TOKEN-LEVEL TRUST REGIONS UNDER REUSE: COMPONENT EFFECTS, STABILITY, AND INTERVENTION

Anonymous authors
Paper under double-blind review

ABSTRACT

Token-level trust regions constrain policy updates in reinforcement learning of large language models, but local constraints do not by themselves explain accuracy gains or stability under repeated use of generated answers. We study these questions in TROLL through component comparisons, update-budget controls and same-checkpoint interventions. On Qwen3-1.7B/GSM8K, removing gradients through an active projection causes no detectable accuracy loss, and at low reuse unclipped policy gradient recovers most of the gain over default-range clipping. At the published radius, two passes over each batch collapse in all three tested seeds, reaching 55.47% mean final accuracy, whereas allocating the same number of optimizer updates to twice as many fresh answers keeps all three runs healthy and reaches 86.83%. Smaller radii reduce this instability. From three shared alarm checkpoints, removing the second pass improves final GSM8K accuracy by 28.20–47.08 points over continuing two passes and by 3.11–3.64 points over the starting checkpoints. These interventions establish the value of changing reuse at those states, separately from choosing when to act. To explore when to intervene, we formulate Co-Drift Guard (CDG), a candidate label-free rule based on joint growth of entropy and within-step pre-projection KL, and separately evaluate model retention, reuse switching and detection across configurations. The results show that the benefit of removing the second pass does not automatically confer an advantage on alarm-based timing, and that detection performance depends on the training configuration. Our study distinguishes component-level gains, stability under reuse and intervention effects, providing empirical foundations for further study of state-dependent reuse decisions.

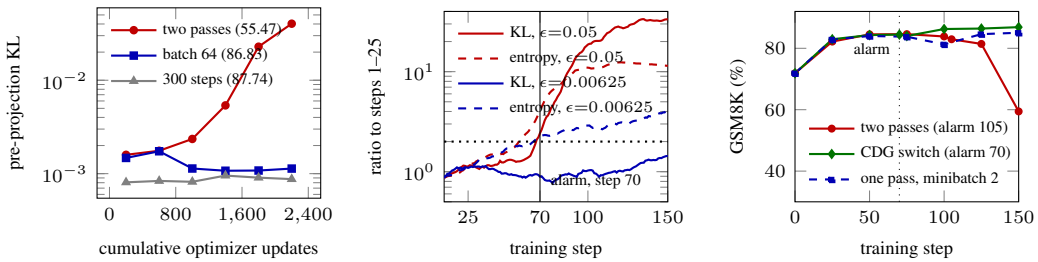
1 INTRODUCTION

Reinforcement learning with verifiable rewards improves a language model by generating answers, checking their correctness and updating the model to favor higher-reward responses (Shao et al., 2024). Generating these answers is a substantial part of training: each prompt can require several long responses before an update begins. Reusing a collected batch for additional passes offers a way to perform more updates without generating another batch. The choice between learning again from existing answers and collecting new ones therefore affects both training cost and optimization.

Policy-update constraints are intended to keep these updates controlled. PPO clips the importance ratio between the current policy and the policy that generated the answers (Schulman et al., 2017). TROLL instead projects each token distribution into a KL trust region around that generating policy (Becker et al., 2026). It backpropagates through this projection. Its radius limits how far the projected training target can move. This constraint applies to the target used within each update, whereas training continues across many batches and repeated passes. Whether such local constraints also prevent performance degradation over the full training trajectory is the question we study.

Our study separates three questions: which components contribute to accuracy, which training conditions preserve stability, and which interventions change a deteriorating trajectory. The component comparisons find no detectable accuracy loss from removing the projection gradient, including at an active-constraint operating point. At low reuse, unclipped policy gradient recovers most of the

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.



(a) (b) (c)

Figure 1: **(a)** Within-step pre-projection KL (3-seed mean per 400 updates) for three ways of spending 2,400 updates at ϵ=0.05 (legend: final GSM8K). The two-pass and batch-64 runs (16 updates per rollout batch each) differ by $1.08\times$ at first and $35.55\times$ at the end (Section 6.2). **(b)** CDG on prospective seed 3: 10-step means relative to steps 1–25. At ϵ=0.05 (red) both cross $2\times$ for three steps at step 70 (GSM8K 83.02), 30 steps (at least 5 on the evaluation grid) before the first 5-point decline. At 0.00625 (blue) entropy reaches $4.05\times$ but KL only $1.43\times$, and no alarm fires. **(c)** Seed 1, ϵ=0.05, 38,400 generated answers per run. Two passes throughout collapse after CDG alarms at step 105. With the switch action, CDG alarms at step 70 and training continues with one pass per rollout batch; it ends 2.43 points above the alarm checkpoint. A single pass over minibatches of 2, with the same updates per answer as two passes, does not collapse (Sections 6.2 and 7.2).

gain over default-range clipping. This ordering changes under repeated passes: published-radius projection runs deteriorate while none of the three PPO-clip runs collapses.

At TROLL's published radius, two passes collapse in all three tested seeds, reaching 55.47% mean final GSM8K accuracy. Allocating the same optimizer-update budget to twice as many fresh answers reaches 86.83% with all three runs healthy. Radius reduction provides another route to stability. To test whether changing reuse can also help after training is underway, we branch three alarm checkpoints into symmetric continuations with one or two passes. Removing the second pass improves final accuracy by 28.20–47.08 points over continuing it and by 3.11–3.64 points over the parent models.

The remaining decision is when to intervene. Some deteriorating trajectories show joint growth in entropy and within-step pre-projection KL, motivating Co-Drift Guard (CDG) as a candidate rule. It compares both signals with their early-training levels and, at sustained co-drift, either retains the alarm model and stops or removes the second pass for subsequent batches. The latter action follows ReMix (Liang et al., 2025), replacing a preset switch time with a training-log trigger. We test the trigger separately from the actions. The initial six-run prospective cohort supports timely model retention, but its two radii already separate outcomes by configuration. Across three new strategy-comparison seeds, alarm switching does not establish an advantage over one pass throughout or a fixed step-50 switch. A separate same-configuration stress test with a detached projection and no regression term yields one early detection, one miss and three no-event alarms in six runs. Together, these results support changing reuse at the tested states while leaving a transferable rule for when to do so unresolved.

Our contributions are:

- **Component attribution (Sections 4 and 5).** Removing projection gradients causes no detectable accuracy loss at an active constraint; at low reuse, unclipped policy gradient recovers most of TROLL's gain over default-range clipping.
- **Stability and controlled intervention under reuse (Section 6).** Equal-update comparisons separate optimizer budget from answer collection, and same-checkpoint branches show that removing the second pass can prevent large subsequent losses while allowing further learning.
- **An evaluation of intervention timing (Section 7).** We test CDG as a candidate rule, distinguish model retention from switching, and report both its initial prospective results and the limitations exposed by three-seed strategy comparisons and a six-seed same-configuration stress test.

Scope. One model (Qwen3-1.7B), task (GSM8K) and trainer (TROLL in verl), 150-step runs and data-order seeds; App. K gives a descriptive Qwen3-8B LoRA replication of the audit.

2 Related work

Trust regions, clipping and projection. Trust-region methods bound each policy update, from TRPO's constrained surrogate (Schulman et al., 2015) to PPO's clipped ratio (Schulman et al., 2017) and its LLM variants GRPO and DAPO (Shao et al., 2024; Yu et al., 2025). Differentiable trust-region projection layers (Otto et al., 2021) replace the clip by an exact projection, and TROLL scales them to token-level KL balls over LLM vocabularies (Becker et al., 2026). Clipping's distortions in LLM RL are documented from several sides: divergence masks replace it (Qi et al., 2026), other work asks whether out-of-region gradients should be dropped or corrected (Yao et al., 2026; Zhou et al., 2026), and entropy collapse under RLVR has its own analysis (Cui et al., 2025). Regressing onto a detached target, as TROLL's auxiliary term does, is classical (Abdolmaleki et al., 2018; Song et al., 2020; Peng et al., 2019). Our audit asks which of these ingredients carries TROLL's value in a controlled setting; our intervention study evaluates both preset and signal-based ways to change reuse.

Minibatch reuse. Common LLM RL recipes take one pass over each rollout batch, and TROLL's own reuse sweep adds fresh minibatches per collection rather than passes (Becker et al., 2026). Skywork-OR1 found reuse harmful under clipping (He et al., 2025), and other work studies how gradient steps per rollout change ratios and clipping (Wang et al., 2026) and how off-policyness accumulates across reuse (Zhang et al., 2026b). We add a controlled contrast under a KL-projection trust region: at equal updates, minibatch size and learning rate, a second pass collapses where fresh updates do not, while the per-step statistics of the two agree early on. We then separate the effect of removing reuse from the problem of detecting when intervention is useful.

Training monitors, early stopping and rollback. PPO implementations stop a batch's updates or roll back the latest update when a KL estimate exceeds a target (Dossa et al., 2021). In RLVR, gradient-increment gating ends a reuse loop when the `lm_head` gradient norm jumps (Miao et al., 2026), while ReMix switches from reuse to on-policy training at a fixed step (Liang et al., 2025). Other label-free monitors use response length to trigger learning-rate decay (Zhang et al., 2026c), combine entropy with other signals under frozen evaluation rules (Wang, 2026), or use relative KL drift to prune hyperparameter trials (Chen et al., 2026). Related approaches include failure management for RL fine-tuning (Zhang et al., 2026a), pretraining rollback after loss spikes (K2 Team et al., 2025), and validation-free checkpoint selection (Nguyen et al., 2025). CDG combines a run-relative baseline with the joint growth of entropy and TROLL's pre-projection KL. Its alarm acts on the remainder of training: it either returns the alarm checkpoint or switches all subsequent batches to one pass. The switch action follows ReMix, with its timing determined by logged signals rather than a fixed step.

3 Background and experimental protocol

TROLL. Given rollout policy π_{old} and current policy π_θ, TROLL projects each token distribution $\pi_\theta(\cdot \mid s_t)$ into the KL ball $\{\pi : \mathrm{KL}(\pi \,\|\, \pi_{\text{old}}(\cdot \mid s_t)) \le \epsilon\}$, yielding $\tilde{\pi}$ via a closed-form REPS dual (Peters et al., 2010) over a top-k sparse support; the policy-gradient loss is evaluated under $\tilde{\pi}$, and gradients flow through the projection by implicit differentiation (Amos & Kolter, 2017) (Eq. (1)). Two hyperparameters matter here: the radius ϵ (published value 0.05) and $K = \text{ppo_epochs} \times \text{train_batch}/\text{mini_batch}$, the number of optimizer updates per rollout batch. The original waves study a *low-reuse* regime (K=4: one pass, minibatch 8; "weak" in the appendix) and a *high-reuse* regime (K=16: two passes over minibatches of 4 prompts; "strong"). Section 6 separates passes from updates.

Models, tasks, arms. We train Qwen3-1.7B (Yang et al., 2025) with full fine-tuning on GSM8K (Cobbe et al., 2021) under verl (Sheng et al., 2025) with GRPO advantages, learning rate 10^{-6}, 32 prompts $\times$ 8 rollouts per batch and 150 global steps, and evaluate every 25 steps on all 1,319 GSM8K test questions, with MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024) as a secondary,

Table 1: Qwen3-1.7B GSM8K accuracy (%), 3 seeds. Low reuse (weak): $K{=}4$, one pass, 150 steps; high reuse (strong): $K{=}16$, two passes, step-100 endpoint (an exploratory horizon chosen after observing the step-150 collapse; disclosed as post-hoc). Δ and 90% CIs are seed-paired against full TROLL ($t_{0.95,2} = 2.920$). "Proj. act." is the projected-token fraction, averaged over steps and the 3 seeds unless marked †; †the $\alpha{=}0$ value is descriptive from the sole locally archived raw M2 log (seed 0), so no 3-seed activity mean is claimed. "n/a" means the arm has no projection. Clipping touched only $\sim$0.5% of $|A|$-mass (proxy) across the weak clip arms. Point estimates with CIs, not equivalence certificates. $^{\diamond}$Final sourced from the runner's completion ledger (raw log not retained; the ledger matches all four retained attribution logs bit-for-bit). The $\alpha{=}0$ mean's $+0.05$ edge over full is driven entirely by seed 2 (86.81, the largest value in the weak block).

Regime	Arm	s0	s1	s2	Mean	Δ vs full [90% CI]	Proj. (%)
weak ($K{=}4$)	full TROLL	86.43	85.75	85.06	**85.75**	—	0.057
weak ($K{=}4$)	detach-only	85.22	85.29	85.75	**85.42**	-0.33 [-1.94, +1.28]	0.069
weak ($K{=}4$)	$\alpha = 0$ (no regression)	85.14	85.44$^{\diamond}$	86.81$^{\diamond}$	**85.80**	+0.05 [-2.56, +2.66]	0.062†
weak ($K{=}4$)	bare no-clip GRPO	85.75	85.22	86.05	**85.67**	-0.08 [-1.63, +1.48]	n/a
weak ($K{=}4$)	PPO-clip	84.00	84.46	83.17	**83.88**	-1.87 [-2.83, -0.91]	n/a
weak ($K{=}4$)	clip $\lvert A\rvert$-mass proxy (%)	0.51	0.55	0.44	**0.50**	n/a	n/a
strong ($K{=}16$)	full TROLL	83.09	81.73	83.70	**82.84**	—	0.836
strong ($K{=}16$)	detach-only	80.29	85.06	84.69	**83.35**	+0.51 [-4.72, +5.73]	0.657
strong ($K{=}16$)	PPO-clip	84.61	87.11	85.29	**85.67**	+2.83 [-0.90, +6.56]	n/a

noisier readout. Seeds vary data order; the rollout engine seed is fixed. Arms are full TROLL, detach-only, $\alpha{=}0$ (regression term removed), bare no-clip GRPO (verl's `vanilla` loss with every clip range set to 10, effectively unclipped) and PPO-clip at verl's default range 0.2. No KL-to-reference penalty or entropy bonus is present in any audit arm (one Section 6 run adds a reference-KL term, marked there). App. J lists every hyperparameter.

Relation to the published TROLL results. TROLL reports 3–10 point gains over clipping on math reasoning and 7–18 on code (Becker et al., 2026). We inherit its radius, solver and sparsification defaults in a smaller recipe (GSM8K, 150 steps), where full TROLL beats clipping by 1.87 points at $K{=}4$.

We report accuracy for each seed and paired differences with 90% confidence intervals for the three-seed comparisons. Training outcomes are classified as HEALTHY, DEGRADED or COLLAPSE using accuracy stability, entropy and response length; thresholds and statistical procedures are given in App. A.

4 Does the gradient need to flow through the projection?

We first examine the gradient path through TROLL's projection. Its per-token loss combines an unclipped policy-gradient term evaluated through the projected distribution with a regression term onto a detached target:

$$\mathcal{L}(\theta) = \mathrm{agg}_t\Big[- \hat{A}_t \, \frac{\tilde{\pi}_t(a_t)}{\pi_{\mathrm{old}}(a_t \mid s_t)}\Big] + \alpha \, \mathrm{agg}_t\Big[\mathrm{KL}\big(\pi_\theta(\cdot \mid s_t) \,\big\|\, \mathrm{sg}[\tilde{\pi}_t]\big)\Big], \qquad \alpha{=}1, \tag{1}$$

where $\tilde{\pi}_t$ is the projection onto the ϵ-ball around π_{old}. The computation uses a renormalized sparse support with target omitted mass 10^{-5}, a 64-logit cap and forced inclusion of the sampled token. Positions already inside the ball retain $\tilde{\pi}_t = \pi_\theta$. The projected-token fraction is the share of response tokens whose initial sparse $\mathrm{KL}(\pi_\theta \| \pi_{\mathrm{old}})$ is at least ϵ. We isolate the projection's gradient path by stopping gradients at its output: full and detached TROLL differ only in whether gradients pass through $\tilde{\pi}_t$ at out-of-ball positions.

Results. At $K{=}4$ detach-only reaches 85.42 against full TROLL's 85.75 (Table 1; -0.33 points, 90% CI $[-1.94, +1.28]$), and at $K{=}16$ (step-100 endpoint) $+0.51$ points ($[-4.72, +5.73]$), with per-seed signs flipping in both regimes. At the published radius, however, the projection binds on only 0.03–0.10% of tokens at $K{=}4$ and 0.46–1.24% at $K{=}16$, so these nulls cannot separate "the gradient is unnecessary" from "the constraint rarely binds". We therefore reran the paired

contrast where the ball is active: $K{=}16$ at $\epsilon{=}0.003125$, 3 seeds, where the full arm projects 7.2–7.8% of tokens (pre-registered activity gate $\geq 3\%$ passed 3/3) and both arms are HEALTHY in 3/3. Detaching changes step-150 accuracy by $+0.83$ points (90% CI $[-0.34, +2.01]$). We detect no benefit from keeping the differentiable path at an active constraint: its 90% interval is $[-2.01, +0.34]$ points for keeping rather than removing the path. A pre-registered factorial crossing detach with $\alpha \in \{0, 1\}$ leaves the interaction unidentified (trajectory-AUC interaction -0.23 points, 90% CI $[-4.76, +4.30]$, with the two α levels run in different cohorts; App. A). Neither contrast establishes equivalence.

5 Where does the gain over clipping come from?

The projection-gradient comparisons do not detect an accuracy benefit from retaining the projection's gradient path. We next separate the other possible sources of TROLL's gain: the projection itself, the regression term and the removal of clipping. The low-reuse block of Table 1 separates these with five arms sharing data, seeds and recipe. Full TROLL reaches 85.75; removing the projection gradient (85.42), the regression term ($\alpha{=}0$, 85.80) or the projection altogether (bare no-clip GRPO, 85.67) leaves accuracy within 0.4 points, with paired CIs containing zero (no-clip$-$full: -0.08, 90% CI $[-1.63, +1.48]$). PPO-clip drops to 83.88, -1.87 points against full ($[-2.83, -0.91]$), at verl's untuned default range 0.2. In this cell the measured advantage of the trust-region package over clipping is recovered by not clipping; the clip gate touches only about 0.5% of the advantage mass per update. MATH-500 transfer does not replicate the ordering (clip is best there, $+2.60$ points against full, 90% CI $[-0.97, +6.17]$), so the attribution is a statement about the trained task.

High reuse reverses the picture. At $K{=}16$ with two passes, the range-10 control collapses in 2/3 seeds by step 150 ($5.91/77.10/69.22$), and a DPPO-style binary divergence mask collapses to 4.7 in a single-seed control. In the attribution cohort both projection arms end at $64.06/62.32/30.02$ (full) and $54.51/46.40/33.74$ (detached) at step 150 (full: $56.18/49.20/61.03$ in the cohort of Table 2), while none of the three PPO-clip seeds collapses ($83.70/86.05/86.43$; App. A). These comparisons show that accuracy gains at low reuse do not directly carry over to repeated passes; training stability also depends on the constraint radius and update organization (Section 6).

6 When do token-level constraints fail under reuse?

The component comparisons show that low-reuse accuracy does not determine stability under repeated passes. We examine how the viable radius changes with reuse, compare organizations of the optimizer budget, and use same-checkpoint branches to test radius reduction and removal of the second pass.

6.1 The viable radius moves with reuse

With two passes at $K{=}16$ (detached path, earlier cohort), the published radius collapses in 3/3 seeds (finals $66.6/57.7/45.4$). A reuse-normalized radius $\epsilon_k = \epsilon_{\text{total}}/K$ with $\epsilon_{\text{total}}{=}0.05$ trains stably in 3/3 ($81.6/83.6/84.2$), and so does a fixed radius matched to the same value (fixed$-$normalized $+0.33$, 90% CI $[-1.49, +2.14]$): the gain lies in radius size, not in the schedule. The viable size moves with reuse. At $K{=}4$, radii up to 0.4 are HEALTHY in 3/3 seeds, while at $K{=}16$ no seed is HEALTHY at 0.0125 and the majority non-COLLAPSE boundary lies between 0.0125 and 0.025 (App. E); between one pass at $K{=}4$ and two passes at $K{=}16$ the largest viable radius shrinks by at least $16\times$ (more than $32\times$ if viable means HEALTHY in 3/3). Locating it took sweeps over two pipelines that disagree on a nominally matched cell (82.18 vs 64.97). The published-radius arms peak at 83.90 on average, above the tuned arms' finals (83.14), but 50–75 steps before a 17–40 point crash: this gap motivates testing whether a label-free stopping rule can preserve early accuracy. The peak is determined from labelled evaluations.

Table 2: Five organizations of optimizer updates at ϵ=0.05 (GSM8K % at the last step, seeds 0/1/2; frozen three-state rule; means from unrounded per-seed values). K: updates per rollout batch (minibatch 4 prompts unless noted). The two-pass and minibatch-2 runs apply 1/16 update per generated answer, the others 1/32. [a]Seed 1 only, later session.

organization	K	passes	updates	answers	per seed	mean	state
one pass, 150 steps	8	1	1,200	38,400	85.60/84.61/85.75	85.32	H 3/3
two passes, 150 steps	16	2	2,400	38,400	56.18/49.20/61.03	**55.47**	C 3/3
one pass, batch 64	16	1	2,400	76,800	86.28/87.04/87.19	86.83	H 3/3
one pass, 300 steps	8	1	2,400	76,800	86.96/87.79/88.48	87.74	H 3/3
one pass, minibatch 2[a]	16	1	2,400	38,400	–/85.06/–	–	H 1/1

6.2 Update organization and the evolution of training statistics

Unless marked, runs in this subsection train from the pretrained model at ϵ=0.05 with the full projection path and are scored on all 1,319 GSM8K test questions. Table 2 arranges five ways of spending updates.

Equal updates, opposite outcomes. The two-pass run and the batch-64 run use the same minibatch, learning rate and 2,400 updates, and their `update_actor` time per step is within 1%. The two-pass run collapses in 3/3 seeds, the batch-64 run is HEALTHY in 3/3. The batch-64 run generates twice as many answers, and before step 75 each step costs $1.243\times$ a two-pass step. The second pass buys early speed only: averaged over seeds it leads the K=8 single-pass run over the same number of generated answers by $+1.26$ and $+1.52$ points at steps 25 and 50, is level at step 75 (-0.08) and trails by 29.85 at step 150.

A matched-update single-pass control. The first four rows change both the number of passes and the ratio of total optimizer updates to generated answers. Using minibatches of 2 prompts with one pass gives 16 updates per batch and the same ratio, 1/16, as the two-pass run. Each answer is still used only once; the ratio is a budget measure, not the number of times an answer participates in training. This seed-1 control ends HEALTHY at 85.06, compared with 59.44 for two passes in the later cohort and 49.20 in the earlier cohort. A one-pass control at twice the learning rate reaches 80.89 (Table 20). The smaller-minibatch result shows that the update-to-answer ratio alone does not determine collapse in this seed; minibatch size and gradient noise also change.

What the per-step statistics show. Compared at equal cumulative updates (Fig. 1a), the two-pass and batch-64 runs agree to within 8% over the first 400 updates in within-step pre-projection KL ($1.08\times$), projected-token fraction ($1.07\times$) and entropy ($1.05\times$); at 2,000–2,400 updates the ratios are $35.55\times$, $146.29\times$ and $6.29\times$. Through step 75 the ball binds on 0.5–0.6% of the two-pass run's tokens while its entropy has already risen from 0.11 to 0.33–0.44. When the guard of Section 7 fires on six local published-radius runs that later decline (one with a reference-KL term of coefficient 0.001), the 10-step mean pre-projection KL is 0.066–0.073 of the radius and the projected fraction 0.70–0.92%: only a small fraction of tokens reaches the projection threshold while the logged statistics rise. This metric measures mean within-step KL relative to the policy that generated the current batch.

A brake, not a repair. We forked two two-pass runs at their step-75 checkpoints and continued each from identical inputs at ϵ=0.05 or at 0.00625 (App. L, Table 15). Tightening ends $+18.35$ and $+35.41$ points above continuing loose, yet 1.37 and 2.12 points below the checkpoint it started from. In these step-75 forks, tightening limits further loss without restoring checkpoint accuracy. Changing the number of passes provides a different intervention from tightening the radius.

Removing the second pass at a shared state. For seeds 601–603, we restore each saved alarm checkpoint into two fresh generation engines, with identical parent model, optimizer and data-cursor state. One continuation retains two passes; the other uses one pass, with radius 0.05 and learning rate unchanged. Final GSM8K accuracy improves by $43.21/28.20/47.08$ points when the second pass is

Algorithm 1 Co-Drift Guard with its two actions (hyperparameters frozen before evaluation).

Require: budget T; passes $E{=}2$; action $\in \{\text{STOP}, \text{SWITCH}\}$; $B{=}25$, $W{=}10$, $M{=}2$, $P{=}3$; $n \leftarrow 0$

for $t = 1, \ldots, T$ **do**
  run one TROLL step with E passes; read h_t and k_t
  if $t = B$ **then** $h_0 \leftarrow \text{mean}(h_{1:B})$, $k_0 \leftarrow \text{mean}(k_{1:B})$; **abort** if either is ≤ 0
  else if $t > B$ **and** $E = 2$ **then**
    $c_t \leftarrow [\text{mean}(h_{t-W+1:t}) \geq Mh_0] \wedge [\text{mean}(k_{t-W+1:t}) \geq Mk_0]$
    $n \leftarrow n + 1$ **if** c_t **else** 0
    if $n = P$ **then** save checkpoint $C^\star$
      if action $=$ STOP **return** $C^\star$ **else** $E \leftarrow 1$ ▷ one pass from $t{+}1$
return final model

removed (Table 19); MATH-500 improves by $9.2/14.4/19.8$ points. The single-pass continuations also exceed their parent models by $3.11/3.64/3.26$ GSM8K points.

7 Can training signals guide intervention?

Co-Drift Guard (CDG) is a candidate rule for choosing intervention times from training logs. It monitors joint growth in entropy and within-step pre-projection KL and triggers either model retention or a switch to one pass.

7.1 Detector and actions

Signals and rule. At global step t, let h_t be the mean token entropy on the training batch (`actor/entropy`) and k_t the mean pre-projection sparse $\text{KL}(\pi_\theta \,\|\, \pi_{\text{old}})$ over response tokens and the K updates of the step (`trpl/initial_kl_mean`). Let h_0, k_0 be their means over steps 1–25 and $\bar{h}_t, \bar{k}_t$ their means over steps $t{-}9, \ldots, t$. Step $t \geq 26$ is *drifting* if $\bar{h}_t \geq 2h_0$ and $\bar{k}_t \geq 2k_0$, and CDG alarms at the first step that closes three consecutive drifting steps and writes a checkpoint there (Algorithm 1); the earliest possible alarm is step 28. A non-positive baseline or a missing signal makes the detection invalid. CDG keeps two rolling means and writes one checkpoint.

Actions. *Stop* ends training and returns the alarm checkpoint. *Switch* retains that checkpoint and uses one pass per rollout batch from the next step, leaving radius, learning rate, optimizer state and the signal baseline unchanged. It returns the final model. The removal of reuse follows ReMix (Liang et al., 2025); the difference tested here is a training-log trigger rather than a preset switch step.

Design choices. Early KL levels depend on update organization: the steps 1–25 means are 1.80–2.97 times those of a $K{=}8$ single pass. CDG therefore normalizes each signal by its own run's baseline. In the archived replay, the entropy condition removes three of KL-only's 13 no-event alarms, all on single-pass runs.

7.2 Evaluation

An *event* is the first periodic evaluation at step ≥ 50 that lies at least 5 GSM8K points below the previous periodic peak. An alarm is *early* if it precedes the event, *late* otherwise, and a *no-event alarm* if the run has no event.

Prospective test with the frozen rule. After the detector parameters and event definition were fixed, six new two-pass $K{=}16$ trajectories were trained for 150 steps: seeds 3, 4 and 5 at $\epsilon{=}0.05$ and 0.00625. Every saved answer was rescored with the frozen scorer, and alarms were reconstructed from the logged metrics (Table 3). All three published-radius runs alarm before the first recorded decline; none of the tight-radius runs alarms. Each alarm checkpoint is within one point of the preceding periodic peak and 17.5–37.5 points above final accuracy. These two radii also separate the observed outcomes by configuration.

Table 3: Prospective test of the frozen rule (GSM8K %). Lead: steps from the alarm to the event (upper bound) and to the last evaluation before it (lower bound). Ratios: 10-step means over the steps 1–25 mean, entropy/KL.

seed	ϵ	alarm	GSM at alarm	event	lead	GSM150	ratios at alarm	max ratios
3	0.05	70	83.02	100	30 / 5	45.49	4.06 / 2.43	12.40 / 33.50
4	0.05	73	85.37	125	52 / 27	67.85	3.57 / 2.03	11.92 / 25.88
5	0.05	97	84.38	125	28 / 3	54.66	5.00 / 2.16	14.71 / 30.06
3/4/5	0.00625	–	–	–	–	82.71/84.91/83.32	–	$\leq$5.33 / $\leq$1.60

Same-configuration stress test. We apply the unchanged rule to six new data-order seeds (701–706) at K=16 and ϵ=0.003125, with the projection gradient detached and the regression term removed (α=0). Archived seeds of this cell had diverged, allowing outcomes to vary across seeds while the configuration remains fixed. All runs continue to step 150 after alarm evaluation; no stop or switch is executed. Of two GSM8K decline events, CDG detects one early and misses one. It alarms on three of the four runs without an event; two end within 2 points of their periodic peak, exceeding the limit of one in the follow-up no-event-alarm criterion (App. M). Returning the alarm checkpoint, or the final model when no alarm occurs, gives 82.93% mean GSM8K accuracy, compared with 81.69% for training to the end and 83.38% for always stopping at step 75. KL alone produces the same alarms; entropy alone alarms on all six runs. In this configuration, the conjunction neither reliably separates outcomes nor improves on KL alone. Per-seed results appear in App. N.

Historical replay and simpler triggers. We apply the frozen detector to 89 archived 1.7B trajectories with TROLL telemetry (App. L, Table 16). Of their 26 decline events, it detects 25 early and one late, with no misses and 10 alarms on runs without a recorded event. This replay includes 31 of the 32 runs inspected during detector design. The other 58 archived runs yield 16 early detections, one late detection and six no-event alarms; their earlier date does not make this a new prospective test. The joint rule trades a small difference in timing for fewer alarms: either signal alone, or an alarm when either signal rises, moves at most one late detection to early but produces 13–42 no-event alarms. Across multipliers 1.5–3 and persistence lengths 1–5, CDG misses no event; response-length and training-reward triggers perform worse. An absolute pre-projection KL threshold of 0.005, chosen after viewing 27 local trajectories, matches CDG there. A threshold of 0.02 is late or misses two of nine declines. A separate configuration-based detector predicts risk from the number of passes and radius alone. Flagging two-pass runs with $\epsilon \geq 0.028$ misses seven of the 26 events. Refitting the radius cutoff to 0.0125 after the sweep misses one and gives 13 false flags. Six of CDG's ten no-event alarms occur in two-pass published-radius runs that ended at step 75 or 100 (Table 14).

Value of stopping. This comparison asks which saved model to return from each of the six prospective trajectories. CDG returns the alarm checkpoint, or the final model if no alarm occurs. Label-based selection scores the candidate checkpoints on 300 GSM8K questions with reference answers and chooses the best-scoring checkpoint. We randomly split the 1,319 test questions into these 300 selection questions and 1,019 reporting questions, repeat the split 1,000 times, and report all policies on the reporting portion (Table 4). This separates checkpoint selection from reporting within each split; the test set had also informed detector design. We also evaluate two fixed choices. One always returns the step-75 checkpoint. The other, a hindsight configuration rule, selects step 75 for two-pass runs with $\epsilon \geq 0.0125$ and the final model otherwise. CDG reaches 83.98%, compared with 83.49% for label-based selection and 84.03% for the hindsight rule. The differences are $+0.49$ points (5–95% range across splits $[-0.07, +1.31]$) and -0.05 points ($[-0.26, +0.18]$), respectively. The split-based ranges summarize checkpoint-selection variability on the six training runs. Always stopping at step 75 matches CDG on the published-radius runs but ends the tight-radius runs earlier, giving 81.92% instead of 83.68%. On the 26 historical event runs, selecting the last periodic checkpoint at or before the alarm gives a median score 0.08 points below the run's peak and 23.58 above its endpoint. For context, the earlier single-pass cohorts reach 85.32–87.74% and the two-pass PPO-clip cohort reaches 85.39% (Table 2 and Section 5); these are separate training cohorts.

Table 4: Returned-model accuracy on 1,019 report questions not used for selection, mean over 1,000 random validation/report splits of the six prospective runs. The oracle picks each run's best checkpoint on the report questions (not usable). The test set also guided CDG's design.

policy	labels	ϵ=0.05 (3)	ϵ=0.00625 (3)	all six
run to the end	0	56.04	83.68	69.86
fixed stop at step 75	0	84.38	81.92	83.15
validation selection	300	84.07	82.91	83.49
CDG (alarm checkpoint)	0	84.28	83.68	**83.98**
configuration rule (hindsight)	0	84.38	83.68	84.03
report-split oracle (upper bound)	1,019	84.55	83.83	84.19

Switching versus simple training strategies. The development switch run on seed 1 ends 2.43 points above its alarm checkpoint (Fig. 1c). To compare whole-run strategies, seeds 601–603 each independently train U (one pass throughout), F (two passes through step 50, then one) and C (two passes until the CDG alarm, then one). Each run generates 38,400 answers over 150 steps; optimizer updates and GPU time are not matched. C alarms at steps 92/80/77 and gains $2.58/3.26/1.90$ GSM8K points after its own alarm. Mean final GSM8K accuracy is $85.67/85.82/84.69$ for U/F/C, and MATH-500 is $61.20/60.93/60.60$. C's GSM8K differences from U are $-1.06/+2.43/-4.32$ points, and from F $-2.43/+2.65/-3.64$. None of the three blocks passes the planned criterion: at least 1 GSM8K point above both alternatives without trailing either by more than 3 MATH-500 points. Seed 602 leads on GSM8K but loses $3.4/3.8$ MATH-500 points to U/F. Their three-seed results do not establish an advantage for alarm timing, and the fixed switch also lacks a consistent advantage over U.

Timing from a common step-50 parent. New two-pass prefixes provide a shared step-50 parent for three continuations: keep two passes, switch immediately to one pass, or wait for CDG and then switch. Immediate switching ends at $87.11/85.75/86.05$ for seeds 601–603, against $43.59/53.00/61.79$ from continued two-pass training. Signal-triggered switching ends at $83.40/85.67/84.99$, with alarms at steps $62/84/62$. All three signal branches outperform continued two-pass training, but trail immediate switching by $3.71/0.08/1.06$ GSM8K points and $2.2/7.8/2.0$ MATH-500 points.

8 Conclusion

TROLL's component-level gains and stability under reuse are distinct. In our experiments, removing the projection gradient causes no detectable accuracy loss, while unclipped policy gradient recovers most of the low-reuse gain over default clipping. Under repeated passes, smaller radii improve stability, and same-checkpoint interventions show that removing the second pass prevents further degradation and enables continued learning. Choosing when to intervene remains unresolved: CDG supports model retention in the initial prospective cohort, but alarm-based switching does not establish an advantage over simple alternatives, and the fixed rule does not reliably distinguish declining from non-declining runs in the same-configuration stress test.

Limitations. The controlled study uses Qwen3-1.7B, GSM8K training, MATH-500 evaluation and one trainer; the 8B study is descriptive. Data-order seeds are not independent model initializations. Same-parent continuations restore model, optimizer and data state with fresh generation engines; removing a pass also reduces optimizer updates. Shared-prefix strategies can diverge before their actions, so their differences do not isolate a precise delay effect. The original two-radius prospective cohort separates outcomes by configuration, and the test set informed detector design. The fixed detector produces both missed events and no-event alarms in the detached, zero-regression stress test. In a separate two-pass run at ϵ=0.05, sparse and full-vocabulary KL closely agree on sampled tokens through step 50. Their mean ratio (mean sparse KL divided by mean full-vocabulary KL) reaches 1.44 at step 75 and 5.23 at step 125, overlapping the alarm period; the contribution of this overestimation to the signals and degradation remains unresolved. Designs for further experiments appear in App. M.

AI USE STATEMENT

We used LLM-based agents mainly to search and summarize literature, to polish and translate the manuscript, and to write and modify code (training patches and analysis scripts). The authors reviewed all AI-assisted work, checked the reported claims against the experimental outputs, and take full responsibility for the manuscript.

REPRODUCIBILITY STATEMENT

Evidence coverage differs across waves. The later comparisons retain per-step logs, configurations, saved evaluation answers and analysis scripts. The new common-parent analysis independently rescored 102,882 GSM8K answer records with zero mismatches; the six-seed stress analysis rescored 59,355 with zero mismatches. These counts include repeated evaluations, not independent questions. MATH-500 scores in these two analyses were read from raw records rather than independently rescored again. Older waves provide console logs and configurations, and two original attribution endpoints (α=0, seeds 1–2) remain completion-ledger-only. The anonymous archive at `https://anonymous.4open.science/r/trd-raw-v1-BBD0` contains the original evidence and the evidence packages for the comparisons added here. Protocol and result sources for the new comparisons are listed in App. N.

ETHICS STATEMENT

This work studies optimization algorithms for language-model post-training on public mathematical reasoning benchmarks and involves no human subjects, private data, or foreseeable dual-use concern beyond those generic to improving LLM training efficiency.

REFERENCES

Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Rémi Munos, Nicolas Heess, and Martin A. Riedmiller. Maximum a posteriori policy optimisation. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL `https://openreview.net/forum?id=S1ANxQW0b`.

Brandon Amos and J. Zico Kolter. Optnet: Differentiable optimization as a layer in neural networks. In Doina Precup and Yee Whye Teh (eds.), *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pp. 136–145. PMLR, 2017. URL `http://proceedings.mlr.press/v70/amos17a.html`.

Philipp Becker, Niklas Freymuth, Serge Thilges, Fabian Otto, and Gerhard Neumann. TROLL: Trust regions improve reinforcement learning for large language models. In *International Conference on Learning Representations (ICLR)*, 2026. arXiv:2510.03817.

Minping Chen, Bowen Xiao, Du Liang, Chuxuan Zeng, and Zeyi Wen. Efficient hyperparameter optimization for LLM reinforcement learning. *arXiv preprint arXiv:2606.03073*, 2026.

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

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

Rousslan Fernand Julien Dossa, Shengyi Huang, Santiago Ontañón, and Takashi Matsubara. An empirical investigation of early stopping optimizations in proximal policy optimization. *IEEE Access*, 9:117981–117992, 2021. doi: 10.1109/ACCESS.2021.3106662.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

K2 Team, Zhengzhong Liu, Liping Tang, Linghao Jin, Haonan Li, Nikhil Ranjan, Desai Fan, Shaurya Rohatgi, Richard Fan, Omkar Pangarkar, Huijuan Wang, Zhoujun Cheng, Suqi Sun, Seungwook Han, Bowen Tan, Gurpreet Gosal, Xudong Han, Varad Pimpalkhute, Shibo Hao, Ming Shan Hee, Joel Hestness, Haolong Jia, Liqun Ma, Aaryamonvikram Singh, Daria Soboleva, Natalia Vassilieva, Renxi Wang, Yingquan Wu, Yuekai Sun, Taylor Killian, Alexander Moreno, John Maggs, Hector Ren, Guowei He, Hongyi Wang, Xuezhe Ma, Yuqi Wang, Mikhail Yurochkin, and Eric P. Xing. K2-V2: A 360-open, reasoning-enhanced LLM. *arXiv preprint arXiv:2512.06201*, 2025.

Jing Liang, Hongyao Tang, Yi Ma, Jinyi Liu, Yan Zheng, Shuyue Hu, Lei Bai, and Jianye Hao. Squeeze the soaked sponge: Efficient off-policy reinforcement finetuning for large language model. *arXiv preprint arXiv:2507.06892*, 2025.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Yuchun Miao, Sen Zhang, Yuqi Zhang, Yaorui Shi, Qi Gu, Xunliang Cai, and Lefei Zhang. When to stop reusing: Dynamic gradient gating for sample-efficient RLVR. *arXiv preprint arXiv:2605.19425*, 2026.

Manh Nguyen, Dung Nguyen, Dai Do, Svetha Venkatesh, and Hung Le. Uncertainty-guided checkpoint selection for reinforcement finetuning of large language models. *arXiv preprint arXiv:2511.09864*, 2025.

Fabian Otto, Philipp Becker, Ngo Anh Vien, Hanna Carolin Maria Ziesche, and Gerhard Neumann. Differentiable trust region layers for deep reinforcement learning. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=qYZD-AO1Vn.

Xue Bin Peng, Aviral Kumar, Grace Zhang, and Sergey Levine. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Jan Peters, Katharina Mülling, and Yasemin Altun. Relative entropy policy search. In Maria Fox and David Poole (eds.), *Proceedings of the Twenty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2010, Atlanta, Georgia, USA, July 11-15, 2010*, pp. 1607–1612. AAAI Press, 2010. doi: 10.1609/AAAI.V24I1.7727. URL https://doi.org/10.1609/aaai.v24i1.7727.

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.

John Schulman, Sergey Levine, Pieter Abbeel, Michael I. Jordan, and Philipp Moritz. Trust region policy optimization. In Francis R. Bach and David M. Blei (eds.), *Proceedings of the 32nd International Conference on Machine Learning, ICML 2015, Lille, France, 6-11 July 2015*, volume 37 of *JMLR Workshop and Conference Proceedings*, pp. 1889–1897. JMLR.org, 2015. URL http://proceedings.mlr.press/v37/schulman15.html.

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. *CoRR*, abs/2402.03300, 2024. doi: 10.48550/ARXIV.2402.03300. URL https://doi.org/10.48550/arXiv.2402.03300.

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. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

H. Francis Song, Abbas Abdolmaleki, Jost Tobias Springenberg, Aidan Clark, Hubert Soyer, Jack W. Rae, Seb Noury, Arun Ahuja, Siqi Liu, Dhruva Tirumala, Nicolas Heess, Dan Belov, Martin A. Riedmiller, and Matthew M. Botvinick. V-MPO: on-policy maximum a posteriori policy optimization for discrete and continuous control. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=SylOlp4FvH.

Peidong Wang, Demi Wang, Xufang Luo, Jiahang Xu, Xiaocui Yang, Shi Feng, Yuqing Yang, and Dongsheng Li. What are key factors for updates in rl for llm reasoning? *arXiv preprint arXiv:2606.22570*, 2026.

Yu Wang. The dark room in the reward channel: Dense prediction rewards collapse GRPO-trained LLM agents – and the channel, not the content, decides what works. *arXiv preprint arXiv:2607.21273*, 2026.

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.

Jiarui Yao, Xiangxin Zhou, Penghui Qi, Wee Sun Lee, Liefeng Bo, and Tianyu Pang. Rethinking the divergence regularization in llm rl. *arXiv preprint arXiv:2606.09821*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru 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, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2503.14476.

Lingzhe Zhang, Tong Jia, Yunpeng Zhai, Liancheng Fang, Kening Zheng, Hongyi Liu, Xiaosong Huang, Philip S. Yu, and Ying Li. Towards robust LLM post-training: Automatic failure management for reinforcement fine-tuning. *arXiv preprint arXiv:2605.04431*, 2026a.

Wenhao Zhang, Yibo Xie, Rui Wang, Jiahua Yang, Lei Jiang, Zibo Yang, Yawei Wang, Jiali Xu, jasperawang, Haoyang Long, Huan Xiong, and alantzhao. Reusing rollouts under policy lag: Prefix-normalized policy optimization for llm reinforcement learning. *arXiv preprint arXiv:2608.01418*, 2026b.

Yaxiang Zhang, Yingru Li, Jiacai Liu, Jiawei Xu, Ziniu Li, Qian Liu, and Haoyuan Li. Beyond precision: Training-inference mismatch is an optimization problem and simple LR scheduling fixes it. *arXiv preprint arXiv:2602.01826*, 2026c.

Table 5: Paired per-seed statistics (n=3; $t_{0.95,2} = 2.920$). The $\pm$1pt TOST fails in both C1 regimes at any realistic seed count (fixed-parameter CI projection first fits at weak $n\approx8$ and strong $n\approx109$; prospective power at the observed effect/variance: weak $n\approx14$ and strong $n\approx245$ for 80%), so point estimates with CIs are the primary readout. The 1.7B C3 test is one-sided paired $t(2) = 3.85$, $p \approx 0.031$. The 8B rows are descriptive diagnostics for 3 fixed seeds, not population-rate, significance, robustness, or equivalence claims.

Contrast	per-seed diffs	mean	interval
C1 weak: detach − full (150 st.)	-1.21, -0.45, +0.68	-0.33	[-1.94, +1.28]
C1 strong: detach − full (100 st., post-hoc)	-2.81, +3.34, +0.99	+0.51	[-4.72, +5.73]
C2: no-clip − full (150 st.)	-0.68, -0.53, +0.99	-0.08	[-1.63, +1.48]
C2: clip − full (150 st.)	-2.43, -1.29, -1.90	-1.87	[-2.83, -0.91]
C3: budget − fixed ϵ (K=16, 150 st.)	+14.94, +25.93, +38.82	+26.56	[+6.41, +46.71]
8B C3: S3 − S2 strict (150 st.)	+47.99, -15.39, +65.35	+32.65	[-72.93, +138.23]‡
8B C3 count: observed (of 3 fixed seeds)	2/3	—	[0.0943, 0.9916]§

‡95% t interval; §95% Clopper–Pearson diagnostic under an IID-binomial model.

Xiangxin Zhou, Jiarui Yao, Penghui Qi, Bowen Ping, Jiaqi Tang, Haonan Wang, and Tianyu Pang. Predictive divergence masks for llm rl. *arXiv preprint arXiv:2607.10848*, 2026.

A Statistical details

Three-state rule. With B the mean of the first two evaluation points (accuracy, and entropy and response length at the nearest training step) and peak the maximum of the 3-point running median of accuracy, a run is *accuracy-stable* if its final accuracy is at least peak-10 and at least $B_{\text{acc}} - 5$. HEALTHY: accuracy-stable and final entropy $\leq \max(1.5B_{\text{ent}}, 0.8)$. COLLAPSE: not accuracy-stable and final entropy $> \max(3B_{\text{ent}}, 1.0)$ or final length $>2.5B_{\text{len}}$ (truncation rate if length saturates). DEGRADED: otherwise.

Equivalence-test precision. Table 5 lists all paired contrasts. A post-hoc power calculation at the observed effect sizes and variances gives the following power for a $\pm$1pt equivalence margin: in the weak regime, power is $\approx27\%$ at n=5 and $\approx53\%$ at n=8, first reaching 80% near $n\approx14$; in the strong regime $n\approx245$ would be needed. We therefore report point estimates and CIs as primary and treat all equivalence-flavoured language as descriptive.

MATH-500 secondary metric. Table 6 gives the weak-regime (K=4, step-150) MATH-500 transfer values per seed with paired 90% CIs against full TROLL ($t_{0.95,2}$=2.920); strong-regime means are $62.0/60.8/61.0$ (full/detach/clip). The GSM8K ordering does not replicate—clip has the best transfer mean—though every CI contains zero; the metric is noisy at this scale and we draw no conclusion from it beyond scoping the C2 attribution to the tested task.

Table 6: Weak-regime MATH-500 transfer (%, eval-only; trained on GSM8K). Paired differences are arm−full per seed; 90% CIs use $t_{0.95,2}$.

arm	s0	s1	s2	mean	paired Δ vs full [90% CI]
full TROLL	59.8	61.0	62.4	61.1	—
detach-only	63.0	61.6	63.4	62.7	$+1.60$ $[-0.76, +3.96]$
α=0	59.0	62.0	60.6	60.5	-0.53 $[-2.93, +1.86]$
no-clip	64.4	62.0	59.0	61.8	$+0.73$ $[-6.02, +7.49]$
PPO-clip	64.8	62.8	63.4	63.7	$+2.60$ $[-0.97, +6.17]$

Strong-regime 150-step horizon (3 seeds). Table 7 completes the pre-registered horizon for the strong regime at the published radius. Both projection arms are non-HEALTHY in $3/3$ seeds under the frozen three-state rule while PPO-clip is HEALTHY in $3/3$; the paired detach−full mean is

-7.25pt with 90% CI $[-24.14, +9.64]$, which a collapse regime makes uninformative by construction and which we therefore do not interpret as a detach-vs-full comparison. The range-10 column is the pre-registered no-constraint control of Section 5.

Table 7: Qwen3-1.7B, K=16, ϵ=0.05 (published radius), step-150 endpoint, GSM8K % on the flexible basis. Seed-0 full/detach/clip values are the archived runs; all other cells are new. States follow the frozen three-state rule.

seed	full TROLL	detach-only	PPO-clip	range-10 (no constraint)
0	64.06	54.51	83.70	5.91
1	62.32	46.40	86.05	77.10
2	30.02	33.74	86.43	69.22
non-COLLAPSE	0/3	0/3	3/3	1/3

8B descriptive intervals. For the frozen 2/3 tally, the Clopper–Pearson 95% interval is $[0.094, 0.992]$; the per-seed S3$-$S2 strict differences $\{+47.99, -15.39, +65.35\}$ give mean $+32.65$ with 90% paired t interval $[-72.93, +138.23]$. These are model-based diagnostics over three fixed seeds, not population inferences.

Binding-radius regression–path factorial. At K=16 and ϵ=0.003125, the factorial crosses the full versus detached projection path with regression weight $\alpha \in \{0, 1\}$. The α=1 cells reuse historical M7-A and M6 runs; all six α=0 arms were executed in a later cohort. The reuse audit checks logged configuration and runner compatibility, but does not certify complete historical patched-source equivalence. The primary outcome is normalized trapezoidal GSM8K AUC over the seven validation points. The same α=1 contrast is $+0.015$pt on AUC and $+0.834$pt on the step-150 endpoint; these are distinct estimands.

path	α	AUC mean	AUC SD	states (s0/s1/s2)
full	1	80.575	0.075	H/H/H
detached	1	80.589	0.284	H/H/H
full	0	81.177	0.500	H/D/D
detached	0	80.962	2.662	D/H/D

The paired interaction $(\text{detach} - \text{full})_{\alpha=0} - (\text{detach} - \text{full})_{\alpha=1}$ is $-2.824/-0.411/+2.546$pt by seed, with mean -0.230pt and 90% CI $[-4.764, +4.305]$. It fails the frozen requirements of a common direction, CI exclusion of zero, and absolute mean above one point. The factorial therefore returns `NO_IDENTIFIED_DIFFERENCE_AT_N3`; this result does not establish equivalence or rule out a masking effect. Endpoint sensitivity gives interactions $-8.415/+0.682/+0.986$pt, mean -2.249pt with 90% CI $[-11.256, +6.757]$, also unidentified. The α=0 execution cohort is DEGRADED-majority in both path cells, whereas the historical α=1 cohort is HEALTHY-majority. Because α is confounded with execution cohort and checkout, we do not attribute this shift to the regression term. An additive cohort shift would cancel in the interaction, but a cohort-by-path effect would not.

B Reuse-aware radius: derivation and limits

For an idealized sequence of equal-energy incremental policy displacements δ_j, let $\frac{1}{2}\|\delta_j\|_F^2 = \epsilon_k$ in local Fisher geometry and let $\bar{\rho}$ be their mean pairwise Fisher cosine. Expanding the squared norm of their sum gives

$$D_K \;\approx\; K\epsilon_k[1 + (K-1)\bar{\rho}], \qquad \epsilon_k \;\approx\; \frac{\Delta}{K[1 + (K-1)\bar{\rho}]}. \tag{2}$$

Hence $1/K$ follows for weakly aligned increments ($\bar{\rho} \approx 0$, or $O(1/K)$); coherent alignment approaches a $1/K^2$ requirement. Mean pairwise cosine satisfies $\bar{\rho} \geq -1/(K-1)$, so the denominator is nonnegative. With nonnegative alignment this surrogate spans exponents between one and two and cannot justify a $1/\sqrt{K}$ rule. Equation 2 is exact only within the local equal-energy model.

The actual method does not constrain δ_j: every detached teacher y_j is projected against one fixed rollout anchor, and the optimizer only tracks that teacher approximately. A more faithful local model is $x_{j+1} = ax_j + (1-a)y_j + u_j$, where $\frac{1}{2}\|y_j\|_F^2 \le \epsilon_k$, $a \in [0,1)$ is residual retention, and u_j collects stale-policy-gradient, optimizer, and tracking residuals. For zero initial displacement in the local anchor coordinates, unrolling yields a bounded teacher term $\sqrt{2\epsilon_k}(1-a^K)$ plus a correlation-dependent residual RMS R_K. Writing $D_K = \frac{1}{2}\mathbb{E}\|x_K\|_F^2$, a sufficient expected-drift certificate for $D_K \le \Delta$, provided $R_K \le \sqrt{2\Delta}$, is

$$\epsilon_k \;\le\; \frac{(\sqrt{2\Delta} - R_K)^2}{2(1-a^K)^2}. \tag{3}$$

Here R_K must upper-bound the effective residual RMS. If $R_K > \sqrt{2\Delta}$, this sufficient certificate provides no admissible radius. That does not establish that the actual feasible set is empty: the bound may be loose or teacher and residual terms may cancel. Setting ϵ_k to zero does not repair the certificate, but can still satisfy the actual drift target in cases the bound does not certify. Detachment also creates a learning-signal lower bound: if at most a fraction β of absolute-advantage mass may lose the direct policy-gradient path, then ϵ_k must exceed the corresponding weighted KL quantile. This constrains a proxy based on absolute-advantage mass, not the norm or direction of the full parameter gradient. Combining the conditions gives a model-dependent certified interval, not a necessary characterization of all feasible radii. Estimating a, residual scale and correlation, and the weighted quantile requires micro-update telemetry absent from the current runs. We use $\epsilon_k = \epsilon_{\text{total}}/K$ (Section 6.1) as an empirically stable default, not a theorem.

C SEED-0 EPSILON–K RESPONSE SURFACE

R056 evaluates 12 detach-only, 150-step arms on Qwen3-1.7B with $K \in \{1, 4, 16\}$ and a pre-specified radius grid. The recovered archive contains 1,800 training rows and 152,796 per-sample validation-rescore rows; all 2,258 referenced manifest entries pass SHA256. The GSM8K trajectory AUCs and frozen health labels are:

K	ϵ	GSM8K AUC	health
1	.003125	79.49	healthy
1	.0125	79.06	healthy
1	.05	79.11	healthy
4	.003125	80.41	healthy
4	.0125	82.54	healthy
4	.025	82.23	healthy
4	.05	80.47	healthy
16	.0015625	78.07	healthy
16	.003125	79.99	healthy
16	.00625	82.10	healthy
16	.0125	81.01	degraded
16	.05	74.25	collapse

The comparison requires every K point to be healthy and the maximum regret from that K's best healthy arm to be at most 1pt. No schedule passes. Fixed-default fails stability at K=16. Fixed-small and linear are healthy at all three K values but have maximum regrets of 2.123pt and 2.104pt; square-root has at most 0.379pt regret where healthy but is degraded at K=16. The frozen verdict is `INCONCLUSIVE_IDENTIFICATION`; we do not fit a post-hoc exponent from this single seed.

R056 is retained as an independent evidence line. At the nominal K=16, ϵ=0.0125, seed-0 cell it reports a DEGRADED trajectory ending at 82.18, whereas M8 reports a COLLAPSE ending at 64.97. The nominal configuration and scoring setup agree, but M8 lacks a run-time manifest for every patched source file, so the historical executions cannot be proven byte-identical. We do not pool their values or use the discrepancy to select either label.

D TARGETED REUSE-RADIUS FOLLOW-UP

The staged R058 follow-up tests whether K is a useful coarse reuse descriptor and whether the conservative 0.003125 radius is locally optimal at $K{=}16$. At $K{=}8$, changing the decomposition from mini-batch 4 times one epoch to mini-batch 8 times two epochs changes GSM8K trajectory AUC by -0.61pt at $\epsilon{=}0.0125$ and -0.33pt at $\epsilon{=}0.00884$; all four arms are healthy and both pass the frozen $\pm$1pt primary gate. Secondary MATH AUC, entropy, and response length do move, so the factorization is not fully interchangeable.

At $K{=}16$ and 150 steps, the paired detach-only comparison is:

seed	AUC(.003125)	AUC(.00625)	difference
1	80.219	81.665	+1.447pt
2	79.903	81.482	+1.579pt

All four trajectories over two paired data-order seeds are healthy; the descriptive paired mean is $+1.513$pt (sample SD 0.094pt). Endpoint effects are not uniformly positive (seed-1 final MATH falls 3.6pt), so the result concerns trajectory AUC. With $n{=}2$, it is directional evidence without a population inference. Moreover, $0.025/\sqrt{K}$ and $0.1/K$ coincide at 0.00625 for $K{=}16$; these runs cannot identify a unique scaling law or an epsilon-by-detachment interaction.

E PRE-REGISTERED RADIUS-BOUNDARY SWEEP

M8 classifies the detached $K{=}4$ cells at $\epsilon \in \{0.1, 0.2\}$ as HEALTHY in 3/3 seeds. At $K{=}16$, $\epsilon{=}0.0125$ gives COLLAPSE/DEGRADED/DEGRADED and $\epsilon{=}0.025$ gives COLLAPSE/DEGRADED/COLLAPSE, placing the frozen majority non-COLLAPSE boundary in $[0.0125, 0.025]$. The K=4 upper boundary was still open, so M8 returned `INDETERMINATE` rather than fitting an exponent. The 0.0125 point is HEALTHY in $0/3$; it is not a recommended healthy radius. Table 8 reports all twelve M8 arms from the recovered raw console logs and their runner-bound completion records.

Table 8: M8: all twelve formal arms. Accuracy and normalized trajectory AUC are percentages; H/D/C denotes HEALTHY/DEGRADED/COLLAPSE under the frozen rule. The 0.0125 cell is non-COLLAPSE-majority but has no HEALTHY seed.

K	ϵ	seed	GSM AUC	GSM final	state
4	0.1	0	81.305	85.747	H
4	0.1	1	82.373	85.216	H
4	0.1	2	82.215	85.747	H
4	0.2	0	81.905	84.913	H
4	0.2	1	82.727	85.519	H
4	0.2	2	82.228	86.657	H
16	0.0125	0	80.307	64.973	C
16	0.0125	1	80.149	78.848	D
16	0.0125	2	80.516	77.786	D
16	0.025	0	79.580	65.353	C
16	0.025	1	81.343	75.588	D
16	0.025	2	77.268	57.316	C

M8E1 extends K=4 to $\epsilon{=}0.4$ under a separate frozen amendment. All three arms remain HEALTHY (GSM8K AUCs 80.730/82.657/81.766; step-150 values 83.70/85.29/85.22), so the required 2/3-collapse gate fires in 0/3 seeds. The mean projected fractions are only 0.00304%, 0.00143%, and 0.00102%, and all three are zero at step 150. The K=4 boundary remains unbracketed; the proposed $[0.2, 0.4]$ boundary and $[8, 32]$ cross-K ratio bracket are not supported. These runs identify neither an accuracy optimum nor a scaling exponent. Adding the 0.4 extension gives a descriptive ratio anchor of 32 and an open interval $[16, \infty)$ under common-boundary, monotonicity and cross-cohort compatibility assumptions. This is not a confidence interval; the frozen verdict remains `INDETERMINATE`.

F EXPLORATORY CONSTRAINT-TIMING FOLLOW-UPS

Design and estimands. P27-C keeps $K{=}16$, Qwen3-1.7B full fine-tuning, the full projection path, $\alpha{=}1$, learning rate 10^{-6}, and the 64-logit sparse cap fixed. It varies radius with *global training step* t; it does not test a scaling law across K. Tight (T) denotes $\epsilon{=}0.003125$ and loose (L) denotes 0.05. TT and LL are constant, while TL and LT75 use 75 steps of each radius in opposite orders (switching at step 76). The first round runs these four configurations and clip $0.2/0.3$ with data-shuffle seeds 101/102. The second round adds constant 0.00625 (FIX) for both seeds and LT50, which switches at step 51, for seed 101 only. Rollout engine seed is zero throughout; these are not independent repetitions of all random sources. The second round follows inspection of the first, so these seeds are not a fresh holdout for selecting a new control rule.

All 15 arms finish 150 global training steps. The frozen V1 reward and evaluation scorer are unchanged. GSM8K has 1,319 evaluation questions and MATH-500 has 500. Trajectory AUC is the trapezoidal integral of accuracy at steps $0/25/50/75/100/125/150$, divided by 150. It averages quality over training progress, not wall-clock time or GPU cost. The step-150 endpoint is reported separately; MATH is a transfer readout, not a second training task. Table 9 includes every formal arm in these rounds.

Table 9: P27-C: all 15 formal arms from the two rounds, frozen V1 scoring. AUC averages seven evaluations over training progress; final is step 150. All values are percentages. LT50 has one data-shuffle seed; all other configurations have two. Comparisons retain the batch-order qualification in text.

configuration	data seed	GSM AUC	GSM final	MATH AUC	MATH final
TT	101	80.939	82.563	62.033	61.400
TT	102	79.618	82.032	60.750	62.400
LL	101	78.443	58.984	59.817	45.200
LL	102	77.982	56.785	59.700	50.600
TL	101	80.086	79.757	61.233	62.200
TL	102	80.831	83.397	61.483	62.800
LT75	101	82.165	82.411	62.583	62.000
LT75	102	81.312	80.970	61.433	62.200
FIX .00625	101	81.179	83.321	62.633	65.800
FIX .00625	102	80.389	83.851	62.033	62.200
LT50	101	81.804	83.245	62.800	62.600
clip .2	101	82.878	85.368	61.700	60.600
clip .2	102	82.803	83.851	60.367	61.000
clip .3	101	82.398	83.851	60.867	60.600
clip .3	102	82.417	85.292	60.733	61.400

Ordering and the constant-radius competitor. LT75$-$TL GSM8K AUC differences are $+2.079/+0.480$pt, mean $+1.279$pt; endpoint differences are $+2.654/-2.426$pt, mean $+0.114$pt. Both schedules end far above LL, whose endpoints are $58.984/56.785\%$. Thus late tightening is not the only tested schedule that avoids LL's endpoint deterioration. Equal tight/loose step counts do not equate measured KL or gradient budgets. A post-hoc time decomposition assigns 0.878pt of the AUC difference to the first 75 steps and 0.401pt to the remainder, so the entire gain cannot be attributed to tightening at step 76.

FIX exceeds TT's AUC by $+0.240/+0.771$pt and its endpoint by $+0.758/+1.820$pt. LT75$-$FIX AUC is $+0.986/+0.922$pt (mean $+0.954$), but its endpoint is $-0.910/-2.881$pt (mean -1.895). The exact AUC mean 0.954006 is below the pre-specified 1pt discussion threshold, and the endpoint loss exceeds the 1pt tradeoff threshold. This does not prove that schedules are ineffective or equivalent; the practical threshold is not a discontinuity in scientific effect.

Earlier switch and clipping. For seed 101, LT50$-$LT75 gives AUC -0.360pt, GSM endpoint $+0.834$pt, and MATH endpoint $+0.600$pt. The endpoint gain misses the pre-specified 1pt probe criterion; the other two criteria pass. Against FIX at the same seed, LT50's GSM endpoint is lower by one question and MATH by 3.2pt. The conjunctive probe screen therefore does not justify automatic expansion. Compared with clip 0.2, FIX averages -2.056pt GSM AUC and -1.024pt

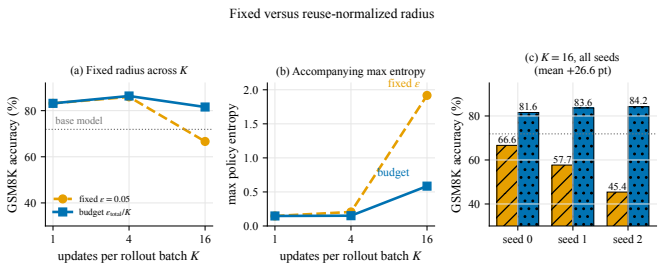


Figure 2: Qwen3-1.7B GSM8K K-sweep (150 steps). Panels (a,b) use one exploratory seed at each $K \in \{1, 4, 16\}$ for fixed ϵ=0.05 versus ϵ_k=$\epsilon_{\text{total}}/K$; the K=1 point is shared because the schedules coincide. Entropy inflation accompanies, but is not claimed to cause, the fixed-radius collapse. Panel (c) shows all 3 K=16 seeds: mean budget−fixed difference +26.6pt (90% paired CI $[+6.41, +46.71]$). Descriptive K-sweep; no scaling-law claim.

GSM endpoint, but +3.2pt MATH endpoint; against clip 0.3 the corresponding differences are $-1.624/-0.986/+3.0$pt. These are task-specific tradeoffs, not evidence of an overall winner under a post-hoc combined metric. The K=16 clip comparisons do not resolve tuning at the original K=4 attribution point.

Evidence qualification. All saved training rewards and evaluation scores were recomputed from raw answers under the frozen scorer; model/data/scorer identities agree across the 15 arms. However, the original strict requirement of identical ordered prompt traces failed. For each same-seed global step the question/answer multiset and multiplicities match; generation-length-based batch balancing changes order and can change optimizer minibatch grouping. The extra multiset check does not retroactively pass the original gate. Scientific reuse and pairing qualification remain pending independent review, and we report the numerical comparisons descriptively. LT50 and LT75 were also independent runs on different instances, not branches from one checkpoint; their evaluations already differ before step 51. Neither these differences nor the step-local initial-KL telemetry identify a cross-step feedback mechanism. With two data-shuffle seeds (one for LT50), these results do not establish a general timing rule, final-model superiority, or the effectiveness of a learned radius controller.

G Additional 1.7B and control figures

Two readings of Fig. 2 matter for Section 6.1: the two schedules are indistinguishable at K=1 and diverge monotonically with reuse, and the fixed-radius failure is an end-of-run collapse rather than a uniformly lower curve—which is why the best-checkpoint accounting of Section 6.1 favors fixed ϵ while the endpoint accounting reverses it.

We read Fig. 3 narrowly. Its no-clip arm ran a separate exploratory code path and is therefore *not* the arm behind the necessity claim: that claim rests on the matched range-10 runs of Table 7 ($2/3$ collapse). What this figure adds is a second, independent failure mode at the same recipe—a DPPO-style binary divergence mask collapses to 4.7 with final entropy 11.2—and the gradient-surgery arms, which carry no claim beyond "the collapse is not specific to the projection".

H Additional 8B figures

The heterogeneity across seeds in Fig. 4 is the reason Appendix K reports per-seed states under the frozen three-state rule instead of averaging: seed 1 provides no evidence about collapse (its S2 never fails), and seeds 0 and 2 fail in qualitatively different ways, only one of which is visible on the flexible basis.

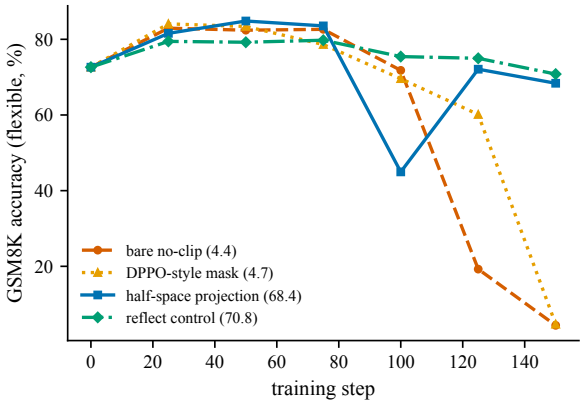


Figure 3: Qwen3-1.7B GSM8K, K=16, fixed ϵ=0.05, single-seed descriptive controls (flexible basis, 150 steps): removing the constraint (bare no-clip, final 4.4) or binary-masking out-of-region tokens (DPPO-style mask, not a DPPO reproduction; final 4.7) both collapse; a half-space gradient-surgery arm (for out-of-region tokens the outward gradient component is projected onto the KL constraint's tangent half-space; labeled "project") and its reflect control (the outward component reflected inward) end far higher. These are gradient-surgery variants from a separate exploratory line — not TROLL's distribution projection — sharing the exact K=16 recipe. Final entropies: no-clip 2.21, mask 11.20.

Table 10: 8B arm codes (Qwen3-8B LoRA rank 32, K=16 unless noted, 150 steps). W1/S1 are full-path; W2/S2/S3/S3P are detached. "Tuned clip" decouples the ranges DAPO-style.

code	regime	seeds	configuration
W1	weak K=4	0,1,2	full TROLL (`loss_mode=trpl`, ϵ=0.05)
W2	weak K=4	0,1,2	W1 + detached projection
W3	weak K=4	0,1,2	PPO-clip (`vanilla`, range 0.2)
W4	weak K=4	0	range-10 vanilla (`clip_ratio/low/high=10`)
S1	strong K=16	0	full TROLL, fixed ϵ=0.05
S2	strong K=16	0,1,2	TRPL detached, fixed ϵ=0.05, no budget
S3	strong K=16	0,1,2	TRPL detached, ϵ_k=$\epsilon_{\text{total}}/K$=0.003125
S3P	strong K=16	0	TRPL detached, fixed ϵ=0.00625, no budget
S4H	strong K=16	0	tuned PPO-clip (low 0.2 / high 0.28)

I Efficiency accounting

Raw per-step medians are confounded by rollout-length differences between arms; we therefore report per-token timing as the primary readout in Table 11.

J Hyperparameters

All values below are read from the archived resolved configs in the released raw logs (not from launch scripts); the release archive lets any row be re-verified. Both waves share the upstream TROLL fork of verl. Within each wave, arms differ only in the flags of Table 10 and Section 3; the cross-wave changes are stated below.

The 8B wave changes the model (Qwen3-8B), actor learning rate (10^{-5}), rollout GPU-memory utilization (0.4 rather than 0.5), and adaptation (LoRA rank 32, α_{LoRA}=64); it keeps K=16/150 steps for the strong arms and K=4 for the weak arms. The batch, sampling, length, aggregation, regularizer, projection, sparsification, and validation settings in the table are unchanged up to the arm flags of Table 10; per-arm resolved configs and golden digests ship in the same archive.

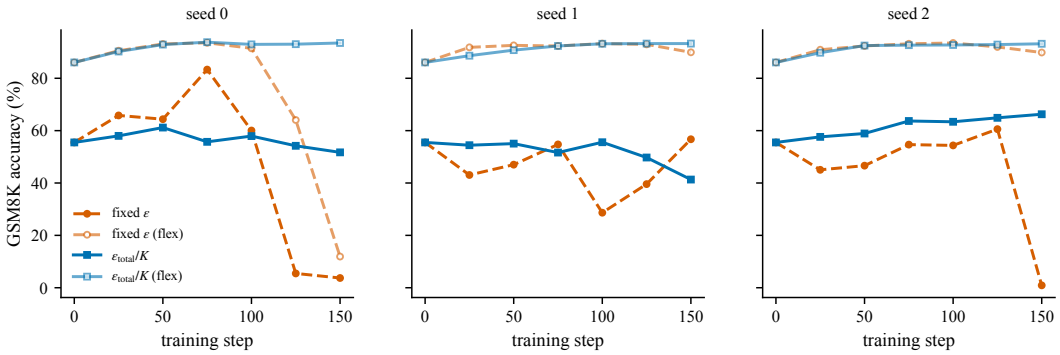


Figure 4: Qwen3-8B LoRA descriptive per-seed curves at K=16 and 150 steps, S2 (fixed ϵ=0.05) vs S3 (ϵ_k=$\epsilon_{\text{total}}/K$): solid is strict, dotted is flexible. Seed 0 shows a joint flexible+completion collapse of S2; seed 1's S2 stays healthy (premise absent); seed 2's S2 collapses on the strict basis while its flexible score remains high.

Table 11: Per-token `update_actor` time from the archived Qwen3-8B LoRA training logs (one run per arm; medians over 150 updates). Observational accounting, NOT a wall-clock or causal speed claim: raw step medians are confounded by rollout-length differences, which is why the per-token normalisation is the primary readout.

Pair	full (ms/tok)	detach (ms/tok)	Δ
W2 (detach) vs W1 (full)	0.3390	0.3365	-0.72%
S2 (detach) vs S1 (full)	0.6944	0.6877	-0.96%

K DESCRIPTIVE 8B REPLICATION AND ITS MEASUREMENT VALIDITY

Frozen-rule readouts. We ran the pre-registered strong-regime readouts on Qwen3-8B (LoRA rank 32) under the frozen per-seed protocol; Table 13 summarizes. For Q3, the pre-registered tally is *observed in 2/3 seeds, premise absent in 1/3*: seeds 0 and 2 collapse under the fixed radius while the budgeted arm does not, and seed 1's fixed-radius arm never collapses, leaving nothing to prevent. Only seed 0 is a joint collapse of flexible score and output completion; seed 2's collapse is a strict-format event with flexible accuracy intact, so we do not claim robust cross-seed collapse prevention. For Q1, the strict pre-registered record reads "observed in 2/3 seeds", but the semantic (flexible) contrast is inactive in 3/3 seeds: *capability equivalence is not identified at 8B*. Two single-seed controls bound the claims: a fixed radius of 0.00625 also trains healthily, and a tuned PPO-clip arm shows a distinct endpoint profile (near-perfect format emission, lowest accuracy conditional on emission). The replication also changes two factors at once relative to the 1.7B study — scale and LoRA adaptation — so no per-factor attribution is possible at 8B.

The audit that scopes all of the above. Fig. 5 explains why these readouts are descriptive. Across the 16 protocol-non-collapsed arms, strict accuracy is $\approx 0.945\times$ the windowed `#### N` emission rate (through-origin fit on the 13 arms with emission $\geq$20%; uncentered R^2=0.9998, centered 0.9975), while accuracy *conditional* on emitting the marker varies only between 91.6 and 95.6%. The training reward does not distinguish correct formats sharing the same final number, so the marker carries no direct, identifiable differential reward preference — which makes format drift possible, though it does not explain why arms drift at different rates. A scorer-window artifact is ruled out ($\leq$ 0.68% of rows on any arm). We read this as a general caution: when the training reward and the evaluation metric disagree about format, the metric measures format emission before it measures capability, and cross-arm "treatment effects" inherit that contamination.

Table 12: 1.7B wave (Qwen3-1.7B, full fine-tuning, GSM8K train).

optimizer / LR	AdamW, 10^{-6}, constant schedule, weight decay 0, grad clip 1.0
batch structure	32 prompts/batch $\times$ n=8 rollouts = 256 responses
reuse K=4 (weak)	mini-batch 8, `ppo_epochs` 1, 150 steps
reuse K=16 (strong)	mini-batch 4, `ppo_epochs` 2; 100 steps (B1) / 150 (Section 6.1)
sampling	temperature 1.0, top-p 1.0, top-k off; thinking mode disabled
lengths	prompt $\leq$ 1024, response $\leq$ 1024 tokens
configured precision	actor/reference model dtype FP32; rollout engine bfloat16
loss aggregation	masked token-mean
regularizers	entropy coef 0; KL-to-reference off in loss and reward
PPO-clip arm	`loss_mode=vanilla`, clip range 0.2 (verl default, untuned)
no-clip arm	`vanilla`, clip ratio/low/high 10 (effectively unclipped; not exact)
TRPL	α=1, ϵ=0.05 unless stated; token-level (`project_full_sequence=False`)
TRPL dual solver	grid over $\log \eta$ with $\eta \in [10^{-7}, 10^{2}]$, 8 points, $\leq$ 20 refinements
sparsification	targets omitted mass $\leq 10^{-5}$ subject to cap 64, plus forced sampled token; dropped-token default 10^{-12}
validation	GSM8K test (1,319) + MATH-500 (500) every 25 steps; greedy, one decode/problem
seeds	data seeds $\{0, 1, 2\}$ (`data.seed`); no other seed varied

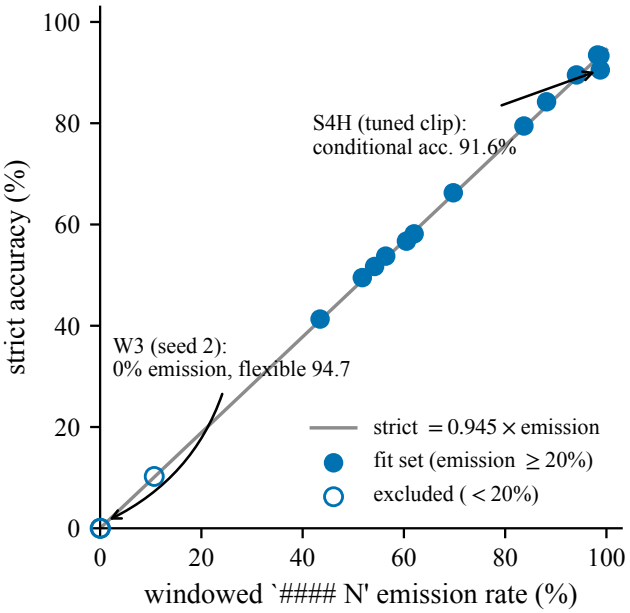


Figure 5: Measurement audit of the 8B strict readout (step 150). Across the 16 protocol-non-collapsed Qwen3-8B LoRA arms, strict accuracy is dominated by the rate at which the exact `#### N` terminator is emitted: a through-origin fit on the 13 arms with emission $\geq$20% gives strict $\approx$ 0.945$\times$emission (uncentered R^2=0.9998; centered R^2=0.9975). Open markers are outside the fit set. W3 (seed 2) emits the marker on 0% of answers yet scores 94.7 on the flexible basis; S4H emits it almost always but has the lowest accuracy conditional on emission (91.6%). The analysis concerns GSM8K terminator emission in the 8B LoRA runs.

L Co-Drift Guard: further details

The ten no-event alarms. Table 14 lists every alarm on an archived run without an event. Under the frozen event definition all ten are false alarms, and this is the count reported in the main text. Six of these alarms fall on runs whose horizon ended during a loose two-pass phase (the 100-step and 75-step runs), two on tight-then-loose runs 37 and 27 steps after their switch to the loose radius at

Table 13: Qwen3-8B LoRA descriptive replication (L2 wave, step 150; 3 fixed seeds, no population-rate or significance claim). S2: fixed radius ϵ=0.05 at K=16; S3: ϵ_k=$\epsilon_{\text{total}}/K$. States and C3 verdicts follow the frozen pre-registered three-state rule (strict basis); strict / flexible / windowed '#### N' emission are shown side by side because the strict basis is dominated by emission (Fig. 5). S3P (fixed ϵ=0.00625, no budget) and S4H (tuned PPO-clip) are pre-registered, seed-0, descriptive, non-population controls. Frozen tally: C3 observed in 2/3 seeds, premise absent in 1/3; capability equivalence is *not identified* at 8B.

		S2 (fixed ϵ)			S3 ($\epsilon_{\text{total}}/K$)		
Seed	C3 verdict	strict	flex	emit	strict	flex	emit
0 (collapse / degraded)	observed	3.7	11.9	5.2	51.7	93.5	54.2
1 (healthy / degraded)	premise absent	56.7	89.9	60.5	41.3	93.3	43.4
2 (collapse / healthy)	observed	0.9	89.8	0.9	66.3	93.2	69.7
S3P: fixed ϵ=0.00625 (healthy)			—		53.8	92.6	56.4
S4H: tuned clip (degraded)			—		90.5	90.5	98.8

step 76, and two on ϵ=0.003125 runs that the three-state rule labels DEGRADED but that have no 5-point event.

Table 14: Archived runs on which CDG alarms without a subsequent event.

run (cohort)	ϵ	passes	horizon	alarm step	end / peak GSM8K
B1 full, seed 0	0.05	2	100	92	83.09 / 84.76
B1 full, seed 1	0.05	2	100	77	81.73 / 85.37
B1 detach, seed 0	0.05	2	100	93	80.29 / 83.55
B1 detach, seed 2	0.05	2	100	90	84.69 / 84.84
factorial full α=0, seed 1	0.003125	2	150	130	82.11 / 82.26
factorial detach α=0, seed 2	0.003125	2	150	97	85.22 / 86.35
reuse order AABB, K=32	0.05	2	75	64	81.12 / 84.99
reuse order ABAB, K=32	0.05	2	75	57	79.76 / 82.71
tight-then-loose, seed 101	$0.003125 \rightarrow 0.05$	2	150	113	79.76 / 83.02
tight-then-loose, seed 102	$0.003125 \rightarrow 0.05$	2	150	103	83.40 / 83.70

Why a run-relative baseline (Section 7). The healthy level of k_t depends on how training is organized: its steps 1–25 mean is $1.94\times$ that of a K=8 single pass for two passes at K=16, $1.80\times$ for the batch-64 run and $2.97\times$ for two passes at K=32. An absolute level is sensitive to the value chosen (Section 7.2); a multiple of the run's own baseline scales with the organization without choosing one.

What each signal contributes (Section 7). KL carries the detection: alone it has the same 25/1/0 record in replay (Table 16) and stays at 1.34–$1.60\times$ on the tight prospective runs, whose entropy reaches 3.98–$5.33\times$. Adding entropy removes three of KL-only's 13 no-event alarms, all on single-pass runs.

Provenance of the replay corpus (Section 7.2). Choosing the signals involved logs of three of the older cohorts outside the design set; only 7 detections come from cohorts never consulted, and those carry all 6 of that set's no-event alarms. All ten of CDG's no-event alarms count as false alarms under the frozen event definition.

Absolute and relative frames on the prospective runs. Rolling back to the last periodic checkpoint at or before the alarm gives 83.90 on the three published-radius runs with the frozen relative rule and 81.22 with the earlier absolute rule (≥ 0.002 for five steps), which fires at steps 62, 29 and 48. Scoring the alarm checkpoint itself raises the relative rule to 84.26. With three runs on a 25-step grid, these differences are descriptive only.

Same-checkpoint forks (Section 6.2). Table 15 lists all 12 fork arms. Continuing at the published radius ends 19.2 to 51.7 points below the fork checkpoint, and tightening ends 0 to 8.3 points below it. The tightened arms of one session (v11) include a MATH-500 format collapse.

Table 15: Same-checkpoint forks at step 75 of two two-pass $K{=}16$, $\epsilon{=}0.05$ runs (GSM8K % at step 150). The 12 arms come from four launch sessions; bold entries share a session and form the within-session pairs quoted in the text.

parent	step-75 checkpoint	tightened (0.00625 or 0.003125)	continued at 0.05
201	84.16	**82.79** / 82.03 / 75.89 / 80.29†	**64.44** / 61.18 / 64.97
202	83.47	**81.35** / 83.47†	**45.94** / 31.77 / 47.31

† radius 0.003125; all other tightened arms 0.00625.

Sensitivity and other label-free triggers (Section 7.2). On the 89 replayed runs, the conjunction with $W{=}10$ and $B{=}25$ gives early/late/missed and no-event counts of 25/1/0 and 19, 18, 17 for $P{=}1, 3, 5$ at $M{=}1.5$; 25/1/0 and 11, 10, 10 at $M{=}2$; 23/3/0 and 10, 9, 9 at $M{=}2.5$; and 22/4/0 and 9, 9, 8 at $M{=}3$. The window, baseline range and first eligible step were not varied, because they were chosen on these runs. On the same runs a response-length surge ($1.5\times$) catches 22 declines early, is late or misses on 4 and raises 13 no-event alarms (4 on single-pass runs), and a training-reward drop raises 51 no-event alarms (23 on single-pass runs). The three alarms that a KL-only rule adds to CDG's ten fall on single-pass runs ($K{=}8$, 4 and 1).

Replay of simpler triggers (Section 7.2). Table 16 compares CDG with its single signals and their disjunction on the 89 archived runs.

Table 16: Triggers replayed on the 89 archived runs with TROLL telemetry (26 with an event, 63 without). "No-event" alarms are false alarms under the frozen event definition.

trigger	early	late	missed	no-event alarms	quiet
CDG (frozen: AND, $M{=}2$, $P{=}3$)	25	1	0	**10**	53
KL signal only	25	1	0	13	50
entropy signal only	26	0	0	39	24
either signal (OR)	26	0	0	42	21

M PLANNED FOLLOW-UP EXPERIMENTS

This appendix records follow-up designs and their pass and kill criteria. It was not frozen by hash. A variant of E2 has since been run at $\epsilon{=}0.003125$ on seeds 701–706, with the projection gradient detached and $\alpha{=}0$, without the 0.0125 cell or the single-pass controls. Against the criteria listed below, this cell meets the limits on missed alarms, returned-model accuracy and regret, but exceeds the limit on no-event alarms: two alarmed runs end within 2 points of their periodic peak. The experiment recorded alarm-time evaluations and continued training without saving model weights. E1 was not run as specified; continuations from the seed 601–603 alarm checkpoints are reported in Section 6. The remaining designs below retain their original specifications.

E3: second pass vs updates per answer. One pass with batch 32, minibatch 2 and one epoch (16 fresh updates per rollout batch, 1/16 update per generated answer), $\epsilon{=}0.05$, full projection path, 3 new seeds, 150 steps. *Pass:* HEALTHY in at least 2/3 seeds and no CDG alarm on those seeds; the text keeps "second pass". A single-seed development run of this cell (seed 1) is reported in Section 6.2; it does not replace the three new seeds. *Kill:* an event in at least 2/3 seeds; Section 6.2 is restated in terms of updates per generated answer. Any CDG miss on this cell is reported as a miss and narrows Section 7.2.

E2: seed-varying prospective test. Two passes at $K{=}16$, 3 new seeds at $\epsilon{=}0.0125$ and 3 at $\epsilon{=}0.003125$ (first stage), extended to 6 seeds per cell plus 3 single-pass $K{=}8$ false-alarm controls

(second stage), which also log `lm_head` gradient norms and per-minibatch KL for E4. *Pass* requires all of: at most one late or missed alarm among runs with an event; at most one alarm on runs without an event that end within 2 points of their peak; mean CDG accuracy at least the mean run-to-end accuracy minus 0.5; mean regret to the periodic oracle at most 1.5 points on alarmed runs. The configuration-rule column is reported without a criterion. *If it fails*, the abstract's prospective-value sentence reports the failure and Section 7.2 is narrowed to configuration-determined regimes.

E4: trigger baselines (no new training). On the E2 logs and the 27 local trajectories: gradient-increment z-score gating, `target_kl`-style stops at standard levels, absolute KL levels fixed before E2, entropy only, response-length surge, training-reward drop and a fixed switch to a single pass at step 50. *Kill for the diagnosis framing:* a per-step statistic at a level fixed before E2 separates event from no-event seeds with a margin of at least $1.25\times$ (CDG's smallest KL margin on tight runs) and leads as early as CDG.

E1: continuation from alarm checkpoints. Restore test first. Then resume the prospective alarm checkpoints of seeds 3, 4 and 5 (steps 70, 73, 97) with one pass per rollout batch at $\epsilon{=}0.05$ to step 150. After a 10-step grace period the rule is re-armed and re-baselined on the continuation's own first 10 steps (same $2\times$/10/3 rule). Online runs that switch to one pass at the alarm without a restart (seeds 1 and 601–603) are reported in Section 7.2. *Pass:* mean GSM8K at step 150 at least 83.26 (the alarm-checkpoint mean minus 1) and no seed ending 5 or more points below its alarm checkpoint.

E5, E6: transfer. E5 applies the frozen rule, with an approximate KL in place of the projection's pre-projection KL, to two-pass $K{=}16$ training with clipping at 0.2 and without clipping. E6 repeats E2's first stage on a second training task or a larger model. Both are reported descriptively unless their criteria are frozen in a later amendment of this appendix before launch.

N Additional intervention and detection experiments

The common-parent experiments use the fixed reuse-fork protocol; each restored sibling shares the parent model, optimizer and data cursor and restarts the generation engine. The three full-strategy blocks instead start from the pretrained model independently. These cohorts are not pooled as repeated measurements of a single policy effect. The unchanged detector is evaluated separately on the six-seed detached, zero-regression stress configuration. All three common-step-50 blocks are complete.

Table 17: New common-step-50 continuations. GSM8K / MATH-500 at step 150 (%). These parents are newly trained prefixes, not the historical alarm parents.

seed	continuation	alarm	GSM8K	MATH-500
601	continue two passes	60	43.594	22.2
601	one pass after 50	–	87.111	62.2
601	alarm then one pass	62	83.397	60.0
602	continue two passes	93	52.995	38.8
602	one pass after 50	–	85.747	63.4
602	alarm then one pass	84	85.671	55.6
603	continue two passes	67	61.789	56.0
603	one pass after 50	–	86.050	62.2
603	alarm then one pass	62	84.989	60.2

The historical alarm-state paired effects on MATH-500 are $+9.2/+14.4/+19.8$ points for one versus two passes (seeds 601–603). The whole-strategy comparison requires C to lead both U and F by at least one GSM8K point without trailing either by more than three MATH-500 points; no seed satisfies this joint development criterion. This is a prespecified useful-difference check, not a test of statistical significance. Across the six stress-test runs, the fixed step-50 score has descriptive ROC-AUC 0.25 on two events and four non-events; the small sample does not justify reversing the score. Seed 704 has a MATH-500 decline without the frozen GSM8K event, and remains a no-event alarm under the primary definition.

Table 18: Unchanged CDG on six new data-order seeds: detached projection, $\alpha = 0$, $\epsilon = 0.003125$, two passes. Events are the frozen GSM8K five-point periodic-peak declines; alarms alone do not terminate these runs.

seed	GSM150	alarm	event	KL-only alarm	outcome
701	83.321	135	–	135	no-event alarm
702	77.938	–	150	–	miss
703	75.967	100	125	100	early detection
704	82.638	97	–	97	no-event alarm
705	84.989	94	–	94	no-event alarm
706	85.292	–	–	–	correct silence

Table 19: Same-alarm-state continuations. GSM8K accuracy (%) at step 150. Both branches restore the same model, optimizer and data state; the one-pass branch uses fewer remaining updates.

seed	parent step	parent	two passes	one pass	difference
601	92	82.714	42.608	85.823	+43.215
602	80	82.714	58.150	86.353	+28.203
603	77	80.895	37.073	84.155	+47.081

Table 20: Development controls (seed 1) and complete U/F/C strategy blocks (seeds 601–603). U uses one pass throughout, F switches after step 50, and C switches after the alarm. Each run generates 38,400 answers; updates and GPU time differ.

seed	organization	passes	updates	alarm	GSM at alarm	GSM150	MATH150
1	minibatch 4	2 throughout	2,400	105	82.94	59.44	54.4
1	minibatch 4	switch	1,760	70	84.46	86.88	59.6
1	minibatch 2	1 throughout	2,400	–	–	85.06	59.2
1	minibatch 4, $2\times$ lr	1 throughout	1,200	–	–	80.89	61.6
601	minibatch 4	1 throughout	1,200	–	–	86.35	61.8
601	minibatch 4	2 until step 50	1,600	–	–	87.72	59.8
601	minibatch 4	switch	1,936	92	82.71	85.29	63.2
602	minibatch 4	1 throughout	1,200	–	–	83.55	62.8
602	minibatch 4	2 until step 50	1,600	–	–	83.32	63.2
602	minibatch 4	switch	1,840	80	82.71	85.97	59.4
603	minibatch 4	1 throughout	1,200	–	–	87.11	59.0
603	minibatch 4	2 until step 50	1,600	–	–	86.43	59.8
603	minibatch 4	switch	1,816	77	80.89	82.79	59.2