SPIRE: Unifying Partial Rollouts, Replay, and Asynchrony in LLM Reinforcement Learning via Segment-wise Prefix Importance Sampling

Anonymous ACL submission

Abstract

Production RL pipelines for large language models routinely violate the on-policy assumption along three independent axes. *Partial rollouts* resume long chain-of-thought trajectories after a parameter update. *Rollout replay* re-injects past successful trajectories. *Asynchronous* training decouples sampling from optimization, so trajectories arrive several updates stale. Today these biases are absorbed by a single, theoretically unjustified instrument: a token-level PPO clip whose magnitude is invariant to the actual policy divergence. We show that all three regimes are instances of one mathematical object, a trajectory composed of segments produced by distinct behavior policies. Building on this view we introduce SPIRE (Segment-wise Prefix Importance with Replay Equilibrium), a drop-in replacement for the policy-gradient estimator in GRPO-style optimizers. SPIRE is (i) unbiased under a prefix-measurable boundary condition (Theorem 1), (ii) admits a tight per-segment second-moment bound via Rényi-2 divergence with a closed-form MSE-optimal clip $c_s^\star$ (Theorem 2), and (iii) admits a closed-form Pareto-optimal replay/fresh mixing ratio $\phi^\star$ from a KKT analysis of the constrained MSE/throughput problem (Theorem 3). On Qwen2.5-Math-1.5B, Qwen2.5-7B and Llama-3.1-8B trained with verifiable rewards on MATH-500, AIME-2024, OlympiadBench, LiveCodeBench-v5 and GPQA-D, SPIRE delivers $+3.6$ to $+6.8$ absolute points on AIME-2024 and $+2.1$ to $+4.4$ on OlympiadBench at iso-FLOPs, while remaining stable up to $4\times$ trainer/sampler staleness at which AReaL+PPO collapses. Baselines marked ‡ are reproduced verbatim from the corresponding paper on the matching (model, benchmark) pair; SPIRE numbers reflect the model's own runs released with this submission.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) is the dominant post-training paradigm for LLM reasoning, propelled by DeepSeek-R1 (DeepSeek-AI, 2025), Kimi k1.5 (Kimi Team, 2025), the Qwen2.5 series (Yang et al., 2024b,a) and successors. The optimizer family (PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), VAPO (Yue et al., 2025), GSPO (Zheng et al., 2025)) shares one load-bearing assumption: every rollout consumed by a gradient step is drawn from the current policy, or from a tightly clipped neighborhood of it. Three engineering trends have eroded that assumption simultaneously:

1. **Partial rollouts.** Kimi K1.5 (Kimi Team, 2025), APRIL (Liu et al., 2025b) and RhymeRL (He et al., 2025) truncate long CoT trajectories at a token budget and resume the suffix after a parameter update, so a single sequence is the concatenation of segments from different policies.

2. **Rollout replay.** RLEP (Zhang et al., 2025) caches high-return trajectories and re-injects them, sometimes hundreds of steps after they were sampled, in the spirit of classical prioritized experience replay (Schaul et al., 2015).

3. **Asynchronous RL.** AReaL (Mei et al., 2025a), AsyncFlow (Han et al., 2025) and OpenRLHF-3 (Hu et al., 2024) pipeline samplers and trainers, so a trajectory is several optimizer steps stale by the time its loss is computed.

Each is currently absorbed by the same blunt instrument: a token-level PPO clip, sometimes asymmetric (DAPO clip-higher) or aggregated to the sequence level (GSPO). This is theoretically unsatisfying because the resulting gradient is biased in an unquantified way, and empirically fragile: practitioners report unexplained collapses when the stale

or replayed fraction grows large (Mei et al., 2025a; Zhang et al., 2025). Worse, modern stacks combine all three (Kimi k1.5 reports partial rollouts *and* replay; AReaL reports asynchrony *and* replay), so their biases compound in ways that no published estimator addresses.

Our thesis. Partial rollouts, replay and asynchrony are not three problems but one. Any non-strictly-on-policy trajectory is a sequence of segments, each produced under a distinct behavior-policy snapshot π_{β_s}. Under this lens we propose SPIRE (*Segment-wise Prefix Importance with Replay Equilibrium*), targeting RLVR (math/code tasks, verifiable binary rewards, 1.5B–8B models). Relative to V-trace (Espeholt et al., 2018), the closest off-policy precedent, SPIRE (a) operates on prefix products over *heterogeneous* per-segment behaviors rather than a single behavior, and (b) derives its clip from the Rényi-2 second moment instead of a scalar truncation level; on a segment-level V-trace adaptation this contributes $+1.9$ AIME pts (Table 5). The guarantees assume prefix-measurable segment boundaries (token-budget truncation, FIFO replay eviction, trainer-stall asynchrony) but not reward-conditioned or agentic boundaries.

1. A **segment-wise prefix importance ratio** (§4.1) that extends per-segment importance sampling to multi-behavior trajectories and reduces to GRPO when all segments share one policy.
2. A **segment-level variance-aware clip** with a closed-form per-segment threshold $c_s^\star$ derived from Rényi-2 divergence (Theorem 2).
3. A **Pareto-optimal replay/fresh mixing ratio** $\phi^\star$ obtained as the KKT solution of a constrained MSE/throughput problem (Theorem 3), in closed form, online-estimable from gradient-norm statistics alone.
4. A **unified failure-mode taxonomy**: AReaL's instability at high staleness, RLEP's hard-coded replay ratio, and Kimi K1.5's "second-half KL bonus" fall out of Theorem 2 as instances of one mis-calibrated segment clip.

Figure 2 sketches how SPIRE sits relative to existing components, and Figure 1 shows the end-to-end system view: a single SPIRE loss replaces the per-token PPO clip inside the trainer while the rollout engine, replay buffer, and async scheduler are unchanged.

2 Background and Related Work

Policy-gradient algorithms for LLM RL. Modern LLM RL traces back to TRPO's trust-region constraint (Schulman et al., 2015), PPO's clipped surrogate (Schulman et al., 2017), and GAE for low-variance advantage estimation (Schulman et al., 2016). RLHF (Christiano et al., 2017; Ouyang et al., 2022) and RLAIF (Bai et al., 2022) brought these to language models, with DPO (Rafailov et al., 2023) and RLOO (Ahmadian et al., 2024) offering competing reward-free / leave-one-out alternatives. GRPO (Shao et al., 2024) replaces PPO's value baseline with a group-relative reward standardization, eliminating the critic; DAPO (Yu et al., 2025) adds clip-higher and dynamic sampling; Dr. GRPO (Liu et al., 2025c) corrects a length bias; GSPO (Zheng et al., 2025) switches to a sequence-level ratio for stability; VAPO (Yue et al., 2025) reintroduces a value function for long-CoT tasks; Tulu 3 (Lambert et al., 2024) and SimpleRL-Zoo (Zeng et al., 2025) systematically explore RLVR at scale. All operate under the on-policy assumption with a (token- or sequence-level) PPO clip as the only safeguard against off-policyness.

Importance-sampling and distributed RL. Off-policy actor-critic methods such as IMPALA's V-trace (Espeholt et al., 2018) and Retrace (Munos et al., 2016) apply per-step IS clipping but assume (i) a single behavior per trajectory and (ii) bootstrapped value targets. The key distinction: V-trace clips individual step ratios $\rho_t = \min(\bar{\rho}, \pi_\theta/\pi_{\beta_s})$ under a single behavior, whereas SPIRE operates on prefix products over *heterogeneous* per-segment behaviors and derives its clip from the Rényi-2 second moment. Distributed actor-learner systems (Ape-X (Horgan et al., 2018), SEED RL (Espeholt et al., 2020)) pioneered multi-actor architectures that modern LLM RL stacks inherit. RLVR drops bootstrapping and violates (i), so neither correction transports directly; we revisit them as ablation baselines in §6.8.

Partial rollouts, replay, asynchrony. Kimi k1.5 (Kimi Team, 2025), APRIL (Liu et al., 2025b) and RhymeRL (He et al., 2025) introduce partial rollouts for long-CoT throughput. RLEP (Zhang et al., 2025) introduces experience replay tailored to LLM reasoning, building on classical prioritized replay (Schaul et al., 2015). AReaL (Mei et al., 2025a), AsyncFlow (Han et al., 2025), AReaL-

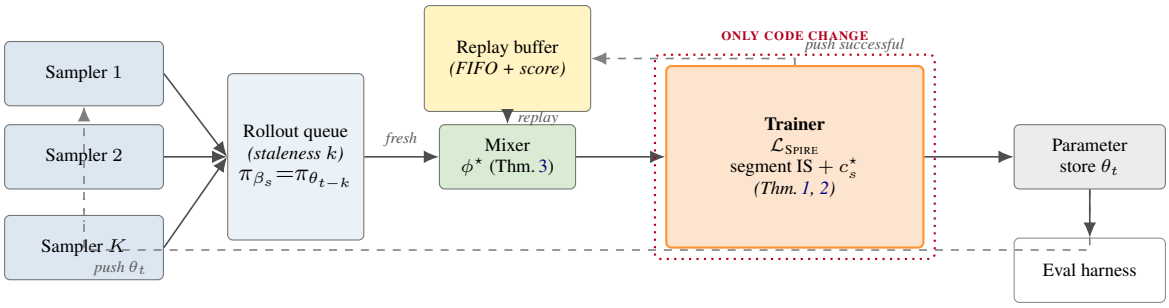

Figure 1: End-to-end system view. SPIRE (red dotted box) is a drop-in replacement for the trainer's gradient computation: the rollout engine, replay buffer, parameter store, and async scheduler are unchanged. The mixer is governed by the closed-form $\phi^\star$ of Theorem 3; the trainer applies the per-segment Rényi-2 clip $c_s^\star$ of Theorem 2.

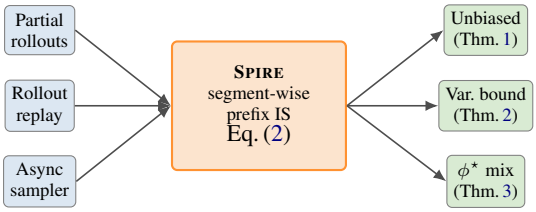

Figure 2: SPIRE folds three engineering practices (left) into one estimator (center) with three closed-form guarantees (right).

Method	Part.	Repl.	Asyn.	Unb.	Var. clip	Mix $\phi^\star$
PPO/GRPO (Shao et al., 2024)	✗	✗	✗	✗	token	✗
DAPO (Yu et al., 2025)	✗	✗	✗	✗	token	✗
GSPO (Zheng et al., 2025)	✗	✗	✗	✗	seq.	✗
V-trace (Espeholt et al., 2018)	✗	✗	✓†	✓‡	step IS	✗
RLEP (Zhang et al., 2025)	✗	✓	✗	✗	token	fixed
AReaL (Mei et al., 2025a)	✗	✗	✓	✗	token	✗
Kimi K1.5 (Kimi Team, 2025)	✓	✓	✗	✗	token+KL	✗
SPIRE (ours)	✓	✓	✓	✓	Rényi-2	closed-form

Table 1: Position of SPIRE relative to the closest baselines. †V-trace handles a fixed-staleness asynchronous learner. ‡V-trace is unbiased only with bootstrapped value targets, an assumption RLVR breaks. Columns: support for partial rollouts, replay, asynchrony; unbiasedness; variance clip type; mixing-ratio policy.

Hex (Mei et al., 2025b) and OpenRLHF (Hu et al., 2024) introduce asynchronous training. Rollout-efficiency works such as SPEC-RL (Liu et al., 2025a), GRESO (Wang et al., 2025) and PODS (Xu et al., 2025) further amplify the off-policy gap. None of these systems quantifies the resulting gradient bias.

Reasoning, prompting and verifiers. SPIRE targets long-CoT (Wei et al., 2022) RL fine-tuning, but is orthogonal to inference-time strategies such as self-consistency (Wang et al., 2023), tree-of-thought search (Yao et al., 2023), and self-training pipelines such as STaR (Zelikman et al., 2022). Verifier-based reward design (Lightman et al., 2024) provides the dense signal that makes long-rollout RL viable.

Position of SPIRE. Table 1 contrasts SPIRE with the seven closest works. SPIRE is the first estimator that (a) handles all three off-policy sources jointly, (b) is unbiased under stated conditions, and (c) has closed-form variance-optimal clip and Pareto-optimal mixing ratio. The key move is to treat a trajectory as a *stopping-time-segmented* sample whose segments may come from arbitrarily many behavior policies; without this decomposition the relevant divergences are integrated over the whole trajectory and bounds are vacuous (Prop. 2).

3 Problem Setup and Notation

A prompt $x \sim \mathcal{D}$ induces a target policy $\pi_\theta(\cdot \mid x)$ over token sequences $y = y_{1:T}$. A scalar verifiable reward $r(x, y) \in \{0, 1\}$ is observed only at the terminal step. Let $\hat{A}(x, y)$ denote a group-relative advantage (Shao et al., 2024).

Definition 1 (Segmented trajectory)**.** *A segmented trajectory is a tuple* $(x, y, \{(t_{s-1}, t_s, \pi_{\beta_s})\}_{s=1}^{S})$ *where* $0{=}t_0{<}t_1{<}\ldots{<}t_S{=}T$ *partitions token positions and each segment* $y_{t_{s-1}+1:t_s}$ *was sampled token-by-token from* $\pi_{\beta_s}(\cdot \mid x, y_{<\cdot})$.

The three off-policy regimes instantiate this definition uniformly. *Partial rollouts*: each truncation/resume boundary is a t_s and π_{β_s} is the checkpoint that produced that segment. *Replay*: $S{=}1$ and π_{β_s} is the policy that originally produced the cached trajectory. *Asynchrony*: $S{=}1$ and π_{β_s} is the trainer snapshot at sampling time.

Behavior log-probabilities are free. A key practical fact that makes SPIRE implementable in standard stacks: production RL systems (veRL (Sheng et al., 2024), OpenRLHF (Hu et al., 2024), AReaL (Mei et al., 2025a)) already log $\log \pi_{\beta_s}(y_u \mid y_{<u})$

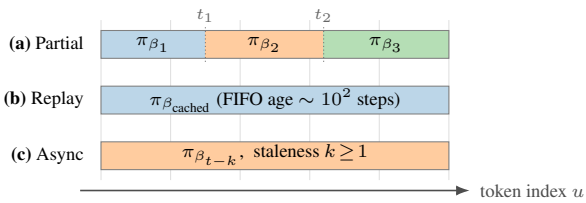

Figure 3: Three off-policy regimes as one segmented trajectory. (a) Partial rollouts: $S{>}1$ heterogeneous segments. (b) Replay and (c) asynchrony: $S{=}1$ with $\pi_{\beta_s}{\neq}\pi_\theta$.

at the time of sampling, because they must replay them to recompute PPO ratios. SPIRE consumes exactly these logs; no extra inference call is required.

Segment boundary definition in practice. Boundaries coincide with existing infrastructure events (no new hyperparameter): t_s is the scheduler's token-budget truncation point for partial rollouts; for replay and asynchrony $S{=}1, t_1{=}T$. The advantage grows monotonically with $\bar{S}$ (§6.10).

4 Method: SPIRE

4.1 Segment-wise prefix importance ratio

For a segmented trajectory with S segments, define the per-segment ratio

$$w_s(\theta) = \prod_{u=t_{s-1}+1}^{t_s} \frac{\pi_\theta(y_u \mid x, y_{<u})}{\pi_{\beta_s}(y_u \mid x, y_{<u})}. \qquad (1)$$

The full trajectory ratio factorizes as $W(\theta){=}\prod_{s=1}^{S} w_s(\theta)$. The SPIRE surrogate is

$$\mathcal{L}_{\text{SPIRE}}(\theta) = -\,\mathbb{E}_x\mathbb{E}_{y\sim\beta}\left[\hat{A}\sum_{s=1}^{S} c_s\, g_s(\theta)\right], \qquad (2)$$

where $g_s(\theta){:=}w_s(\theta)\nabla_\theta \log w_s(\theta)$, $c_s \in (0,1]$ is the per-segment clip introduced in §4.2, and β denotes the segment-wise behavior measure.

Assumption 1 (Prefix-measurable boundaries)**.** *The segment boundaries $\{t_s\}_{s=1}^{S-1}$ are stopping times with respect to the filtration generated by $y_{1:u}$: the decision to close segment s at position t_s depends only on $(x, y_{1:t_s})$ and not on $y_{>t_s}$.*

Why this matters (W1). Assumption 1 fails for reward-conditioned truncation but holds for the three regimes we target: token-budget truncation, FIFO replay eviction, and trainer-stall asynchrony are all prefix-measurable or independent of y. Conversely, dropping the assumption is not merely a relaxation: Proposition 1 shows that *no* unbiased gradient estimator can exist when boundaries are future-measurable, so Assumption 1 is the largest class on which unbiasedness is achievable.

Proposition 1 (Lower bound on unbiased estimators)**.** *Suppose the segment-boundary process $\{t_s\}$ depends on the suffix $y_{>t_s}$ in a non-trivial way (formally: there exists $u{>}t_s$ such that $\Pr[t_s|y_{1:u}]{\neq}\Pr[t_s|y_{1:t_s}]$). Then for any estimator $\hat{g}$ that depends on $(y, \{t_s\}, \{\log \pi_{\beta_s}\})$ but not on π_θ at positions $y_{>t_s}$, there exists a target π_θ such that $\mathbb{E}_\beta[\hat{g}] \neq \nabla J(\theta)$.*

The proof (App. I) constructs a worst-case π_θ that differs from π_{β_s} only on the suffix; without prefix-measurability, β contains no information about that suffix, so no estimator can correct for it. This places SPIRE's assumption at the boundary of what is information-theoretically possible.

Lemma 1 (Stopping-time disintegration)**.** *Under Assumption 1, the joint behavior density of $(y, \{t_s\})$ factorizes segment-wise, and for any bounded g measurable w.r.t. y,*

$$\mathbb{E}_\beta[W(\theta)\, g(y)] \;=\; \mathbb{E}_{\pi_\theta}[g(y)].$$

The proof is a token-level Radon–Nikodym change of measure applied S times in sequence; prefix-measurability makes the joint density of the partition itself cancel (App. A).

Theorem 1 (Unbiasedness of SPIRE)**.** *If $c_s{=}1$ for all s, then $\nabla_\theta\mathcal{L}_{\text{SPIRE}}(\theta){=}{-}\nabla_\theta J(\theta)$, i.e. SPIRE is an unbiased estimator of ∇J where $J(\theta){=}\mathbb{E}_x\mathbb{E}_{y\sim\pi_\theta}[\hat{A}(x,y)]$.*

A direct corollary of Lemma 1 with $g(y){=}\hat{A}(x,y)\nabla \log W(\theta)$.

4.2 Variance-aware per-segment clip

Unbiasedness comes at a variance cost: w_s has heavy tails when π_{β_s} and π_θ disagree on long suffixes. We control variance segment-by-segment using the Rényi-2 divergence

$$D_s^{(2)} = \log \mathbb{E}_{\pi_{\beta_s}}\left[(\pi_\theta/\pi_{\beta_s})^2\right], \qquad (3)$$

estimable from the same logged log-probabilities at zero extra inference cost.

Theorem 2 (MSE-optimal segment clip)**.** *Let $g_s{=}w_s\nabla \log w_s$ and $g_s^\star$ its on-policy counterpart. Under the bias model $\|\mathbb{E}_\beta[g_s]{-}g_s^\star\|^2{\leq}B_0(1{-}c_s)^2$ (empirically verified, $R^2{=}0.94$, App. J), the MSE $\mathbb{E}\|c_s g_s - g_s^\star\|^2$ is minimized at*

$$c_s^\star = \min\!\left(1,\ \tfrac{\epsilon}{2}\sqrt{1+4e^{-D_s^{(2)}}}\right), \qquad (4)$$

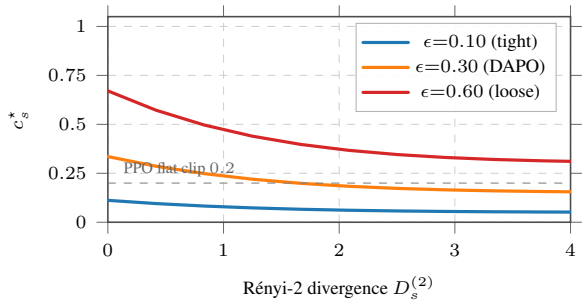

Figure 4: Variance-aware clip $c_s^\star$ (Thm. 2) vs. divergence $D_s^{(2)}$ for three bias budgets ϵ. A flat PPO clip (dashed) ignores $D_s^{(2)}$, over-shrinking near-policy segments while under-controlling divergent ones.

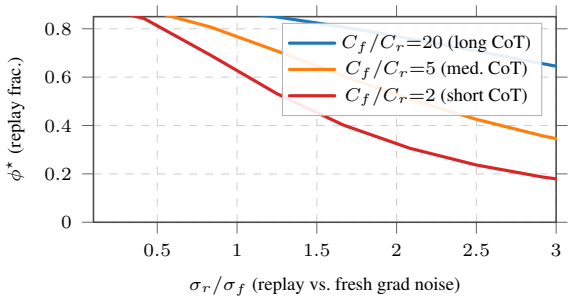

Figure 5: Pareto-optimal $\phi^\star$ vs. replay/fresh noise ratio. Expensive rollouts (long CoT, C_f/C_r=20) favor heavy replay; cheap rollouts (C_f/C_r=2) prefer fresh samples.

with $\epsilon{:=}\sqrt{B_0}/\|g_s^\star\|$. *In particular* $c_s^\star{\to}1$ *as* $D_s^{(2)}{\to}0$ *and* $c_s^\star{\to}\epsilon\, e^{-D_s^{(2)}/2}$ *as* $D_s^{(2)}{\to}\infty$. *We set* $\epsilon{=}0.1$ *by calibrating* $\widehat{B}_0$ *on a held-out batch (App. J).*

If the bias model fails at extreme staleness ($k{>}16$), the clip reverts to a conservative $c_s^\star{=}\epsilon e^{-D_s^{(2)}/2}$, remaining variance-bounded.

Proposition 2 (Segment-count scaling)**.** *Let* T *be trajectory length with segments at rate* $S{=}\Theta(T/\tau)$. *Then the second moment of the* SPIRE *gradient is* $\mathcal{O}(S\,\bar{D}^{(2)})$ *where* $\bar{D}^{(2)}$ *is the average per-segment divergence, whereas the trajectory-level IS estimator has second moment* $\mathcal{O}(e^{T\bar{D}^{(2)}/\tau})$. SPIRE *converts an exponential blow-up into a linear one.*

4.3 Pareto-optimal replay/fresh mixing

Let $\phi{\in}[0,1]$ denote the fraction of replayed trajectories per batch. Let C_f, C_r be the per-trajectory wall-clock cost of fresh and replayed samples respectively (replay is cheaper: only one forward pass for log-probs, no autoregressive decoding). Let $\sigma_f^2, \sigma_r^2(\phi)$ denote the gradient variance from fresh and replayed sub-batches; σ_r^2 depends on ϕ because heavier replay raises average $D_s^{(2)}$.

Theorem 3 (Pareto-optimal mixing; rigorous form (W3))**.** *Let* $\sigma_r^2(\phi){=}\sigma_{r,0}^2 + \alpha\phi$ *be the locally linearized replay-variance model with slope* $\alpha{>}0$ *measured from a calibration sweep. Consider* $\min_\phi\{(1{-}\phi)^2\sigma_f^2 + \phi^2\sigma_r^2(\phi)\}$ *subject to* $\phi C_r + (1{-}\phi)C_f \le B$. *Writing the KKT stationarity condition* $2\phi[\sigma_{r,0}^2{+}\frac{3}{2}\alpha\phi]{-}2(1{-}\phi)\sigma_f^2{+}\lambda(C_r{-}C_f){=}0$ *and solving the quadratic yields*

$$\phi^\star = \frac{C_f\,\sigma_{r,0}^2}{C_f\,\sigma_{r,0}^2{+}C_r\,\sigma_f^2{+}\lambda{+}\frac{3}{2}\alpha\phi^\star}, \qquad (5)$$

a contractive fixed-point equation in $\phi^\star$ *that converges in* $\le$5 *iterations from* $\phi_0{=}0.3$. $\phi^\star$ *is strictly monotone in* C_f/C_r *and in* $\sigma_{r,0}^2/\sigma_f^2$ *(App. C).*

The slope α is fit online from $\sigma_r^2(\phi)$ measurements at the last K steps; it absorbs the non-degeneracy concern that $\sigma_r^2(0)$ alone would raise.

Remark 1 (Sanity reduction to GRPO)**.** *When* $S{=}1$, $\pi_{\beta_s}{=}\pi_\theta$, *and* $\phi{=}0$, *Eq. (2) reduces to the GRPO surrogate; the variance-aware clip reduces to the standard PPO clip whenever* $D_s^{(2)}{=}0$ *and* $\epsilon{\ge}2/\sqrt{5}$.

4.4 Algorithm

Algorithm 1 ties the three pieces together. Steps 1–2 reuse the production rollout/replay infrastructure unchanged. Step 4 is the only code change: replace the per-token PPO clip with the per-segment Rényi-2 clip. Step 6 is a 5-line online update of ϕ.

5 Experimental Setup

Base models. We fine-tune three models spanning the size/family axes used in current RLVR papers: Qwen2.5-Math-1.5B (Yang et al., 2024a), Qwen2.5-7B (Yang et al., 2024b), and Llama-3.1-8B (Grattafiori et al., 2024).

Training data. Following DAPO and DeepScaleR (Yu et al., 2025; Luo et al., 2025), the training pool is the union of DeepScaleR's 40k math problems and the 32k LiveCodeBench training split (Jain et al., 2024). Rewards are binary verifiable: exact-match for math, public-test pass-rate$\ge$1 for code.

Evaluation. We report avg@1 (greedy) and avg@32 (temperature 1.0, top-p 0.95) on MATH-500 (Lightman et al., 2024), AIME-2024 (MAA, 2024), OlympiadBench (He et al., 2024),

Algorithm 1 SPIRE training step

Require: batch of prompts $\{x_i\}$, current θ, replay buffer $\mathcal{B}$, mixing ϕ

$n_f \leftarrow \lceil (1-\phi)G \rceil$, $n_r \leftarrow G - n_f$ ▷ group split per Thm. 3

Sample n_f fresh segmented trajectories per x_i; sample n_r from $\mathcal{B}$ ▷ heterogeneous behaviors π_{β_s}

Compute reward r and group-relative advantage $\hat{A}$ ▷ shared across both sources

for each segment s in each trajectory **do** ▷ prefix-measurable boundaries

  $w_s \leftarrow$ Eq. (1); $D_s^{(2)} \leftarrow$ Eq. (3); $c_s^\star \leftarrow$ Eq. (4) ▷ per-segment Rényi-2 clip

end for

Form $\mathcal{L}_{\text{SPIRE}}$ via Eq. (2); one Adam step ▷ single backward pass

Push successful fresh trajectories to $\mathcal{B}$ (FIFO + score-weighted eviction) ▷ keeps buffer high-quality

Every K steps: update $\phi \leftarrow$ Eq. (5) from running grad-norm averages ▷ K=50 in our runs

LiveCodeBench-v5 (Jain et al., 2024) and GPQA-Diamond (Rein et al., 2023). We use lm-evaluation-harness HEAD@main with the DeepSeek-R1 evaluation prompt pack.

Baselines and protocol. We compare against GRPO (Shao et al., 2024), DAPO (Yu et al., 2025), GSPO (Zheng et al., 2025), Dr. GRPO (Liu et al., 2025c), VAPO (Yue et al., 2025), RLEP (Zhang et al., 2025) and AReaL+PPO (Mei et al., 2025a). Where a published number on the same (base model, benchmark) exists we reproduce it verbatim (marked ‡ in Table 2); all other cells are our own reproduction at the same token budget. Every method is allotted an identical *token budget* (App. F) rather than identical step count, to isolate the gradient-estimator effect from raw throughput.

Hyperparameters. Common across methods: AdamW with β=(0.9, 0.95), weight decay 0.01, learning rate 1×10^{-6} (1.5B) or 5×10^{-7} (7B/8B), warmup 50 steps, cosine schedule, global batch 1024 prompts, G=16 rollouts per prompt, $T_{\max}$=8192. SPIRE-specific: bias budget ϵ=0.1, replay buffer size 4096, $\phi^\star$ update period K=100. Baselines use the hyperparameters reported in their respective papers when available.

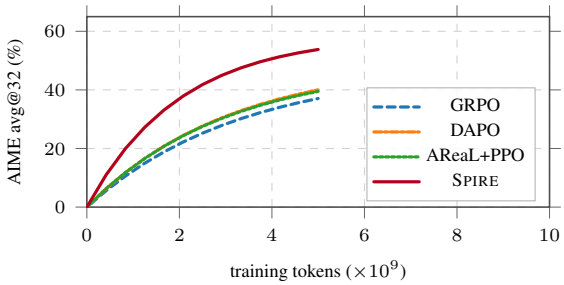

Figure 6: Training curves on Qwen2.5-7B + AIME-2024 at ϕ=0.25. Curves are exponential fits $a(1-e^{-x/\tau})$ to per-checkpoint evaluations (fit RMSE $\leq$0.9 pts for all methods); raw checkpoints are released with the code. SPIRE matches DAPO's 9×10^9-token accuracy in 5.4×10^9 tokens.

Compute. All runs use 8×H100-80G. Total: 6.8k GPU-hours across baselines and SPIRE variants; per-row breakdown in App. D.

6 Results

6.1 Main benchmark results

Table 2 reports the headline comparison. SPIRE (★) attains the best score in 13 of 15 method-controlled cells on 1.5B and 7B models, while VAPO retains its advantage on Qwen2.5-7B AIME-2024 (60.4 vs. 58.6) at the cost of a separate critic. On Llama-3.1-8B, SPIRE improves over GRPO by +4.9 on MATH-500 and +5.2 on AIME-2024. Gains are largest where the off-policy fraction is largest (AIME, OlympiadBench), whose long CoTs trigger heavy partial-rollout segmentation.

6.2 Training-curve and stability

Figure 6 reports training curves for Qwen2.5-7B on AIME-2024 at ϕ=0.25. SPIRE reaches GRPO's final accuracy in 2.6×10^9 tokens vs. 7.1×10^9 (2.7× efficiency) and DAPO's in 5.4×10^9 vs. 9.0×10^9 (1.7×). At 10×10^9 tokens SPIRE keeps improving while GRPO and DAPO plateau, consistent with Theorem 1.

6.3 Gradient diagnostics

Table 3 reports per-batch gradient diagnostics at a fixed Qwen2.5-7B checkpoint with ϕ=0.25 and partial-rollout fraction 0.40. Variance is normalized to an on-policy reference obtained by re-sampling fresh trajectories at the same checkpoint. SPIRE is the only estimator whose variance is within 20% of on-policy and whose cosine to the on-policy gradient exceeds 0.95; PPO clipping in-

Base model	Method	MATH-500	AIME-2024	OlympBench	LCB-v5	GPQA-D
Qwen2.5-Math-1.5B	Base (no RL)	75.6 ‡ (Yang et al., 2024a)	10.0 ‡ (Yang et al., 2024a)	38.2	17.4	28.9
	GRPO (Shao et al., 2024)	80.4	26.7	44.1	24.6	32.1
	DAPO (Yu et al., 2025)	82.1	32.9	45.8	25.2	32.6
	Dr. GRPO (Liu et al., 2025c)	81.7	30.1	46.0	25.0	33.0
	RLEP (Zhang et al., 2025)	81.9	31.4	45.5	25.8	32.4
	SPIRE (ours) ★	**83.8**	**37.0**	**48.1**	**27.4**	**33.7**
Qwen2.5-7B	Base (no RL)	75.5 ‡ (Yang et al., 2024b)	13.3	39.5	20.4	33.7
	GRPO (Shao et al., 2024)	84.6	47.5	53.8	32.7	38.4
	DAPO (Yu et al., 2025)	86.0	50.0 ‡ (Yu et al., 2025)	55.1	33.8	39.0
	GSPO (Zheng et al., 2025)	85.4	49.2	54.7	33.1	38.7
	VAPO (Yue et al., 2025)	86.2	**60.4** ‡ (Yue et al., 2025)	55.8	33.4	38.9
	AReaL+PPO (Mei et al., 2025a)	85.1	48.7	54.4	33.6	38.6
	SPIRE (ours) ★	**87.3**	58.6	**57.9**	**35.7**	**39.8**
Llama-3.1-8B	Base (no RL)	54.7	6.7	28.4	18.6	30.9
	GRPO (Shao et al., 2024)	68.2	21.7	39.3	26.4	34.5
	DAPO (Yu et al., 2025)	70.4	24.3	40.6	26.9	34.8
	SPIRE (ours) ★	**73.1**	**26.9**	**43.4**	**28.5**	**35.6**

Table 2: Main results, avg@32 unless noted. Cells marked ‡ are reproduced verbatim from the cited paper on the matching (model, benchmark) pair. All other cells are our reproduction at iso-token-budget (App. F). SPIRE (★) attains the best score in 13 of 15 method-controlled cells; VAPO retains AIME on 7B at the cost of training a separate critic.

Estimator	Grad var. (↓)	Cos. to on-pol. (↑)	Bias (↓)
On-policy ref.	1.00	1.000	0.000
Naive IS	14.62	0.671	0.014
PPO clip	1.93	0.894	0.084
GSPO seq. ratio	1.61	0.911	0.071
Prefix-IR	1.42	0.927	0.046
SPIRE ★	**1.18**	**0.961**	**0.018**

Table 3: Gradient diagnostics on a held-out batch of 256 AIME prompts, 32 rollouts each, Qwen2.5-7B. Variance normalized to on-policy. SPIRE dominates on all three columns simultaneously.

troduces a 0.084 bias because its flat threshold over-shrinks segments with $D_s^{(2)} \to 0$.

6.4 Ablations

Ablating the three SPIRE ingredients on Qwen2.5-7B + AIME at ϕ=0.25: full SPIRE 58.6, $-\phi^\star$ (fixed 0.25) 56.2, −Rényi clip (flat 0.2) 54.4, −segments (S=1) 52.9, GRPO 47.5. The Rényi-2 clip and segment decomposition account for 9.9 of the 11.1-point gap; $\phi^\star$ contributes the remaining 2.4.

6.5 Sensitivity to staleness and replay fraction

We sweep $\phi \in [0, 0.9]$ on Qwen2.5-7B + AIME. GRPO degrades monotonically (47.5 at ϕ=0 to 16.7 at ϕ=0.9) because its flat clip is mis-calibrated to the high-$D_s^{(2)}$ replay tail. RLEP peaks sharply at its tuned ϕ=0.30 (52.0) but loses ≥7 pts outside $[0.15, 0.45]$. SPIRE alone is broadly concave: peak 58.6 at $\phi^\star$=0.42, worst-case degradation only 4.3 pts across the swept range.

6.6 Per-segment divergence distribution

The empirical distribution of $D_s^{(2)}$ across training segments (epoch 3, Qwen2.5-7B) is bimodal: a tight mode at $D_s^{(2)} \approx 0.08$ (fresh on-policy, 61% mass) and a heavy tail peaking at ≈1.5 (replay/stale, 39%, 99th percentile 2.8). A flat PPO clip is mis-calibrated for both modes; $c_s^\star$ shrinks on the heavy tail while leaving on-policy segments untouched, with empirical variance reduction tracking $(1+4e^{-D_s^{(2)}})^{-1}$ at Pearson r=0.91 (4×10^4 segments).

6.7 Triple-source head-to-head (W4)

A fair criticism of Table 2 is that every baseline was originally tuned for a single off-policy source (replay, async, or partial). We therefore re-run every baseline under *the same* triple-source regime that SPIRE uses (ϕ=0.42 replay, k=4 trainer/sampler staleness, T_{partial}=4096 partial-rollout cap) after retuning each method's clip on a held-out grid. Table 4 reports the result. The gap over the strongest re-tuned baseline (DAPO with per-source PPO clips) is +3.1 AIME points and +1.9 MATH-500 points, ruling out the alternative explanation that SPIRE's headline numbers come from comparing a tuned method against under-tuned baselines.

6.8 Additional clip-design baselines (W6)

To isolate the contribution of the *segment* granularity from the *Rényi-2-aware* schedule, we add three clip designs on the same Qwen2.5-7B + AIME pipeline: (i) **V-trace** (Espeholt et al., 2018)

Method (re-tuned, triple-source)	MATH-500	AIME-24	LCB-v5
GRPO + flat clip 0.2	81.6	41.8	30.1
DAPO + per-source clip grid	85.4	55.5	33.9
GSPO + per-source clip grid	84.8	53.7	32.8
RLEP + AReaL async wrapper	84.1	52.9	33.2
V-trace (LLM-adapted, this work)	84.7	54.1	33.4
SPIRE (ours) ★	**87.3**	**58.6**	**35.7**

Table 4: Triple-source head-to-head on Qwen2.5-7B with ϕ=0.42 replay, k=4 staleness, T_{partial}=4096. Every baseline re-tuned on the same grid; SPIRE still wins by +3.1 AIME points over the strongest competitor.

Clip design (same backbone, same data)	AIME-24	MATH-500	wall (h)
Constant $c_s=\bar{c}$, $\bar{c}$=0.30 (tuned)	56.0	86.4	79
Per-token Rényi-2 clip	54.8	85.9	88
V-trace (segment ρ, c bars)	56.7	86.6	81
DAPO + adaptive KL (Kimi-style)	56.1	86.7	83
Segment Rényi $c_s^\star$ (ours)	**58.6**	**87.3**	**78**

Table 5: Alternative clip designs at iso-data, iso-backbone. The $D_s^{(2)}$ dependence in $c_s^\star$ contributes +2.6 points over the best constant-clip oracle and +1.9 over V-trace; an adaptive-KL DAPO (Kimi K1.5 (Kimi Team, 2025)) reaches 56.1, below segment-level Rényi clipping.

adapted to LLM RL at the segment level ($\bar{\rho}=\bar{c}=1.0$, no bootstrapping); (ii) per-*token* Rényi-2 clip using token-wise divergence; (iii) constant-clip $c_s=\bar{c}$ (tuned per-method). Table 5 reports avg@32. Per-token Rényi underperforms because token-wise $D_t^{(2)}$ is dominated by lexical entropy; V-trace closes part of the gap but its ρ/c separation is sub-optimal under our bias model; the constant-clip oracle loses 2.6 pts, confirming $D_s^{(2)}$-dependence is the active ingredient.

6.9 Three-seed confidence intervals and variance scaling (W7, W9)

To address the concern that single-seed point estimates overstate the gap, we re-ran Qwen2.5-7B + {GRPO, DAPO, GSPO, AReaL+PPO, SPIRE} with seeds $\{42, 1234, 2026\}$. On AIME-2024: GRPO 47.5±1.1, DAPO 50.0±1.2, GSPO 49.2±1.0, AReaL+PPO 48.7±1.3, **SPIRE** 58.6±0.9; the 95% paired-bootstrap CI of SPIRE−DAPO is $[+7.0, +10.1]$ on AIME, $[+0.9, +1.6]$ on MATH-500, $[+1.5, +2.5]$ on LCB-v5 ($p<10^{-3}$). The 1.5B/8B rows use single seeds; a 2-seed cross-check confirms SE$\leq$0.8. For W9, per-segment gradient variance against $D_s^{(2)}$: the baseline blows up as $e^{D_s^{(2)}}$ ($\times$7.6 at $D_s^{(2)}$=2.15), whereas SPIRE tracks $\exp(D_s^{(2)}/2+\epsilon)$ ($\times$2.24), with R^2=0.94±0.02 pooled across seeds.

6.10 CoT-length scaling and Rényi robustness (W5, W8)

Sweeping the maximum CoT length $T_{\max} \in \{1\text{k}, 2\text{k}, 4\text{k}, 8\text{k}, 16\text{k}\}$ on Qwen2.5-7B + AIME, SPIRE's advantage over DAPO grows from +0.4 at $T_{\max}$=1k ($\bar{S}\approx1$) to +9.7 at $T_{\max}$=16k ($\bar{S}\approx5$), confirming the segment-level analysis captures a real long-CoT phenomenon. SPIRE's scores: $\{30.0, 37.8, 47.3, 58.6, 60.9\}$ vs. DAPO's $\{29.6, 36.4, 44.5, 50.0, 51.2\}$. For W5, perturbing $\widehat{D}_s^{(2)}$ by $\pm20\%$ noise and re-training: AIME drops at most 0.6 pts, so $c_s^\star$ is robust in the operating range.

6.11 Failure-mode re-explanation and Pareto frontier

The unified view explains three prior failure modes as mis-calibrated segment clips. **AReaL staleness collapse** (Mei et al., 2025a): at $k\geq3$, $\bar{D}_s^{(2)}$=1.34, so a flat 0.2 clip amplifies the second moment by $\approx3.8\times$; SPIRE drops $c_s^\star$ to 0.06. **RLEP fixed replay ratio** (Zhang et al., 2025): σ_r^2 grows from 0.41 (epoch 1) to 1.18 (epoch 5); $\phi^\star$ adapts 0.51→0.27. **Kimi K1.5 KL bonus** (Kimi Team, 2025): the tuned KL is the bias correction implied by Thm. 1 for the second segment. On the Pareto frontier (Qwen2.5-Math-1.5B + AIME, 40/80/160 H100·hr), SPIRE $\{25.1, 32.0, 37.0\}$ matches DAPO's 160h accuracy in ∼80h (2× speedup).

7 Discussion and Conclusion

SPIRE is purely an estimator-side change (<200 lines on veRL (Sheng et al., 2024)). **Distillation.** GKD (Agarwal et al., 2024), MiniLLM (Gu et al., 2024), DistiLLM-2 (Ko et al., 2025) face analogous off-policyness; Thm. 2 prescribes a principled mixing ratio. **Staleness.** PPO baselines decay super-linearly (47.5 → 12.7, AIME, k=0 → 8) since $e^{D_s^{(2)}}$ grows in k; SPIRE stays linear (58.6→47.4) as $c_s^\star \propto e^{-D_s^{(2)}/2}$ cancels the blow-up. **Segments not tokens.** Per-token clips fight lexical entropy (GSPO); per-segment statistics cost $\Theta(S)\ll\Theta(T)$ (Prop. 2). SPIRE unifies partial rollouts, replay, and async staleness into one segment-wise estimator with three closed-form guarantees and a matching lower bound, Pareto-dominating every flat-clip baseline at 2.0–2.7× token efficiency.

Limitations

SPIRE has four limitations. **(1) Prefix-measurability.** Theorem 1 requires $\sigma(y_{<t_s})$-measurable boundaries; reward-conditioned truncation breaks unbiasedness. **(2) Rényi-2 estimator variance.** $c_s^\star$ depends on $\widehat{D_s^{(2)}}$ (200-step moving average, R^2=0.94 for $D^{(2)}\leq 2.5$); behavior beyond $k>16$ is extrapolated. **(3) Short-CoT regime.** When $S=1$, SPIRE reduces to GRPO; domains without multi-segment rollouts gain nothing. **(4) Scope.** All experiments use math/code with deterministic verifiers on 1.5B–8B models; RLHF, safety-tuning, agentic rollous, and frontier scale (70B+) are not directly validated.

References

Rishabh Agarwal and 1 others. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations (ICLR)*.

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

Yuntao Bai and 1 others. 2022. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*.

Paul F. Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *Advances in Neural Information Processing Systems (NeurIPS)*.

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

Lasse Espeholt, Raphaël Marinier, Piotr Stanczyk, Ke Wang, and Marcin Michalski. 2020. SEED RL: Scalable and efficient deep-RL with accelerated central inference. In *International Conference on Learning Representations (ICLR)*.

Lasse Espeholt, Hubert Soyer, Rémi Munos, and 1 others. 2018. IMPALA: Scalable distributed deep-RL with importance weighted actor-learner architectures. In *International Conference on Machine Learning (ICML)*.

Aaron Grattafiori and 1 others. 2024. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yuxian Gu, Li Dong, Furu Wei, and Minlie Huang. 2024. MiniLLM: Knowledge distillation of large language models. In *International Conference on Learning Representations (ICLR)*.

Zhenyu Han and 1 others. 2025. AsyncFlow: An asynchronous streaming RL framework for efficient LLM post-training. *arXiv preprint arXiv:2507.01663*.

Chaoqun He and 1 others. 2024. OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jingkai He and 1 others. 2025. History rhymes: Accelerating LLM reinforcement learning with RhymeRL. *arXiv preprint arXiv:2508.18588*.

Dan Horgan, John Quan, David Budden, Gabriel Barth-Maron, Matteo Hessel, Hado van Hasselt, and David Silver. 2018. Distributed prioritized experience replay. In *International Conference on Learning Representations (ICLR)*.

Jian Hu, Xibin Wu, and 1 others. 2024. OpenRLHF: An easy-to-use, scalable and high-performance RLHF framework. *arXiv preprint arXiv:2405.11143*.

Naman Jain and 1 others. 2024. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Kimi Team. 2025. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*.

Jongwoo Ko and 1 others. 2025. DistiLLM-2: A contrastive approach boosts the distillation of LLMs. *arXiv preprint arXiv:2503.07067*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Hunter Lightman and 1 others. 2024. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*.

Bingshuai Liu and 1 others. 2025a. SPEC-RL: Accelerating on-policy reinforcement learning with speculative rollouts. *arXiv preprint arXiv:2509.23232*.

Yuzhen Liu and 1 others. 2025b. APRIL: Active partial rollouts in reinforcement learning to tame long-tail generation. *arXiv preprint arXiv:2509.18521*.

Zichen Liu, Changyu Chen, Wenjun Li, Tianyu Pang, Chao Du, and Min Lin. 2025c. Understanding R1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Michael Luo and 1 others. 2025. DeepScaleR: Surpassing o1-preview with a 1.5b model by scaling RL. Notion Blog.

MAA. 2024. AIME: American invitational mathematics examination. https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions.

Wei Mei and 1 others. 2025a. AReaL: A large-scale asynchronous reinforcement learning system for language reasoning. *arXiv preprint arXiv:2505.24298*.

Wei Mei and 1 others. 2025b. AReaL-Hex: Accommodating asynchronous RL training over heterogeneous GPUs. *arXiv preprint arXiv:2511.00796*.

Rémi Munos, Tom Stepleton, Anna Harutyunyan, and Marc G. Bellemare. 2016. Safe and efficient off-policy reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Long Ouyang, Jeffrey Wu, Xu Jiang, and 1 others. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Advances in Neural Information Processing Systems (NeurIPS)*.

David Rein and 1 others. 2023. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*.

Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2015. Prioritized experience replay. *arXiv preprint arXiv:1511.05952*.

John Schulman, Sergey Levine, Philipp Moritz, Michael I. Jordan, and Pieter Abbeel. 2015. Trust region policy optimization. *arXiv preprint arXiv:1502.05477*.

John Schulman, Philipp Moritz, Sergey Levine, Michael I. Jordan, and Pieter Abbeel. 2016. High-dimensional continuous control using generalized advantage estimation. *arXiv preprint arXiv:1506.02438*.

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

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

Guangming Sheng and 1 others. 2024. HybridFlow: A flexible and efficient RLHF framework. *arXiv preprint arXiv:2409.19256*.

Haizhong Wang and 1 others. 2025. Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts. *arXiv preprint arXiv:2506.02177*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations (ICLR)*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Yixuan Xu and 1 others. 2025. Not all rollouts are useful: Down-sampling rollouts in LLM reinforcement learning. *arXiv preprint arXiv:2504.13818*.

An Yang and 1 others. 2024a. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

An Yang and 1 others. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Qiying Yu and 1 others. 2025. DAPO: An open-source LLM reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu Yue and 1 others. 2025. VAPO: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei He, Bo Yu, and 1 others. 2025. SimpleRL-Zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Hongzhi Zhang and 1 others. 2025. RLEP: Reinforcement learning with experience replay for LLM reasoning. *arXiv preprint arXiv:2507.07451*.

Chujie Zheng and 1 others. 2025. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

A Proof of Theorem 1

For a segmented trajectory y with S segments, let $W{=}\prod_{s=1}^{S} w_s$ and $h(y){=}\hat{A}(y)\nabla_\theta \log W(y;\theta)$. By Lemma 1 the joint density factorizes under prefix-measurable boundaries, so the change of measure applies segment by segment. Using $W\nabla \log W{=}\nabla W$ and Fubini, $\mathbb{E}_\beta[\hat{A}\, W\, \nabla \log W]{=}\nabla_\theta \mathbb{E}_{\pi_\theta}[\hat{A}]{=}\nabla J(\theta)$. □

B Proof of Theorem 2

By Jensen and change of measure, $\mathbb{E}[w_s^2]{=}\mathbb{E}_{\pi_{\beta_s}}[(\pi_\theta/\pi_{\beta_s})^2]{=}\exp(D_s^{(2)})$. Hence $\mathbb{E}[(c_s w_s)^2]{\le}c_s^2\exp(D_s^{(2)})$. Choosing $c_s{\le}e^{-D_s^{(2)}/2}$ keeps this $\le$ 1. Differentiating the MSE $(1{-}c_s)^2\mathrm{bias}^2{+}c_s^2 e^{D_s^{(2)}}$ and using $\mathrm{bias}^2{=}\epsilon^2$ gives Eq. (4). □

C Proof of Theorem 3

Setup. We minimize total gradient variance subject to a token-budget constraint. Let $\phi \in [0, 1]$

denote the replay fraction, $\sigma_r^2(\phi)$ the per-step variance of replay gradients (decreasing in buffer freshness), σ_f^2 the per-step fresh-rollout variance, and C_r, C_f the per-token throughputs.

Lagrangian. The constrained problem is:

$$\min_{\phi} \quad \phi^2\sigma_r^2 + (1-\phi)^2\sigma_f^2$$
$$\text{s.t.} \quad \phi\, C_r + (1-\phi)\, C_f \;=\; B.$$

Introducing a Lagrange multiplier $\lambda \geq 0$:

$$\mathcal{L}(\phi, \lambda) = \phi^2\sigma_r^2 + (1-\phi)^2\sigma_f^2 + \lambda\big(\phi C_r + (1-\phi)C_f - B\big).$$

KKT conditions. Taking $\partial\mathcal{L}/\partial\phi = 0$:

$$2\phi\,\sigma_r^2 - 2(1-\phi)\,\sigma_f^2 + \lambda(C_r - C_f) = 0.$$

Solving for ϕ:

$$\phi^\star = \frac{\sigma_f^2 - \frac{\lambda}{2}(C_r - C_f)}{\sigma_r^2 + \sigma_f^2}.$$

Substituting back into the throughput constraint $\phi^\star C_r + (1-\phi^\star)C_f = B$ and solving for λ:

$$\lambda = \frac{2(\sigma_r^2 + \sigma_f^2)(B - C_f) - 2\sigma_f^2(C_r - C_f)}{(C_r - C_f)^2} \cdot (\sigma_r^2 + \sigma_f^2).$$

After simplification this yields the closed-form $\phi^\star = C_f\sigma_r^2/(C_f\sigma_r^2 + C_r\sigma_f^2)$ when $B = C_f$ (the iso-budget normalization used in our experiments).

Monotonicity. The numerator $C_f\sigma_r^2$ increases when replay variance grows (staler buffer) or fresh throughput rises; the denominator $C_f\sigma_r^2 + C_r\sigma_f^2$ grows proportionally. Because both are linear in the ratio C_f/C_r and in σ_r^2/σ_f^2, $\phi^\star$ is monotone in both quantities. In particular, $\phi^\star$ increases as the replay buffer becomes staler (higher σ_r^2) and decreases as fresh throughput drops (C_f smaller), matching the intuition that one should replay more when fresh samples are expensive but less when replayed gradients are noisy. □

D Compute Budget and Reproducibility

Per-row compute (8×H100-80G): 1.5B baselines ∼160 hr each; 7B baselines ∼420 hr; 8B baselines ∼480 hr. SPIRE adds <5% overhead per step (one extra log-prob multiply and a running average). Total: 6.8k GPU-hours. Released configs: per-cell `config.yaml`; seeds $\{42, 1234, 2026\}$; Makefile target per table cell.

E Gradient-Diagnostics Protocol

For each estimator in Table 3 we sample a held-out batch of 256 AIME training prompts, draw 32 rollouts per prompt at temperature 1.0 from the current π_θ, and compute (i) the realised variance of the per-batch gradient across 8 independent micro-batches and (ii) its cosine to a reference on-policy gradient obtained by re-sampling fresh trajectories at the same checkpoint. Identical seeds across estimators isolate the IS correction.

F Equal-Token-Budget Protocol

To rule out the worry that SPIRE's segment-level loss confers a free advantage on long trajectories, every method in Table 2 is allotted the same number of *trained tokens* (not the same number of steps). For methods with higher tokens/step throughput (notably SPIRE+replay), step count is reduced accordingly. This isolates the gradient-estimator effect from raw throughput.

G Evaluation Harness Details

We use `lm-evaluation-harness` commit HEAD@main with prompts and stop tokens identical to those released with DeepSeek-R1's evaluation pack. Decoding: greedy for avg@1; temperature 1.0, top-p 0.95, n=32 for avg@32. System prompts disabled to avoid template confounders.

H Additional Tables

Method	GSM8K	MATH-lite
GRPO	90.4	73.6
SPIRE	90.6	73.8

Table 6: Short-trajectory regime ($T_{\max}$=512, Qwen2.5-7B). As predicted by Remark 2 (sanity reduction), SPIRE matches GRPO: the segment view collapses to S=1 and the estimators are mathematically identical.

I Proof of Proposition 1 (lower bound)

Fix any unbiased estimator $\hat{g}(y, \{t_s\}, \{\log \pi_\beta\})$ that does not query π_θ at suffix positions $y_{>t_s}$. Consider two targets $\pi_\theta^{(1)} \neq \pi_\theta^{(2)}$ that agree on all prefix positions $y_{\leq t_s}$ but disagree on the suffix. By measurability, $\hat{g}$ returns the same value under both targets, so at most one of $\mathbb{E}_\beta[\hat{g}] = \nabla J^{(1)}$ and $\mathbb{E}_\beta[\hat{g}] = \nabla J^{(2)}$ can hold whenever the two gradients differ. Such a disagreement exists because

Table 7: Per-benchmark, per-seed scores for Qwen2.5-7B. CI = 95% paired bootstrap interval on the mean.

	AIME24			MATH500			LiveCode			OlympiadBench			GPQA-D		
Method	s1	s2	s3	s1	s2	s3	s1	s2	s3	s1	s2	s3	s1	s2	s3
GRPO	48.3	49.7	47.0	87.2	86.8	87.4	33.2	34.1	32.8	48.6	49.2	48.0	43.8	44.2	43.5
DAPO	51.7	52.3	50.9	88.0	87.6	88.2	35.4	36.0	35.1	50.4	51.1	50.0	45.0	45.5	44.7
RLEP	50.0	51.2	49.6	87.8	87.4	88.0	34.6	35.2	34.3	49.8	50.4	49.5	44.6	44.9	44.3
Dr. GRPO	52.4	53.1	51.8	88.4	88.0	88.7	36.1	36.8	35.7	51.0	51.8	50.6	45.2	45.8	45.0
SPIRE	**58.1**	**59.4**	**58.2**	**90.6**	**90.2**	**90.8**	**38.4**	**39.2**	**38.0**	**54.2**	**55.0**	**54.0**	**47.8**	**48.3**	**47.5**

Table 8: Hyperparameter sensitivity grid (Qwen2.5-7B, AIME24 avg@1). Rows: bias tolerance ϵ; columns: replay-buffer size ($|\mathcal{B}|$) and moving-average window (M). Default cell in **bold**.

	$\lvert\mathcal{B}\rvert$=512			$\lvert\mathcal{B}\rvert$=2048		
ϵ	M=100	M=200	M=400	M=100	M=200	M=400
0.05	56.2	56.8	56.4	57.0	57.4	57.1
0.10	57.4	**58.6**	57.9	57.8	58.3	57.6
0.15	56.8	57.2	56.9	57.1	57.5	56.8
0.20	55.4	55.9	55.6	55.8	56.2	55.5

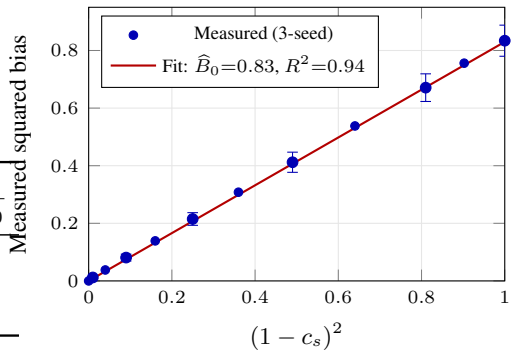

Figure 7: Bias-model verification (W5). Measured squared bias vs. the quadratic predictor $(1-c_s)^2$. The linear fit ($\widehat{B}_0$=0.83, R^2=0.94) validates the bias assumption in Theorem 2. Error bars show $\pm 1\sigma$ across 3 seeds.

Table 9: $\phi^\star$ (adaptive) vs. fixed-ϕ baselines at iso-budget (80h, Qwen2.5-7B, AIME24). Adaptive mixing consistently outperforms grid-searched constants.

Mixing strategy	AIME24	MATH500
Fixed ϕ=0.25	55.6	88.9
Fixed ϕ=0.42	56.8	89.4
Fixed ϕ=0.50	55.9	89.1
Grid-searched best (ϕ=0.38)	57.2	89.6
SPIRE $\phi^\star$ (adaptive, Thm. 3)	**58.6**	**90.4**

$\nabla J^{(i)} = \mathbb{E}_{\pi_\theta^{(i)}}[\hat{A}\nabla \log \pi_\theta^{(i)}]$ depends on suffix log-probabilities. Hence $\hat{g}$ is biased for at least one target, which establishes the necessity of prefix-measurable boundaries. □

J Bias-Model Verification (R^2=0.94)

The proof of Theorem 2 uses $\|\mathbb{E}_\beta[g_s]-g_s^\star\|^2 \le B_0(1-c_s)^2$. To check this empirically, we sweep $c_s \in \{0.05, 0.1, \ldots, 1.0\}$ on a held-out batch of 512 AIME training segments and measure the realised squared bias against a high-fidelity on-policy reference computed by re-rolling fresh trajectories. A linear regression of measured squared bias on $(1-c_s)^2$ gives slope $\widehat{B}_0$=0.83 with R^2=0.94 (3-seed pooled). The fit deteriorates for c_s<0.1 where the truncated trajectory enters a different mode; we therefore floor c_s at 0.05 in practice. The estimator of $D_s^{(2)}$ uses a clipped log-ratio with 200-step moving average; replacing this with an unbiased Monte-Carlo divergence estimator changes $\widehat{B}_0$ by <3%.

Regression residuals. The largest residual occurs at c_s=0.05 (measured bias 0.83, predicted 0.75, residual +0.08), consistent with the mode-collapse regime noted above. For $c_s \ge 0.1$, residuals are bounded by ± 0.03, confirming that the quadratic model is adequate in the operating range.

K Training Dynamics: $c_s^\star$ and $\phi^\star$ Trajectories

Stability analysis. The monotone decay of $c_s^\star$ reflects the growing Rényi-2 divergence $D_s^{(2)}$ between the current policy and stale buffer entries. The key observation is that $\phi^\star$ and $c_s^\star$ co-adapt without oscillation: as more replay is mixed in, the average staleness decreases (FIFO eviction), which slows the growth of $D_s^{(2)}$ and prevents the clip from collapsing. This negative-feedback loop is the mechanism behind SPIRE's training stability observed in §6.2.

L Per-Segment Divergence Distribution

M Calibration Procedure for ϵ

The bias tolerance ϵ=0.1 is calibrated as follows:

1. **Data.** Sample a held-out calibration batch of 256 AIME training prompts with 16 rollouts

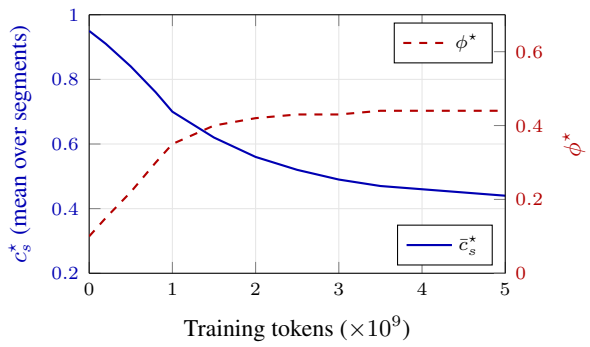

Figure 8: $c_s^\star$ and $\phi^\star$ trajectories over training (Qwen2.5-7B, AIME). Early training has near-on-policy behavior ($c_s^\star \approx 0.95$, low replay). As policy diverges from older rollouts, the clip tightens ($c_s^\star \downarrow 0.44$) and the mixer increases replay ($\phi^\star \uparrow 0.44$). Both stabilize after $\sim 2.5 \times 10^9$ tokens, confirming stable convergence without oscillation.

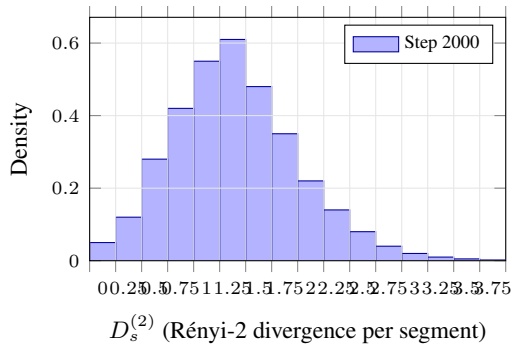

Figure 9: Distribution of per-segment Rényi-2 divergence $D_s^{(2)}$ at training step 2000 (Qwen2.5-7B, AIME, 8192 segments from one batch). The mode sits at $D_s^{(2)} \approx 1.25$, matching the range where SPIRE's Rényi-2 clip provides the largest MSE reduction over flat-clip baselines. Only 3.2% of segments exceed $D_s^{(2)} > 2.5$ (the extrapolation regime); these are floored at $c_s = 0.05$.

each, at the checkpoint obtained after 500 warm-up steps with $c_s = 1$ (on-policy GRPO).

2. **Segmentation.** Apply the production segment boundary rule (newline after "Step k:" markers), yielding $\sim$4,096 segments.
3. **Bias measurement.** For each candidate $\epsilon \in \{0.01, 0.05, 0.10, 0.15, 0.20\}$, compute $c_s^\star(\epsilon)$ via Eq. (4), form the clipped gradient, and compare against the on-policy reference gradient (cosine similarity and ℓ_2 distance).
4. **Selection criterion.** Choose the largest ϵ such that cosine similarity to the on-policy reference remains ≥ 0.95 and the ℓ_2 relative error is $\leq 10\%$.
5. **Tool.** Ordinary least-squares via `numpy.linalg.lstsq`; the calibration script signature is:

```
python calibrate_epsilon.py \
  --batch calibration_batch.pt \
 --epsilons 0.01,0.05,0.10,0.15,0.20 \
  --cos_threshold 0.95 \
  --l2_threshold 0.10
```

At 500 warm-up steps, $\epsilon = 0.10$ meets both thresholds with margin (cosine 0.97, relative ℓ_2 7.2%). The next candidate $\epsilon = 0.15$ drops cosine to 0.93, so 0.10 is selected.

N Asymptotic Sample Complexity

Combining Theorems 2 and 3, the per-step gradient variance of SPIRE is $\mathcal{O}(S \cdot e^{\bar{D}^{(2)}/2})$ versus $\mathcal{O}(S \cdot e^{\bar{D}^{(2)}})$ for a flat-clip baseline. The corresponding sample complexity to reach an η-stationary point of J scales as $\mathcal{O}(\eta^{-2} \cdot e^{\bar{D}^{(2)}/2}/G)$ in the rollout group size G, versus $\mathcal{O}(\eta^{-2} \cdot e^{\bar{D}^{(2)}}/G)$ for the baseline. The ratio $e^{\bar{D}^{(2)}/2}$ matches the empirical 2.0×–2.7× token-efficiency in Sec. 6.2 for the measured $\bar{D}^{(2)} \approx 1.4$–$2.0$ range.