic, or reference-model forward pass.

Sample mean gives each trajectory weight $1/B$, whereas token mean weights it by $T_i / \sum_j T_j$. The exact reduction formulas and optional token-selection details are in Secs. B and C.

Algorithm 1 The one-pass FLASHREINFORCE update.

Require: completed trajectories $\mathcal{B}$, current policy π_θ, admission threshold δ

1. Batch $A_i \leftarrow R_i - B^{-1} \sum_j R_j$

2. Trust
$\rho_{i,t} \leftarrow \exp(\log \pi_\theta(a_{i,t} \mid h_{i,t}) - \log \mu_i(a_{i,t} \mid h_{i,t}))$
$\bar{D}_i \leftarrow T_i^{-1} \sum_t d_{i,t}$
$m_i \leftarrow \mathbf{1}[\bar{D}_i \le \delta]$

3. Update take one optimizer step on the sample-mean loss in Eq. (11), with A_i, m_i, and μ_i held fixed

The local-gradient identity motivates conditional-action correction. Batch centering, fixed-mask admission, and trajectory normalization define the practical surrogate.

4 EXPERIMENTS

We first evaluate sustained asynchronous training with DeepSeek-R1-Distill-Qwen-1.5B, then compare FLASHREINFORCE with critic-free baselines on reasoning and agent tasks. Unless stated otherwise, FLASHREINFORCE uses the one-pass update in Alg. 1. Published results in Tabs. 1 and 4 come from Wu et al. (2026), whose Contrastive-REINFORCE with Negative Token Filtering (C-RF + NTF) contrasts successful and failed trajectories and masks high-probability tokens in failures. Their baselines were trained synchronously with veRL (Sheng et al., 2024).

4.1 EVALUATION SETTINGS AND ROLLOUT BUDGETS

We evaluate long-CoT reasoning, mathematical reasoning, Python tool use, and ALFWorld. We report avg@k (mean accuracy over k rollouts per problem) and equally weighted benchmark means. Rollout budgets accompany the results; full datasets, sampling protocols, and batch configurations are in Sec. A.

4.2 MAIN RESULTS

4.2.1 STABLE ASYNCHRONOUS LONG-COT TRAINING AT LAG 4

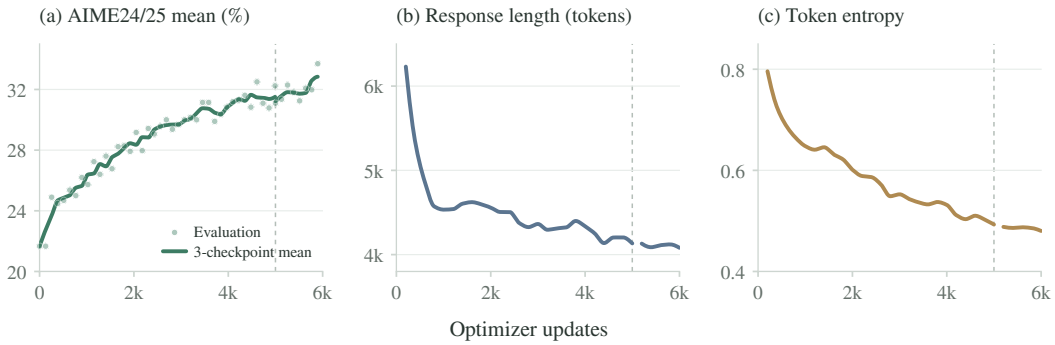


Figure 2: DeepSeek-R1-Distill-Qwen-1.5B with sequence trust at policy lag ≈ 4, through 6,000 cumulative updates. The dashed line marks continuation from the step-5,000 checkpoint (Sec. A.1). (a) AIME24/25 mean avg@32 accuracy: dots mark evaluations and curves show a three-checkpoint moving average within each segment. (b,c) Training response length and token entropy, averaged in consecutive 200-update windows.

In the reported long-CoT run, FLASHREINFORCE remains stable through 6,000 cumulative updates at policy lag ≈ 4, improving its AIME24/25 mean from 21.7 to 33.7 at the last evaluation within this horizon (Fig. 2). Response length decreases and entropy settles as learning progresses. MoE tool-use experiments provide further stability evidence at lag eight (Sec. 4.2.4). Section 4.3.3 discusses the trust comparison.

4.2.2 COMPARISON WITH CRITIC-FREE REASONING BASELINES

Table 1: Five-benchmark comparison on Qwen2.5-Math-1.5B. FLASHREINFORCE reports its peak-mean checkpoint at step 2,000 using 256k rollouts; published baselines from Wu et al. (2026) use 512k. Boldface marks the best reported task score and mean.

Method	G	Step	Rollouts	MATH-500	AMC23	Minerva	AIME25	Olympiad	Mean
Base model	–	0	0	35.1	30.9	6.7	5.4	19.4	19.5
C-RF + NTF (Wu et al., 2026)	1	1,000	512k	69.6	55.5	18.4	9.4	23.9	35.3
GRPO (Wu et al., 2026)	16	1,000	512k	**71.0**	**57.1**	18.2	10.0	24.9	36.3
FLASHREINFORCE	1	2,000	256k	69.2	52.5	**22.8**	**10.8**	**34.5**	**38.0**

FLASHREINFORCE exceeds the best baseline mean by 1.7 points with half the training trajectories (Tab. 1).

4.2.3 MULTI-TURN MATHEMATICAL TOOL USE

Table 2: Python-tool accuracy (%) at matched steps and rollout budgets. Calls per trajectory are averaged over the 100 updates ending at each checkpoint. In (b), FLASHREINFORCE uses $\delta = 10^{-2}$ without routing replay at lag ≈ 8; GRPO uses lag ≈ 1. Boldface marks the better task score or mean.

Method	Step	AMC23	Minerva	AIME25	Mean	Tool calls
(a) Qwen2.5-7B-Instruct						
GRPO	600	51.2	**32.3**	7.5	30.3	0.00
FLASHREINFORCE	600	**60.6**	28.8	**21.7**	**37.0**	3.25
(b) Qwen3-30B-A3B (MoE), avg@4						
GRPO	800	93.8	40.4	46.7	60.3	1.19
FLASHREINFORCE	800	**95.6**	**44.9**	**60.8**	**67.1**	1.94

On Qwen2.5-7B-Instruct, GRPO stops calling the tool within roughly 200 steps; FLASHREINFORCE maintains 3.25 calls per trajectory and leads the mean by 6.7 points (Tab. 2).

4.2.4 SCALING TO A 30B MOE AT LAG 8

On Qwen3-30B-A3B, FLASHREINFORCE leads GRPO by 6.8 points at the same 102.4k-rollout budget, despite policy lag ≈ 8 versus ≈ 1 (Tab. 2). To examine longer-run stability, we compare three FLASHREINFORCE configurations at lag eight with GRPO at lag one (Fig. 3). The routing-replay variant, R3, uses $\delta = 10^{-3}$; full training settings are in Sec. A.2.2.

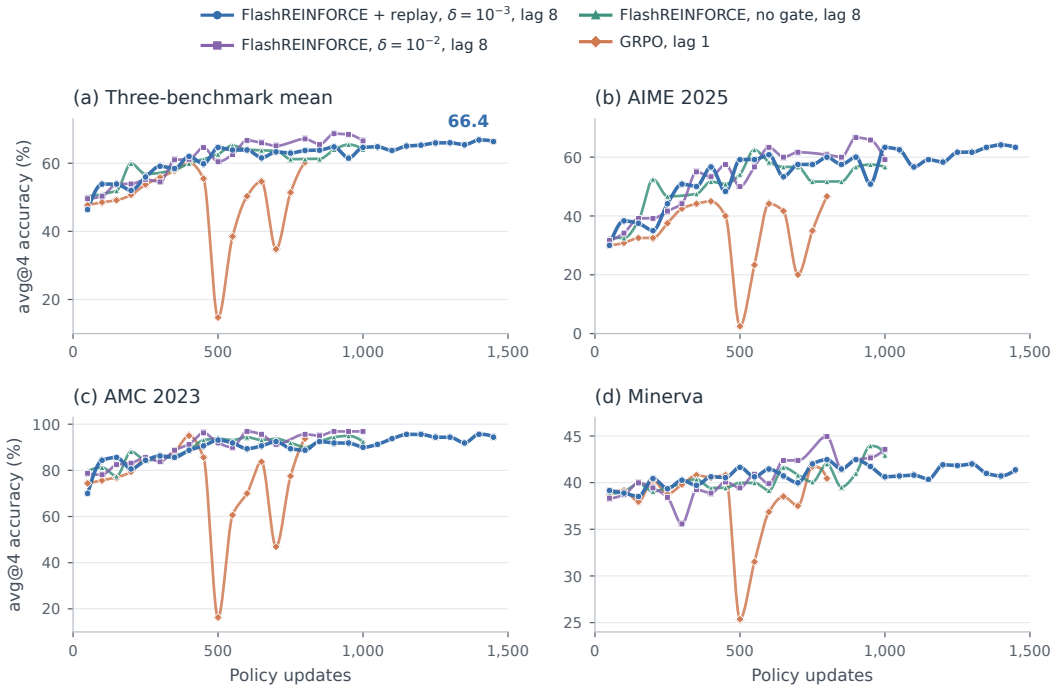


Figure 3: Qwen3-30B-A3B Python-tool accuracy: FLASHREINFORCE at lag eight and GRPO at lag one. Markers show avg@4 evaluations, joined by shape-preserving interpolation; the mean equally weights the three benchmarks. R3 is shown through 1,450 policy updates.

R3 maintains a three-benchmark mean of 63.7–66.8 over policy updates 1,000–1,450, ending at 66.4. GRPO suffers two sharp drops before recovering to 60.3 at step 800. At the shared step-1,000 checkpoint, all three FLASHREINFORCE configurations sustain tool use (Tab. 3).

Table 3: MoE configuration comparison after 1,000 policy updates, all at policy lag eight. Accuracy is avg@4 (%), with an equally weighted three-task mean; calls per trajectory and truncation rate (%) are averaged over updates 901–1,000. Replay denotes MoE routing replay.

Gate δ	Replay	AMC23	Minerva	AIME25	Mean	Calls	Trunc.
10^{-2}	Off	96.9	43.6	59.2	66.5	1.76	7.2
10^{-3}	On	90.0	40.6	63.3	64.7	1.95	2.8
No gate	Off	92.5	42.9	56.7	64.0	1.80	7.6

4.2.5 Interactive decision making in ALFWorld

Table 4: ALFWorld success rate (%). Published rows are from Table 2 of Wu et al. (2026); FlashReinforce is evaluated at step 200 after 12.8k training trajectories. Boldface marks the best value.

Method	Source	Seen	Unseen
GRPO, $G = 8$	Published	78.6	76.8
C-RF + NTF, $G = 1$	Published	90.5	86.3
FlashReinforce, $G = 1$	Ours	**98.3**	**96.5**

FlashReinforce improves seen/unseen success by 7.8/10.2 points over the strongest published baseline (Tab. 4). C-RF also uses $G = 1$, so prompt diversity alone does not explain the gain.

4.3 Ablation studies

4.3.1 Fresh one-batch updates

We compare collecting 128 completed trajectories per update with collecting 512 under one behavior policy and processing four sequential minibatches of 128. Both schedules use each trajectory once with the same IS correction and gate. In the sequential schedule, the remaining three minibatches encounter a learner that has advanced one to three updates beyond their behavior policy.

At equal data and optimization budget (102.4k trajectories and 800 steps), fresh updates reach 28.45 three-task mean accuracy versus 26.80 for sequential minibatches (Tab. 5).

Table 5: Fresh one-batch updates versus sequential minibatches on Qwen2.5-Math-1.5B. Both rows use 102.4k trajectories and 800 optimizer steps; the metric is the mean accuracy on AMC23, Minerva, and AIME25.

Update schedule	Trajectories	Updates	Three-task mean
Fresh batch for every update	102.4k	800	**28.45**
Four sequential minibatches per rollout batch	102.4k	800	26.80

4.3.2 Mean baseline and loss reduction

Signed feedback from the mean baseline. We examine negative feedback from reward-zero trajectories. At a fixed batch mean, omitting their negative-advantage terms leaves a positive-only $0/1$-equivalent update, up to the batch-wide scale $1 - \bar{R}$. Panel (a) of Tab. 6 compares signed and positive-only feedback.

Signed feedback reaches 36.2 mean accuracy with 3.30 tool calls per trajectory; the positive-only variant reaches 9.8 and makes no tool calls.

Table 6: Ablations on Qwen2.5-7B-Instruct tool use. Panel (a) compares the learning signal at step 700; panel (b) summarizes observed late-training states after step 900.

(a) Signed-feedback ablation

Learning signal	AMC23	Minerva	AIME25	Mean	Tool calls
Batch-centered signed advantage	**57.5**	**30.2**	**20.8**	**36.2**	**3.30**
Positive-only $0/1$-equivalent update	12.5	16.9	0.0	9.8	0.00

(b) Late-training continuation after step 900

State	Extra updates	Reward	Length	Tools	Trunc. (%)
Step-900 reference	0	0.430	2,573	3.87	18.8
Sample-mean continuation	~ 150	**0.477**	**2,518**	**4.01**	**17.2**
Token-mean collapse	~ 150	0.403	3,947	2.18	45.3

Loss-reduction continuation. To test trajectory weighting (Eq. (18)), we continue both reductions from the step-900 checkpoint for roughly 150 steps. Sample mean maintains reward and tool use, whereas token mean enters a long-response collapse: responses grow to 3,947 tokens and truncation rises to 45.3%, alongside lower reward and fewer tool calls (Tab. 6, panel (b)). This favors averaging within each trajectory over weighting by length.

4.3.3 Sequence-level trust region

DPPO shows that *IS alone can fail as training continues*: its PG-IS baseline collapses as training–inference mismatch grows, even at very small learning rates (Qi et al., 2026, Sec. 5.1 and Figure 3). Our Qwen2.5-Math comparison further shows that masking granularity matters: token-level masking destabilizes training, while sequence-level admission remains stable (Tab. 7).

Behavior correction and sequence trust. On ALFWorld, removing both IS and sequence trust drops step-40 seen/unseen success from 81.0/77.6 to 35.0/27.6. This joint ablation evaluates their combined contribution; the following comparisons examine the gate's design.

Admission granularity. We compare token-local and sequence-level admission at the same threshold, $\delta = 1 \times 10^{-3}$, to test the granularity of the trust decision.

Table 7: Sequence- versus token-local trust on Qwen2.5-Math-1.5B, using the same divergence proxy and threshold. Token-local keeps t iff $d_{i,t} \leq \delta$; sequence-level thresholds the mean.

Trust region	Last step	AMC23 peak $\rightarrow$ last	Train reward	Entropy	Outcome
Sequence-level	**2,160**	**55.6 $\rightarrow$ 53.1**	**0.23**	**0.20**	**Stable**
Token-local	1,237	51.9 $\rightarrow$ 26.3	0.18 $\rightarrow$ 0.01	0.37 $\rightarrow$ 5.6	Collapsed

The token-local collapse coincides with rising entropy and falling reward. Section B explains the structural distinction: token gating leaves some contributions at the same stored histories, while sequence admission removes the rejected trajectory's complete contribution.

On R1 at lag four, sequence trust reaches 33.7 AIME24/25 mean accuracy at step 5,896, while continued training without trust remains at 30.6 with higher measured drift (Tab. 8). The with-trust continuation reaches a new AIME24 peak of 38.85 at step 5,896 (Tab. 10, appendix).

Table 8: R1 trust ablation at policy lag 4 with token IS. Accuracy: AIME24/25 avg@32 mean. Train–inference KL is the sampled-action proxy between learner and rollout policies, averaged over updates 4,801–5,000 (first two rows) or 5,801–6,000 (last two). Experimental settings are in Sec. A.1.

Setting	Eval step	Accuracy (%)	Train–inference KL ($\times 10^{-4}$)
With sequence trust	4,992	**32.2**	**6.73**
Without trust	4,992	30.6	7.38
With sequence trust (continued)	5,896	**33.7**	**8.96**
Without trust (continued)	5,888	30.6	9.45

Gate threshold. Both $\delta = 1 \times 10^{-3}$ and $\delta = 3 \times 10^{-3}$ retain stable Qwen2.5-Math training through 4,000 steps (Sec. A.3.3). The 3×10^{-3} comparison run rejects only 0.13% of trajectories, consistent with screening drift outliers while retaining ordinary IS correction.

Gate aggregation. Arithmetic and geometric means each lead on one reported benchmark (Sec. A.3.4); we retain the arithmetic mean.

5 Related Work

Baselines for language-model RL. REINFORCE uses score-function updates, with action-independent baselines reducing variance (Williams, 1992; Sutton et al., 2000; Greensmith et al., 2004). RLOO and GRPO use sibling-rollout baselines (Ahmadian et al., 2024; Shao et al., 2024); REINFORCE++, DAPO, and CISPO refine normalization, clipping, and sampling (Hu et al., 2025; Yu et al., 2025; MiniMax et al., 2025). C-RF + NTF protects high-probability tokens from negative feedback (Wu et al., 2026). SAO uses a learned value baseline for asynchronous single-rollout training (Hou et al., 2026), while SPO maintains per-prompt statistics (Xu & Ding, 2025).

Trust regions and divergence control. Natural policy gradient and TRPO motivate local divergence control, while PPO uses sampled-ratio clipping (Amari, 1998; Kakade, 2001; Schulman et al., 2015; 2017). DPPO studies divergence-based proximal updates (Qi et al., 2026); GSPO uses sequence ratios and cumulative-token policy optimization studies prefix correction (Zheng et al., 2025; Zhang et al., 2026b). Seq-MIS and Trust Region Masking motivate sequence-level admission (Liu et al., 2025a; Li et al., 2026).

Asynchronous and off-policy learning. IMPALA established importance-corrected actor–learner training (Espeholt et al., 2018). Asynchronous RLHF, AReaL, and LlamaRL study throughput, staleness, and synchronization for language models (Noukhovitch et al., 2025; Fu et al., 2025; Wu et al., 2025). GiGPO studies long-horizon credit assignment (Feng et al., 2025). Training–inference mismatch work explains why recomputed "old" probabilities may differ from the probabilities that generated a token (Zhong et al., 2026; Zhang et al., 2025; 2026a).

6 Conclusion

FlashREINFORCE combines batch-centered feedback, sequence trust, and sample-mean weighting in a critic-free single-rollout update. Our experiments demonstrate stable long-CoT training through 6,000 asynchronous updates at lag ≈ 4, improved reasoning accuracy with half the rollouts, and tool use with a 30B MoE at lag eight. Component comparisons clarify the complementary roles of fresh updates, signed feedback, trajectory weighting, and sequence admission.

Limitations

The sampled-action proxy can miss probability shifts among unobserved actions (Sec. B). Our interactive evaluations cover Python tool use and ALFWorld; extending FlashREINFORCE to environments such as OSWorld (Xie et al., 2024) remains future work.

AI USE STATEMENT

Generative AI tools were used to assist with manuscript editing, language polishing, and consistency checks. They were not used to run experiments or generate the reported data. The authors independently verified the analysis and technical claims, reviewed all AI-assisted text, and take full responsibility for the submission.

ETHICS STATEMENT

This work does not involve human participants or private personal data. It studies training efficiency for mathematical reasoning and tool-using language models. Such capability improvements can support useful automation but may also increase the impact of unsafe deployments; model users remain responsible for task-specific evaluation, access control, and monitoring.

REPRODUCIBILITY STATEMENT

The update is fully specified in Sec. 3 and Alg. 1; training, evaluation, rollout-budget, checkpoint, and long-horizon details are reported in Secs. A, A.3 and 4.1. Code and configurations will be released after the anonymous review period, as stated on the first page.

REFERENCES

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting REINFORCE style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 2024.

Shun-ichi Amari. Natural gradient works efficiently in learning. *Neural Computation*, 10(2):251–276, 1998.

DeepSeek-AI. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Lasse Espeholt, Hubert Soyer, Rémi Munos, Karen Simonyan, Volodymyr Mnih, Tom Ward, Yotam Doron, Vlad Firoiu, Tim Harley, Iain Dunning, Shane Legg, and Koray Kavukcuoglu. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. *arXiv preprint arXiv:2505.10978*, 2025.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Evan Greensmith, Peter L. Bartlett, and Jonathan Baxter. Variance reduction techniques for gradient estimates in reinforcement learning. *Journal of Machine Learning Research*, 5:1471–1530, 2004.

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

Jian Hu. Group versus batch baseline under a fixed sampling budget. alphaXiv preprint, 2026. URL https://www.alphaxiv.org/abs/2609.group-batch-means-sampling-budget. Version 2.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. REINFORCE++: Stabilizing critic-free policy optimization with global advantage normalization. *arXiv preprint arXiv:2501.03262*, 2025.

Sham M. Kakade. A natural policy gradient. In *Advances in Neural Information Processing Systems 14*, 2001.

Yingru Li, Jiacai Liu, Jiawei Xu, Yuxuan Tong, Ziniu Li, Qian Liu, and Baoxiang Wang. Trust region masking for long-horizon LLM reinforcement learning. *arXiv preprint arXiv:2512.23075*, 2026.

Jiacai Liu, Yingru Li, Yuqian Fu, Jiawei Wang, Qian Liu, and Yu Shen. When speed kills stability: Demystifying RL collapse from the training–inference mismatch. Technical article, 2025a. URL https://richardli.xyz/rl-collapse. Introduces Sequence-Level Masked Importance Sampling (Seq-MIS).

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

MiniMax et al. MiniMax-M1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and more efficient off-policy RL for language models. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

Qwen Team. Qwen3.5: Towards native multimodal agents. Technical blog and model release, 2026. URL https://qwen.ai/blog?id=qwen3.5.

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*, 2015.

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

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

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

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew J. Hausknecht. ALFWorld: Aligning text and embodied environments for interactive learning. In *The Ninth International Conference on Learning Representations*, 2021.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. In *Advances in Neural Information Processing Systems 12*, 2000.

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

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4):229–256, 1992.

Bo Wu, Sid Wang, Yunhao Tang, Jia Ding, Eryk Helenowski, Liang Tan, Tengyu Xu, Tushar Gowda, Zhengxing Chen, Chen Zhu, Xiaocheng Tang, Yundi Qian, Beibei Zhu, and Rui Hou. LlamaRL: A distributed asynchronous reinforcement learning framework for efficient large-scale LLM training. *arXiv preprint arXiv:2505.24034*, 2025.

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

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. OSWorld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *arXiv preprint arXiv:2404.07972*, 2024.

Zhongwen Xu and Zihan Ding. Single-stream policy optimization. *arXiv preprint arXiv:2509.13232*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, et al. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yifan Zhang et al. Reliable RL scaling requires accounting for prefill-decode kernel mismatch, 2026a. Technical report.

Yuheng Zhang, Chenlu Ye, Shuowei Jin, Changlong Yu, Wei Xiong, Saurabh Sahu, and Nan Jiang. Rethinking importance sampling in LLM policy optimization: A cumulative token perspective. *arXiv preprint arXiv:2605.07331*, 2026b.

Ziyang Zhang, Xinheng Ding, Jiayi Yuan, Rixin Liu, Huizi Mao, Jiarong Xing, and Zirui Liu. Deterministic inference across tensor parallel sizes that eliminates training–inference mismatch. *arXiv preprint arXiv:2511.17826*, 2025.

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.

Tianle Zhong, Neiwen Ling, Yifan Pi, Zijun Wei, Tianshu Yu, Geoffrey Fox, Peng Wu, and Xiao Yu. Diagnosing training inference mismatch in LLM reinforcement learning. *arXiv preprint arXiv:2605.14220*, 2026.

A Experimental Details and Additional Results

Implementation details. For numerical stability, training computes the learner-to-behavior log-ratio in float32, clamps it to $[-30, 30]$, and exponentiates it. The policy loss differentiates directly through this clamped ratio, with advantages, behavior probabilities, and selection masks held fixed. Its gradient agrees with the detached-ratio formulation in Eq. (11) inside the clamp interval; tokens outside the interval have zero policy-loss gradient. The numerical guard preserves the batch-size and trajectory-length denominators.

Evaluation protocols and rollout budgets. We use avg@k for mean per-sample accuracy over k rollouts per problem; benchmark means weight tasks equally.

Long-CoT mathematical reasoning. We train DeepSeek-R1-Distill-Qwen-1.5B (DeepSeek-AI, 2025) on DPPO's 1,460-problem MATH subset (Qi et al., 2026), at policy lag ≈ 4 with 128 trajectories per update. We report mean AIME24/25 avg@32 accuracy; hyperparameters and trust-comparison details appear in Sec. A.1.

Reasoning baseline comparison. We train Qwen2.5-Math-1.5B (Yang et al., 2024) on 7.5k deduplicated DAPO-Math prompts (Yu et al., 2025) and report avg@16 on MATH-500, AMC23, Minerva, AIME2025, and OlympiadBench. Settings are in Tab. 13.

Multi-turn tool use. We train Qwen2.5-7B-Instruct and the 30B MoE Qwen3-30B-A3B with a Python tool (Qwen Team, 2025; Yang et al., 2025). Each update uses 128 rollouts: 128 prompts sampled once for FlashREINFORCE, or 32 prompts sampled four times for GRPO. We evaluate committed answers on AMC23, Minerva, and AIME2025; settings are in Sec. A.2.

Interactive decision making. We train Qwen2.5-7B-Instruct on 1,024 ALFWorld games with 64 trajectories per update and evaluate 140 seen and 134 unseen games, matching the published protocol (Shridhar et al., 2021; Wu et al., 2026). Settings are in Tab. 17.

A.1 Long-CoT reasoning: DeepSeek-R1-Distill-Qwen-1.5B

A.1.1 Training and evaluation settings

Table 9: Hyperparameters for the DeepSeek-R1-Distill-Qwen-1.5B trust comparison.

Data and sampling	
Training data	`sail/Sanity-Test-R1D-1.5B`: 1,460 MATH problems with initial solve rates of 20–80% (Qi et al., 2026)
Token budget	Prompt: 1,024; response: 8,192; total: 9,216 tokens
Rollout sampling	Temperature 1.0; top-$p = 1.0$
Evaluation	AIME24 and AIME25; 30 problems each; 32 samples per problem; every 128 updates from step 0
Evaluation sampling	Temperature 0.6; top-$p = 0.95$; 8,192 generated tokens
Update and optimization	
Online update	128 prompts, one rollout each; batch-mean baseline; sample-mean loss; token IS; no negative-token filtering
Optimizer	AdamW; peak lr 10^{-6}; weight decay 0.1; gradient clip 1.0
LR schedule	Cosine decay with extended horizon; 3% initial warmup; minimum lr 10^{-7}
Asynchronous collection and sequence trust	
Collection	512-prompt generation requests; 128-trajectory learner batches; queue depth 8 (policy lag ≈ 4)
Trust threshold cap	5×10^{-3}

A.1.2 TRUST COMPARISON AND CONTINUED TRAINING

Table 10: AIME avg@32 accuracy (%). (a) Trust comparison through 5,000 updates. (b) Continued training within 6,000 cumulative updates; approximate steps align nearby evaluations across arms. Mean equally weights AIME24 and AIME25; Trunc. is the percentage of evaluation responses reaching the 8,192-token cap. Boldface marks the with-trust run's AIME24 peak.

(a) With versus without sequence trust

	With sequence trust			Without trust (IS only)		
Step	**AIME24**	**AIME25**	**Mean**	**AIME24**	**AIME25**	**Mean**
0	22.71	20.63	21.67	22.71	20.42	21.56
1,024	28.13	23.33	25.73	27.60	24.06	25.83
2,048	31.98	26.35	29.17	28.85	25.63	27.24
3,072	32.08	27.92	30.00	33.13	26.46	29.79
3,968	35.42	26.25	30.83	34.58	26.77	30.68
4,992	36.15	28.33	32.24	33.85	27.40	30.63

(b) Continued training beyond 5,000 updates

	With sequence trust				Without trust (IS only)			
Step (≈)	**AIME24**	**AIME25**	**Mean**	**Trunc.**	**AIME24**	**AIME25**	**Mean**	**Trunc.**
5,100	34.79	27.92	31.35	42%	31.98	26.25	29.11	48%
5,300	35.10	29.48	32.29	41%	33.96	28.33	31.15	47%
5,400	36.56	27.08	31.82	40%	32.81	27.19	30.00	48%
5,500	35.00	27.50	31.25	41%	33.96	26.98	30.47	48%
5,600	35.94	28.23	32.08	41%	32.60	27.81	30.21	49%
5,800	35.31	28.65	31.98	41%	31.67	28.02	29.84	47%
5,900	**38.85**	28.54	33.70	39%	33.33	27.81	30.57	45%

A.1.3 BATCH-MEAN VERSUS GROUP-MEAN ADVANTAGES

We compare 128×1 sampling (128 prompts, one rollout each) with 32×4 sampling (32 prompts, four rollouts each) on DeepSeek-R1-Distill-Qwen-1.5B using the same 1,460-problem training set. Both use the FLASHREINFORCE loss, including token importance sampling, sequence trust, and sample-mean reduction, at policy lag 4 with $\delta = 5 \times 10^{-3}$. Training uses AdamW with peak learning rate 10^{-6} and weight decay 0.1. The 128×1 configuration centers rewards over the learner batch; 32×4 uses Dr. GRPO's within-prompt mean subtraction without standard-deviation normalization (Liu et al., 2025b). We compare approximately 4,000 optimizer updates.

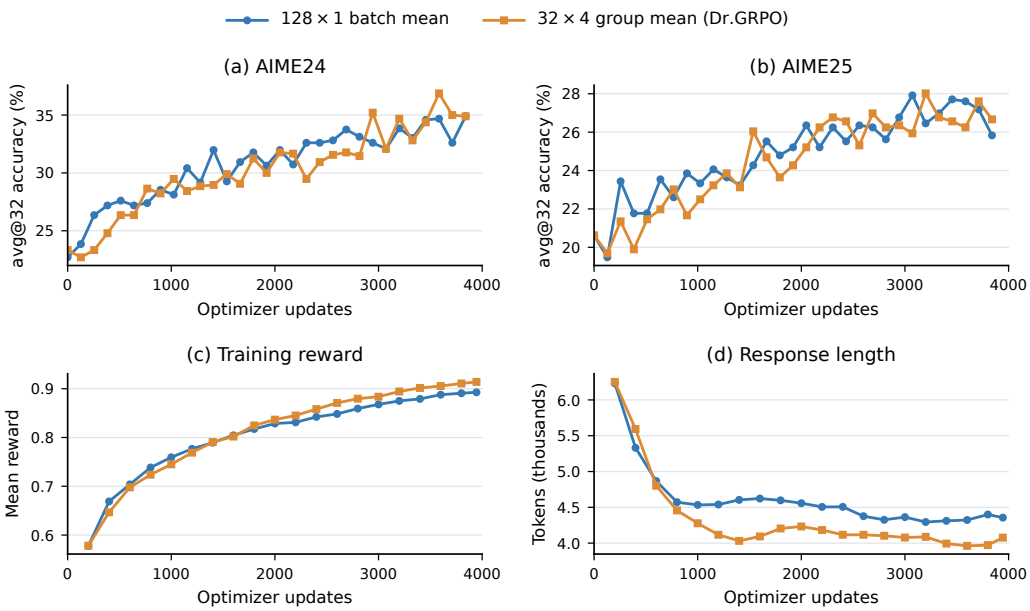


Figure 4: Batch-mean and group-mean training under the FLASHREINFORCE loss, with records through update 3,945 and AIME evaluations through update 3,840. AIME24 and AIME25 report avg@32 accuracy at measured checkpoints. Training reward and response length use consecutive 200-update means and the final 145-update window; lines join measured points.

The two configurations follow broadly similar learning trajectories through approximately 4,000 updates (Fig. 4). Across seven paired evaluations after update 3,000, group centering is ahead by 0.22 percentage points in mean AIME24/25 accuracy. Late in training, group centering has a modest training-reward advantage and shorter responses, while benchmark accuracy remains close and differences vary across tasks and checkpoints.

Connection to the group–batch tradeoff. Group centering adapts the baseline to each prompt, whereas batch centering estimates a shared baseline from more observations and covers more prompts per update. The modest performance differences are consistent with the competing effects of local-baseline quality, estimation noise, and prompt coverage analyzed for on-policy leave-one-out estimators by Hu (2026). This tradeoff provides a natural explanation for why batch centering remains competitive as training progresses.

A.2 PYTHON TOOL USE

A.2.1 QWEN2.5-7B-INSTRUCT: TRAINING AND EVALUATION SETTINGS

Table 11: Configuration of the Qwen2.5-7B-Instruct Python-tool experiment.

Setting	Value
Training data	DAPO-style math prompts
Online batch	128 prompts × one rollout
Interaction budget	10 tool turns; 8,192 tokens at comparison
Generation budget	6,144 tokens across turns
Reward	Exact committed answer
Optimization	AdamW; cosine lr 10^{-6}
Sequence threshold	$\delta = 1 \times 10^{-3}$
Evaluation	AMC23, Minerva, AIME25; temperature 0.7; top-$p = 0.7$

A.2.2 QWEN3-30B-A3B: TRAINING AND EVALUATION SETTINGS

Table 12: Configuration of the Qwen3-30B-A3B Python-tool experiments.

Setting	Value
Model	Qwen3-30B-A3B; thinking disabled
Training data	Deduplicated 7.5k DAPO-Math prompts, formatted for tool use
Rollouts per update	FLASHREINFORCE: 128 prompts $\times 1$; GRPO: 32 prompts $\times 4$
Policy lag	FLASHREINFORCE: ≈ 8; GRPO: ≈ 1
Interaction budget	Up to 10 turns; 8,192 generated tokens per turn; 18,432 total tokens
Rollout sampling	Temperature 1.0; top-$p = 1.0$
Optimization	AdamW; initial lr 10^{-6} with cosine decay; weight decay 0.1
Main comparison	$\delta = 10^{-2}$; routing replay off
Longer stability run	$\delta = 10^{-3}$; routing replay on (R3)
GRPO update	Group-normalized advantages; PPO clipping 0.2; routing replay off
Reference KL	Disabled in all configurations
Evaluation	AMC23, Minerva, AIME25; avg@4; temperature 0.7; top-$p = 0.7$; every 50 updates

A.3 MATHEMATICAL REASONING: QWEN2.5-MATH-1.5B

A.3.1 TRAINING AND EVALUATION SETTINGS

Table 13: Qwen2.5-Math baseline-comparison configuration. Boldface marks the defining one-pass design choices.

Model and Data			
Model	Qwen2.5-Math-1.5B	Prompts	Deduplicated 7.5k DAPO-Math
Token budget	2,048 prompt; 4,096 response	Online batch	128 prompts; **one rollout each**
Optimization			
Update	**One full-batch step; no mini-batches or reuse**	Optimizer	AdamW; lr 10^{-6}; wd 0.1
Schedule	Cosine; gradient clip 1.0	Actor lag	No explicit cap; induced by asynchronous completion
Sampling and Control			
Rollout	Temperature 1.0; top-$p = 1.0$	Evaluation	Temperature 0.7; top-$p = 0.7$
Token selection	None in the baseline comparison	Sequence threshold	$\delta = 3 \times 10^{-3}$

A.3.2 CHECKPOINT SCORES THROUGH 2,000 UPDATES

Table 14: Qwen2.5-Math checkpoints through step 2,000, using $\delta = 3 \times 10^{-3}$ without negative-token filtering. Panel (a) reports aggregate metrics; panel (b) gives per-benchmark avg@16 accuracy. Training reward is averaged over the preceding 200 updates. Evaluation uses temperature 0.7 and training rollouts use 1.0. Boldface highlights the Table 1 checkpoint in (a) and column bests in (b).

(a) Aggregate dynamics

Step	Rollouts	Five-task mean	Train reward	Eval length	Trunc.
1,000	128k	35.94	0.230	790	3.8%
1,500	192k	37.29	0.259	824	4.9%
2,000	**256k**	**37.97**	**0.270**	**867**	**6.0%**

(b) Per-benchmark accuracy

Step	MATH	AMC	Minerva	AIME	Olympiad
1,000	66.4	50.5	21.3	9.0	32.5
1,500	68.2	51.6	**22.8**	10.2	33.6
2,000	**69.2**	**52.5**	**22.8**	**10.8**	**34.5**

The five-benchmark mean rises from 35.94 at step 1,000 to 37.97 at step 2,000, the checkpoint reported in Tab. 1.

A.3.3 SEQUENCE-GATE THRESHOLD ABLATION

We examine sensitivity to the gate threshold by comparing the $\delta = 1 \times 10^{-3}$ and $\delta = 3 \times 10^{-3}$ configurations on Qwen2.5-Math-1.5B. Both runs use one rollout per prompt, 128 trajectories per update, sample-mean reduction, and no negative-token filtering. We evaluate the same four checkpoints on all five reasoning benchmarks with 16 samples per prompt, temperature 0.7, and top-$p = 0.7$.

Table 15: Gate-threshold configurations on Qwen2.5-Math-1.5B. Entries are five-benchmark mean avg@16 accuracy (%); the four checkpoints use 128k, 256k, 384k, and 512k training rollouts. Boldface marks the higher mean at each checkpoint.

δ	Step 1,000	Step 2,000	Step 3,000	Step 4,000
1×10^{-3}	**36.04**	36.89	36.93	37.09
3×10^{-3}	35.94	**37.97**	**37.75**	**37.58**

Both configurations remain stable through 4,000 updates. The looser threshold yields higher means at steps 2,000–4,000 in these runs, consistent with using the gate as a safeguard against large proxy-drift outliers.

A.3.4 SEQUENCE-GATE AGGREGATION ABLATION

We examine whether geometric aggregation improves on the arithmetic-mean gate in Eqs. (7) and (8). This aggregation ablation compares geometric-mean gating at two thresholds with the earlier $\delta = 1 \times 10^{-3}$ arithmetic-mean reference.

Let $d_{i,t}$ be the tokenwise Bernoulli KL in Eq. (6) and T_i the number of policy tokens in trajectory i. The geometric variant replaces the arithmetic mean $\bar{D}_i = T_i^{-1} \sum_{t=1}^{T_i} d_{i,t}$ with the geometric mean of the same KL values and applies the sequence gate

$$D_i^{\text{geo}} = \left(\prod_{t=1}^{T_i} d_{i,t} \right)^{1/T_i}, \qquad m_i^{\text{geo}} = \mathbf{1}[D_i^{\text{geo}} \le \delta]\,. \tag{12}$$

The mask m_i^{geo} replaces m_i in Eq. (11) and is held fixed during the update: an admitted trajectory contributes all its policy-token terms, while a rejected trajectory contributes none.

Table 16: Sequence-gate aggregation ablation. Runs are at 1,100–1,250 policy updates; AMC23 is evaluated at step 1,100.

Sequence gate	δ	**AMC23@1,100**	**Minerva**
Arithmetic-mean gating	1×10^{-3}	53.1	19.5
Geometric-mean gating	1×10^{-3}	50.6	20.1
Geometric-mean gating	3×10^{-3}	49.4	19.6

In Tab. 16, the arithmetic-mean reference achieves higher AMC23 accuracy, while geometric-mean gating gives small gains on Minerva. Geometric aggregation therefore offers no consistent improvement across the two reported benchmarks.

A.4 ALFWORLD: QWEN2.5-7B-INSTRUCT

A.4.1 TRAINING AND EVALUATION SETTINGS

Table 17: Configuration of the Qwen2.5-7B-Instruct ALFWorld experiment.

Setting	**Value**
Training data	1,024 training games
Online batch	64 games $\times$ one rollout
Interaction budget	50 environment actions; 16,384 total tokens
Generation budget	8,192 tokens across turns
Reward	Terminal environment success
Optimization	AdamW; constant lr 10^{-6}
Sequence threshold	$\delta = 1 \times 10^{-3}$
Evaluation	140 seen + 134 unseen games; temperature 1.0; top-$p = 0.7$

B TOKEN-LEVEL VERSUS SEQUENCE-LEVEL TRUST REGIONS

The Qwen2.5-Math comparison in Tab. 7 shows token-local collapse while sequence-level admission remains stable. This appendix gives two reasons for making one decision per trajectory: local-surrogate error accumulates policy movement across the sequence, and removing one token's score term leaves later stored histories unchanged. We use the fixed horizon T from Sec. 2.2, with a shared prompt distribution and environment for π and μ; T_i denotes an observed trajectory's length.

Local surrogate. Recall from Sec. 2.2 the surrogate $\mathcal{L}_\mu$ of Eq. (2), the behavior continuation value Q_t^μ, and the per-position movements ε_t, κ_t, and $\bar{\kappa}$. Let $A_t^\mu(h,a) = Q_t^\mu(h,a) - \mathbb{E}_\mu[R \mid h_t = h]$ be the behavior advantage, with $|A_t^\mu| \le A_{\max}$, and let $f_t(h) = \mathbb{E}_{a\sim\pi(\cdot|h)}[A_t^\mu(h,a)]$. Then Eq. (2) becomes

$$\mathcal{L}_\mu(\pi) = J(\mu) + \sum_{t=1}^{T} \mathbb{E}_{h\sim d_\mu^t}\big[f_t(h)\big], \tag{13}$$

Proposition B.1 (Exact local gradient)**.** *For a differentiable policy satisfying $\pi_\theta(\cdot \mid h) \ll \mu(\cdot \mid h)$ at every behavior-reachable history, Eq. (3) holds, and at $\pi_\theta = \mu$ the expression equals $\nabla_\theta J(\pi_\theta)$.*

Proof. The baseline term vanishes because $V_t^\mu(h)$ is action-independent and $\sum_a \nabla_\theta \pi_\theta(a \mid h) = 0$. Differentiate the remaining inner expectation of Eq. (2) with d_μ^t and Q_t^μ held fixed: $\nabla_\theta \sum_a \pi_\theta(a \mid h) Q_t^\mu(h,a) = \mathbb{E}_{a\sim\mu}[\rho_t Q_t^\mu(h,a) \nabla_\theta \log \pi_\theta(a \mid h)]$, using $\nabla \pi_\theta = \pi_\theta \nabla \log \pi_\theta$ and absolute continuity for the change of measure. Replacing $Q_t^\mu(h_t, a_t)$ by the sampled R, whose conditional expectation it is, gives Eq. (3). At $\pi_\theta = \mu$ every ratio is one and the expression is the REINFORCE gradient of J. □

The following bound uses the same performance-difference and coupling route as the classical bound in Li et al. (2026, Sec. 3.1), retaining per-position divergences.

Proposition B.2 (Surrogate error is governed by accumulated drift)**.**

$$|J(\pi) - \mathcal{L}_\mu(\pi)| \le 4A_{\max} \sum_{t=1}^{T} \varepsilon_t \sum_{u<t} \varepsilon_u \le 2A_{\max} \Big(\sum_{t=1}^{T} \varepsilon_t \Big)^2 \le A_{\max} T^2 \bar{\kappa}. \tag{14}$$

Proof. The finite-horizon performance-difference identity gives $J(\pi) - J(\mu) = \sum_t \mathbb{E}_{h\sim d_\pi^t}[f_t(h)]$, so

$$J(\pi) - \mathcal{L}_\mu(\pi) = \sum_{t=1}^{T} \Big(\mathbb{E}_{d_\pi^t}[f_t] - \mathbb{E}_{d_\mu^t}[f_t] \Big). \tag{15}$$

Since $\mathbb{E}_{a\sim\mu}[A_t^\mu(h,a)] = 0$, the function f_t satisfies $|f_t(h)| \le 2A_{\max} D_{\mathrm{TV}}(\pi(\cdot \mid h), \mu(\cdot \mid h)) \le 2A_{\max}\varepsilon_t$. Coupling the two processes with shared environment randomness and a maximal coupling of the two action distributions at every history they share, the histories first disagree at step u with probability at most ε_u, so $D_{\mathrm{TV}}(d_\pi^t, d_\mu^t) \le \sum_{u<t} \varepsilon_u$. Each term of Eq. (15) is bounded by $2\|f_t\|_\infty D_{\mathrm{TV}}(d_\pi^t, d_\mu^t)$, which gives the first inequality. The second follows from $\sum_t \varepsilon_t \sum_{u<t} \varepsilon_u \le \frac{1}{2}(\sum_t \varepsilon_t)^2$. For the third, Pinsker gives $\varepsilon_t \le \sqrt{\kappa_t/2}$ and Cauchy–Schwarz gives $\sum_t \sqrt{\kappa_t/2} \le \sqrt{T \sum_t \kappa_t/2} = T\sqrt{\bar{\kappa}/2}$. □

Under $\varepsilon_t \le \varepsilon$, the first bound gives $2A_{\max} T(T-1)\varepsilon^2$, recovering that classical coupling form. The per-position TV bound accumulates movement across all earlier positions, and for a fixed horizon the final KL bound depends on the sum $\sum_t \kappa_t = T\bar{\kappa}$. This motivates monitoring policy movement over complete trajectories. The computable statistic and the admission rule below are practical design choices guided by this accumulated-drift view.

Computable sequence trust approximation. The worst-case quantity $\bar{\kappa}$ is not directly available to the learner because it requires full behavior distributions and a supremum over behavior-reachable histories. For the observed length T_i of trajectory i, define the corresponding worst-case average $\bar{\kappa}_i = T_i^{-1} \sum_{t=1}^{T_i} \sup_h D_{\mathrm{KL}}(\mu_i(\cdot \mid h) \,\|\, \pi_\theta(\cdot \mid h))$ and the realized-history statistic

$$D_i^{\mathrm{full}} = \frac{1}{T_i} \sum_{t=1}^{T_i} D_{\mathrm{KL}}(\mu_i(\cdot \mid h_{i,t}) \,\|\, \pi_\theta(\cdot \mid h_{i,t})). \tag{16}$$

At every stored history, $d_{i,t}$ is the KL after projecting the vocabulary onto the sampled action versus all remaining actions. The data-processing inequality and the definition of the history-wise supremum therefore give

$$\bar{D}_i \le D_i^{\text{full}} \le \bar{\kappa}_i. \tag{17}$$

By Eq. (17), rejecting $\bar{D}_i > \delta$ identifies a trajectory whose mean full KL also exceeds δ. Passing the screen provides no corresponding upper bound: probability mass can shift among unobserved actions while leaving the sampled-action probability unchanged. The proxy supplies a low-cost drift screen using stored behavior probabilities.

Trajectory-normalized practical surrogate. Let $\ell_{i,t}$ denote the per-token loss in Eq. (11), including the fixed advantage and sequence mask, and let $L_i = T_i^{-1} \sum_{t=1}^{T_i} \ell_{i,t}$ be its within-trajectory mean. Using the same token losses, the two reductions are

$$\mathcal{L}_{\text{sample}} = \frac{1}{B} \sum_{i=1}^{B} L_i, \qquad \mathcal{L}_{\text{token}} = \frac{\sum_{i=1}^{B} \sum_{t=1}^{T_i} \ell_{i,t}}{\sum_{j=1}^{B} T_j} = \sum_{i=1}^{B} \frac{T_i}{\sum_{j=1}^{B} T_j} L_i. \tag{18}$$

Sample mean assigns each trajectory weight $1/B$; token mean assigns $T_i / \sum_j T_j$, giving longer failures greater weight even at the same mean token loss. The token-level signal is unchanged. Optional token-selection details appear in Sec. C.

The sample-mean update replaces the local surrogate's token sum with a within-trajectory average, changing its length weighting. Table 6 compares sample-mean and token-mean reductions from the same checkpoint.

With the advantage and sequence mask held fixed, the detached-ratio loss in Eq. (11) has negative gradient Eq. (10), since

$$\nabla_\theta \, \rho_{i,t}(\theta) = \rho_{i,t}(\theta) \, \nabla_\theta \log \pi_\theta(a_{i,t} \mid h_{i,t}) = \nabla_\theta \big[\operatorname{stopgrad}(\rho_{i,t}) \log \pi_\theta(a_{i,t} \mid h_{i,t}) \big]. \tag{19}$$

Token gating. A token-local rule uses $g_{i,t} = \mathbf{1}[d_{i,t} \le \delta]$ and, with the gate held fixed, replaces the complete trajectory contribution by

$$G_i^{\text{tok}} = \frac{A_i}{B T_i} \sum_{t=1}^{T_i} g_{i,t} \rho_{i,t} \nabla_\theta \log \pi_\theta(a_{i,t} \mid h_{i,t}). \tag{20}$$

This operation deletes selected score terms but does not alter the stored prefix $h_{i,t}$ of any later retained position. In particular, dropping position u leaves every later learner probability $\pi_\theta(a_{i,t} \mid h_{i,t})$ evaluated at the same behavior-generated history. The remainder of a trajectory can therefore keep contributing even when its sequence mean exceeds δ. Sequence admission instead multiplies the complete token-score sum by m_i: every admitted trajectory satisfies $\bar{D}_i \le \delta$, and every rejected trajectory contributes nothing.

B.1 Expected drift without a history-wise supremum

The local surrogate also admits bounds in terms of expected drift over behavior histories. The following result retains population expectations in the KL chain-rule argument reviewed by Li et al. (2026, Sec. 3.1 and Appendix B.1).

Proposition B.3 (Expected-history surrogate error)**.** *Fix policies μ, π, a common prompt distribution and environment, and a fixed horizon T with absorbing padding. Assume $|A_t^\mu(h,a)| \le A_{\max}$ on the relevant histories. Define*

$$e_u = \mathbb{E}_{h\sim d_\mu^u}[D_{\mathrm{TV}}(\mu(\cdot \mid h), \pi(\cdot \mid h))], \qquad k_u = \mathbb{E}_{h\sim d_\mu^u}[D_{\mathrm{KL}}(\mu(\cdot \mid h) \,\|\, \pi(\cdot \mid h))]. \tag{21}$$

For the unmasked, unnormalized surrogate in Eq. (13),

$$|J(\pi) - \mathcal{L}_\mu(\pi)| \le 2A_{\max} \sum_{u=1}^{T-1} (T-u) e_u, \tag{22}$$

$$|J(\pi) - \mathcal{L}_\mu(\pi)| \le 2A_{\max} \sum_{t=1}^{T} \sqrt{\frac{1}{2} \sum_{u<t} k_u}. \tag{23}$$

The KL bound is informative when the corresponding divergences are finite.

Proof. Let $f_t(h) = \mathbb{E}_{a\sim\pi(\cdot|h)}[A_t^\mu(h,a)]$, so $\|f_t\|_\infty \le A_{\max}$. The performance-difference decomposition in Eq. (15) gives

$$|J(\pi) - \mathcal{L}_\mu(\pi)| \le 2A_{\max} \sum_{t=1}^{T} D_{\mathrm{TV}}(d_\pi^t, d_\mu^t).$$

Write P_π^u and P_μ^u for the history-extension kernels, including the shared environment. Adding and subtracting $d_\mu^u P_\pi^u$, using contraction of TV under a common kernel, and averaging conditional TV under d_μ^u yields

$$D_{\mathrm{TV}}(d_\pi^{u+1}, d_\mu^{u+1}) \le D_{\mathrm{TV}}(d_\pi^u, d_\mu^u) + e_u.$$

The initial history distributions agree, so induction gives $D_{\mathrm{TV}}(d_\pi^t, d_\mu^t) \le \sum_{u<t} e_u$. Summing proves Eq. (22). For complete histories, the KL chain rule gives $D_{\mathrm{KL}}(d_\mu^t \| d_\pi^t) = \sum_{u<t} k_u$; common environment transitions contribute zero conditional KL. Applying Pinsker at each t proves Eq. (23). □

The expectations e_u and k_u range over the original behavior distribution, including trajectories rejected by a training gate. The fixed horizon T includes absorbing padding.

C Optional Negative-Token Filtering

C.1 Selector and motivation

Trajectory-level negative advantages also penalize useful intermediate steps in failed rollouts. Sample-mean reduction addresses length weighting (Sec. 4.3.2) but does not identify which steps caused failure. Here we examine how retaining different fractions of tokens within failed trajectories affects reasoning accuracy and tool use.

Wu et al. (2026) describe how group-relative advantages allow positive and negative rollouts of the same prompt to partially cancel gradients on shared supporting tokens. Single-rollout training lacks this within-prompt cancellation. Sample-mean reduction addresses the separate effect of trajectory length on update weighting.

Without token-level supervision, we rank tokens by policy entropy, following evidence that high-entropy positions carry disproportionate learning signal (Wang et al., 2025). Let $F_i = \mathbf{1}[R_i \leq 0]$ denote the failure label in the binary-reward runs, and let $r_{i,t} \in \{1, \ldots, T_i\}$ be the rank of token t by decreasing entropy. For retention fraction $q \in [0, 1]$, define

$$m_{i,t}^{\text{tok}} = \mathbf{1}[F_i = 0 \text{ or } r_{i,t} \leq \lceil qT_i \rceil]\,. \tag{24}$$

Successful trajectories retain all policy tokens; failed trajectories retain their $\lceil qT_i \rceil$ highest-entropy positions. The selector is held fixed during the update and multiplies the corresponding token term in Eq. (10); it applies to binary-reward settings with an explicit failure label.

C.2 Entropy diagnostic

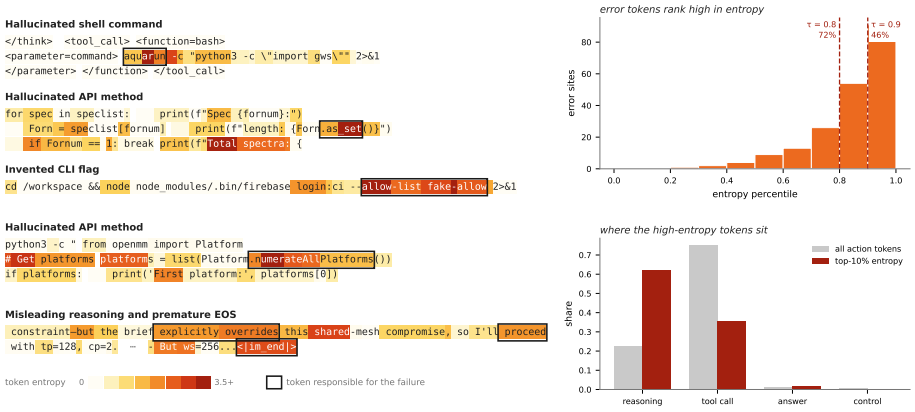


Figure 5: **Entropy diagnostic for the optional selector. Left:** reward-zero Qwen3.5-4B terminal-task fragments (Qwen Team, 2026); boxes mark failure-causing spans. **Right:** 72% of annotated sites exceed the within-trajectory 80th entropy percentile (versus 20% for length-matched random spans), and 46% exceed the 90th. Top-decile entropy tokens concentrate in reasoning text rather than format-control tokens.

Figure 5 shows that annotated failure sites are concentrated at high-entropy positions, supporting entropy as a ranking heuristic for this diagnostic.

C.3 Mathematical-reasoning ablation

Table 18: Negative-token filtering on Qwen2.5-Math-1.5B at $\delta = 1 \times 10^{-3}$. Entries are five-benchmark mean avg@16 accuracy (%); $q = 1$ applies no filtering. Boldface marks column bests.

Setting	q	Step 1,000	Step 2,000	Step 3,000	Step 4,000
No filtering	1.0	36.04	36.89	36.93	**37.09**
Light entropy filtering	0.9	36.03	**37.91**	**37.40**	37.01
Aggressive entropy filtering	0.2	**37.04**	36.80	–	–

Light filtering reaches a peak mean of 37.91 versus 37.09 for the unfiltered reference. The baseline comparison in Tab. 1 instead uses $\delta = 3 \times 10^{-3}$ without token filtering.

C.4 PYTHON TOOL-USE ABLATION: QWEN2.5-7B-INSTRUCT

Table 19: Negative-token filtering on Qwen2.5-7B-Instruct Python tool use at $\delta = 1 \times 10^{-3}$. Best mean is the peak three-benchmark accuracy (%); $q = 1$ applies no filtering. Tool calls are averaged over steps 601–700. Boldface marks column bests.

Setting	q	**Best mean**	**Best step**	**Calls 601–700**
No filtering	1.0	37.0	600	3.30
Light entropy filtering	0.9	**39.4**	700	**3.72**
Aggressive entropy filtering	0.2	32.2	200	0.03
Full failure-token filtering	0.0	13.1	100	0.00

Light negative-token masking offers a modest advantage.