

Rollout-Level Advantage-Prioritized Experience Replay for GRPO

Gyeongtae Yoo¹ Sanghyeok Park² Soohyuk Jang¹ Ik-hwan Kim¹

Sungroh Yoon^{1,2,3†}

¹Department of Electrical and Computer Engineering, Seoul National University

²Interdisciplinary Program in AI, Seoul National University

³AIIS, ASRI, INMC, and ISRC, Seoul National University

sryoon@snu.ac.kr

Abstract

Reinforcement learning from verifiable rewards with GRPO is a standard approach for post-training reasoning LLMs. It remains sample inefficient. Each rollout is used for a single gradient update and then discarded. Naive replay is not well suited in this setting because LLM policies drift quickly per gradient step. Stored rollouts therefore become stale and can destabilize training. We propose a rollout-level replay buffer for GRPO that stores and samples individual rollouts rather than whole groups. The buffer bounds staleness through age eviction. Any rollout older than $\tau_{\max}$ training steps is removed. The buffer also preserves on-policy data via fresh-anchored composition. Each batch keeps its fresh on-policy rollouts and then concatenates replay rollouts drawn separately from the buffer. We prioritize replay by per-rollout advantage magnitude and recycle individual rollouts whose advantages are large. Across three Qwen3-Base scales on five math benchmarks, our method outperforms GRPO and naive replay baselines. Gains are positive at every scale and reach +1.66 pp on the five-benchmark average at 4B. Under an AES metric that jointly measures accuracy and token efficiency, our method is the only condition with a positive margin over GRPO at every scale.

1 Introduction

Reinforcement learning from verifiable rewards (RLVR) with Group Relative Policy Optimization (GRPO) (Shao et al., 2024; Guo et al., 2025) is a standard approach for post-training reasoning LLMs. Producing a group of candidate rollouts per training prompt accounts for most of GRPO’s training cost (Baroian and Berger, 2026; Zheng et al., 2025b). Yet each rollout drives only a single gradient update before it is discarded. This is especially wasteful for rollouts that carry a strong learning

signal. One example is a rare correct rollout in an otherwise-incorrect group.

Experience replay is one option, but it does not transfer directly to GRPO. Policies drift quickly per gradient step, and stored rollouts become stale within a few updates (Fu et al., 2025; Zheng et al., 2025a). Such staleness can destabilize training, so explicit staleness control is required in any GRPO replay design.

We propose a **rollout-level** replay buffer for GRPO, where individual rollouts are stored and sampled. The buffer bounds staleness via **age eviction**. Each rollout’s age is the number of training steps elapsed since it was generated. Any rollout older than $\tau_{\max}$ steps is removed. The policy lag of any reused rollout is therefore bounded by $\tau_{\max}$ steps. This step-level bound holds independently of buffer size or model scale, unlike a capacity-based rule.

The buffer-entry and eviction policy alone do not specify how the gradient batch is constructed. A single unified buffer of fresh and stale rollouts can displace the fresh on-policy rollouts at a given step. This is the standard DQN and PER pattern, and we refer to it as **pool composition**. We instead adopt the **fresh-anchored composition** used in prior GRPO replay work (Zhan et al., 2026; Li et al., 2025). The fresh rollouts are retained in full. Replay rollouts are drawn separately and then concatenated on top. Each gradient step then includes an on-policy anchor. Age eviction and fresh-anchored composition together bound worst-case policy lag.

Within each group, the standard normalization yields a per-rollout group-relative advantage A_i . Rollouts with the largest absolute advantage $|A_i|$ account for most of the learning signal (Lin et al., 2025; Zhu et al., 2026a). These are the within-group minority rollouts in skewed groups. We adopt $|A_i|$ as the replay priority signal and sample individual rollouts directly across the buffer.

[†]Corresponding author.

This contrasts with prior GRPO replay schemes that operate at the granularity of the query. Those schemes store and sample the G rollouts of each prompt together as a single bound unit, a setting we refer to as **query-level** replay (Zhan et al., 2026; Li et al., 2025).

We train on the DeepScaleR-Preview prompt set (Luo et al., 2025b) at three Qwen3-Base scales of 0.6B, 1.7B, and 4B. Our method outperforms GRPO and two naive replay baselines on five math benchmarks. Gains are positive at every scale and reach +1.66 pp on the five-benchmark average at 4B. The contributions of our work are as follows.

- We propose a GRPO replay buffer that combines age eviction, fresh-anchored composition, and rollout-level $|A_i|$ priority. Our method outperforms GRPO and naive replay baselines at every scale. Five-benchmark-average gains are +0.61 pp at 0.6B, +0.60 pp at 1.7B, and +1.66 pp at 4B. See Section 4, Section 5.2, and Table 1.
- Per-axis ablations at 1.7B justify each component of the method. A moderate age cap of $\tau_{\max} = 10$ performs best and trades off replay volume against staleness. Fresh-anchored composition is necessary at scale. Its pool-composition alternative mixes stale rollouts with fresh ones and falls below GRPO at 4B. Rollout-level $|A_i|$ priority further leads on PASS@ K . See Section 5.3 and Tables 2, 3, and 4.
- Our method also improves the accuracy efficiency score (AES) over GRPO at every scale. AES jointly scores accuracy and generation-token budget as a single efficiency value. A length-by-correctness decomposition further analyzes the gain. See Section 6 and Tables 5 and 6.

2 Related Work

Experience Replay with DQN and PER Experience replay originated with DQN (Mnih et al., 2015). DQN trains a value network on transitions sampled from a FIFO buffer so that the network draws past experience rather than only the latest trajectory. Prioritized Experience Replay (Schaul et al., 2016) extends DQN with sampling proportional to a power-scaled priority. The priority is typically the TD-error and focuses updates on transitions with the largest learning signal. Both methods are built for value-based RL with millions of small refreshable transitions. That regime does not hold

for GRPO. Our buffer adopts the DQN-style FIFO and the PER backbone, and adds GRPO-specific adaptations developed in Section 4.

Experience Replay for LLM RL Applying replay to GRPO violates two PER assumptions. First, the group-relative advantage A_i cannot be refreshed without regenerating the entire group. Second, LLM policies drift quickly per gradient step (Fu et al., 2025; Zheng et al., 2025a), so even recent rollouts can be substantially stale. The closest architectural neighbors are ExGRPO (Zhan et al., 2026), RePO (Li et al., 2025), RLEP (Zhang et al., 2025a), Buffer Matters (Wan et al., 2026), and Fatemi (2026). All five mix on-policy fresh rollouts with replayed rollouts in each GRPO update. This on-policy anchor design is the same one that our fresh-anchored composition in Section 4.1 adopts. The five works differ from our method along the buffer-entry and priority axes, since all five replay at the query level. ExGRPO prioritizes by correctness times entropy, RLEP by verified-correct retention, and Buffer Matters by re-evaluating historically difficult samples. Fatemi (2026) prioritizes by empirical solve rate. This is the maximum-variance regime equivalent to high σ_g .

Two other works apply a DQN-style replay framing for LLM RL. Arnal et al. (2026) maintains a FIFO trajectory buffer with uniform or correctness-weighted sampling. That work uses an importance-ratio correction in place of an on-policy anchor. FreshPER (Ma et al., 2026) applies a soft exponential age decay under reward-magnitude priority. Both differ from our method along the priority and staleness-control axes. Arnal et al. (2026) also differs on composition. Beyond these closest neighbors, the GRPO replay literature includes prompt-level reuse such as Prompt Replay (Baroian and Berger, 2026) and PSPO (Zhu et al., 2026b). It also includes filter-replay frameworks such as EFRame (Wang et al., 2025a) and DOTS (Sun et al., 2025), and diversity-preserving replay such as DyJR (Li et al., 2026). These operate at the prompt level or interleave selection with replay, and are orthogonal to our rollout-level design.

Priority Signal Design Two angles in recent GRPO work inform priority signal design. The first is absolute advantage as a selection signal. CPPO (Lin et al., 2025) prunes completions with low $|A|$ since the high $|A|$ tail accounts for the bulk of the useful learning signal. DPPO (Zhu et al.,

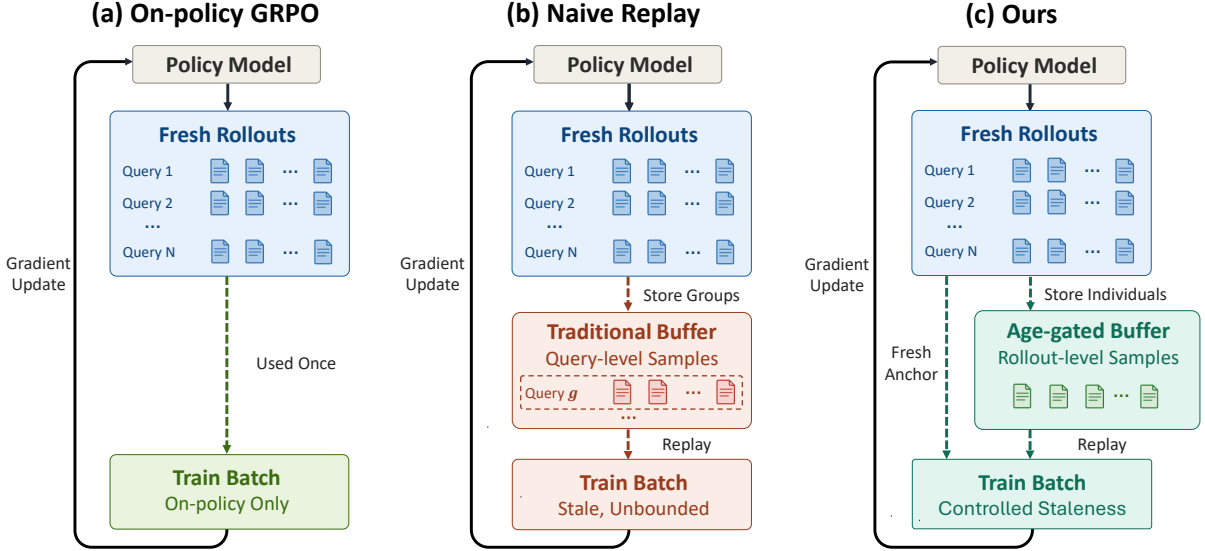


Figure 1: RL post-training paradigms for LLMs. Panel (a) is strictly on-policy GRPO that discards each rollout. Panel (b) is classical replay that stores whole groups under a capacity bound with unbounded staleness. Panel (c) is our method, which anchors every update with a fresh batch and replays individual rollouts from an age-gated buffer under $|A_i|$ priority.

2026a) adds an importance-sampling correction under the same principle. PODS (Xu et al., 2025) downsamples each GRPO step’s rollouts by reward variance. It keeps the highest and lowest reward extremes within a group as a within-step analogue of high $|A_i|$ selection. The second angle is within-group reward variance, widely used as a difficulty or informativeness proxy. DAPO (Yu et al., 2025) drops zero-variance groups from training, a convention we also adopt in Section 3. VCRL (Jiang et al., 2025) schedules training by group reward variance. Online difficulty filtering (Bae et al., 2026; Gao et al., 2025) biases sampling toward prompts whose solve rate is bounded away from 0 and 1. This is the maximum-variance regime under binary rewards and is equivalent to high σ_g . The same regime is adopted as a replay-side priority by Fatemi (2026). We use the query-level σ_g as the baseline against our rollout-level $|A_i|$ priority. Our work applies the first angle to replay sampling, since rollout-level $|A_i|$ priority targets the within-group minority rollouts that query-level σ_g cannot resolve. Appendix L expands on the classical-replay and priority-signal threads and summarizes the position of our method in the broader design space.

3 Preliminaries

We study GRPO (Shao et al., 2024) under binary rewards.

GRPO At training step t we pick a prompt q and ask the current policy π_t to generate a group of G candidate answers $\{o_1, \dots, o_G\}$. Each o_i is a rollout. An automatic verifier assigns each rollout a binary reward $r_i \in \{-1, +1\}$. The group-relative advantage of rollout i is

$$A_i = \frac{r_i - \mu_g}{\sigma_g + \epsilon_\sigma}, \quad (1)$$

where μ_g and σ_g are the within-group reward mean and standard deviation, and ϵ_σ is a small numerical-stability constant. Groups with identical rewards yield $A_i = 0$ for all members and provide no learning signal. We therefore drop such groups, so the replay buffer holds only nonzero-advantage rollouts. This keeps buffer management simple. GRPO updates π_θ via a PPO-style clipped surrogate with importance ratio $\rho_i = \frac{\pi_\theta(o_i|q)}{\pi_t(o_i|q)}$,

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q, \{o_i\}} \left[\frac{1}{G} \sum_{i=1}^G \min(\rho_i A_i, \text{clip}(\rho_i, 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}) A_i) \right]. \quad (2)$$

Here π_t denotes the behavior policy at the rollout’s generation step. π_θ denotes the current target policy at the update step. ρ_i is applied per token following the verl and DAPO convention. For replay, $\pi_t(o_i|q)$ is cached with each rollout at generation time, so ρ_i reflects drift from the rollout’s birth step to

Algorithm 1 Rollout-level $|A_i|$ replay training step at t . Fresh rollouts always anchor the update, and the replay draw recycles past rollouts under an $|A_i|$ -prioritized buffer with age eviction.

Require: Current policy π_t , fresh-rollout count B_{fresh} , replay ratio r , max age τ_{max}

Phase 1. Fresh rollout generation

- 1: Sample B_{fresh} fresh rollouts from π_t and compute $\{A_i\}$ following Section 3
- 2: **Filter.** drop zero-variance groups, leaving B'_{fresh} surviving rollouts ▷ zero-variance filter

Phase 2. $|A_i|$ -prioritized replay sampling

- 3: Draw $r B'_{\text{fresh}}$ rollouts from the buffer via Equation 3 ▷ rollout-level

Phase 3. Mixed-policy gradient update

- 4: $\pi_{t+1} \leftarrow$ GRPO update on the fresh survivors concatenated with the replay draw ▷ fresh-anchored

Phase 4. Buffer update with age-gated eviction

- 5: Add the fresh survivors to the buffer with birth step t
 - 6: Evict every rollout with birth step $< t - \tau_{\text{max}}$ ▷ age eviction
-

the current training step. We adopt the Clip-Higher scheme of DAPO (Yu et al., 2025) and set $\epsilon_{\text{low}} = 0.2$ and $\epsilon_{\text{high}} = 0.28$.

Prioritized Replay Sampling We use the PER scheme (Schaul et al., 2016) with a fixed-capacity FIFO buffer that evicts the oldest entry when full. Section 4.1 extends this with age eviction. A unit i with priority $p_i \geq 0$ is drawn with probability

$$P(i) = \frac{p_i^\alpha}{\sum_j p_j^\alpha}, \quad (3)$$

where the sum runs over all units currently in the buffer. The exponent $\alpha \in [0, 1]$ interpolates between uniform sampling at $\alpha = 0$ and fully proportional sampling at $\alpha = 1$. We set the priority to the advantage magnitude $p_i = |A_i|$. Section 4.2 motivates this choice.

4 Adapting Experience Replay to GRPO

Applying experience replay to GRPO raises two issues. First, replayed rollouts are stale relative to the current policy. Second, the group structure of GRPO aggregates per-rollout learning signals into a single query-level value. We address the first issue with age eviction and fresh-anchored composition, which together bound policy lag. Section 4.1 describes this design. We address the second issue with rollout-level buffer entries and advantage-magnitude priority. This selects the high-signal rollouts that query-level replay does not isolate, as described in Section 4.2. Figure 1 contrasts our method with standard on-policy GRPO and classical replay. The full procedure is given in Algorithm 1.

Replay makes the update only approximately on-policy, and the replayed portion of each gradient

step is a biased estimator of the on-policy GRPO gradient. We do not correct that bias. The controls below bound that departure rather than remove it, and the cached $\pi_t(o_i | q)$ of a replayed rollout enters only through the clipped ratio, which limits its influence without reweighting it to the current policy. Our update therefore does not recover the on-policy objective.

4.1 Age Eviction and Fresh-Anchored Composition

Age Eviction To bound per-rollout staleness, each stored rollout carries a birth step t_b that records the training step at which it was generated. Its age at the current step t is $t - t_b$. We impose age eviction and remove any rollout whose age exceeds τ_{max} , so no update reuses a rollout beyond a fixed age. Bounding by age rather than buffer capacity keeps the limit invariant to model scale and data difficulty. Zero-variance filtering removes a scale-dependent fraction of generated groups, so the effective batch B'_{fresh} varies with model scale. Appendix B reports the per-scale values. Any fixed capacity therefore induces different steady-state ages per setting. In contrast, τ_{max} matches the per-step drift of LLM post-training (Fu et al., 2025; Zheng et al., 2025a) without per-configuration tuning. A FIFO capacity can only approximate this because B'_{fresh} shifts both across settings and during a single run as the solve-rate distribution of the policy changes.¹ A brief warmup with replay disabled at the start of training allows the buffer to populate and the model to acquire basic instruction-

¹A capacity cap of 30,000 is retained as a backstop. At the default $\tau_{\text{max}} = 10$, age eviction is the binding constraint at all scales. The steady-state buffer size $\tau_{\text{max}} \cdot B'_{\text{fresh}}$ is about 10,000 even at 4B.

Model	Method	MATH-500		AIME25		AIME26		HMMT-F25		HMMT-F26		avg	
		avg@8	pass@8	avg@16	pass@16	avg@16	pass@16	avg@16	pass@16	avg@16	pass@16	avg	pass
Qwen3-0.6B	GRPO	52.71	74.27	1.67	14.44	0.28	3.33	0.35	4.44	0.88	10.10	11.18 $\pm$ 0.19	21.32 $\pm$ 0.40
	uniform	52.89	76.87	1.94	12.22	0.90	7.78	0.21	3.33	0.57	6.06	11.30 $\pm$ 0.17	21.25 $\pm$ 1.24
	σ_g	52.32	76.87	2.50	14.44	0.90	6.67	0.28	4.44	1.14	9.09	11.43 $\pm$ 0.38	22.30 $\pm$ 0.92
	Ours	53.53	76.87	2.08	15.56	1.18	5.56	0.62	5.56	1.52	9.09	11.79 $\pm$ 0.25	22.52 $\pm$ 0.78
Qwen3-1.7B	GRPO	67.58	84.93	5.56	26.67	2.85	15.56	0.21	3.33	3.09	9.09	15.86 $\pm$ 0.10	27.92 $\pm$ 0.61
	uniform	67.50	84.93	5.01	24.84	3.86	13.76	0.38	4.32	2.11	9.09	15.77 $\pm$ 0.41	27.39 $\pm$ 0.63
	σ_g	67.43	85.40	4.03	16.67	4.24	20.00	0.21	3.33	3.09	11.11	15.80 $\pm$ 0.40	27.30 $\pm$ 2.09
	Ours	68.87	86.07	5.83	30.00	4.44	17.78	0.69	6.67	2.46	11.11	16.46 $\pm$ 0.18	30.32 $\pm$ 1.98
Qwen3-4B	GRPO	83.31	92.80	17.57	38.89	17.29	32.22	8.89	20.00	11.81	26.26	27.77 $\pm$ 0.98	42.03 $\pm$ 1.28
	uniform	82.86	92.93	17.19	38.19	15.13	32.95	8.58	27.38	10.93	24.96	26.94 $\pm$ 1.70	43.28 $\pm$ 0.68
	σ_g	82.25	92.67	16.60	42.22	14.65	33.33	6.39	20.00	9.85	24.24	25.95 $\pm$ 0.86	42.49 $\pm$ 3.12
	Ours	84.52	94.13	19.38	43.33	19.44	45.56	10.14	31.11	13.70	30.30	29.43 $\pm$ 2.25	48.89 $\pm$ 3.71

Table 1: Cross-scale main results comparing GRPO, two naive replay baselines, and our fresh-anchored rollout-level $|A_i|$ method. The two naive baselines are uniform and σ_g under pool composition. All replay variants share $r = 0.5$, $\tau_{\max} = 10$, and $\alpha = 0.5$. MATH-500 uses $k = 8$, and AIME and HMMT use $k = 16$. The avg columns are the unweighted five-benchmark mean, reported as mean $\pm$ standard deviation across seeds. **Bold** marks the best per column within each scale.

following before replay activates. The warmup length is given in Section 5.1.

Fresh-Anchored Composition Age eviction caps the staleness of each individual rollout, but it does not specify how fresh and replay rollouts combine into the gradient batch. At each training step we have fresh on-policy rollouts sampled from the current policy π_t and the age-bounded buffer of past rollouts. We adopt fresh-anchored composition, following recent GRPO replay schemes that pair fresh on-policy rollouts with separately drawn replay rollouts. The contrasting mode merges fresh and stale rollouts into a single pool, which is the standard DQN and PER pattern. We call this pool composition and include it as a naive baseline in Section 5.2, then ablate it against fresh-anchored composition in Section 5.3.1. Our formulation parameterizes the mix by an explicit replay ratio r . The fresh rollouts are retained in full. A replay portion of size $r \cdot B'_{\text{fresh}}$ is drawn separately from the buffer and concatenated on top, where B'_{fresh} is the post-filter fresh rollout count. Each gradient step then retains a fresh on-policy anchor that bounds policy lag independently of the replay portion. The pool-composition baseline uses the same total batch size $(1 + r) \cdot B'_{\text{fresh}}$ but draws the entire batch from a unified pool of fresh and stale rollouts. Any difference between the two modes in Section 5.3.1 is therefore attributable to composition mode rather than to replay budget.

4.2 Rollout-Level $|A_i|$ Priority

$|A_i|$ Priority Even with age eviction bounding policy lag (Section 4.1), each replayed rollout is by construction at least slightly off-policy. The

priority signal must therefore focus the replay budget on rollouts whose learning signal still justifies the off-policy cost. We expect a useful priority to favor rollouts with (i) a large birth-step gradient coefficient and (ii) low representation in fresh batches. The per-rollout advantage magnitude $|A_i|$ satisfies both. It is the per-rollout coefficient in the GRPO objective (Equation 2), the same role TD-error plays in value-based PER. High- $|A_i|$ rollouts are also the within-group minority of skewed groups (Appendix E) that fresh batches see sparsely, because such groups arise from prompts the policy rarely solves cleanly. In contrast, σ_g would emphasize balanced groups that fresh batches already cover densely. Recent GRPO work uses $|A|$ similarly (Lin et al., 2025; Zhu et al., 2026a; Xu et al., 2025). Because sampling from the buffer follows $P(i) \propto p_i^\alpha$ (Equation 3), replay draws high- $|A_i|$ rollouts more often than the on-policy distribution would, and that reweighting is the purpose of the buffer rather than a side effect. We consequently do not undo it with a PER importance-sampling weight, which at full strength targets the same expected update as uniform sampling from the buffer, the $\alpha = 0$ setting that Table 4 reports as rollout-level uniform and places below $|A_i|$ priority on both metrics. We set the strength of the reweighting with α instead and temper it at $\alpha = 0.5$, because full proportional sampling at $\alpha = 1.0$ overshoots and costs pass accuracy (Appendix C).

Rollout-Level Buffer Entries $|A_i|$ priority requires a buffer-entry decision. Conventional GRPO replay schemes (Zhan et al., 2026; Ma et al., 2026) bind the G rollouts of a query into one buffer entry, so all G rollouts share a single priority. This forces

the priority signal to be query-level, such as uniform or a group-aggregate like σ_g . A query-level signal cannot resolve the within-group minority that holds the $|A_i|$ signal. Combined with pool composition (Section 4.1), this configuration is the naive baseline in Table 1. We therefore deviate from this convention and store individual rollouts. Each rollout carries its own birth-step $|A_i|$ and is sampled directly by Equation 3. Section 5.3.2 confirms that the gain comes from $|A_i|$'s within-group variation, not from the buffer-entry shift. That coefficient stays at its birth-step value together with the group statistics (μ_g, σ_g) , so it is estimated under an earlier solve-rate distribution while the gradient it multiplies is taken under the current policy, and rollout-level draws break the within-group centering that keeps the fresh portion mean-zero. The same $\tau_{\max}$ that bounds policy lag also bounds how stale the coefficient can be, since a replayed advantage is never more than $\tau_{\max}$ steps out of date.

5 Experiments

5.1 Experimental Setup

We train three scales of the Qwen3-Base family at 0.6B, 1.7B, and 4B. Training uses the DeepScaleR-Preview (Luo et al., 2025b) prompt set across all conditions. The four conditions are GRPO, the two naive replay baselines, and our method. We hold all training hyperparameters constant except for the replay-buffer design. The cross-scale main comparison of Section 5.2 and the analyses of Section 6 are run with three training seeds, and the ablations of Section 5.3 use a single seed. Training hyperparameters and the seed policy are listed in Appendix A.

Replay Buffer Our method instantiates Algorithm 1 with fresh-anchored composition from Section 4.1 and rollout-level $|A_i|$ priority from Section 4.2. The priority is $p_i = |A_i| + \epsilon$, and we sample per Equation 3 with $\alpha = 0.5$. Age eviction uses $\tau_{\max} = 10$ and the replay ratio is $r = 0.5$. We use a 20-step warmup during which replay is disabled. The naive replay baselines are uniform and σ_g . They share the same $\tau_{\max}$, r , α , and warmup as our method. They differ from our method in three ways. They use pool composition, they store at the query level, and their priority signal is uniform or σ_g rather than $|A_i|$. A replayed rollout's advantage A_i and its group statistics (μ_g, σ_g) are frozen at the birth-step values, since refreshing them would require regenerating the entire group. We ablate

Method	Accuracy		Clipping (%)	
	avg	pass	fresh	replay
$\tau_{\max} = 1$	16.49	29.59	0.055	0.266
$\tau_{\max} = 10$ (Ours)	16.58	32.35	0.060	0.403
$\tau_{\max} = 30$	16.01	27.34	0.058	0.654
$\tau_{\max} = 60$	15.28	26.66	0.058	0.875

Table 2: Age-eviction horizon $\tau_{\max}$ ablation at Qwen3-1.7B-Base. The clipping columns report the fraction of importance ratios clipped during the PPO update, decomposed into fresh and replay batch portions. **Bold** marks the best.

$\tau_{\max}$ in Section 5.3.1 and jointly ablate r and α in Appendix C.

Benchmarks and Decoding We evaluate on five math-reasoning benchmarks, namely MATH-500, AIME25, AIME26, HMMT-F25, and HMMT-F26. Appendix A (Benchmarks and Decoding paragraph) gives the per-benchmark sizes, decoding settings, and per-benchmark k values. That appendix also describes the five-benchmark aggregate used in the ablation tables from Table 2 through Table 4 and the full seed policy.

5.2 Cross-Scale Main Comparison

Table 1 reports the main result. The table separates the benefit of our method from the no-replay GRPO reference and from the naive replay baselines. The naive baselines apply uniform and σ_g priorities under pool composition with query-level buffer entries. The naive baselines have their batch size matched to fresh-anchored ours, so any gap to our method is attributable to composition and rollout-level $|A_i|$ priority rather than to replay budget.

Our method improves over both baseline families at every scale. The gain over GRPO on the five-benchmark average is +0.61 pp at 0.6B, +0.60 pp at 1.7B, and +1.66 pp at 4B. The gain also extends to PASS@K. Our method exceeds GRPO by +6.86 pp on pass, and the best naive baseline by +2.49 pp on avg and +5.61 pp on pass, as shown in the rightmost block of Table 1.

Naive replay gives no consistent gain over GRPO under either uniform or σ_g priority, and at 4B it falls below the no-replay baseline by 0.83 and 1.82 pp on the five-benchmark average. Reusing rollouts is therefore not beneficial on its own, and without a priority that selects which rollouts to reuse it can cost accuracy. The design-space ablations in Section 5.3 examine which axes of our design separate it from these baselines.

Method, Composition	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B	
	avg	pass	avg	pass	avg	pass
GRPO (no replay)	11.20	21.32	15.75	27.64	26.71	40.92
Pool composition	11.29	21.02	16.00	27.08	23.54	39.08
Fresh-anchored (Ours)	11.58	23.24	16.58	32.35	31.06	50.98

Table 3: Cross-scale composition-mode comparison between fresh-anchored composition for ours and pool composition. Both are under rollout-level $|A_i|$ priority. avg and pass are five-benchmark aggregates under Table 1’s k convention. **Bold** marks the better composition at each scale.

5.3 Ablations

We validate each design choice of Section 4 at 1.7B, with the composition ablation extended to 0.6B and 4B. We sweep alternative variants along two axes. The first axis is staleness handling and includes age eviction ($\tau_{\max}$) and fresh-anchored composition. Section 5.3.1 covers this axis. The second axis is the rollout-level priority signal, comparing $|A_i|$ against uniform and σ_g . Section 5.3.2 covers this axis. Each table reports five-benchmark aggregate avg and pass under the k convention of Table 1.

5.3.1 Age Eviction and Fresh Anchoring

Table 2 sweeps the age-eviction horizon $\tau_{\max} \in \{1, 10, 30, 60\}$ at 1.7B with other axes at default. Allowing samples to age past $\tau_{\max} = 10$ reduces both metrics. As $\tau_{\max}$ grows from 10 to 60, avg drops from 16.58 to 15.28 and pass drops from 32.35 to 26.66. Aging past the cap also induces asymmetric importance-ratio clipping. Replay-side clipping grows monotonically with the cap and reaches 0.88% at $\tau_{\max} = 60$. Fresh-side clipping is approximately constant. The asymmetric clipping alongside the performance drop indicates that an unmanaged buffer accumulates policy lag rather than stochastic noise.² The per-step clipping trajectory is in Figure 5 of Appendix J. At the opposite end, $\tau_{\max} = 1$ matches our default on avg with 16.49 against 16.58, but is -2.76 pp lower on pass. At $\tau_{\max} = 1$ the buffer holds at most one step’s worth of rollouts at any time, so the replay portion has little to recycle and contributes minimal additional signal. We adopt $\tau_{\max} = 10$ as the smallest age at which the buffer accumulates enough rollouts to make the priority-driven replay meaningful, while keeping policy lag bounded.

²All clipping rates stay below 1%. Over 99% of importance ratios therefore fall within the DAPO clip-higher window from $1 - 0.2$ to $1 + 0.28$.

Method	Level	avg	pass
GRPO (no replay)	n/a	15.75	27.64
uniform	query	15.94	27.48
	rollout	16.19	28.96
σ_g	query	15.43	29.04
	rollout	16.04	29.16
Ours with $ A_i $	rollout	16.58	32.35

Table 4: Priority signal and level ablation at Qwen3-1.7B-Base. All rows share fresh-anchored composition, $r = 0.5$, and $\tau_{\max} = 10$, so the rollout-level uniform row is an informative-sampling control that matches our method’s replay volume and differs only in the priority signal. The $|A_i|$ signal is rollout-only by design. avg and pass are five-benchmark aggregates under Table 1’s k convention. **Bold** marks the best per column.

Age eviction alone does not fully bound policy lag at scale. The composition mode also affects performance across the three tested scales. Table 3 fixes $\tau_{\max} = 10$ and replaces fresh-anchored composition with pool composition. The effect is scale-dependent. At 0.6B the avg gap between the two compositions is negligible at 0.29 pp and the pass gap is modest at $+2.22$ pp. Both gaps widen from 0.6B to 4B. At 1.7B the pass gain over GRPO is removed under pool composition, with 27.08 against the GRPO value of 27.64. At 4B both avg and pass fall below GRPO outright, with avg 23.54 against the GRPO value of 26.71.

5.3.2 Rollout-Level $|A_i|$ Against Query-Level Baselines

Table 4 ablates the priority signal at 1.7B across two axes. The signal axis covers uniform, σ_g , and $|A_i|$. The level axis covers query and rollout. Uniform and σ_g are normally query-level signals. They can also be applied at the rollout level, which gives a small lift over their query-level counterparts. The larger effect is the priority signal itself. Rollout-level $|A_i|$ leads on both avg and pass, with the widest gap on pass at $+3.19$ pp over the next-best σ_g . This indicates that rollout-level buffer entries alone are only a precondition. The gain comes from a signal that varies within a group. $|A_i|$ provides such variation by construction, while σ_g is group-constant and does not. Rollout-level $|A_i|$ and rollout-level uniform draw the same number of replay rollouts from a buffer that admits only nonzero-advantage rollouts, so the $+0.39$ pp avg and $+3.39$ pp pass gap between them reflects which rollouts are selected rather than how many informative ones are added. Replay-free infor-

Method	AES per Qwen3-Base scale		
	0.6B	1.7B	4B
<i>Against untrained base model</i>			
GRPO	-0.146	+0.474	-0.067
Ours	+0.058	+0.718	+0.585
<i>Against no-replay GRPO baseline</i>			
uniform	+0.016	+0.008	-0.001
σ_g	-0.245	+0.046	-0.075
Ours	+0.177	+0.221	+0.292

Table 5: Token-based AES at three Qwen3-Base scales. The top block is referenced to the untrained Base init and the bottom block to the no-replay GRPO baseline. Reference values are listed in Appendix D. Naive baselines use pool composition with query-level priority, and **bold** marks the best AES per scale.

mative sampling such as dynamic sampling and oversample-and-filter raises the share of nonzero-advantage groups by generating more rollouts, so it pays on generation while replay pays on the learner. Holding the number of informative rollouts fixed, as this control does, therefore separates prioritization from that filtering within a fixed generation budget. We adopt rollout-level $|A_i|$.

6 Response-Length and Efficiency Analysis

The effect of replay in this setting is not limited to accuracy. We additionally evaluate each post-trained model with AES (Luo et al., 2025a). AES was originally proposed to score reasoning-model efficiency as the joint trade-off between accuracy and generation length.

6.1 Accuracy Efficiency Score (AES)

AES combines normalized changes in accuracy and generation length into a single efficiency value,

$$\text{AES} = \Delta L + \beta \Delta \text{Acc}. \quad (4)$$

The length term $\Delta L = (L_{\text{ref}} - L)/L_{\text{ref}}$ is positive when the method generates shorter responses than the reference. The accuracy term $\Delta \text{Acc} = (A - A_{\text{ref}})/A_{\text{ref}}$ is positive when it is more accurate. A larger AES is therefore better on both axes. The weight is $\beta = 3$ when $\Delta \text{Acc} \geq 0$ and $\beta = 5$ otherwise. This asymmetry penalizes accuracy drops more than it rewards gains. Because both terms are measured against the reference, the same method scores differently under different references. Table 5 reports two such references. The top block uses the untrained base model and measures the absolute efficiency of post-training. The bottom block

Scale	$\mu_{\checkmark}/\mu_{\times}$		Δ (%)
	GRPO	Ours	
Qwen3-0.6B	0.139	0.185	+33.6
Qwen3-1.7B	0.224	0.277	+23.4
Qwen3-4B	0.119	0.147	+25.2

Table 6: Ratio of mean response length on correct rollouts to mean response length on incorrect rollouts, pooled across the five-benchmark set. Lengths are pooled over rollouts, and Δ is the relative change of that ratio from GRPO to ours. A positive Δ means correct rollouts grew longer relative to incorrect ones.

uses the no-replay GRPO baseline and measures the incremental efficiency of replay over no-replay. GRPO itself sits at zero in the bottom block by construction. The two blocks therefore answer distinct questions and should not be compared directly across blocks. Under this metric our method is the highest at every scale under both references.

Absolute Efficiency (Base Reference) GRPO’s AES against Base is negative at 0.6B and 4B, where its accuracy gain over Base is offset by length inflation, and our method is above it at all three scales.

Incremental Efficiency (GRPO Reference) Under the GRPO reference, rollout-level $|A_i|$ leads at every scale and is the only condition positive at all three. The two naive baselines instead straddle zero. Replay therefore improves token efficiency only under the $|A_i|$ priority. The 0.6B margin is the smallest of the three, in line with the higher share of all-wrong groups at 0.6B (Appendix B), which leaves fewer mixed-reward groups from which high- $|A_i|$ candidates can be drawn. The full 12-configuration AES ranking at 1.7B across the α , τ_{max} , and r sweep is in Appendix K.

Where the Efficiency Gain Comes From The AES gain is a redistribution of length rather than a uniform reduction in generation tokens. Correct rollouts became longer and incorrect ones shorter, so the ratio of mean response length on correct rollouts to that on incorrect ones rises at every scale against the no-replay GRPO baseline, by +33.6, +23.4, and +25.2% (Table 6, with the two sides in Appendix D). One reading of this follows from what $|A_i|$ selects. A correct rollout reaches high $|A_i|$ only in a mostly wrong group, and an incorrect rollout only in a mostly correct one. Replay therefore reinforces the rare correct solutions of hard prompts, which are the long ones, and suppresses the rare failures of easy prompts. Appendix I mea-

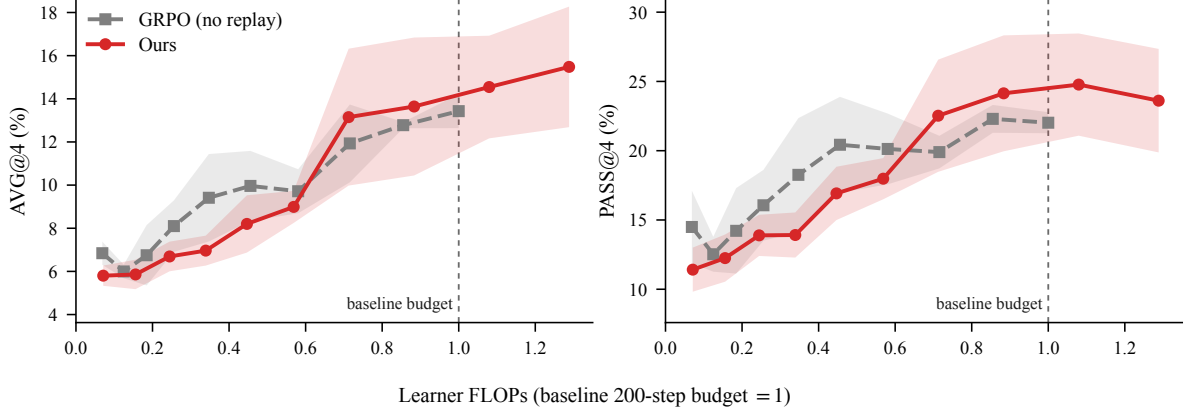


Figure 2: Accuracy against learner FLOPs at Qwen3-4B-Base over three training seeds, normalized so each seed’s baseline 200-step budget is 1. The left panel is $\text{AVG}@K$ and the right is $\text{PASS}@K$. Points are evaluated every 20 training steps on the four contest benchmarks (AIME25, AIME26, HMMT-F25, HMMT-F26) at $k = 4$, and bands are ± 1 standard deviation across seeds.

sures this composition on the untrained base models, where the correct side is the minority in 99% of the 4B groups that carry the highest priority. We offer this as a conjecture rather than a tested mechanism, since we do not vary the two directions independently.

6.2 Compute-Matched Efficiency

We additionally analyze the 4B runs at matched compute. Learner FLOPs are proportional to the trained tokens of each step, so we compare our method and the no-replay baseline at equal trained tokens rather than at equal training steps. Figure 2 plots both conditions over three training seeds. At the baseline’s full budget, marked in the figure, our method leads by +0.75 pp on $\text{AVG}@K$ and +2.50 pp on $\text{PASS}@K$. It continues to its own 200 steps at 1.29 times that budget and ends +2.05 pp and +1.59 pp above the baseline. Both conditions draw the same fresh rollouts per step and our method trains more tokens per step, so at the marked budget it has seen only 86% of the baseline’s training prompts. Leading on less new data supports the premise behind the buffer, that a single pass leaves learning signal in the rollouts GRPO discards. Appendices G and H report the FLOPs convention, the step-wise curve, and the measured overhead.

7 Conclusion

We presented an experience replay buffer for GRPO with two main findings. First, the naive replay baselines we test give no consistent gain over

GRPO across three Qwen3-Base scales of 0.6B, 1.7B, and 4B on DeepScaleR-Preview, and at 4B they fall below it on the five-benchmark average. Reuse without an appropriate priority can therefore cost accuracy rather than add to it.

Second, our method combines fresh-anchored composition with age eviction and rollout-level $|A_i|$ priority. It yields a positive margin at every scale, reaching +1.66 pp on the five-benchmark average at 4B. On the contest benchmarks at 4B the pass-side gains reach +13.3 pp on AIME26 and +11.1 pp on HMMT-F25.

Per-axis ablations support each design choice. Age eviction bounds policy lag through a step-level cap, and fresh-anchored composition contributes the on-policy anchor. At 1.7B, rollout-level $|A_i|$ outperforms both uniform and σ_g on both metrics. Each baseline is tested at query and rollout level. The widest pass gap is +3.19 pp over rollout-level σ_g . Fresh-anchored composition further adds +0.58 pp avg and +5.27 pp pass over pool composition.

Beyond accuracy, our method is the only condition with a positive token-budget AES margin over GRPO at every scale. A correctness-conditioned length decomposition in Section 6.1 shows the model lengthens correct rollouts relative to incorrect ones, raising their length ratio by +23 to +34% across scales. These results indicate that rollout-level replay modifies the trained model’s response distribution, not only its accuracy.

Limitations

Our study is confined to math reasoning with binary verifiable rewards. We use DAPO-style GRPO with asymmetric clipping as the underlying trainer. We do not study replay under non-verifiable rewards such as learned reward models, nor under other RL algorithms such as PPO or REINFORCE. Whether the rollout-level $|A_i|$ priority and age eviction generalize to those settings is an open question.

Our update is biased with respect to the on-policy GRPO objective, and Section 4 states the three sources. Replayed advantages stay frozen at their birth-step values, the prioritized draw is deliberately left uncorrected because an importance-sampling weight would cancel the reweighting toward high- $|A_i|$ rollouts that motivates replay in the first place, and replayed rollouts carry a policy lag of up to $\tau_{\max}$ steps. Age eviction, the fresh anchor, and clipping bound these departures rather than remove them. The $\alpha = 0$ setting of Table 4 is the in-expectation endpoint of a fully corrected draw, but we do not run the weighted estimator itself, we do not refresh replayed advantages, and we do not characterize the bias analytically. Our evidence for the trade-off is empirical. Refreshing replayed advantages online, in the spirit of online difficulty filtering (Bae et al., 2026; Gao et al., 2025), is left to future work.

We do not compare against replay-free informative sampling. Dynamic sampling and oversample-and-filter fill an update with nonzero-advantage groups by generating additional rollouts, so their cost falls on generation, whereas replay adds no generation and pays on the learner instead. A fair comparison therefore has to be matched on total compute rather than on training steps, which we leave to future work. The uniform-replay control of Table 4 isolates prioritization from informative filtering within replay, but it does not establish whether the same accuracy is reachable by spending the extra budget on generation.

No run is reported beyond global step 200, so longer-horizon dynamics are not characterized. These include failure modes that may emerge only after substantially more updates and the eventual ceiling of the recycling benefit. Similarly, the body scaling sweep is limited to Qwen3-Base at 0.6B, 1.7B, and 4B trained on the DeepScaleR-Preview prompt set, and the compute-matched comparison of Section 6.2 is measured at 4B alone. Transfer to larger model scales, to multi-turn or agentic tasks,

and to non-math domains is not validated.

Cross-scale trends within the tested range are not perfectly monotonic, and the range is bounded on both sides. The 0.6B model in particular shows smaller margins than 1.7B and 4B on several metrics. A plausible contributing factor is a scale-dependent shift in prompt difficulty, as suggested by Appendix B. Only 34% of generated groups are mixed-reward at 0.6B during training and this rises to 49% at 4B. The rare-correct rollouts that high- $|A_i|$ priority targets are therefore scarce in absolute count at smaller scales. At the other end, a single 8B run under a reduced training batch shows no gain on the five-benchmark average against its own no-replay baseline. Appendix I reports that run together with a start-of-training solve-count distribution that has moved away from the region $|A_i|$ targets, which is the same misalignment seen from the opposite direction. RLVR is sensitive to the alignment between prompt difficulty and current policy capability. Replay may amplify this sensitivity because the buffer keeps recycling high- $|A_i|$ candidates from the same difficulty distribution rather than rebalancing it. We do not study a remedy in this work. Difficulty-adaptive prompt selection or buffer composition is a direction for future work.

All evaluation is automatic via a programmatic answer checker. We do not run human studies on the quality of the responses produced by our method. The asymmetric length pattern in Section 6 therefore reflects a verifiable-reward view of efficiency rather than a perceived-quality view.

The ablation tables use a single training seed of 42 per run. Evaluation uses a single primary $k = 8$ sampling wave at seed 42 for every benchmark. The $k = 8$ wave is kept as the MATH-500 budget, and a second $k = 8$ wave at seed 43 is added for AIME and HMMT to reach $k = 16$. The same two seeds are used for every condition at every scale. Any seed-dependent fluctuation in the per-example samples is therefore shared across the comparisons in the body tables. The detailed seed policy and the per-wave sample_id ranges are in Appendix A (Training and Sampling Seeds).

Ethical Considerations

The contribution of this work is a sample-efficient training procedure for GRPO on verifiable-reward math reasoning. The work is foundational and not tied to a specific deployed application. Any effi-

ciency improvement in RLVR post-training is dual-use in the sense that it lowers the cost of producing reasoning models. The method could therefore be applied for both beneficial and harmful purposes by downstream developers. Misuse scenarios specific to math reasoning include automated solving of homework or examination problem sets, which constitutes academic dishonesty. They also include large-scale generation of synthetic competition-style solutions, which could contaminate future training corpora if posted publicly without provenance markers. We release no new model or dataset artifact with this work, which limits direct deployment risk. See Appendix A for the artifacts and licenses used. The compute footprint of the full sweep is roughly 2,500 H100-hours, as reported in Appendix A. This is modest relative to large-scale post-training pipelines but compounds at scale. We report it transparently to support reproducibility and discourage redundant re-runs. We do not study replay under non-verifiable rewards. No claim is made about transfer to settings with learned reward models or human-feedback signals.

Acknowledgements

This work was supported by the BK21 FOUR program of the Education and Research Program for Future ICT Pioneers, Seoul National University in 2026, Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO.RS-2021-II211343, Artificial Intelligence Graduate School Program (Seoul National University)], the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A3B1077720), and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [NO. RS-2022-II220959]. This research was also conducted as part of the Sovereign AI Foundation Model Project (Data Track), organized by the Ministry of Science and ICT (MSIT) and supported by the National Information Society Agency (NIA), S. Korea (Grant No. 2026-AIData-WII01). This work was also supported by the National AI Project of the Ministry of Science and ICT and the AIIS.

References

Marcin Andrychowicz, Filip Wolski, Alex Ray, Jonas Schneider, Rachel Fong, Peter Welinder, Bob Mc-

Grew, Josh Tobin, Pieter Abbeel, and Wojciech Zaremba. 2017. [Hindsight experience replay](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.

Charles Arnal, Vivien Cabannes, Taco Cohen, Julia Kempe, and Rémi Munos. 2026. [Efficient RL training for LLMs with experience replay](#). *arXiv preprint arXiv:2604.08706*.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, Jeongyeon Nam, and Donghyun Kwak. 2026. [Online difficulty filtering for reasoning oriented reinforcement learning](#). In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 700–719, Rabat, Morocco. Association for Computational Linguistics.

Andrei Baroian and Rutger Berger. 2026. [Prompt replay: Speeding up GRPO with on-policy reuse of high-signal prompts](#). *arXiv preprint arXiv:2603.21177*.

Brian R. Bartoldson, Siddarth Venkatraman, James Diefenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kaikhura. 2025. [Trajectory balance with asynchrony: Decoupling exploration and learning for fast, scalable LLM post-training](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.

Marc G. Bellemare, Will Dabney, and Rémi Munos. 2017. [A distributional perspective on reinforcement learning](#). In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, volume 70 of *Proceedings of Machine Learning Research*, pages 449–458. PMLR.

Albin Cassirer, Gabriel Barth-Maron, Eugene Brevdo, Sabela Ramos, Toby Boyd, Thibault Sottiaux, and Manuel Kroiss. 2021. [Reverb: A framework for experience replay](#). *arXiv preprint arXiv:2102.04736*.

Xinyue Chen, Che Wang, Zijian Zhou, and Keith W. Ross. 2021. [Randomized ensembled double Q-learning: Learning fast without a model](#). In *International Conference on Learning Representations (ICLR)*.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. 2023. [RAFT: Reward rAnked Fine-Tuning for generative foundation model alignment](#). *Transactions on Machine Learning Research*.

Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Yingwei Ma, Jue Chen, Binhua Li, Zhi Jin, Fei Huang, Yongbin Li, and Ge Li. 2025. [RL-PLUS: Countering capability boundary collapse of LLMs in reinforcement learning with hybrid-policy optimization](#). *arXiv preprint arXiv:2508.00222*.

- Pierluca D’Oro, Max Schwarzer, Evgenii Nikishin, Pierre-Luc Bacon, Marc G. Bellemare, and Aaron Courville. 2023. [Sample-efficient reinforcement learning by breaking the replay ratio barrier](#). In *International Conference on Learning Representations (ICLR)*.
- Mehdi Fatemi. 2026. [Prioritized replay for RL post-training](#). *arXiv preprint arXiv:2601.02648*.
- William Fedus, Prajit Ramachandran, Rishabh Agarwal, Yoshua Bengio, Hugo Larochelle, Mark Rowland, and Will Dabney. 2020. [Revisiting fundamentals of experience replay](#). In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, volume 119 of *Proceedings of Machine Learning Research*, pages 3061–3071. PMLR.
- Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jia-shu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. 2025. [AReAL: A large-scale asynchronous reinforcement learning system for language reasoning](#). *arXiv preprint arXiv:2505.24298*.
- Scott Fujimoto, David Meger, and Doina Precup. 2020. [An equivalence between loss functions and non-uniform sampling in experience replay](#). In *Advances in Neural Information Processing Systems 33 (NeurIPS)*, Online. Curran Associates, Inc.
- Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, and 1 others. 2024. [A framework for few-shot language model evaluation](#). GitHub repository.
- Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. 2025. [Prompt curriculum learning for efficient LLM post-training](#). *arXiv preprint arXiv:2510.01135*.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. [DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning](#). *Nature*, 645(8081):633–638.
- Zhenyu Han, Ansheng You, Haibo Wang, Kui Luo, Guang Yang, Wenqi Shi, Menglong Chen, Sicheng Zhang, Zeshun Lan, Chunshi Deng, Huazhong Ji, Wenjie Liu, Yu Huang, Yixiang Zhang, Chenyi Pan, Jing Wang, Xin Huang, Chunsheng Li, and Jianping Wu. 2025. [AsyncFlow: An asynchronous streaming RL framework for efficient LLM post-training](#). *arXiv preprint arXiv:2507.01663*.
- Harvard–MIT Mathematics Tournament. 2025. [HMMT february 2025](#). Harvard–MIT Mathematics Tournament, February 2025.
- Harvard–MIT Mathematics Tournament. 2026. [HMMT february 2026](#). Harvard–MIT Mathematics Tournament, February 2026.
- Matteo Hessel, Joseph Modayil, Hado van Hasselt, Tom Schaul, Georg Ostrovski, Will Dabney, Dan Horgan, Bilal Piot, Mohammad Azar, and David Silver. 2018. [Rainbow: Combining improvements in deep reinforcement learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pages 3215–3222. AAAI Press.
- Zhang-Wei Hong, Tao Chen, Yen-Chen Lin, Joni Pajarinen, and Pulkit Agrawal. 2022. [Topological experience replay](#). In *International Conference on Learning Representations (ICLR)*.
- 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, Wei Shen, Jason Klein Liu, Zilin Zhu, Weixun Wang, Songlin Jiang, Haoran Wang, Hao Chen, Bin Chen, Weikai Fang, Xianyu, Yu Cao, Haotian Xu, and Yiming Liu. 2024. [OpenRLHF: An easy-to-use, scalable and high-performance RLHF framework](#). *arXiv preprint arXiv:2405.11143*.
- Hugging Face. 2025. [Math-verify: Mathematical answer verification](#). GitHub repository.
- David Isele and Akansel Cosgun. 2018. [Selective experience replay for lifelong learning](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32. AAAI Press.
- Guochao Jiang, Wenfeng Feng, Guofeng Quan, Chuzhan Hao, Yuewei Zhang, Guohua Liu, and Hao Wang. 2025. [VCRL: Variance-based curriculum reinforcement learning for large language models](#). *arXiv preprint arXiv:2509.19803*.
- Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. [Scaling laws for neural language models](#). *arXiv preprint arXiv:2001.08361*.
- Steven Kapturowski, Georg Ostrovski, John Quan, Rémi Munos, and Will Dabney. 2019. [Recurrent experience replay in distributed reinforcement learning](#). In *International Conference on Learning Representations (ICLR)*.
- Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. [Efficient memory management for large language model serving with PagedAttention](#). In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*.
- Thanh-Long V. Le, Myeongho Jeon, Kim Vu, Viet Dac Lai, and Eunho Yang. 2026. [No prompt left behind](#).

- Exploiting zero-variance prompts in LLM reinforcement learning via entropy-guided advantage shaping. In *The Fourteenth International Conference on Learning Representations (ICLR)*.
- Nicolas Le Roux, Marc G. Bellemare, Jonathan Lebesold, Arnaud Bergeron, Joshua Greaves, Alex Fr chet, Carolyne Pelletier, Eric Thibodeau-Laufer, S ndor Toth, and Sam Work. 2025. **Tapered off-policy REINFORCE: Stable and efficient reinforcement learning for large language models**. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Long Li, Zhijian Zhou, Tianyi Wang, Weidi Xu, Zuming Huang, Wei Chu, Zhe Wang, Shirui Pan, Chao Qu, and Yuan Qi. 2026. **DyJR: Preserving diversity in reinforcement learning with verifiable rewards via dynamic Jensen-Shannon replay**. *arXiv preprint arXiv:2603.16157*.
- Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. 2025. **RePO: Replay-enhanced policy optimization**. *arXiv preprint arXiv:2506.09340*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. **Let’s verify step by step**. In *The Twelfth International Conference on Learning Representations (ICLR)*.
- Timothy P. Lillicrap, Jonathan J. Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. **Continuous control with deep reinforcement learning**. In *International Conference on Learning Representations (ICLR)*.
- Long-Ji Lin. 1992. **Self-improving reactive agents based on reinforcement learning, planning and teaching**. *Machine Learning*, 8(3–4):293–321.
- Zhihang Lin, Mingbao Lin, Yuan Xie, and Rongrong Ji. 2025. **CPPO: Accelerating the training of group relative policy optimization-based reasoning models**. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Yitao Liu, Chenglei Si, Karthik R. Narasimhan, and Shunyu Yao. 2025a. **Contextual experience replay for self-improvement of language agents**. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14179–14198, Vienna, Austria. Association for Computational Linguistics.
- Zichuan Liu, Jinyu Wang, Lei Song, and Jiang Bian. 2025b. **Sample-efficient LLM optimization with reset replay**. *arXiv preprint arXiv:2508.06412*.
- Haotian Luo, Li Shen, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Naiqiang Tan, Xiaochun Cao, and Dacheng Tao. 2025a. **O1-Pruner: Length-harmonizing fine-tuning for O1-like reasoning pruning**. *arXiv preprint arXiv:2501.12570*.
- Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. 2025b. **DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL**. Notion Blog. <https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2>.
- Weiyu Ma, Yongcheng Zeng, Yan Song, Xinyu Cui, Jian Zhao, Xuhui Liu, and Mohamed Elhoseiny. 2026. **Freshness-aware prioritized experience replay for LLM/VLM reinforcement learning**. *arXiv preprint arXiv:2604.16918*.
- Hanyi Mao, Qianjia Xiao, Lei Pang, and Haixiao Liu. 2025. **Clip your sequences fairly: Enforcing length fairness for sequence-level RL**. *arXiv preprint arXiv:2509.09177*.
- Mathematical Association of America. 2025. **AIME 2025 problems**. American Invitational Mathematics Examination, February 2025.
- Mathematical Association of America. 2026. **AIME 2026 problems**. American Invitational Mathematics Examination, February 2026.
- MiniMax. 2025. **MiniMax-M1: Scaling test-time compute efficiently with lightning attention**. *arXiv preprint arXiv:2506.13585*.
- Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharmashan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. **Human-level control through deep reinforcement learning**. *Nature*, 518(7540):529–533.
- Evgenii Nikishin, Max Schwarzer, Pierluca D’Oro, Pierre-Luc Bacon, and Aaron Courville. 2022. **The primacy bias in deep reinforcement learning**. In *Proceedings of the 39th International Conference on Machine Learning (ICML)*, volume 162 of *Proceedings of Machine Learning Research*, pages 16828–16847. PMLR.
- Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. 2025. **Asynchronous RLHF: Faster and more efficient off-policy RL for language models**. In *The Thirteenth International Conference on Learning Representations (ICLR)*.
- Guido Novati and Petros Koumoutsakos. 2019. **Remember and forget for experience replay**. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97 of *Proceedings of Machine Learning Research*, pages 4851–4860. PMLR.
- Noam Razin, Hattie Zhou, Omid Saremi, Vimal Thilak, Arwen Bradley, Preetum Nakkiran, Joshua M.

- Susskind, and Etai Littwin. 2024. [Vanishing gradients in reinforcement finetuning of language models](#). In *The Twelfth International Conference on Learning Representations (ICLR)*.
- David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy P. Lillicrap, and Greg Wayne. 2019. [Experience replay for continual learning](#). In *Advances in Neural Information Processing Systems 32 (NeurIPS)*, Vancouver, Canada. Curran Associates, Inc.
- Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver. 2016. [Prioritized experience replay](#). In *International Conference on Learning Representations (ICLR)*.
- 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, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. [HybridFlow: A flexible and efficient RLHF framework](#). *arXiv preprint arXiv:2409.19256*.
- Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. [Efficient reinforcement finetuning via adaptive curriculum learning](#). *arXiv preprint arXiv:2504.05520*.
- Avi Singh, John D. Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J. Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron Parisi, Abhishek Kumar, Alex Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Elsayed, Hanie Sedghi, and 22 others. 2024. [Beyond human data: Scaling self-training for problem-solving with language models](#). *Transactions on Machine Learning Research*.
- Samarth Sinha, Jiaming Song, Animesh Garg, and Stefano Ermon. 2022. [Experience replay with likelihood-free importance weights](#). In *Proceedings of the 4th Annual Learning for Dynamics and Control Conference (L4DC)*, volume 168 of *Proceedings of Machine Learning Research*, pages 110–123. PMLR.
- Yifan Sun, Jingyan Shen, Yibin Wang, Tianyu Chen, Zhendong Wang, Mingyuan Zhou, and Huan Zhang. 2025. [Improving data efficiency for LLM reinforcement fine-tuning through difficulty-targeted online data selection and rollout replay](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- Hado van Hasselt, Yotam Doron, Florian Strub, Matteo Hessel, Nicolas Sonnerat, and Joseph Modayil. 2018. [Deep reinforcement learning and the deadly triad](#). *arXiv preprint arXiv:1812.02648*.
- Xu Wan, Yansheng Wang, Wenqi Huang, and Mingyang Sun. 2026. [Buffer matters: Unleashing the power of off-policy reinforcement learning in large language model reasoning](#). *arXiv preprint arXiv:2602.20722*. Introduces “Batch Adaptation Policy Optimization (BAPO)”; the BAPO acronym here is distinct from the BAPO (Balanced Policy Optimization with Adaptive Clipping) of Xi et al. (2025).
- Chen Wang, Lai Wei, Yanzhi Zhang, Chenyang Shao, Zedong Dan, Weiran Huang, Yuzhi Zhang, and Yue Wang. 2025a. [EFrame: Deeper reasoning via exploration-filter-replay reinforcement learning framework](#). *arXiv preprint arXiv:2506.22200*.
- Zhenting Wang, Guofeng Cui, Yu-Jhe Li, Kun Wan, and Wentian Zhao. 2025b. [DUMP: Automated distribution-level curriculum learning for RL-based LLM post-training](#). *arXiv preprint arXiv:2504.09710*.
- Bo Wu, Sid Wang, Yunhao Tang, Jia Ding, Eryk Helenowski, Liang Tan, Tengyu Xu, Tushar Gowda, Zhengxing Chen, Chen Zhu, Xiaocheng Tang, Yundi Qian, Beibei Zhu, and Rui Hou. 2025. [LlamaRL: A distributed asynchronous reinforcement learning framework for efficient large-scale LLM training](#). *arXiv preprint arXiv:2505.24034*.
- Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Zhihao Zhang, Honglin Guo, Xun Deng, Zhikai Lei, Miao Zheng, Guoteng Wang, Shuo Zhang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, and 2 others. 2025. [BAPO: Stabilizing off-policy reinforcement learning for LLMs via balanced policy optimization with adaptive clipping](#). *arXiv preprint arXiv:2510.18927*.
- Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, and Hanze Dong. 2025a. [A minimalist approach to LLM reasoning: from rejection sampling to reinforce](#). *arXiv preprint arXiv:2504.11343*.
- Wei Xiong, Chenlu Ye, Baohao Liao, Hanze Dong, Xinxing Xu, Christof Monz, Jiang Bian, Nan Jiang, and Tong Zhang. 2025b. [Reinforce-Ada: An adaptive sampling framework under non-linear RL objectives](#). *arXiv preprint arXiv:2510.04996*.
- Yixuan Even Xu, Yash Savani, Fei Fang, and J. Zico Kolter. 2025. [Not all rollouts are useful: Down-sampling rollouts in LLM reinforcement learning](#). *arXiv preprint arXiv:2504.13818*.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. 2025. [Learning to reason under off-policy guidance](#). In *Advances in Neural Information Processing Systems (NeurIPS)*.
- 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, and 41 others. 2025. [Qwen3 technical report](#). *arXiv preprint arXiv:2505.09388*.

- Jiarui Yao, Yifan Hao, Hanning Zhang, Hanze Dong, Wei Xiong, Nan Jiang, and Tong Zhang. 2025. [Optimizing chain-of-thought reasoners via gradient variance minimization in rejection sampling and RL](#). *arXiv preprint arXiv:2505.02391*.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gao-hong Liu, Juncai Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, and 17 others. 2025. [DAPO: An open-source LLM reinforcement learning system at scale](#). In *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*.
- Runzhe Zhan, Yafu Li, Zhi Wang, Xiaoye Qu, Dongrui Liu, Jing Shao, Derek F. Wong, and Yu Cheng. 2026. [ExGRPO: Learning to reason from experience](#). In *Proceedings of the International Conference on Learning Representations (ICLR)*.
- Hongzhi Zhang, Jia Fu, Jingyuan Zhang, Kai Fu, Qi Wang, Fuzheng Zhang, and Guorui Zhou. 2025a. [RLEP: Reinforcement learning with experience replay for LLM reasoning](#). *arXiv preprint arXiv:2507.07451*.
- Ruiqi Zhang, Daman Arora, Song Mei, and Andrea Zanette. 2025b. [SPEED-RL: Faster training of reasoning models via online curriculum learning](#). *arXiv preprint arXiv:2506.09016*.
- Shangdong Zhang and Richard S. Sutton. 2017. [A deeper look at experience replay](#). *arXiv preprint arXiv:1712.01275*.
- Yuheng Zhang, Wenlin Yao, Changlong Yu, Yao Liu, Qingyu Yin, Bing Yin, Hyokun Yun, and Lihong Li. 2025c. [Improving sampling efficiency in RLVR through adaptive rollout and response reuse](#). *arXiv preprint arXiv:2509.25808*.
- Rui Zhao and Volker Tresp. 2018. [Energy-based hindsight experience prioritization](#). In *Proceedings of the 2nd Conference on Robot Learning (CoRL)*, volume 87 of *Proceedings of Machine Learning Research*, pages 113–122. PMLR.
- Haizhong Zheng, Jiawei Zhao, and Beidi Chen. 2025a. [Prosperity before collapse: How far can off-policy RL reach with stale data on LLMs?](#) *arXiv preprint arXiv:2510.01161*.
- Haizhong Zheng, Yang Zhou, Brian R. Bartoldson, Bhavya Kailkhura, Fan Lai, Jiawei Zhao, and Beidi Chen. 2025b. [Act only when it pays: Efficient reinforcement learning for LLM reasoning via selective rollouts](#). In *Advances in Neural Information Processing Systems*, volume 38, pages 124321–124346. Curran Associates, Inc.
- Yinmin Zhong, Zili Zhang, Xiaoniu Song, Hanpeng Hu, Chao Jin, Bingyang Wu, Nuo Chen, Yukun Chen, Yu Zhou, Changyi Wan, Hongyu Zhou, Yimin Jiang, Yibo Zhu, and Daxin Jiang. 2025. [StreamRL: Scalable, heterogeneous, and elastic RL for LLMs with disaggregated stream generation](#). *arXiv preprint arXiv:2504.15930*.
- Haodong Zhu, Yangyang Ren, Yanjing Li, Mingbao Lin, Linlin Yang, Xuhui Liu, Xiantong Zhen, Haiguang Liu, and Baochang Zhang. 2026a. [Unbiased dynamic pruning for efficient group-based policy optimization](#). *arXiv preprint arXiv:2603.04135*.
- Xinxin Zhu, Ying He, Haowen Hou, Ruichong Zhang, Nianbo Zeng, Yulin Peng, Jiongfang Fang, and F. Richard Yu. 2026b. [PSPO: Prompt-level prioritization and experience-weighted smoothing for efficient policy optimization](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 29186–29194. AAAI Press.

A Full Experimental Details

Artifacts and Licenses We use only publicly released artifacts. The base models are the Qwen3-Base family (Yang et al., 2025) at 0.6B, 1.7B, and 4B parameters, released under the Apache 2.0 license. The training prompts are from DeepScaleR-Preview (Luo et al., 2025b), released under the MIT license. Evaluation uses MATH-500 (Lightman et al., 2024) along with AIME25 (Mathematical Association of America, 2025), AIME26 (Mathematical Association of America, 2026), HMMT-F25 (Harvard-MIT Mathematics Tournament, 2025), and HMMT-F26 (Harvard-MIT Mathematics Tournament, 2026). The MATH-500 source problems are MIT-licensed via the upstream MATH benchmark and the PRM800K repository. The AIME and HMMT problems are publicly released by the Mathematical Association of America and the Harvard-MIT Mathematics Tournament, respectively. All prompts and responses are in English. None of these resources contain personal information about individuals. Our use is consistent with the research-use terms of the original releases. We release no new dataset or pretrained model artifact with this paper.

Benchmarks and Decoding We evaluate on the five math-reasoning benchmarks listed in the Artifacts and Licenses paragraph above. All five are evaluation-only. The sizes are 500 problems for MATH-500, 30 for AIME25, 30 for AIME26, 30 for HMMT-F25, and 33 for HMMT-F26. AIME and HMMT are reported per year, comparing AIME25 against AIME26 and HMMT-F25 against HMMT-F26 to show per-benchmark variance. For each prompt we draw samples at temperature 0.6 and top- p at 0.8. MATH-500 is evaluated at $k = 8$. AIME and HMMT are evaluated at $k = 16$ as two disjoint waves of 8. Per benchmark we report $\text{AVG}@K$ for mean accuracy and $\text{PASS}@K$ for any-correct rate. The ablation tables from Table 2 to Table 4 report a five-benchmark aggregate. This aggregate is the unweighted mean of MATH-500, AIME25, AIME26, HMMT-F25, and HMMT-F26. The main result in Table 1 shows the per-benchmark cells directly. The k -wave construction and the full seed policy are detailed in the Training and Sampling Seeds paragraph below.

Training Set and Hyperparameters The training set is DeepScaleR-Preview (Luo et al., 2025b), filtered to prompts of at most 1024 tokens, leav-

Setting	Value
<i>Common across scales</i>	
Train batch size	256
Group size G	8
Max prompt length	1024 tokens
Max response length	8192 tokens
Learning rate	1×10^{-6}
PPO mini-batch size	128
Clip $\epsilon_{\text{low}}, \epsilon_{\text{high}}, c_{\text{dual}}$	0.2, 0.28, 10.0
Advantage stabilizer ϵ_{σ} (Eq. 1)	10^{-6}
Optimizer	AdamW ($\beta_1=0.9, \beta_2=0.999$)
Weight decay	0.01
Gradient clip	1.0 (L2 norm)
LR warmup	none
KL penalty β	0 (kl_coef=0, use_kl_loss=False)
<i>Per-scale at 0.6B, 1.7B, and 4B</i>	
PPO micro-batch per GPU	4, 4, 2

Table 7: RL training hyperparameters. The shared block applies to all conditions and the per-scale row varies across 0.6B, 1.7B, and 4B. ϵ_{low} and ϵ_{high} are the DAPO clip-higher window from Section 3. c_{dual} is the verl dual-clip bound on negative-advantage ratios, and the KL term is fully disabled.

ing 40,306 prompts. Table 7 lists the RL training hyperparameters used across all conditions. The per-scale row reports the value that varies by model size. The tabulated results are all at global step 200, which is fixed across conditions to equalize the rollout budget. GRPO uses a group size of $G = 8$ rollouts per prompt at every scale. This is the per-prompt sampling count behind the $\{A_i\}$ computation in Section 3 and the G -rollout group in Algorithm 1.

Compute All training runs use NVIDIA H100 80GB SXM5 GPUs in a single-node configuration. The 0.6B and 1.7B runs use four H100s per run, and the 4B runs use eight. A single run to step 200 takes approximately 10, 7, and 14 wall-clock hours at 0.6B, 1.7B, and 4B. These correspond to roughly 40, 28, and 112 H100-hours per run. The 0.6B wall-clock exceeds the 1.7B wall-clock because the 0.6B rollouts are on average substantially longer than the 1.7B rollouts. The mean rollout lengths at step 200 are approximately 2,086 and 1,191 tokens respectively, as reported in Table 12. Rollout generation dominates per-step time at fixed GPU count. The full sweep reported in the paper covers our method, the naive baselines, and the ablations across the three scales, for 29 distinct configurations. It consumes roughly 1,200 H100-hours to step 200, or about 2,500 H100-hours including extended-training runs and the roughly 30% overhead from failed or restarted jobs. All runs use bfloat16 mixed precision and gradient checkpointing, with a rollout-side GPU memory utilization of 0.6.

Software Training uses the verl framework (Sheng et al., 2024) at v0.7.0.dev0, extended with custom modifications for the replay buffer described in this paper. Rollout generation during GRPO updates uses vLLM (Kwon et al., 2023) at v0.11.0 with bfloat16 precision, tensor-parallel size 1, a maximum model length of 9,216 tokens, and chunked prefill. The training-side runtime is PyTorch v2.8.0 on CUDA 12.8 with FlashAttention v2.8.1, FlashInfer v0.3.1, Hugging Face transformers v4.57.6, and Ray v2.51.1 for distributed orchestration. Evaluation uses vLLM v0.14.1 with Hugging Face transformers v5.0.0. The evaluation maximum response length is 8,192 tokens at 0.6B and 1.7B and 16,384 tokens at 4B. Sampling settings are described in the Benchmarks and Decoding paragraph above. Each prompt is rendered as a two-message chat with system content “You are a helpful assistant.” and user content “Problem : {question}\n\nPlease reason step by step, and put your final answer within \boxed{ }.”. The chat is applied via `tokenizer.apply_chat_template(..., add_generation_prompt=True)`. All scales use the tokenizer of the corresponding Qwen3-Base checkpoint.

Verifier Answers are extracted from the final `\boxed{...}` expression within the trailing 500 characters of the completion. This truncation bounds verifier latency without changing correctness when the final boxed answer is near the end of the response. Training-time reward verification uses a Minerva-style code-based comparator adapted from lm-evaluation-harness (Gao et al., 2024). The extracted answer is normalized through LaTeX cleanup, unit-word removal, fraction and square-root shorthand expansion, and digit-comma stripping. The normalized answer is then compared by exact string match to the similarly normalized ground truth, yielding a reward of ± 1 . Final evaluation grading uses math-verify (Hugging Face, 2025) at v0.9.0 with the default extractor and grader. Both the prediction and the ground truth are parsed with `math_verify.parse`, wrapping the operand in a LaTeX math environment if not already inside one. The answer is accepted when `math_verify.verify(gt, pred)` returns true. If either side fails to parse, we fall back to raw string equality.

Training and Sampling Seeds The cross-scale main comparison of Table 1 is run three times per

condition, with training seeds 42, 101, and 202. Its per-benchmark cells are the mean over those three runs, and its avg and pass columns add the standard deviation across them. The AES and length analyses of Section 6.1 and the matched-FLOPs curves of Section 6.2 use the same three runs. The ablations use training seed 42 alone. Evaluation sampling uses one primary $k = 8$ wave at seed 42 for every benchmark. Seed 42 is the legacy default of our evaluation harness. This wave is the sole source of MATH-500’s $k = 8$ result. For AIME and HMMT we additionally draw a second $k = 8$ wave at seed 43 and concatenate the two waves to obtain the reported $k = 16$ statistics. The two seeds yield disjoint `sample_id` ranges 0 through 7 and 8 through 15 that do not overlap. The same two seeds are used for every condition at every scale. The conditions are GRPO, the naive baselines, and our method. Any seed-dependent fluctuation in the per-example samples is therefore held constant across the comparisons in Tables 1 through 5. This is not a source of differential bias between methods.

B Effective Batch Size After Zero-Variance Filtering

Table 8 reports the per-step surviving fresh-rollout count after zero-variance filtering for the GRPO baseline at the three Qwen3-Base scales. The count is averaged over the first 200 training steps, which is the snapshot point used throughout the paper. Each step issues the same fresh-rollout budget $B_{\text{fresh}} = 2048$, which is the train batch of 256 prompts at $G = 8$ rollouts each and matches the B_{fresh} symbol in Algorithm 1. The fraction surviving the zero-variance filter varies by scale. The surviving rollouts are those whose group has mixed correct and incorrect rewards. Just over three-fifths of the smallest model’s batch falls in all-wrong groups, leaving a mixed-reward fraction of approximately 34% at 0.6B. The larger models hold 44% to 49% mixed-reward groups. The dropped fraction shifts toward all-correct groups as model scale increases.

A capacity-only buffer-eviction rule would therefore induce a scale-dependent staleness profile. Larger models fill the buffer faster and evict sooner, while smaller models do the opposite. Age eviction with horizon τ_{max} , as described in Section 4.1, removes this dependence by bounding worst-case rollout age independently of buffer fullness.

Scale	B_{fresh}	B'_{fresh}	All-pos (%)	All-neg (%)
Qwen3-0.6B	2048	692.0	4.75	61.48
Qwen3-1.7B	2048	901.6	8.57	47.42
Qwen3-4B	2048	999.2	21.09	30.10

Table 8: Effective batch size after zero-variance filtering for the GRPO baseline at three Qwen3-Base scales, averaged over training steps 0 to 200. B_{fresh} is the pre-filter rollout count from Algorithm 1. B'_{fresh} is the mean surviving count belonging to mixed-reward groups. “All-pos” and “All-neg” are the percentages of groups filtered out for being all-correct and all-wrong.

Config	avg	pass
GRPO (no replay)	15.75	27.64
Ours at $r = 0.5, \alpha = 0.5$	16.58	32.35
$r = 0.5, \alpha = 1.0$	15.96	27.39
$r = 1.0, \alpha = 0.5$	15.99	27.99
$r = 1.0, \alpha = 1.0$	16.53	30.67

Table 9: Replay-ratio and priority-strength sweep at Qwen3-1.7B-Base. avg and pass are five-benchmark aggregates under Table 1’s k convention. **Bold** marks the best.

C Replay-Ratio and Priority-Strength Sweep

Table 9 sweeps the replay ratio r and priority-strength exponent α at Qwen3-1.7B-Base. Our setting at $r = 0.5$ and $\alpha = 0.5$ leads on both metrics. It is essentially tied for best on $\text{AVG}@K$, within 0.05 pp of the next-best $r = 1.0, \alpha = 1.0$, and has a larger margin of +1.68 pp on $\text{PASS}@K$ over the next-best. We adopt the setting on the $\text{PASS}@K$ margin. $\text{PASS}@K$ rewards solving a problem on at least one of k samples, which is the axis our rollout-level $|A_i|$ priority targets by recycling the within-group minority rollouts that fresh batches see sparsely (Section 4.2).

D AES References and the Length Decomposition

Table 10 lists the reference values used in the AES computation of Table 5. Table 11 gives the two correctness-conditioned sides whose ratio Table 6 reports in Section 6.1. Both tables share the benchmark set, the k convention, and the decoding settings of the post-training evaluation in Section 5.1. For AES, the Base init values are the untrained Qwen3-Base checkpoints, and the GRPO values are the no-replay GRPO baseline at step 200.

Reference	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B	
	Acc (%)	L	Acc (%)	L	Acc (%)	L
Base init	9.52	2162	13.53	2007	17.36	2452
GRPO	11.18	3609	15.86	2090	27.77	7031

Table 10: AES reference values reporting five-benchmark aggregate accuracy in percent and per-rollout response length in Qwen3 tokens. These values serve as A_{ref} and L_{ref} in Table 5.

Scale	GRPO		Ours	
	$\mu(\text{len} \mid \checkmark)$	$\mu(\text{len} \mid \times)$	$\mu(\text{len} \mid \checkmark)$	$\mu(\text{len} \mid \times)$
Qwen3-0.6B	526	3820	661	3631
Qwen3-1.7B	507	2262	555	2009
Qwen3-4B	1047	8844	1110	7546

Table 11: Mean response length in tokens conditioned on correctness, pooled across the five-benchmark set. These are the two sides whose ratio Table 6 reports. That table forms the ratio within each run before averaging, so dividing the entries here does not reproduce its columns exactly.

E Rollout-Level $|A_i|$ vs. Query-Level σ_g in Binary RLVR

We compare the two priority signals that recur throughout the paper. These are the rollout-level absolute advantage $|A_i|$ and the query-level standard deviation σ_g . The comparison is in the regime of binary verifiable rewards $r_i \in \{-1, +1\}$ that our experiments use. Both quantities admit closed forms parameterized by the group’s correct count k . We drop the stabilizer ϵ_σ of Equation 1, which is smaller than σ_g by six orders of magnitude for any mixed group.

Proposition 1 (Closed-form $|A_i|$ and σ_g in binary RLVR). *Consider a group of G rollouts under a binary verifiable reward $r_i \in \{-1, +1\}$ with correct count k and $0 < k < G$. The group mean and standard deviation are*

$$\mu_g = \frac{2k}{G} - 1, \quad (5)$$

$$\sigma_g = 2\sqrt{\frac{k}{G}\left(1 - \frac{k}{G}\right)}, \quad (6)$$

and the group-relative advantage $A_i = (r_i - \mu_g)/\sigma_g$ takes exactly two values per group,

$$|A_i^{\text{correct}}| = \sqrt{\frac{G-k}{k}}, \quad (7)$$

$$|A_i^{\text{incorrect}}| = \sqrt{\frac{k}{G-k}}. \quad (8)$$

For any mixed group with $0 < k < G$, the side with fewer rollouts carries strictly the larger $|A_i|$. We call this side the minority.

Proof. With $r_i \in \{-1, +1\}$, the group statistics are

$$\begin{aligned}\mu_g &= \frac{k \cdot (+1) + (G-k) \cdot (-1)}{G} = \frac{2k}{G} - 1, \\ \sigma_g^2 &= \mathbb{E}[r^2] - \mu_g^2 = 1 - \left(\frac{2k}{G} - 1\right)^2 \\ &= 4 \frac{k}{G} \left(1 - \frac{k}{G}\right),\end{aligned}$$

yielding Equations 5 and 6. Substituting into $A_i = (r_i - \mu_g)/\sigma_g$,

$$\begin{aligned}A_i^{\text{correct}} &= \frac{(+1) - (2k/G - 1)}{\sigma_g} \\ &= \frac{2(G-k)/G}{2\sqrt{(k/G)(1-k/G)}} \\ &= \sqrt{\frac{G-k}{k}},\end{aligned}\tag{9}$$

$$\begin{aligned}A_i^{\text{incorrect}} &= \frac{(-1) - (2k/G - 1)}{\sigma_g} \\ &= \frac{-2k/G}{2\sqrt{(k/G)(1-k/G)}} \\ &= -\sqrt{\frac{k}{G-k}},\end{aligned}\tag{10}$$

yielding the absolute values stated. Their ratio is

$$\frac{|A_i^{\text{correct}}|}{|A_i^{\text{incorrect}}|} = \frac{G-k}{k},\tag{11}$$

which exceeds 1 iff $k < G - k$, that is, iff the correct rollouts are the minority. The smaller of $\{k, G - k\}$ therefore always indexes the side with the larger $|A_i|$. $\square$

Implication for Priority Design σ_g in Equation 6 depends only on the group’s correct count k . It is therefore a *single* value shared by all G rollouts of the same group. Using σ_g as a replay priority therefore promotes or demotes an entire group uniformly, with no within-group discrimination. In contrast, $|A_i|$ from Equations 7 and 8 varies *within* a group. The minority side always carries strictly the larger $|A_i|$ by the ratio in Equation 11. The minority rollouts of skewed groups are the rare-correct or rare-incorrect examples. They therefore receive a structurally higher priority under rollout-level $|A_i|$ than under query-level σ_g sampling. This is the formal basis for the rollout-level $|A_i|$ priority adopted in Section 4.2.

Beyond Binary Rewards Proposition 1 relies on $r_i \in \{-1, +1\}$, or equivalently any two-valued encoding. The closed forms in Equations 7 and 8 and the strict minority-side dominance no longer hold under continuous or partial-credit rewards. The broader *intuition* is that rollout-level $|A_i|$ tracks the rollouts carrying the most informative gradient

Method	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B	
	L	$\Delta L\%$	L	$\Delta L\%$	L	$\Delta L\%$
GRPO (no replay)	2086	n/a	1191	n/a	3443	n/a
Ours ($ A_i $ rollout)	1752	-16.0	1099	-7.7	3156	-8.3

Table 12: Training-time response length at step 200 for our method against the no-replay GRPO baseline across three Qwen3-Base scales. L is the wandb-reported response_length/mean and is the endpoint of Figure 3. $\Delta L\%$ is the change of our method relative to the per-scale GRPO baseline.

signal. This intuition is expected to extend to non-binary settings. The within-group ranking property exploited here would have to be restated.

F Cross-Scale Training-Time Response Length

We report the training-time response length of our method and the no-replay GRPO baseline over the first 200 training steps at the three Qwen3-Base scales. Figure 3 plots the wandb-logged mean response length over training. Table 12 gives the step-200 endpoint of each curve along with the relative change of our method against the per-scale GRPO baseline.

G FLOPs Accounting

A forward-backward update costs about $6N$ FLOPs per trained token and a forward-only pass about $2N$ per token, where N is the non-embedding parameter count (Kaplan et al., 2020). Both conditions run the same model at a given scale, so N cancels and every FLOPs ratio in Section 6.2 reduces to a ratio of token counts taken from the training logs. Matching learner FLOPs rather than total FLOPs is the conservative direction, because our method also generates fewer tokens than the baseline at equal learner FLOPs, so matching total FLOPs would move the comparison further in its favor.

Figure 4 replots the same evaluations against training steps, which is the view a reader looking for a conventional learning curve expects.

H System Overhead

Replay rollouts are appended as additional gradient-accumulation micro-steps rather than a larger per-device micro-batch, so Table 13 shows peak GPU memory unchanged or slightly lower at every scale, with the buffer held in driver host memory at a fixed 0.5 MiB per stored rollout. The added cost

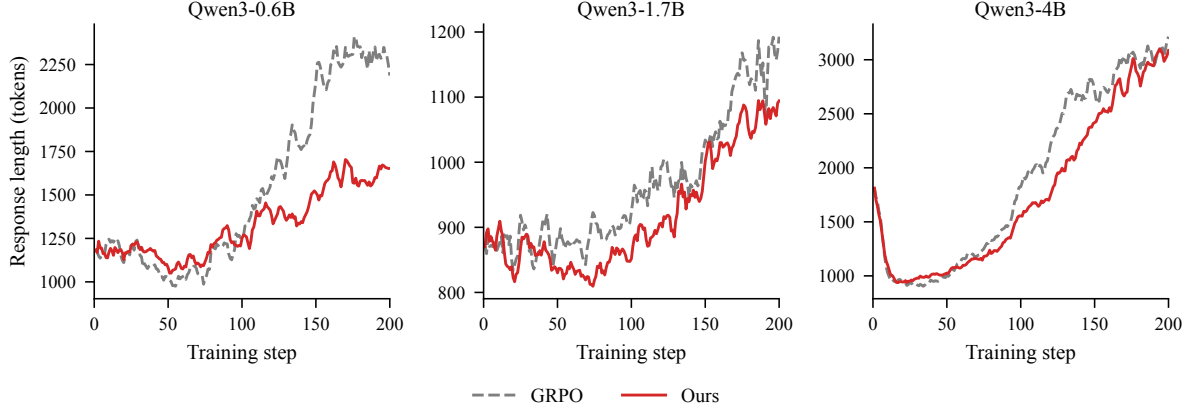


Figure 3: Training-time response length over the first 200 steps for our method and the no-replay GRPO baseline at three Qwen3-Base scales. The curves are plotted as 5-step rolling means of response_length/mean. The step-200 endpoint of each curve is the value tabulated in Table 12.

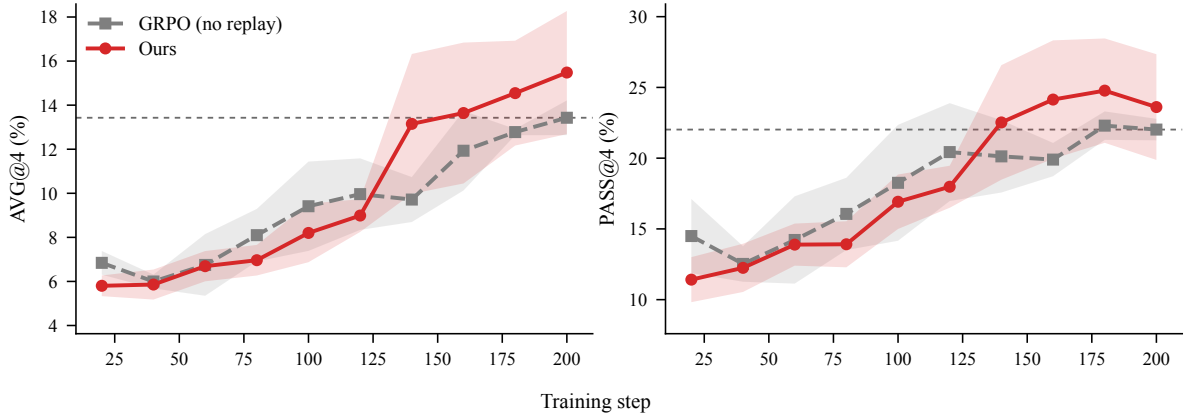


Figure 4: The points of Figure 2 against training steps rather than learner FLOPs, at Qwen3-4B-Base. The dashed horizontal line marks the baseline’s converged accuracy, which our method reaches at step 151 on $\text{AVG}@K$ and step 138 on $\text{PASS}@K$.

Metric	Qwen3-0.6B		Qwen3-1.7B		Qwen3-4B	
	GRPO	Ours	GRPO	Ours	GRPO	Ours
Step time (s)	191.0	165.8	106.6	114.7	193.9	241.4
update stage (s)	16.3	26.2	19.8	31.2	40.0	93.0
Peak GPU memory (GB)	85.2	84.3	94.5	93.5	68.6	68.6
Buffer peak occupancy	n/a	7,840	n/a	9,568	n/a	12,344
Buffer host RAM (GiB)	n/a	3.8	n/a	4.7	n/a	6.0

Table 13: Learner overhead of replay against the no-replay GRPO baseline, averaged over training steps 1 to 200 from the training logs. Hardware configurations differ across scales, so conditions should be compared within a scale rather than across scales.

falls on the update stage, and per-step wall time changes by -13% , $+8\%$, and $+24\%$ at 0.6B, 1.7B, and 4B, since generation time falls as our method produces shorter responses.

I The 8B Result and the Start-of-Training Solve-Count Distribution

Table 14 reports the solve-count distribution of the four base models before any training. A group whose minority side holds m of the 8 rollouts gives each of those rollouts $|A_i| = \sqrt{(8-m)/m}$, so this distribution fixes the distribution of priorities the buffer draws from. At 4B the correct side is the rare one in 99% of the groups carrying the highest priority. From 0.6B to 4B the mass leaving $k = 0$ lands at $k \leq 3$, where that is the case, and the region grows from 21.2% to 40.6% of prompts. From 4B to 8B it does not: the region is flat at 39.3% while $k \geq 4$ more than doubles from 9.8% to 22.1%.

Table 15 reports a Qwen3-8B-Base run against its own no-replay baseline, where the five-

	Qwen3-0.6B	Qwen3-1.7B	Qwen3-4B	Qwen3-8B
$k = 0$	74.24	63.30	49.54	38.00
$k = 1$	12.14	15.28	18.96	17.48
$k = 2$	5.74	8.20	12.70	12.48
$k = 3$	3.30	5.62	8.98	9.38
$k = 4$	2.24	3.46	6.10	7.92
$k = 5$	1.54	2.24	2.64	6.28
$k = 6$	0.68	1.48	0.90	5.12
$k = 7$	0.10	0.40	0.18	2.74
$k = 8$	0.02	0.02	0.00	0.60
$\mathbb{E}[k]$	0.55	0.85	1.16	1.88
Mixed, $1 \leq k \leq 7$	25.7	36.7	50.5	61.4
of which $k \leq 3$	21.2	29.1	40.6	39.3
of which $k \geq 4$	4.6	7.6	9.8	22.1

Table 14: Solve-count distribution of the four base models before any training, as the percentage of 5,000 training prompts whose 8 rollouts contain k correct answers, measured under the training sampling settings. Mixed groups are those the zero-variance filter keeps. Within them, $k \leq 3$ are the groups whose minority side is correct, which is where the highest $|A_i|$ priorities sit, and $k \geq 4$ are the rest. Each bin has a standard error of at most 0.7 pp.

	MATH-500	AIME25	AIME26	HMMT-F25	HMMT-F26	avg	pass
GRPO	86.45	17.50	19.17	12.08	14.02	29.84	44.45
Ours	86.62	19.17	19.17	11.25	13.64	29.97	43.83

Table 15: Qwen3-8B-Base at global step 200 against its own no-replay GRPO baseline on the five-benchmark set. Per-benchmark cells and the avg column are $\text{AVG}@K$, and the pass column is $\text{PASS}@K$, both at $k = 8$ on every benchmark. The run uses a train batch of 128 and a PPO mini-batch of 64 rather than the 256 and 128 of the body results, and it uses a single training seed. Its pass column is therefore not comparable to Table 1, which uses $k = 16$ on the contest benchmarks. All other training and decoding settings match Appendix A.

benchmark average differs by +0.13 pp against +1.66 pp at 4B. That run uses a train batch of 128 and a PPO mini-batch of 64 rather than 256 and 128, is evaluated with $k = 8$ on all five benchmarks rather than $k = 16$ on the contest sets, and uses a single training seed. We can only conjecture that the stronger 8B base model shifts the solve-count distribution away from the region $|A_i|$ targets and that this contributes to the absent margin. One run under a different batch does not establish it.

J Importance-Ratio Clipping over Training

Figure 5 shows the per-step importance-ratio clipping fraction over the first 200 training steps for the four age caps swept in Table 2. The curves separate fresh-side and replay-side clipping. The four age caps share a fresh-side profile that is essentially $\tau_{\max}$ -invariant. On the left panel, all four series fluctuate within roughly 0.06%. Replay-side clipping grows monotonically with the age cap and accumulates over training. On the right panel, $\tau_{\max}=60$ ends near 0.9%, roughly three times the $\tau_{\max}=1$ value at the same step. This asymmetry

is the dynamic counterpart to the policy-lag reading of Section 5.3.1. What grows with $\tau_{\max}$ is specifically the replay-side clipping rate, while the on-policy fresh side is unaffected.

K Full AES Ranking (12 Configurations)

Table 16 reports a 12-configuration 1.7B AES sweep against the no-replay GRPO baseline. The sweep covers three axes all under fresh-anchored composition. These are priority-strength $\alpha \in \{0, 0.5, 1.0\}$, max-age $\tau_{\max} \in \{1, 10, 30, 60\}$, and replay-ratio $r \in \{0.5, 1.0\}$. The pool-composition naive baselines of Table 5 are reported there and are not duplicated here. Our proposed ($|A_i|, \alpha=0.5, \tau_{\max}=10$) leads the ranking at +0.266. This is ahead of the next-best ($|A_i|, \alpha=0.5, \tau_{\max}=1$) at +0.248 and every other tested configuration. The tested configurations include alternative $(\alpha, \tau_{\max}, r)$ choices of the same $|A_i|$ priority and the uniform and σ_g priority baselines. The sweep uses a single training seed of 42, so the 1.7B Ours row here at +0.266 is the seed-42 value of the corresponding cell in Table 5, which averages three seeds. Both use the same

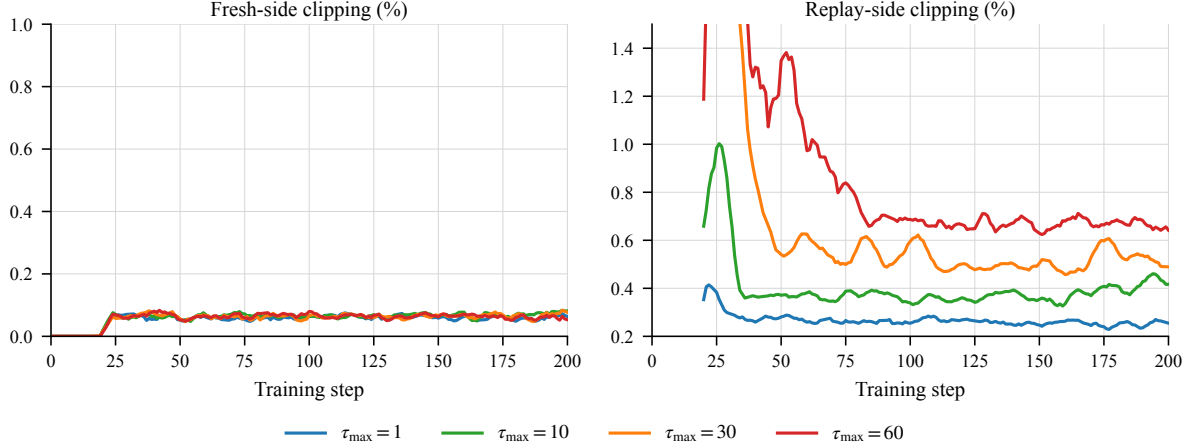


Figure 5: Importance-ratio clipping rates over training at Qwen3-1.7B-Base for the four age caps of Table 2. The curves are plotted as 5-step rolling means on the fresh side in the left panel and the replay side in the right panel. Replay-side clipping grows monotonically with the age cap while fresh-side clipping stays $\tau_{\max}$ -invariant. This provides the dynamic counterpart to Table 2’s step-200 means.

five-benchmark cache and aggregation reported in Appendix D, but the reference here is the seed-42 GRPO run at 15.75 accuracy and 2174 tokens rather than the three-seed mean listed there.

L Extended Related Work

Section 2 positions our method against its closest neighbors along three lines. The three lines are classical experience replay (Mnih et al., 2015; Schaul et al., 2016), replay schemes for LLM RL (Zhan et al., 2026; Li et al., 2025; Zhang et al., 2025a; Wan et al., 2026; Fatemi, 2026; Arnal et al., 2026; Ma et al., 2026), and priority signal design (Lin et al., 2025; Zhu et al., 2026a; Xu et al., 2025; Yu et al., 2025; Jiang et al., 2025; Bae et al., 2026; Gao et al., 2025). This appendix expands on the classical and priority strands, then summarizes where our method sits in the overall design space.

L.1 Classical and Deep-RL Experience Replay

Experience replay originated with Lin (1992), which stored past transitions and re-presented them to a connectionist Q-learner. Deep replay developed with DQN (Mnih et al., 2015) and was extended to continuous control by DDPG (Lillicrap et al., 2016). PER (Schaul et al., 2016) introduced priority as an explicit component. It samples proportional to $|\delta_i|^\alpha$ for TD error δ_i with importance-sampling corrections. The α knob is the same one we ablate in Appendix C. The max-priority-on-insertion convention of Section 3 is inherited

unchanged. Rainbow (Hessel et al., 2018) identified PER as one of the higher-impact DQN components. Distributional value learning (Bellemare et al., 2017) is orthogonal in that it reshapes the target rather than the sampling distribution, much as GRPO already adopts a distributional view at the group level.

A second classical line addresses storage and eviction. HER (Andrychowicz et al., 2017) relabels failed trajectories with achieved goals. Energy-based hindsight prioritization (Zhao and Tresp, 2018) combines PER-style sampling with HER-style relabeling at the trajectory level, paralleling our move from transitions to rollouts. LAP (Fujimoto et al., 2020) proves an equivalence between PER sampling and implicit loss reweighting and removes the need for importance-sampling correction. The analogous correction is unnecessary here because $|A_i|$ in GRPO is itself the policy-gradient coefficient rather than a Bellman-target proxy. ReF-ER (Novati and Koumoutsakos, 2019) drops gradient contributions whose ratio falls outside a trust region. LFIW (Sinha et al., 2022) replaces importance-sampling ratios with a likelihood-free density-ratio estimator. Both express the principle that a buffer implicitly defines a neighborhood of stale behavior policies. Our age eviction with horizon $\tau_{\max}$ is a coarser combinatorial alternative that bounds the worst-case ratio under bounded per-step drift without estimating ratios or densities. Selective replay (Isele and Cosgun, 2018) and topological replay (Hong et al., 2022) address what to

Rank	Method	Acc. (%)	L (tokens)	ΔL (%)	ΔAcc (%)	AES
1	Ours ($ A_i $ rollout, default)	16.58	1938	+10.9	+5.2	+0.266
2	$ A_i $ rollout ($\tau_{\max}=1$)	16.49	1940	+10.8	+4.7	+0.248
3	$ A_i $ rollout ($r=1.0$)	15.99	1788	+17.8	+1.5	+0.224
4	$ A_i $ rollout ($\tau_{\max}=30$)	16.01	1805	+17.0	+1.6	+0.219
5	uniform rollout (default)	16.19	2121	+2.5	+2.8	+0.107
6	$ A_i $ rollout ($\alpha=1.0$)	15.96	2034	+6.5	+1.3	+0.105
7	uniform query (default)	15.94	2065	+5.0	+1.2	+0.087
8	σ_g rollout (default)	16.04	2143	+1.4	+1.8	+0.069
9	$ A_i $ rollout ($\tau_{\max}=60$)	15.28	1745	+19.7	-3.0	+0.047
10	uniform rollout ($\tau_{\max}=30$)	15.18	1698	+21.9	-3.6	+0.038
11	σ_g query (default)	15.43	2001	+8.0	-2.0	-0.022
12	uniform rollout ($\tau_{\max}=1$)	16.45	2584	-18.8	+4.4	-0.056

Table 16: Full token-based AES ranking of 12 1.7B configurations against the no-replay GRPO baseline at step 200 with Acc = 15.75% and $L = 2174$ tokens. All rows are under fresh-anchored composition with AES formulation and 5-component aggregation following Table 5. Default replay parameters are ($\alpha=0.5$, $\tau_{\max}=10$, $r=0.5$) for $|A_i|$ and σ_g rows and ($\alpha=0$, $\tau_{\max}=10$, $r=0.5$) for uniform rows. Parenthetical qualifiers list only non-default values, and **bold** marks the best.

store. We instead place selection in the priority signal.

A third strand ties most directly to our composition choice. Zhang and Sutton (2017) in “Combined Experience Replay” notes that buffer capacity is sensitive because small buffers correlate data and large ones make it stale. That work always includes the most recent transition in every minibatch. Fresh-anchored composition generalizes this from a single transition to a full GRPO batch. CLEAR (Rolnick et al., 2019) likewise mixes on-policy and replay updates each step with a behavioral-cloning regularizer. We rely on the GRPO surrogate clip in its place.

Modern deep RL has also shown that the replay ratio is an important hyperparameter for sample efficiency. REDQ (Chen et al., 2021), SR-SAC (D’Oro et al., 2023), and the primacy-bias analysis of Nikishin et al. (2022) stress the per-update axis. Fedus et al. (2020) characterizes the joint effect of capacity and replay ratio in value-based deep RL. LoRR (Liu et al., 2025b) ports periodic-reset ideas to LLM RL. In GRPO the per-rollout cost dominates the per-update cost by orders of magnitude, so the relevant ratio is gradient updates per generated rollout. LLM policies drift more per step than small actor networks (Fu et al., 2025; Zheng et al., 2025a). This is the failure mode the age cap is designed to prevent. For stability, the deadly-triad analysis of van Hasselt et al. (2018) identifies replay as a load on the bootstrapping side. GRPO does not bootstrap. The off-policy axis is still present whenever stored rollouts differ from the current policy, which is the same concern $\tau_{\max}$

bounds. Distributed replay systems such as Ape-X (Horgan et al., 2018), R2D2 (Kapturowski et al., 2019), and Reverb (Cassirer et al., 2021) treat actor-learner lag as a scaling concern. Their core abstractions of insertion-time priority, FIFO with optional eviction, and a configurable insert-to-sample ratio map onto our buffer directly.

L.2 Priority Signals and Sample Selection in GRPO

Priority signal design for GRPO inherits two open questions from the broader literature. The first is which scalar identifies high-signal data. The second is at what granularity that scalar is defined. The first question appears in the TD-error of PER (Schaul et al., 2016), the loss equivalence of LAP (Fujimoto et al., 2020), the density-ratio of LFIW (Sinha et al., 2022), and the gradient-norm analysis of Razin et al. (2024) that links low reward variance to vanishing gradients. In GRPO, $|A_i|$ is the policy-gradient coefficient magnitude itself. CPPO (Lin et al., 2025) and DPPO (Zhu et al., 2026a) prune low- $|A|$ rollouts on the fresh side. PODS (Xu et al., 2025) keeps within-step reward extremes via σ_g -based downsampling. We extend this $|A_i|$ lineage to the replay side by recycling individual rollouts with large absolute advantage after the single-pass update has discarded them.

The second question is granularity. It runs from the prompt level (Baroian and Berger, 2026; Gao et al., 2025; Jiang et al., 2025; Shi et al., 2025; Bae et al., 2026; Zheng et al., 2025b; Le et al., 2026; Zhang et al., 2025b; Wang et al., 2025b; Xiong et al., 2025a,b) to the query level used by

most GRPO replay schemes including ExGRPO and [Fatemi \(2026\)](#). Our method operates at the within-group rollout level. Within-group difficulty signals such as σ_g ([Jiang et al., 2025](#); [Fatemi, 2026](#)) are reasonable query-level proxies but do not resolve the within-group minority rollouts where the gradient magnitude concentrates. We argue this point in Section 4.2 and report results in Section 5.3.2. The prompt-level branch composes orthogonally with our buffer. A prompt selector chooses what to roll out at all, while ours chooses what to reuse after generation. CPPO, DPPO, and PODS are the fresh-side analogues that prune what enters the update, while we prioritize what re-enters it.

L.3 Positioning Summary

Across these strands, our method occupies a specific point in a multidimensional design space.

- **Storage unit.** Rollout in ours, against transition in [Schaul et al. \(2016\)](#), trajectory in [Andrychowicz et al. \(2017\)](#), and query or group in [Zhan et al. \(2026\)](#); [Li et al. \(2025\)](#); [Zhang et al. \(2025a\)](#); [Wan et al. \(2026\)](#).
- **Priority signal.** $|A_i|$, the policy-gradient coefficient magnitude itself, against TD error in [Schaul et al. \(2016\)](#), correctness in [Dong et al. \(2023\)](#); [Zhang et al. \(2025a\)](#); [Singh et al. \(2024\)](#); [Xiong et al. \(2025a\)](#), correctness times entropy in [Zhan et al. \(2026\)](#), difficulty or σ_g in [Jiang et al. \(2025\)](#); [Fatemi \(2026\)](#), and gradient-variance in [Yao et al. \(2025\)](#).
- **Composition.** Fresh-anchored concat, against pooled in [Mnih et al. \(2015\)](#); [Schaul et al. \(2016\)](#), mixed-batch in [Zhang et al. \(2025a\)](#); [Zhan et al. \(2026\)](#); [Li et al. \(2025\)](#); [Wan et al. \(2026\)](#); [Arnal et al. \(2026\)](#); [Yan et al. \(2025\)](#); [Dong et al. \(2025\)](#), and in-context replay in [Liu et al. \(2025a\)](#).
- **Staleness control.** Age eviction $\tau_{\max}$, against capacity-only in [Mnih et al. \(2015\)](#); [Schaul et al. \(2016\)](#), statistical clipping or importance-sampling bounds in [Le Roux et al. \(2025\)](#); [MiniMax \(2025\)](#); [Zheng et al. \(2025a\)](#); [Xi et al. \(2025\)](#); [Mao et al. \(2025\)](#), system-level queue depth in [Fu et al. \(2025\)](#); [Han et al. \(2025\)](#); [Hu et al. \(2024\)](#); [Zhong et al. \(2025\)](#); [Wu et al. \(2025\)](#); [Noukhovitch et al. \(2025\)](#), soft recency bias in [Bartoldson et al. \(2025\)](#), and reuse-count limit in [Zhang et al. \(2025c\)](#).

To our knowledge, no prior work occupies this combination of rollout-level $|A_i|$ priority with fresh-anchored composition under age eviction.

Per-axis ablations in Section 5.3 indicate that each choice contributes. Removing the age cap increases replay-side clipping as shown in Table 2. Removing fresh anchoring drops below GRPO at 4B as shown in Table 3. Reducing priority to query-level σ_g removes the within-group variation of the signal and costs the most on $\text{PASS}@K$ as shown in Table 4. The LLM-RL replay-buffer design space is tightly coupled, and the combination of choices matters more than any single axis.

M Use of AI Assistants

We used Claude Code from Anthropic as the primary AI assistant. It supported routine code edits to training, evaluation, and analysis scripts. It also provided grammar, phrasing, and editing assistance on the manuscript. No AI tool was used to generate research ideas, experimental design, hypotheses, or scientific claims. All results, analyses, and final wording were produced and verified by the authors. This use is consistent with the ACL policy on AI writing assistance.