Extended Abstract

Motivation Online reinforcement learning for language model reasoning samples a fresh group of rollouts for every prompt at every optimizer step and scores each rollout with a verifier. Because RLOO uses no value network, generating and verifying these fresh rollouts is the dominant training cost, and every step discards the rollouts it just paid for. This project asks how much of that cost can be removed by reusing recent rollouts across steps, and how much stale-policy bias the reuse introduces before the saving stops being worthwhile. The efficiency axis is the number of fresh verifier evaluations per run, one per newly generated response.

Method RLOO is normally an on-policy, critic-free policy-gradient method. For each prompt it samples a group of K responses and uses the leave-one-out mean reward of the others as each response's baseline, which removes the value network. This work adapts RLOO into a semi-off-policy method by storing each sampled group in a bounded-age replay buffer, tagged with its behavior log-probabilities and sampling step. Every update mixes fresh groups with reused groups from the buffer, and each reused response is corrected by a one-sided clipped importance weight $w_i = \min(c, \exp(\log \pi_\theta(y_i \mid x_i) - \log \mu(y_i \mid x_i)))$, where μ is the older behavior policy and π_θ is the current policy. The full group is kept intact so the leave-one-out baseline still holds, and no value network is added. Three knobs control the design. The replay ratio ρ is the number of reused groups per fresh group in a batch of fixed size, the maximum replay age L caps how stale a reused group may be, and the clip ceiling c bounds how much any single stale rollout can move the gradient. Replay buffers are standard off-policy, but adapting RLOO to replay whole leave-one-out groups under a clipped importance weight is the contribution.

Implementation The policy is Qwen2.5-0.5B, fine-tuned on Countdown 3-to-4 arithmetic, combining given numbers with the four operations to reach a target, with a deterministic equation checker as the verifier. Each RLOO update uses 128 prompt groups with $K = 8$ responses for 100 optimizer steps. The first sweep fixes $\rho = 1$ and varies $L \in \{1, 2, 3\}$ and $c \in \{1, 3, 10\}$. A second sweep fixes $L = 2$ and varies $\rho \in \{0.5, 1, 1.5, 2\}$ and $c \in \{1, 3, 10\}$ to isolate the replay-ratio axis, against a fresh-only $\rho = 0$ baseline. Canonical evaluation reports the mean over five seeded $K = 16$ repeats on 50 held-out prompts, where a prompt is solved at pass@k when any of its first k samples passes the verifier. A best-of-N probe extends sampling to $K = 64$ for inference-time scaling.

Results Replay at $\rho = 1$ cuts fresh verifier evaluations from 102.4K to 51.7K per run, about half, while several replay settings match or exceed the fresh-only pass@16 of 0.756. In the $L \times c$ sweep the replay age and the clip ceiling interact, no single clip is best across ages, yet the tight clip $c = 1$ gives the highest evaluation reward at every age. The $\rho \times c$ sweep shows the same interaction along the replay ratio. At the aggressive $\rho = 2$ setting, which uses 35.1K evaluations or about one third of the baseline, a loose clip $c = 10$ reaches pass@16 0.796 while a tight clip $c = 1$ at the same budget reaches 0.832. In the probe, the strongest $\rho = 1$ setting reaches the fresh-only $K = 64$ quality by $K = 16$, a fourfold inference-time cut, and the aggressive $c = 1$ setting reaches pass@64 0.844, tying the top large-K band.

Discussion The least expected result concerns the clip. A small ceiling does not starve the update of signal. It bounds the rare high-weight stale rollouts that would otherwise dominate the gradient, which raises the effective sample size and keeps many rollouts contributing. A loose clip suffices at low replay ratios, but aggressive replay requires a more conservative clip. Diversity adds a caution. Replay narrows answer-span diversity relative to fresh-only RLOO, yet the strongest settings are not the most diverse, so high pass@k depends on whether samples land on correct solution modes rather than on raw breadth.

Conclusion Age-limited, clip-bounded replay lets verifier-based RLOO reuse recent rollouts and match or exceed fresh-only high-k quality at substantially lower fresh-evaluation cost, while keeping the critic-free leave-one-out baseline and adding no value network. The recommendation is two-tiered. For balanced quality, $\rho = 1$, $L = 1$, $c = 1$ gives the highest reward and correction rate at half the fresh budget. For the low-budget frontier, $\rho = 2$, $L = 2$, $c = 1$ reaches the top of the $K = 64$ band at about one third of the fresh budget. The main lesson is that replay ratio, replay age, and clip ceiling interact, and more aggressive replay calls for a more conservative clip.

How Much Stale Rollout Reuse Can Verifier-Based RLOO Tolerate? A Semi-Off-Policy Replay Study on Countdown Reasoning

Amulya Parthasarathy
Department of Computer Science
Stanford University
amulyasp@stanford.edu

Abstract

Online reinforcement learning for language model reasoning samples a fresh group of rollouts for every prompt at every update and scores each rollout with a verifier, which makes verifier-scored generation the dominant cost of training. This work studies whether an importance-weighted RLOO objective can reuse recent rollouts from a bounded-age replay buffer, and how much stale-policy bias the reuse introduces. The extension converts on-policy RLOO into a semi-off-policy method through a replay buffer and a clipped importance weight, while keeping the critic-free leave-one-out baseline and adding no value network. Three settings control the tradeoff, the replay ratio ρ, the maximum replay age L, and the importance-weight clip ceiling c. On the Countdown arithmetic task with a Qwen2.5-0.5B policy, replay at $\rho = 1$ removes about half of the fresh verifier evaluations, from 102.4K to 51.7K per run, while several replay settings match or exceed the fresh-only pass@16 of 0.756. The replay age and the clip ceiling interact, and no single clip is best across ages. A tight clip of $c = 1$ gives the highest evaluation reward at every age at $\rho = 1$, consistent with a gradient-democratization effect that bounds the influence of rare high-weight stale rollouts. A best-of-N probe shows that the strongest replay setting reaches the fresh-only pass@k level at $K = 64$ by $K = 16$, a fourfold reduction in inference-time sampling. A more aggressive $\rho = 2$ setting reaches one third of the fresh budget, and its large-K quality depends on the clip, with the tightest clip recovering the top of the $\rho = 1$ band at $K = 64$.

1 Introduction

Reinforcement learning has become the standard final stage for aligning and sharpening large language models, from human-feedback methods (Ouyang et al., 2022) to verifier-based methods that replace a learned reward model with a programmatic checker. On reasoning tasks with a checkable answer, the reward is a deterministic indicator of correctness, and policy-gradient methods such as PPO (Schulman et al., 2017), GRPO (Shao et al., 2024), and RLOO (Ahmadian et al., 2024) optimize the policy against that signal. RLOO is attractive in this setting because it removes the value network entirely and replaces it with a leave-one-out baseline computed over the group of sampled responses, which lowers memory and implementation cost while remaining a strong optimizer for language model fine-tuning.

The default project for this course builds the pipeline of supervised fine-tuning, preference optimization, and on-policy RLOO on the Countdown arithmetic task. The milestone report established that supervised fine-tuning already produces a high pass@16 of about 0.76 with a low pass@1 of about 0.30, that preference optimization improves the ranking of the existing candidate set without enlarging it, and that on-policy RLOO lifts single-sample accuracy substantially while leaving the

high-k ceiling roughly unchanged. Figure 1 summarizes that pipeline. The present report takes the on-policy RLOO baseline as its starting point and studies an extension.

On-policy RLOO has a structural inefficiency. Every optimizer step samples a fresh group of K responses per prompt, scores each response with the verifier, applies one gradient update, and then discards the rollouts. Because generation and verification are the expensive operations, the fresh-rollout requirement dominates the cost of training. A natural question is whether recent rollouts can be reused across optimizer steps rather than discarded, which would reduce the number of fresh, verifier-scored rollouts per update. Reuse, however, makes the data off-policy, because a reused rollout was sampled by an older policy than the one being updated. This work measures how much of that staleness verifier-based RLOO can tolerate before the stale-policy bias outweighs the efficiency gain.

The extension stores rollout groups in a bounded-age replay buffer and mixes reused groups with fresh ones in each RLOO update. A clipped importance weight corrects for the drift between the policy that generated a reused rollout and the policy being updated, much like the ratio clipping in PPO (Schulman et al., 2017) and the clipped importance sampling used in off-policy actor-critic methods (Su et al., 2017). The result is a semi-off-policy variant of RLOO that keeps the leave-one-out baseline and adds no additional learned model. Three knobs are studied over a grid of experiments. This report makes three contributions.

- A semi-off-policy extension of RLOO that reuses recent rollouts through a bounded-age replay buffer and a one-sided clipped importance weight, while keeping the critic-free leave-one-out baseline.
- A controlled sweep over the replay ratio, the maximum replay age, and the importance-weight clip ceiling, evaluated with a seeded canonical protocol of five repeats at $K = 16$, that quantifies the efficiency-quality tradeoff of stale rollout reuse.
- An analysis of the sweeps that identifies a tight clip as the strongest setting for evaluation reward, shows that replay age and clipping interact and that the same interaction holds along the replay ratio so that more aggressive replay calls for a tighter clip, and reports a best-of-N probe in which the strongest replay setting reaches the fresh-only $K = 64$ quality at $K = 16$.

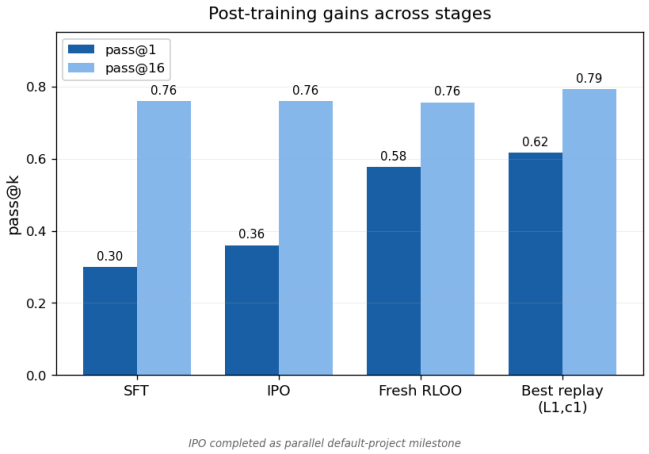


Figure 1: Post-training across the project pipeline. Single-sample accuracy (pass@1) rises from SFT through preference optimization and fresh RLOO, while the high-k capability ceiling (pass@16) stays roughly flat. SFT and IPO are milestone single evaluations. Fresh RLOO and the best replay setting are five-seed canonical means. The flat pass@16 with rising pass@1 is the expected signature that RLOO improves access to correct answers rather than enlarging the candidate set.

2 Related Work

Reinforcement learning for language models Human-feedback alignment popularized policy-gradient fine-tuning of language models with a learned reward model and a PPO optimizer (Ouyang et al., 2022; Schulman et al., 2017). On tasks with a checkable answer, the learned reward model can be replaced by a verifier, which removes reward-model error and gives a deterministic correctness signal. Group-relative methods such as GRPO (Shao et al., 2024) and REINFORCE-style methods such as RLOO (Ahmadian et al., 2024) estimate a baseline from a group of sampled responses rather than from a learned value network, which simplifies the optimizer. This work builds directly on RLOO and keeps its leave-one-out baseline.

Importance sampling and clipping Reusing data from an older policy requires a correction for the distribution mismatch. Importance sampling provides an unbiased correction by reweighting each sample by the ratio of its probability under the target and behavior policies, at the cost of high variance when the ratio is large. Clipped and truncated importance ratios are the standard variance-control device, and they appear both in the PPO clipped surrogate (Schulman et al., 2017) and in off-policy actor-critic methods with experience replay (Su et al., 2017). The clipped importance weight used here is a one-sided ceiling, which caps the upward influence of a reused rollout while leaving small ratios unchanged.

Off-policy and replay for language model RL Recent work has begun to relax the strictly on-policy assumption for language model RL. AGRO unifies on-policy and off-policy fine-tuning under a single objective based on generation consistency (Tang et al., 2025). Trajectory Balance with Asynchrony populates experience replay buffers with asynchronous off-policy actors and learns from them with a principled off-policy objective, reporting speed and quality gains and showing that recency-prioritizing sampling helps as generation is scaled (Bartoldson et al., 2025). Replay-Enhanced Policy Optimization adds a replay buffer of off-policy samples to GRPO and reports efficiency and quality gains on mathematical reasoning (Li et al., 2025). These methods target group-relative or actor-critic objectives, and applying a bounded-age replay buffer to RLOO, where the full K-response leave-one-out group must be kept intact, does not appear to have been studied. The present study is narrower in scope and complementary in aim. Rather than proposing a new objective, it takes the existing RLOO objective, adds a bounded-age replay buffer with a clipped importance weight, and measures how the efficiency-quality tradeoff moves as the replay ratio, the replay age, and the clip ceiling vary.

3 Method

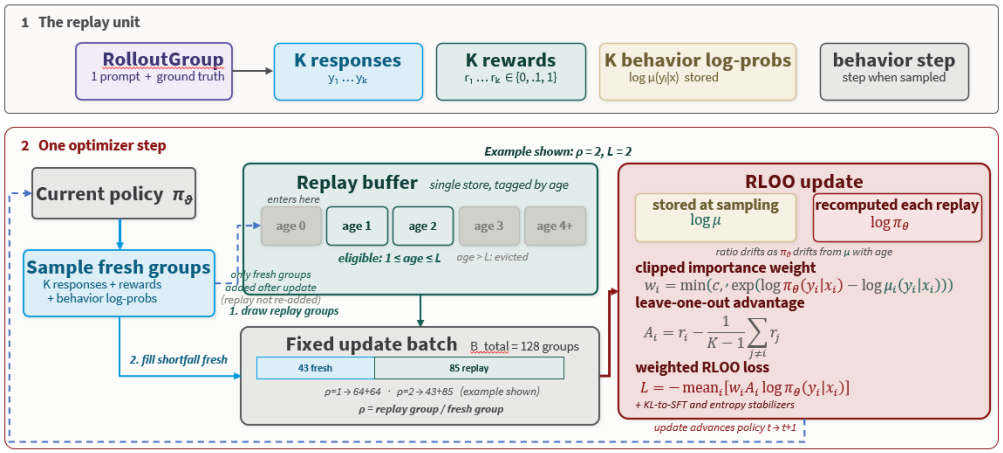


Figure 2: Replay-RLOO overview. Each optimizer step draws fresh rollout groups from the current policy and reuses recent groups from a bounded-age replay buffer. The behavior log-probabilities are stored when a group is sampled, the current log-probabilities are recomputed at each reuse, and a one-sided clipped importance weight corrects for the drift. The leave-one-out advantage is computed within each group of K responses, so no value network is required.

3.1 Background: RLOO

RLOO optimizes a policy π_θ against a scalar reward by sampling a group of K responses $y_1, \ldots, y_K$ for each prompt x and using the mean reward of the other responses in the group as a per-response baseline. With reward r_i for response y_i, the leave-one-out advantage is

$$A_i \;=\; r_i \;-\; \frac{1}{K-1}\sum_{j \neq i} r_j. \tag{1}$$

The baseline is computed from the group itself, so RLOO uses no learned value function. On Countdown the reward is the verifier outcome, which is one for a valid equation that uses each number once and reaches the target, 0.1 for a well-formed equation that misses the target, and zero when no valid equation is produced.

3.2 Replay buffer and the three knobs

The extension stores each sampled rollout group in a replay buffer, tagged with the optimizer step at which it was generated. A group is fresh at age zero, and its age increases by one after each subsequent step. At every update the batch is assembled from fresh groups drawn from the current policy and reused groups drawn from the buffer. Three knobs govern the reuse.

The replay ratio ρ is the number of reused groups per fresh group in a batch of fixed total size. At $\rho = 0$ the batch is entirely fresh, at $\rho = 1$ half of the batch is reused, and at $\rho = 2$ two thirds of the batch is reused. The maximum replay age L caps how stale a reused group may be. A group is eligible for reuse while its age is at least one and at most L, and a group older than L is evicted. The reused groups in each batch are sampled uniformly at random without replacement from the eligible groups. At $L = 1$ only the previous step contributes reusable groups, and at $L = 3$ groups up to three steps old remain eligible. The clip ceiling c bounds the importance weight that corrects for staleness, as described next.

3.3 Clipped importance weight

A reused group was generated by an older behavior policy μ, which is the policy at the step when the group was sampled, while the gradient updates the current policy π_θ. The sequence-level importance weight reweights each reused response to correct for this drift, and a one-sided ceiling caps it,

$$w_i \;=\; \min\Big(c,\; \exp\big(\log \pi_\theta(y_i \mid x_i) \;-\; \log \mu(y_i \mid x_i)\big)\Big). \tag{2}$$

The behavior log-probability $\log \mu(y_i \mid x_i)$ is stored when the group is first sampled, and the current log-probability $\log \pi_\theta(y_i \mid x_i)$ is recomputed at each reuse. Fresh groups have an importance weight of one by construction, since the behavior policy equals the current policy at age zero. The ceiling c caps the upward influence of a reused rollout. A small ceiling such as $c = 1$ clips many rollouts and bounds the contribution of any single stale rollout, while a very large ceiling such as $c = 1000$ leaves the raw ratio almost unclipped.

3.4 Objective

The replay-RLOO loss is the importance-weighted policy-gradient term over the assembled batch of B responses, regularized by a Kullback-Leibler penalty toward the frozen SFT reference and an entropy bonus,

$$\mathcal{L}(\theta) \;=\; -\frac{1}{B}\sum_{i=1}^{B} w_i\, A_i\, \log \pi_\theta(y_i \mid x_i) \;+\; \beta_{\mathrm{KL}}\, \mathrm{KL}[\pi_\theta \,\|\, \pi_{\mathrm{ref}}] \;-\; \beta_H\, H[\pi_\theta]\,. \tag{3}$$

The full group of K responses is kept intact for the leave-one-out baseline, and the importance weight is applied only to the policy-gradient term. Figure 2 shows the data flow for one optimizer step. The construction reuses recent rollouts and clips their influence in the same spirit as the PPO ratio clip (Schulman et al., 2017) and off-policy actor-critic replay (Su et al., 2017), while keeping the critic-free RLOO baseline. Appendix B gives the full buffer mechanics, the per-step order of operations, and the complete training and evaluation settings.

4 Experimental Setup

Task and model The task is Countdown 3-to-4 arithmetic, in which the model must combine three or four given numbers with the four basic operations to reach a target value. Correctness is judged by a deterministic equation checker that validates the arithmetic and the use of each number once. The policy is Qwen2.5-0.5B, initialized from a supervised fine-tuning checkpoint trained on the course dataset. Both the trainable policy and the frozen reference are initialized from that checkpoint.

Training configuration Each RLOO update samples $K = 8$ responses per prompt with a batch of 128 prompt groups and runs for 100 optimizer steps. Training uses a learning rate of 10^{-5}, a KL coefficient of 10^{-3}, an entropy coefficient of 10^{-3}, and rollout sampling at temperature 1.0 with top-p 1.0. The default clip ceiling for the on-policy baseline is $c = 10$, which is inert when $\rho = 0$ because all importance weights equal one. The replay buffer is swept over $L \in \{1, 2, 3\}$ and $c \in \{1, 3, 10\}$ at $\rho = 1$, giving a three by three experiment grid. A fresh-only $\rho = 0$ run is the baseline, and a more aggressive $\rho = 2$ run at $L = 2$ and $c = 10$ probes the budget frontier. A second sweep fixes the replay age at $L = 2$ and varies the replay ratio over $\rho \in \{0.5, 1, 1.5, 2\}$ and the clip ceiling over $c \in \{1, 3, 10\}$ to isolate the replay-ratio axis. Its $\rho = 1$ row reuses the $L = 2$ cells of the first sweep and its $\rho = 2$, $c = 10$ cell is the aggressive $\rho = 2$ run at the loose clip, so most of its boundary is shared with the runs above. A short $c = 1000$ run is used only as a near-unclipped stress probe and is not a full comparable run.

Evaluation protocol Functional correctness is reported with pass@k under two conventions. Both draw samples from the policy at temperature 0.6 with top-p 0.95 and top-k 20, lower than the temperature 1.0 used for the training rollouts. The canonical evaluation uses 50 held-out prompts and the ordered first-k convention, in which a prompt counts as solved at pass@k when any of its first k sampled responses passes the verifier. It reports the mean over five independent seeded repeats at $K = 16$, with 95 percent confidence-interval half-widths. A separate best-of-N probe extends sampling to $K = 64$ with 64 samples per prompt in each of five seeded repeats and reports the unbiased pass@k estimator

$$\text{pass@}k \;=\; 1 - \binom{n - c_{\text{correct}}}{k} \Big/ \binom{n}{k}, \tag{4}$$

where n is the number of sampled responses per prompt and c_{correct} is the number of correct responses. Each of the five seeded repeats draws $n = 64$ samples per prompt, and the estimator is computed within a repeat and then averaged across the repeats. The probe is read as a distinct inference-time scaling analysis rather than as a substitute for the canonical metric. The primary efficiency axis is the number of fresh verifier evaluations per 100-step run, with one evaluation per newly generated training response. Each fresh rollout incurs exactly one verifier evaluation, so the fresh-evaluation count equals the number of freshly generated rollouts. Appendix B derives the fresh-evaluation budget for each replay ratio.

Compute Training and evaluation ran on a single H100 GPU on Azure and on Modal, with a wall-clock cost of about 210 minutes per 100-step run.

5 Results

The fresh-only baseline reaches a canonical pass@16 of 0.756 and a pass@1 of 0.576 at 102.4K fresh verifier evaluations. The main result is that replay at $\rho = 1$ reaches comparable or higher pass@16 at 51.7K fresh evaluations, roughly half the baseline budget. Table 1 reports the pipeline and the headline replay settings, and the rest of this section examines the sweeps, with the full results tables in Appendix A.

Table 1: Performance across the pipeline and the key replay settings. SFT and IPO are milestone single evaluations. Fresh-only RLOO and the replay settings are five-seed canonical means at $K = 16$. The fresh-evaluation column is the verifier budget per 100-step run.

Setting	(ρ, L, c)	Fresh evals	pass@1	pass@16
SFT (milestone)	—	—	0.30	0.76
IPO (milestone)	—	—	0.36	0.76
Fresh-only RLOO	$(0, -, -)$	102.4K	0.576	0.756
Best balanced	$(1, 1, 1)$	51.7K	0.616	0.792
L=1 high-k	$(1, 1, 3)$	51.7K	0.612	0.816
L=2 high-k	$(1, 2, 10)$	51.7K	0.596	0.820
Strong $\rho = 1$ L=3 high-k	$(1, 3, 1)$	51.7K	0.584	0.820
L=3 late recovery	$(1, 3, 3)$	51.7K	0.620	0.784
Aggressive, c=1	$(2, 2, 1)$	35.1K	0.600	0.832
Aggressive, c=10	$(2, 2, 10)$	35.1K	0.560	0.796

5.1 Quantitative Evaluation

Sample-efficiency frontier Figure 3 plots canonical pass@16 against fresh verifier evaluations. The fresh-only baseline sits at the full budget of 102.4K with pass@16 0.756. Every $\rho = 1$ cell sits at 51.7K, and the spread of pass@16 across the design grid runs from 0.748 to 0.820. Several cells exceed the fresh-only baseline at half the budget, and the best $\rho = 1$ cells reach 0.820. The aggressive $\rho = 2$ setting at the loose clip $c = 10$ reaches 35.1K evaluations, about one third of the baseline budget, with a canonical pass@16 of 0.796, while a tight clip $c = 1$ at the same budget reaches 0.832 (Figure 5 and Table 3). On this canonical axis the fresh-only baseline is Pareto-dominated, since replay reaches equal or higher pass@16 at lower cost. The right panel of Figure 3 shows that this is a property of the clip rather than of the replay ratio alone, since only c changes between the two points at the same budget.

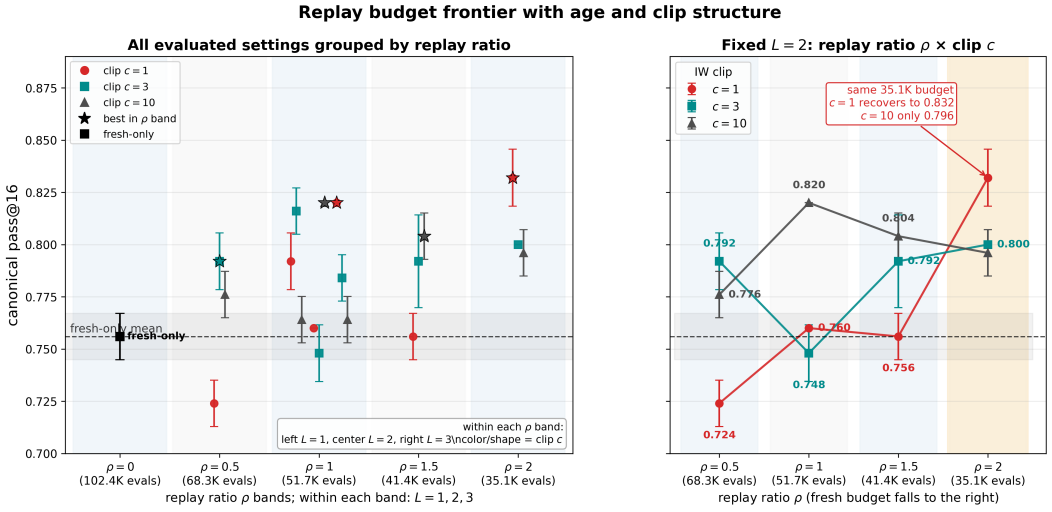


Figure 3: Replay-budget frontier with age and clip structure, canonical pass@16 over five seeded repeats with 95 percent confidence-interval half-widths. Left, all evaluated settings grouped by replay ratio ρ, where within each band the available subcolumns correspond to replay age $L = 1, 2, 3$ and the marker and color encode the clip ceiling c, with a star on the best cell in each band and the fresh-only baseline at $\rho = 0$. Right, the fixed replay age $L = 2$ sweep showing how the clip interacts with the replay ratio. At the aggressive $\rho = 2$ budget of 35.1K fresh evaluations, one third of the fresh-only budget, a tight clip recovers quality that a loose clip loses, with $c = 1$ reaching pass@16 0.832 while the same budget at $c = 10$ reaches 0.796.

The replay-age sweep Figure 4 shows the three by three experiment grid for pass@16, evaluation reward, and answer-span diversity at $\rho = 1$, where evaluation reward is the mean verifier score on the canonical set. Two patterns stand out. First, replay age and clipping interact, and there is no single best clip across ages. The best clip for pass@16 is $c = 3$ at $L = 1$, $c = 10$ at $L = 2$, and $c = 1$ at $L = 3$. Second, evaluation reward is highest at $c = 1$ for every replay age, with the largest reward of 0.661 at $L = 1$ and $c = 1$. The cell at $L = 1$ and $c = 1$ also has the highest correction rate, which makes it the best-balanced operating point on the reward axis even though its pass@16 of 0.792 is below the high-k leaders at 0.820.

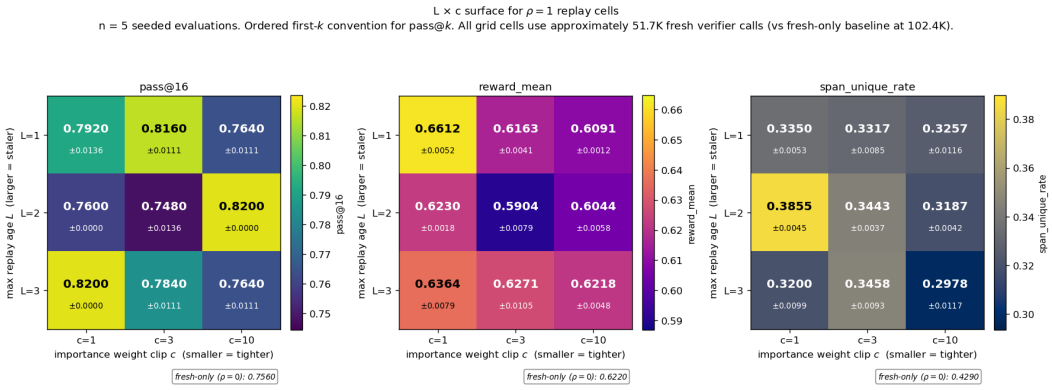
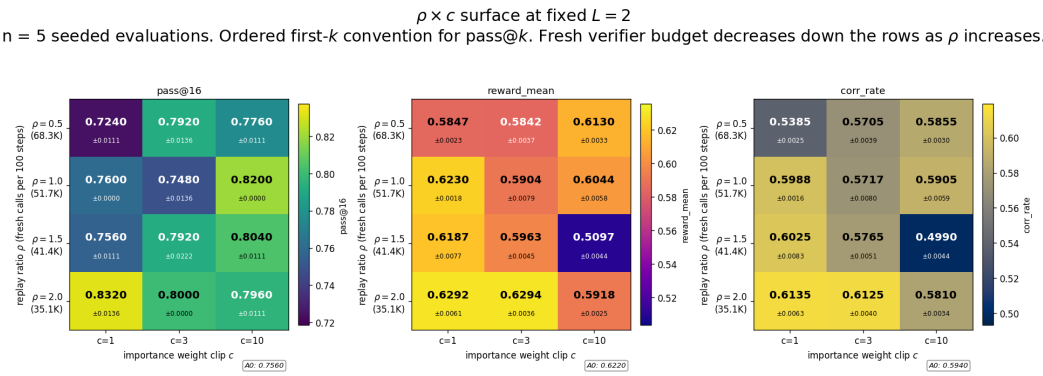


Figure 4: Replay age by clipping sweep at $\rho = 1$. All cells use about 51.7K fresh verifier evaluations. Rows vary the maximum replay age L and columns vary the importance-weight clip ceiling c. Values are five-seed means with 95 percent confidence-interval half-widths. The fresh-only baseline is off the grid because it has no replay, and it anchors each panel as a reference value. The best clip for pass@16 depends on the replay age, while the highest evaluation reward occurs at the tight clip $c = 1$ at every age.

The replay-ratio sweep Figure 5 fixes the replay age at $L = 2$ and varies the replay ratio and the clip ceiling, which isolates the interaction between how much replay enters each batch and how hard the importance weight is clipped. The replay ratio and the clip ceiling interact in the same direction the replay age and the clip ceiling do. At the lower ratios $\rho = 0.5$ and $\rho = 1$ the loose clip $c = 10$ stays competitive on reward and correction rate. At the higher ratios $\rho = 1.5$ and $\rho = 2$ the loose clip degrades reward, correction rate, and pass@1 while leaving pass@16 inside the noisy top band. The $\rho = 1.5$, $c = 10$ cell is the clearest case, with reward and correction rate near 0.510 and 0.499 while its pass@16 stays at 0.804 and its pass@1 falls to 0.456. At $\rho = 2$ the tight clips recover the average quality the loose clip loses. The $c = 3$ and $c = 1$ settings reach a reward near 0.629 and a correction rate near 0.613, against 0.592 and 0.581 at $c = 10$. The $c = 1$ setting also gives the highest observed canonical $K = 16$ pass@16 in the sweep at 0.832, but that value sits at the top of a tight band of pass@16 scores from 0.796 to 0.832 that fifty held-out prompts cannot reliably separate, so the robust signal is the reward and correction-rate recovery rather than the single pass@16 maximum. Put plainly, a more aggressive replay ratio requires a more conservative clip.

Inference-time scaling Figure 6 reports the best-of-N probe at K up to 64. The strongest $\rho = 1$ replay setting at $L = 3$ and $c = 1$ reaches a pass@k of 0.813 at $K = 16$, which matches the fresh-only baseline pass@k of 0.804 at $K = 64$, so the strongest $\rho = 1$ setting reaches the fresh-only $K = 64$ quality with a fourfold smaller inference-time sample budget. Among the aggressive $\rho = 2$ cells the clip ceiling sets the large-K behavior. The loose clip $c = 10$ is the weakest curve at every K and reaches 0.776 at $K = 64$. The tight clip $c = 1$ leads the aggressive cells at every K and reaches 0.844 at $K = 64$, which ties the strongest $\rho = 1$ cells while using one third of the fresh budget. The $c = 3$ setting falls between the two, above $c = 10$ through the middle of the curve but below the $\rho = 1$ band at large K. The aggressive frontier therefore trades quality for budget only under a loose clip, and the tight clip recovers the large-K scaling at one third of the fresh cost.



A0 ($\rho = 0$ fresh-only baseline, 102.4K fresh calls) anchors each panel as a reference value. Fresh verifier budget falls as ρ rises, from 68.3K at $\rho = 0.5$ to 35.1K at $\rho = 2.0$. The $\rho = 1.0$ row reproduces the $L = 2$ row of the L × c figure and is the cross-figure consistency anchor.

Figure 5: Replay ratio by clipping sweep at fixed replay age $L = 2$. Rows vary the replay ratio ρ and columns vary the importance-weight clip ceiling c. Panels report pass@16, evaluation reward, and correction rate as five-seed means with 95 percent confidence-interval half-widths. The $\rho = 1$ row coincides with the $L = 2$ row of Figure 4. At the higher replay ratios the loose clip $c = 10$ loses reward and correction rate and the tight clips recover it, which mirrors the age-by-clip interaction at fixed $\rho = 1$.

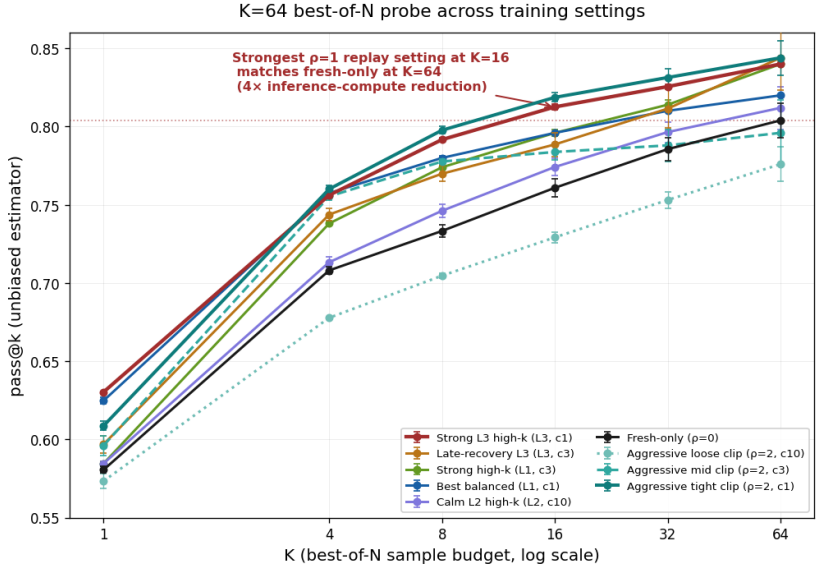


Figure 6: Best-of-N probe with the unbiased pass@k estimator at K up to 64. The strongest $\rho = 1$ replay setting reaches the fresh-only $K = 64$ quality level by $K = 16$, a fourfold reduction in inference-time sampling. The aggressive $\rho = 2$ setting splits by clip, with the tight clip $c = 1$ reaching the top band and the loose clip $c = 10$ scaling most weakly.

5.2 Qualitative Analysis

Training stability and the clip mechanism Figure 7 overlays the clip sweep at $\rho = 1$ and $L = 1$ across four training signals. The tight clip $c = 1$ binds on a large fraction of rollouts and, by bounding each reused rollout, raises the normalized effective sample size of the replay batch and produces the strongest reward trajectory. The default clip $c = 10$ binds only on sparse outliers and remains stable. The near-unclipped $c = 1000$ probe collapses the effective sample size, which is why it is treated as a stress probe rather than a comparable run. The Kullback-Leibler divergence to the

SFT reference grows smoothly and stays controlled across the useful settings. This dynamics view explains the counterintuitive result. A small ceiling does not remove signal, it democratizes the gradient by preventing a few rare high-weight stale rollouts from dominating the update, which keeps the policy moving coherently.

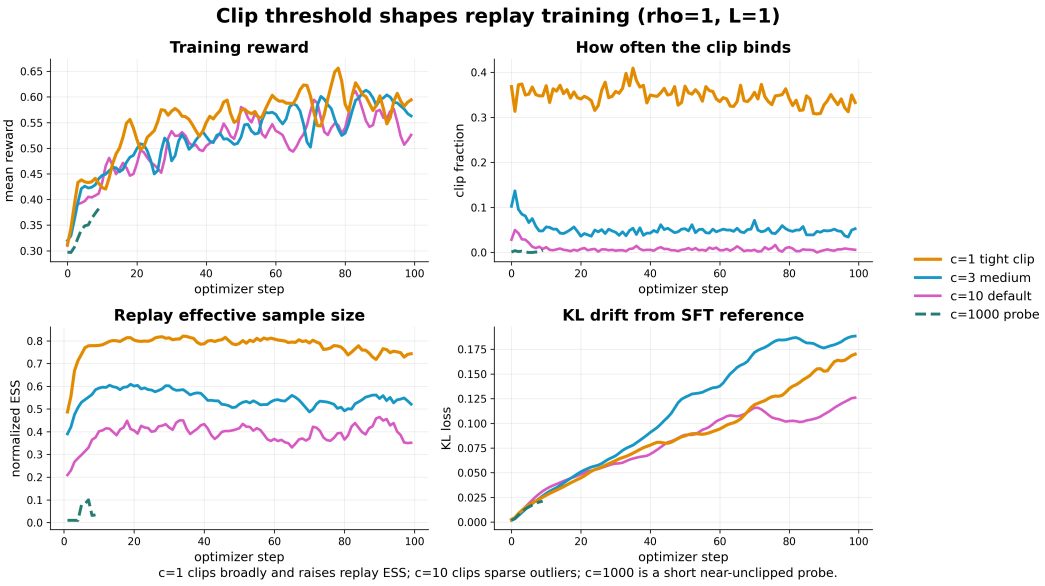


Figure 7: Clip sweep at fixed $\rho = 1$ and $L = 1$ across training reward, clip fraction, replay effective sample size, and Kullback-Leibler divergence to the SFT reference. The tight clip $c = 1$ binds broadly and holds the highest effective sample size, which gives the strongest reward trajectory. The near-unclipped $c = 1000$ run is a short stress probe.

Diversity is necessary but not sufficient Figure 8 reports answer-span diversity. The fresh-only baseline is the most diverse at a span-unique rate of 0.429, and every replay setting narrows diversity to the range 0.298 to 0.386. Reuse therefore concentrates the answer distribution, which is expected because the buffer repeats a finite set of past generations. Diversity alone does not predict pass@16, however. The two cells with the highest pass@16 of 0.820 are among the less diverse replay settings (Figure 4), while a more diverse replay cell does not reach the top of the pass@16 range. The useful effect of a setting is therefore not the raw breadth of its answers but whether its samples land on correct solution modes.

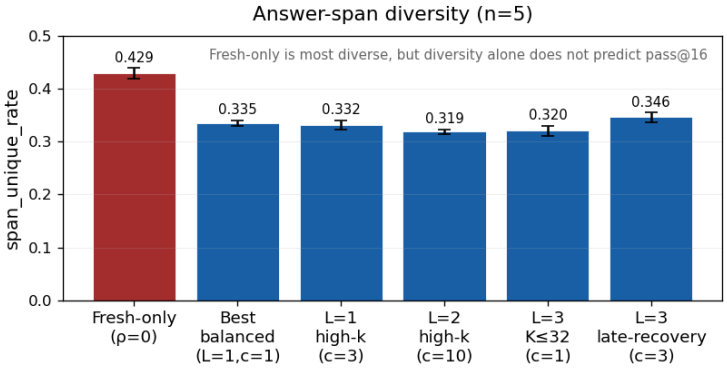


Figure 8: Answer-span diversity over five seeds. The fresh-only baseline is the most diverse. The strongest high-k replay settings are less diverse, which shows that diversity must land on correct solution modes rather than broaden the answer set.

6 Discussion

Verifier-based RLOO tolerates a substantial amount of stale rollout reuse, and that tolerance is a frontier rather than a fixed ceiling. The frontier is set jointly by the replay ratio, the replay age, and the clip ceiling. Across the full 100-step clipped sweeps with $c \in \{1, 3, 10\}$, no run collapsed outright. At $\rho = 1$, with half of each batch reused from recent groups, Replay-RLOO recovers the fresh-only pass@16 and several settings exceed it at about half the fresh verifier-evaluation budget. At $\rho = 2$ with $L = 2$, it reaches one third of that budget. The price of pushing reuse higher is that the clip must become more conservative. Quality degradation under a loose clip is clearest when reuse is high, especially at $\rho = 1.5, L = 2, c = 10$, where pass@1 falls to 0.456 while pass@16 stays high. Tight clipping recovers the aggressive frontier. At $\rho = 2, L = 2, c = 1$, the run reaches the top of the project's $K = 64$ band. The operational rule is therefore simple: more aggressive replay requires a more conservative clip.

The clip is part of the learning rule, not a stabilizer added afterward. A tight ceiling could have starved the update by suppressing useful off-policy correction, but that is not what happens in this setting. In the $\rho = 1$ age sweep, $c = 1$ gives the highest evaluation reward at every replay age, and the $L = 1, c = 1$ setting gives the best reward and correction rate in the headline table. Figure 7 shows the engineering mechanism. As the policy drifts from the behavior policy, a few reused rollouts acquire large raw importance ratios. A loose ceiling lets those few rollouts dominate the gradient and concentrate the update on a small number of stale successes. A tight ceiling caps their weight, raises the effective sample size, and keeps more groups contributing to the update. In this implementation, the clip ceiling is therefore as central as the age cap for controlling replay quality.

Replay narrows answer-span diversity, but lower diversity does not by itself predict weaker high-k performance. The strongest high-k settings are not the most diverse, and some diverse settings do not give the strongest pass@16. What matters is whether the retained modes include correct solutions, not only how wide the answer distribution is. This is why the diversity metrics are useful diagnostics, but not sufficient success criteria.

Four caveats bound the result. First, the study covers one task, one model size, and mostly one training trajectory per setting, with evaluation variance controlled by five seeded repeats rather than repeated training runs. Second, the canonical tables and the $K = 64$ probe use different pass@k conventions, ordered first-k and the unbiased estimator, so they are complementary views rather than one interchangeable scale. Third, Countdown's verifier is cheap and deterministic, so the fresh-evaluation axis is best read as the algorithmic budget for newly generated, verifier-scored completions. In settings with slower learned verifiers, the same axis should translate more directly into wall-clock verifier cost. Fourth, the method is deliberately simple: it uses a one-sided sequence-level clip, one behavior log-probability per response, uniform sampling of eligible groups, and synchronous age-limited replay.

Future work Three experiments would test how far the result generalizes. Repeating the headline settings over several training seeds would separate knob effects from training noise and resolve small gaps between the top settings that a fifty-prompt evaluation set cannot. Running the same sweeps on a second task and a larger policy would test whether the clip-by-ratio interaction and the budget savings hold beyond Countdown and a half-billion-parameter model. Using a slower learned verifier would test whether fresh verifier evaluations become a direct wall-clock saving rather than an algorithmic proxy. On the method side, token-level behavior-log-probability logging, prioritized replay sampling, and asynchronous replay are the natural refinements to the current full-group, uniformly sampled, synchronous buffer.

7 Conclusion

Age-limited, clip-bounded replay lets verifier-based RLOO reuse recent rollout groups and match or exceed fresh-only high-k quality at substantially lower fresh-evaluation cost, while keeping the leave-one-out baseline intact and adding no value network. The clip ceiling is not a secondary stabilizer. No single clip is globally best across replay ages or replay ratios, and the safe clip becomes more conservative as replay becomes more aggressive. Tight clipping works by capping a small number of high-ratio stale rollouts and keeping more groups active in the update. The best-of-N probe sharpens the efficiency claim. The strongest $\rho = 1$ replay curve reaches the fresh-only $K = 64$

quality already at $K = 16$. The practical operating point is $\rho = 1, L = 1, c = 1$, which gives the best balanced reward and correction-rate result at about half the fresh-evaluation budget. The low-budget frontier is $\rho = 2, L = 2, c = 1$, which reaches the top large-K band observed in the project at about one third of that budget. The final engineering rule is simple. Full-group RLOO replay is viable when replay age is bounded and the clip is tightened as reuse increases.

Changes from Proposal The project direction is unchanged from the proposal. The realized study builds the required SFT, IPO, and on-policy RLOO pipeline, adds replay only on the RLOO path, and studies the same semi-off-policy replay design across the three proposed knobs, the replay ratio ρ, the replay age L, and the clip ceiling c, with the full-group replay unit and the clipped importance weight as proposed. The refinements are on the evaluation side, a seeded canonical protocol of five repeats for variance control and the optional best-of-N stretch carried out as a $K = 64$ inference-time probe.

Collaboration and AI tool use. ChatGPT and Claude were used to

1. explore background material on IPO/DPO, RLOO, and replay buffers
2. research the instrumentation needed for the replay-buffer extension
3. troubleshoot implementation errors and infrastructure issues, including interpreting error messages and reasoning about potential fixes
4. edit prose for clarity and grammar
5. assist with table formatting and plot generation for visualization

All project ideas, technical decisions, experimental design, and analysis are the author's own.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to Basics: Revisiting REINFORCE-Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Lun-Wei Ku, Andre Martins, and Vivek Srikumar (Eds.). Association for Computational Linguistics, Bangkok, Thailand, 12248–12267. doi:10.18653/v1/2024.acl-long.662

Brian R. Bartoldson, Siddarth Venkatraman, James Diffenderfer, Moksh Jain, Tal Ben-Nun, Seanie Lee, Minsu Kim, Johan Obando-Ceron, Yoshua Bengio, and Bhavya Kailkhura. 2025. Trajectory Balance with Asynchrony: Decoupling Exploration and Learning for Fast, Scalable LLM Post-Training. https://arxiv.org/abs/2503.18929

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Joshua Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. *CoRR* abs/2107.03374 (2021). arXiv:2107.03374 https://arxiv.org/abs/2107.03374

Siheng Li, Zhanhui Zhou, Wai Lam, Chao Yang, and Chaochao Lu. 2025. RePO: Replay-Enhanced Policy Optimization. arXiv:2506.09340 [cs.CL] https://arxiv.org/abs/2506.09340

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, S. Koyejo,

S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (Eds.), Vol. 35. Curran Associates, Inc., 27730–27744. https://proceedings.neurips.cc/paper_files/paper/2022/file/b1efde53be364a73914f58805a001731-Paper-Conference.pdf

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG] https://arxiv.org/abs/1707.06347

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y.K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. https://arxiv.org/abs/2402.03300

Pei-Hao Su, Paweł Budzianowski, Stefan Ultes, Milica Gašić, and Steve Young. 2017. Sample-efficient Actor-Critic Reinforcement Learning with Supervised Data for Dialogue Management. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, Kristiina Jokinen, Manfred Stede, David DeVault, and Annie Louis (Eds.). Association for Computational Linguistics, Saarbrücken, Germany, 147–157. doi:10.18653/v1/W17-5518

Yunhao Tang, Taco Cohen, David W. Zhang, Michal Valko, and Rémi Munos. 2025. RL-finetuning LLMs from on- and off-policy data with a single algorithm. arXiv:2503.19612 [cs.LG] https://arxiv.org/abs/2503.19612

A Additional Experiments

Full canonical sweep Table 2 reports the complete seeded canonical results at $K = 16$ for all $\rho = 1$ design cells, the fresh-only baseline, and the aggressive $\rho = 2$ setting. Cell names map to coordinates as follows. At $L = 1$ the cells are $c = 1$, $c = 3$, and $c = 10$. At $L = 2$ and $L = 3$ the columns follow the same clip order. All values are means over five seeds with 95 percent confidence-interval half-widths.

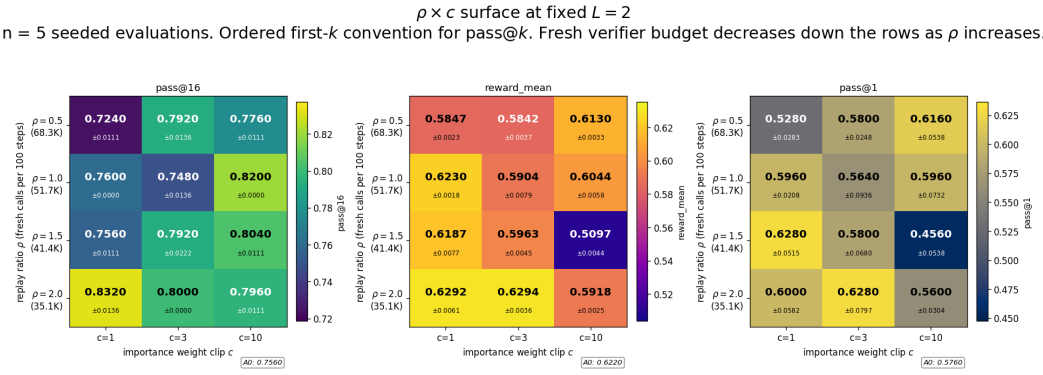
Table 2: Seeded canonical reference at $K = 16$, five seeds, 50 held-out prompts. Reward is the mean verifier reward and correction rate is the mean fraction of fully correct responses.

Setting	(ρ, L, c)	pass@1	pass@16	reward	correction rate
Fresh-only	$(0, -, -)$	0.576 ± 0.051	0.756 ± 0.011	0.622 ± 0.005	0.594 ± 0.005
Best balanced	$(1, 1, 1)$	0.616 ± 0.092	0.792 ± 0.014	0.661 ± 0.005	0.644 ± 0.006
L=1 high-k	$(1, 1, 3)$	0.612 ± 0.022	0.816 ± 0.011	0.616 ± 0.004	0.601 ± 0.005
L=1 default	$(1, 1, 10)$	0.600 ± 0.066	0.764 ± 0.011	0.609 ± 0.001	0.597 ± 0.002
L=2 low-clip	$(1, 2, 1)$	0.596 ± 0.021	0.760 ± 0.000	0.623 ± 0.002	0.599 ± 0.002
L=2 mid-clip	$(1, 2, 3)$	0.564 ± 0.094	0.748 ± 0.014	0.590 ± 0.008	0.572 ± 0.008
L=2 high-k	$(1, 2, 10)$	0.596 ± 0.073	0.820 ± 0.000	0.604 ± 0.006	0.591 ± 0.006
Strong $\rho = 1$ L=3 high-k	$(1, 3, 1)$	0.584 ± 0.092	0.820 ± 0.000	0.636 ± 0.008	0.625 ± 0.009
L=3 late recovery	$(1, 3, 3)$	0.620 ± 0.079	0.784 ± 0.011	0.627 ± 0.011	0.612 ± 0.010
L=3 default	$(1, 3, 10)$	0.604 ± 0.044	0.764 ± 0.011	0.622 ± 0.005	0.610 ± 0.006
Aggressive, c=10	$(2, 2, 10)$	0.560 ± 0.030	0.796 ± 0.011	0.592 ± 0.003	0.581 ± 0.003

Replay-ratio canonical sweep Table 3 reports the seeded canonical results at $K = 16$ for the second sweep, which fixes the replay age at $L = 2$ and varies the replay ratio and the clip ceiling. The $\rho = 1$ rows reproduce the $L = 2$ cells of Table 2, and the $\rho = 2$, $c = 10$ row reproduces the aggressive $c = 10$ cell. The fresh-evaluation budget falls as the replay ratio rises, from 68.3K at $\rho = 0.5$ to 51.7K at $\rho = 1$, 41.4K at $\rho = 1.5$, and 35.1K at $\rho = 2$. All values are means over five seeds with 95 percent confidence-interval half-widths.

Table 3: Seeded canonical reference at $K = 16$ for the replay-ratio sweep at fixed replay age $L = 2$, five seeds, 50 held-out prompts. The $\rho = 1$ rows reproduce the $L = 2$ cells of Table 2, and the $\rho = 2$, $c = 10$ row reproduces the aggressive $c = 10$ cell. Reward is the mean verifier reward and correction rate is the mean fraction of fully correct responses.

Setting	(ρ, L, c)	pass@1	pass@16	reward	correction rate
ρ=0.5, loose	$(0.5, 2, 10)$	0.616 ± 0.054	0.776 ± 0.011	0.613 ± 0.003	0.586 ± 0.003
ρ=0.5, mid	$(0.5, 2, 3)$	0.580 ± 0.025	0.792 ± 0.014	0.584 ± 0.004	0.571 ± 0.004
ρ=0.5, tight	$(0.5, 2, 1)$	0.528 ± 0.028	0.724 ± 0.011	0.585 ± 0.002	0.539 ± 0.003
ρ=1, loose	$(1, 2, 10)$	0.596 ± 0.073	0.820 ± 0.000	0.604 ± 0.006	0.591 ± 0.006
ρ=1, mid	$(1, 2, 3)$	0.564 ± 0.094	0.748 ± 0.014	0.590 ± 0.008	0.572 ± 0.008
ρ=1, tight	$(1, 2, 1)$	0.596 ± 0.021	0.760 ± 0.000	0.623 ± 0.002	0.599 ± 0.002
ρ=1.5, loose	$(1.5, 2, 10)$	0.456 ± 0.054	0.804 ± 0.011	0.510 ± 0.004	0.499 ± 0.004
ρ=1.5, mid	$(1.5, 2, 3)$	0.580 ± 0.068	0.792 ± 0.022	0.596 ± 0.005	0.577 ± 0.005
ρ=1.5, tight	$(1.5, 2, 1)$	0.628 ± 0.052	0.756 ± 0.011	0.619 ± 0.008	0.603 ± 0.008
ρ=2, loose	$(2, 2, 10)$	0.560 ± 0.030	0.796 ± 0.011	0.592 ± 0.003	0.581 ± 0.003
ρ=2, mid	$(2, 2, 3)$	0.628 ± 0.080	0.800 ± 0.000	0.629 ± 0.004	0.613 ± 0.004
ρ=2, tight	$(2, 2, 1)$	0.600 ± 0.058	0.832 ± 0.014	0.629 ± 0.006	0.614 ± 0.006



A0 ($\rho = 0$ fresh-only baseline, 102.4K fresh calls) anchors each panel as a reference value. Fresh verifier budget falls as ρ rises, from 68.3K at $\rho = 0.5$ to 35.1K at $\rho = 2.0$. The $\rho = 1.0$ row reproduces the $L = 2$ row of the L × c figure and is the cross-figure consistency anchor.

Figure 9: Replay ratio by clipping sweep at fixed replay age $L = 2$, the same sweep as Figure 5 with pass@1 in the third panel in place of correction rate. Rows vary the replay ratio ρ and columns vary the importance-weight clip ceiling c. Values are five-seed means with 95 percent confidence-interval half-widths. The loose clip $c = 10$ at the higher replay ratios shows the widest gap between a strong pass@16 and a weak pass@1, most clearly at $\rho = 1.5$ where pass@16 is 0.804 while pass@1 falls to 0.456, which is the single-sample cost of letting stale rollouts move the gradient.

Best-of-N probe Table 4 reports the unbiased pass@k of the probed settings at $K \in \{1, 4, 8, 16, 32, 64\}$ with 64 samples per prompt in each of five seeded repeats. Among the $\rho = 1$ settings, the $L = 3$, $c = 1$ curve is the strongest through $K = 32$, with a near-tie at $K = 4$, and the $L = 3$, $c = 3$ setting and the aggressive $(2, 2, 1)$ setting share the highest pass@64 endpoint at 0.844. Among the aggressive $\rho = 2$ cells the clip ceiling sets the large-K outcome. The loose clip $c = 10$ is the weakest curve at every K, the tight clip $c = 1$ leads the aggressive cells and reaches the top of the $\rho = 1$ band at $K = 64$ at one third of the fresh budget, and the $c = 3$ setting falls between. Two reading notes apply. The canonical $K = 16$ values in Table 2 use the ordered first-k convention over a seeded sample, while the probe uses the unbiased estimator (Chen et al., 2021) over 64 samples per prompt within each repeat averaged across five seeds, so the probe gives a lower-variance complementary estimate at $K = 16$, while the canonical table remains the primary evaluation convention for the heatmaps. The remaining gap between the two conventions is driven

mainly by the difference in the seed namespace rather than by the estimator definition, and it is largest for the $(2, 2, 10)$ cell.

Table 4: Best-of-N probe, unbiased pass@k, five seeds, 64 samples per prompt per repeat.

Setting (ρ, L, c)	K=1	K=4	K=8	K=16	K=32	K=64
Fresh-only $(0, -, -)$	0.581	0.708	0.733	0.761	0.786	0.804
Best balanced $(1, 1, 1)$	0.625	0.758	0.780	0.796	0.810	0.820
L=1 high-k $(1, 1, 3)$	0.585	0.738	0.774	0.796	0.814	0.840
L=2 high-k $(1, 2, 10)$	0.585	0.713	0.746	0.774	0.797	0.812
Strong $\rho = 1$ L=3 high-k $(1, 3, 1)$	0.630	0.756	0.792	0.813	0.826	0.840
L=3 late recovery $(1, 3, 3)$	0.597	0.744	0.770	0.789	0.812	0.844
Aggressive, c=1 $(2, 2, 1)$	0.609	0.760	0.798	0.819	0.831	0.844
Aggressive, c=3 $(2, 2, 3)$	0.596	0.755	0.778	0.784	0.788	0.796
Aggressive, c=10 $(2, 2, 10)$	0.573	0.678	0.705	0.729	0.753	0.776

B Implementation Details

B.1 Software, hardware, and artifacts

The replay-buffer extension runs on the default-project infrastructure without changing it, using vLLM for rollout generation, PyTorch with HuggingFace Transformers for the policy update across two Ray workers, HuggingFace Datasets for data, and Weights and Biases for logging. The only systems-side change is running the main experiments on a single Azure H100 alongside the Modal H100 setup from the default project. PyTorch is installed from the CUDA 12.8 wheels under Python 3.11, with the full dependency list in the repository `pyproject.toml` and the launch commands in the repository scripts for SFT, IPO, fresh-only RLOO, replay RLOO, and evaluation.

Both the trainable policy and the frozen reference for every RLOO run start from the project SFT checkpoint `amulyaparthasarathy/qwen-sft-countdown`, which is public on HuggingFace.[1] The RLOO checkpoints from the sweep are released on the same account. The SFT stage that produces that checkpoint and the separate IPO milestone stage follow the course milestone setup and are not repeated here.

B.2 Training and evaluation settings

Table 5 lists the settings shared by every RLOO run, the three knobs that are swept, and the evaluation setup.

B.3 The replay buffer and the order of each step

The buffer stores whole groups of K responses, not single responses, so the leave-one-out baseline is always computed over a complete group. Each stored group keeps its prompt and ground truth, its K responses, the K verifier rewards, the K behavior log-probabilities, and the step at which it was sampled. A check at construction rejects any group whose response, reward, and log-probability lists are not all the same length, because a mismatch would quietly break the leave-one-out math.

A group can be reused once its age, the current step minus the step it was sampled at, is between one and L. The age test is applied to each group on its own, not as a single oldest-allowed value, which is what stops a just-sampled group from being reused at age zero when the buffer holds a mix of ages. Reused groups are picked uniformly at random without replacement. When there are fewer eligible groups than the batch needs, which happens while the buffer is still filling, the gap is filled with fresh groups. Groups older than L are dropped at the end of the step. The 4096-group capacity limit is only a safety net and is never reached, since with L at most three and 128 fresh groups per step the buffer holds at most $(L + 1) \times 128$, which is 512 groups.

The order within a step is fixed. The step samples fresh groups, samples the reused groups, runs the update, then adds the fresh groups to the buffer and drops the over-age ones. Adding and dropping

[1] `https://huggingface.co/amulyaparthasarathy/qwen-sft-countdown`

Table 5: Settings for the RLOO runs and for evaluation.

RLOO training (shared by every run)	
Policy and reference start point	`amulyaparthasarathy/qwen-sft-countdown`
Responses per group K	8
Prompt groups per update	128
Optimizer steps	100
Learning rate	10^{-5}, constant, no warmup
Weight decay	10^{-4}
KL coefficient	10^{-3}
Entropy coefficient	10^{-3}
Gradient clipping	none
Rollout sampling	temperature 1.0, top-p 1.0, top-k off, 1024 tokens
Swept knobs	
Replay ratio ρ	$\{0, 0.5, 1, 1.5, 2\}$
Maximum replay age L	$\{1, 2, 3\}$
Importance-weight clip ceiling c	$\{1, 3, 10\}$, plus a $c = 1000$ stress probe
Replay buffer capacity	4096 (never reached in practice)
Evaluation	
Sampling	temperature 0.6, top-p 0.95, top-k 20, 1024 tokens
Samples per prompt	16 for the canonical metric, 64 for the probe
Engine	vLLM, maximum model length 2048

happen after the update so that the groups sampled this step cannot be reused in the same step. Reused groups are not added again, since they are already in the buffer. The buffer can be saved and restored for crash recovery, but the random state used for sampling is not saved, so an exact rerun is not guaranteed after a restore.

B.4 How replay lowers the verifier budget

The budget is the number of fresh, verifier-scored rollouts in a run. Let $B = 128$ be the prompt groups per update and $K = 8$ the responses per group. For a requested replay ratio ρ, the number of fresh groups per step after the buffer has filled is

$$B_{\text{fresh}} = \text{round}\left(\frac{B}{1+\rho}\right), \qquad B_{\text{replay}} = B - B_{\text{fresh}}.$$

The first step is fully fresh because the buffer is empty, so over a 100-step run the number of fresh verifier evaluations is

$$K\left(B + 99\, B_{\text{fresh}}\right),$$

which reduces to $100\, B\, K$ in the fresh-only case $\rho = 0$. Table 6 gives the result for each ratio.

Table 6: Fresh-evaluation budget by replay ratio for a 100-step run.

ρ	Fresh groups per step	Replay groups per step	Fresh evaluations
0	128	0	102.4K
0.5	85	43	68.3K
1	64	64	51.7K
1.5	51	77	41.4K
2	43	85	35.1K

There is a limit on how aggressive the ratio can be. Each step adds its fresh groups to the buffer and keeps them for L steps, so the pool of reusable groups holds about $L \times B_{\text{fresh}}$ groups, while the replay demand each step is $B_{\text{replay}} = \rho \times B_{\text{fresh}}$. The demand can be met only when $\rho \leq L$. This is why the ratio sweep fixes $L = 2$, so that ratios up to $\rho = 2$ are supported, and does not treat $\rho = 1.5$ or $\rho = 2$ at $L = 1$ as comparable cells.

B.5 Experiments that were run

Table 7 lists the experiment blocks. The canonical pass@k results for the RLOO cells are in Tables 2 and 3, and the best-of-N curves are in Table 4. The two sweeps share four cells, since the $\rho = 1$ column of the ratio sweep is the same as the $L = 2$ row of the age sweep, and the $\rho = 2$, $c = 10$ cell is the same as the loose-clip aggressive cell.

Table 7: Experiment blocks. Fresh-evaluation budgets are in Table 6.

Block	Configuration	Purpose
Fresh-only RLOO	$\rho = 0$	baseline and budget anchor
$L \times c$ sweep	$\rho = 1$, $L \in \{1, 2, 3\}$, $c \in \{1, 3, 10\}$	how age and clip interact
$\rho \times c$ sweep	$L = 2$, $\rho \in \{0.5, 1, 1.5, 2\}$, $c \in \{1, 3, 10\}$	how ratio and clip interact
Stress probe	$\rho = 1$, $L = 1$, $c = 1000$, short run	near-unclipped failure case
Best-of-N probe	selected cells at $K = 64$	test-time scaling

B.6 How results were measured

The verifier returns one of three values for a response. It returns 1.0 when the equation is well formed, uses each given number once, and reaches the target, 0.1 when the equation is well formed but does not reach the target, and 0 when no valid equation is produced. Evaluation reward is the mean of this value over responses. The correction rate is the fraction of responses that score exactly 1.0. A prompt counts as solved at pass@k when any of its first k responses scores 1.0. Reward is higher than the correction rate whenever well-formed but wrong equations are present, which is why the two columns differ throughout Tables 2 and 3.

The canonical metric is the mean over five seeded repeats at $K = 16$ on a fixed set of fifty held-out prompts, using the ordered first-k rule. The best-of-N probe instead samples 64 responses per prompt and uses the unbiased pass@k estimator. The reported confidence intervals are t-intervals with four degrees of freedom over the five seeds, so they measure variation across generation seeds on the fixed fifty-prompt set. They do not capture the uncertainty from using only fifty prompts, which is a prompt-sampling standard error of roughly 0.06 on pass@k. That spread is why close pass@k values are read as a band and why nearby settings are rerun before being compared. The probe is reported apart from the canonical table because it uses a different seed namespace and a different estimator.

B.7 What was logged to check correctness

Every run logs the training reward, the fraction of training rollouts that are fully correct, the KL drift from the SFT reference, the clip fraction, the mean and maximum of the raw and clipped importance weights, and the normalized effective sample size of the replay batch. These are also broken out by source, fresh against replay, and by age, so age-0, age-1, age-2, and older groups can be followed separately, together with the actual replay ratio reached, the fresh and replay group counts, and the buffer age histogram. The $c = 1000$ setting is a deliberate near-unclipped probe. A few reused responses with very large weights take over the batch and the effective sample size collapses, so it is treated as a stress test rather than a comparable run and is not given a full evaluation.