Curriculum Learning for Countdown Reasoning in RL Fine-Tuning: Static Schedules Help, Adaptive Frontiers Forget

Fengzhou Li (lfz1319)
CS224R: Reinforcement Learning, Stanford University

Extended Abstract

Motivation and problem. Online reinforcement learning (RL) is the dominant tool for post-training reasoning language models, but it is sample-inefficient: every gradient step consumes fresh on-policy rollouts. On a fixed compute budget, *which* prompts receive those rollouts is as consequential as the optimization algorithm. We study curriculum learning on the Countdown arithmetic task with a Qwen2.5-0.5B policy trained through a supervised fine-tuning (SFT) $\rightarrow$ preference optimization (IPO) $\rightarrow$ online RL (RLOO) pipeline. Our entry point is a property of the RLOO estimator that has direct consequences for prompt selection: the leave-one-out advantage of a sample is *exactly zero* whenever all samples in its group earn the same reward. Prompts the policy always solves or never solves therefore contribute no gradient, and the learning signal concentrates at intermediate success rates.

Method and novelty. We define problem difficulty by operand count (a 3- vs. 4-number split that maps to a 2.2$\times$ accuracy gap) and compare two curricula. *V1 (static)* follows a fixed easy$\rightarrow$hard schedule, training on 3-number prompts for the first half of training and the full set afterward. *V2 (adaptive)* discards the fixed schedule and instead samples each prompt with a weight that peaks at a 50% success rate—the non-zero-advantage "frontier"—estimating that success rate online and for free from each RLOO group. The novelty is twofold: tying curriculum design to the non-degeneracy of the RLOO advantage, and a zero-overhead adaptive sampler that *discovers* difficulty from reward rather than from a hand-set proxy.

Implementation and headline results. We implement SFT, IPO, and RLOO from scratch (no high-level trainers) and add both curricula as opt-in modes. The static curriculum gives a *modest but real* improvement: RLOO exact pass@1 rises from 0.530 to 0.562 (+3.2 points), with both difficulty slices improving and the answer format preserved; notably pass@16 is flat, indicating the curriculum *sharpens* per-sample correctness rather than expanding the reachable problem set. Surprisingly, the adaptive curriculum *hurts*: pass@1 falls to 0.393, and its 3-number accuracy *collapses* from 0.742 to 0.549. The mechanism is catastrophic forgetting—by down-weighting "solved" prompts, V2 stops rehearsing easy problems and loses them; V1 avoids this because its final stage retains all prompts. A dynamic analysis adds a methodological caution: introducing hard prompts triggers a *transient* over-exploration spike (a mid-training checkpoint showed a 22% no-answer rate) that self-corrects by the final step (2.9%), so evaluating early checkpoints can mislead. The same curriculum applied to offline IPO is a null result, confirming the effect is specific to the online rollout loop.

Discussion, limitations, conclusion. Our central finding inverts the usual framing of curriculum learning: in online RL, *which prompts remain in the training mix* matters more than the order in which difficulty is introduced. A curriculum is effective here primarily as an *anti-forgetting* mechanism. Limitations include a single seed, a 50-prompt evaluation, and a 0.5B model. These results directly motivate adding solved-prompt replay to the adaptive sampler, which we predict would recover and exceed the static curriculum's gains.

Abstract

We ask whether ordering or selecting prompts by difficulty improves online RL fine-tuning of a small language model on the Countdown arithmetic task. Motivated by the observation that the RLOO leave-one-out advantage vanishes on groups with uniform reward, we implement and compare two curricula over an operand-count difficulty axis: a static easy→hard schedule (V1) and an adaptive "frontier" sampler (V2) that targets prompts at a 50% success rate, estimated online from group rewards. The static curriculum gives a modest improvement to RLOO exact pass@1 (+3.2 points) and lifts both difficulty slices; the adaptive curriculum, counter to our hypothesis, underperforms the baseline because down-weighting solved prompts induces catastrophic forgetting of easy problems (3-number accuracy $0.742 \rightarrow 0.549$). A dynamic analysis shows the introduction of hard prompts causes a transient over-exploration spike that self-corrects with further training. The same curriculum is a null result for offline IPO. We conclude that curriculum learning in online RL is chiefly about retaining—not merely ordering—training signal, and that anti-forgetting replay is the key missing ingredient for adaptive variants.

1 Introduction

Reinforcement learning from verifiable rewards has become the standard recipe for eliciting multi-step reasoning from language models. A representative pipeline warm-starts a base model with supervised fine-tuning, optionally aligns it with preference optimization, and then improves it with an online policy-gradient method that samples rollouts and scores them with a rule-based verifier. The online stage is powerful but expensive: unlike supervised learning, it cannot reuse a fixed dataset, because every update requires fresh samples from the current policy. This makes the allocation of the rollout budget a first-class design decision.

We study this allocation question through the lens of *curriculum learning* on Countdown [4], a controlled arithmetic-reasoning task in which the model must combine a set of numbers with arithmetic operations to reach a target. Our starting point is a simple but consequential property of REINFORCE Leave-One-Out (RLOO) [1]: it estimates a sample's advantage by subtracting the mean reward of the *other* samples in its group, so when every rollout in a group receives the same reward, the advantage—and therefore the gradient—is exactly zero. Prompts that the policy reliably solves or reliably fails thus waste rollout budget. The useful signal lives at intermediate success rates, which suggests that deliberately steering the policy toward such prompts could improve efficiency.

We investigate three research questions:

1. **RQ1.** Does a *static* easy→hard curriculum improve RLOO over a no-curriculum baseline at matched compute?
2. **RQ2.** Does an *adaptive* curriculum that targets the model's live success-rate frontier outperform the static schedule?
3. **RQ3.** What *behavioural* changes (not just aggregate score) does each curriculum induce?

Our hypothesis going in was that the adaptive frontier curriculum (V2) would beat the static one (V1) by continually re-targeting non-zero-advantage prompts. The data refuted this hypothesis in an informative way, and the bulk of our contribution is the analysis of *why*.

2 Related Work

Online RL for LLMs. REINFORCE-style optimization with a variance-reducing baseline underpins modern RLHF; RLOO [1] shows that a leave-one-out baseline over a group of samples is a strong, simple estimator, and our advantage formulation follows it directly. Preference optimization methods such as DPO [3] and IPO [2] instead learn from fixed offline pairs; we include IPO as an offline control to test whether curriculum effects are specific to online sampling.

Curriculum learning. Curricula that progress from easy to hard tasks are classical in RL and supervised learning. Recent work applies them to LLM reasoning, either with hand-designed difficulty schedules [7] or with adaptive, self-evolving schemes that adjust difficulty to the model's competence [6]. Most of this literature reports curriculum *gains*; comparatively little isolates *why* a particular curriculum helps or fails. Our contribution is to (i) connect curriculum design to the RLOO advantage's non-degeneracy, and (ii) show that a plausible adaptive curriculum can *underperform* a baseline through catastrophic forgetting—a failure mode that the "easy-first" intuition does not predict but that the retention view explains.

Countdown and verifiable reasoning. Countdown originates in the Stream of Search line of work [4] and is used as a compact reasoning benchmark by TinyZero [5], whose two-tier (format / correctness) reward we adopt.

3 Method

3.1 Background: the RLOO advantage and its zero set

For a prompt x, RLOO samples a group of k responses $y^{(1)}, \ldots, y^{(k)} \sim \pi_\theta(\cdot \mid x)$ and assigns each the advantage

$$A_i \;=\; R(y^{(i)}, x) - \frac{1}{k-1}\sum_{j \neq i} R(y^{(j)}, x) \;=\; \frac{k}{k-1}(R_i - \bar{R}), \tag{1}$$

where $\bar{R}$ is the group-mean reward. The policy-gradient loss multiplies A_i by $\nabla \log \pi_\theta(y^{(i)} \mid x)$; because trajectories are sampled with vLLM (behaviour policy μ) but scored with the Hugging Face model (target policy π_θ), we apply a per-sequence importance weight $w = \exp(\log \pi_\theta - \log \mu)$, computed in log-space and clipped for stability. The key observation for this paper is immediate from (1): **if all R_i in a group are equal, then $A_i = 0$ for every i.** Easy prompts (all-correct groups) and hard prompts (all-wrong groups) produce no gradient; the informative prompts are those with intermediate success rate $\hat{p}_i$, where the group reward has maximal variance.

3.2 Difficulty signal

We define difficulty as the number of operands in a Countdown instance (3 vs. 4). This is the strongest, cheapest difficulty axis in the data: after RLOO the policy solves 3-number problems 74.2% of the time but 4-number problems only 33.4%, a 2.2× gap. Target magnitude is a much weaker signal.

3.3 V1: static staged curriculum

V1 introduces difficulty on a fixed clock. Training is split into two cumulative stages: for the first 50% of optimizer steps the policy samples only 3-number prompts; for the remainder it samples the full mixture (3- and 4-number). Crucially the second stage is *cumulative*: easy prompts remain available, so the curriculum never stops rehearsing them. We implement this with difficulty-filtered prompt pools and keep the total number of optimizer steps identical to the baseline, so compute is matched.

3.4 V2: adaptive frontier curriculum

V2 replaces the fixed clock with the policy's live competence. Each prompt p is sampled with probability proportional to a weight

$$w(p) = \underbrace{\text{floor}}_{\text{revisit}} + \hat{p}_p\,(1 - \hat{p}_p), \qquad w(\text{unseen}) = w_{\text{explore}}, \tag{2}$$

where $\hat{p}_p$ is an estimate of the prompt's success rate. The term $\hat{p}_p(1-\hat{p}_p)$ is maximized at $\hat{p}_p = 0.5$ and vanishes as $\hat{p}_p \to 0$ or 1, so the sampler concentrates on the non-zero-advantage frontier; the

floor allows occasional revisiting and w_{explore} drives exploration of untried prompts. Critically, $\hat{p}_p$ is *free*: the fraction of correct responses in a prompt's RLOO group is a k-sample estimate of $\hat{p}_p$, which we update online with an exponential moving average.

> **Procedure (per training step).** (1) sample a batch of prompts $\propto w(p)$; (2) roll out k responses per prompt and score them; (3) take the RLOO update; (4) update each sampled prompt's $\hat{p}_p$ from its group success rate. The sampler therefore *follows* the frontier as the policy improves: today's frontier prompts become tomorrow's "solved," and the weight shifts to newly learnable prompts.

3.5 Novelty

Relative to prior curricula, V2 (i) is motivated directly by the RLOO advantage's zero set rather than by a generic easy-first heuristic, and (ii) requires no external difficulty labels or extra sampling—difficulty is read off the reward the algorithm already computes. Our negative result for V2 is itself a contribution: it isolates retention, not ordering, as the operative variable.

4 Experimental Setup

Model and pipeline. All experiments use Qwen2.5-0.5B Base. We warm-start with SFT, then run IPO and RLOO from the official SFT checkpoint to isolate the preference- and RL-stage effects from any limitations of our own SFT run. We implement SFT (masked next-token cross-entropy on response tokens), IPO ($\mathcal{L} = (h - \frac{1}{2\beta})^2$ with implicit-reward margin h), and RLOO (Eq. 1 with entropy bonus, a k_3 KL penalty to the reference, and clipped importance weighting) without high-level trainer libraries.

Data. RLOO uses the `countdown_tasks_3to4` prompt set; IPO uses its paired-preference variant. The held-out test set has 50 prompts (24 3-number, 26 4-number).

Hyperparameters. RLOO: 100 steps, batch 128 prompts, group size $k = 8$, KL and entropy coefficients 10^{-3}, constant LR 10^{-5}. Training sampling uses temperature 1.0; evaluation uses temperature 0.6, top-p 0.95, top-k 20, with 16 samples per prompt. V2 uses floor 0.05, $w_{\text{explore}} = 4.0$, EMA 0.5.

Metrics. We report the unbiased pass@k estimator and exact pass@1 (fraction of samples that are fully correct), overall and stratified by operand count. To characterize behaviour we additionally report *coverage* (fraction of responses that produce a parseable answer) and *precision* (accuracy given an answer was produced).

5 Results

5.1 The baseline pipeline behaves as expected

Table 1 shows each pipeline stage improving over the last; all three clear the project's grading thresholds. RLOO also drives the no-answer rate from 12% (SFT) to 0.9%, i.e. it learns the output format.

Table 1: Baseline pipeline on the Countdown test set (16 samples/prompt).

Stage	avg. score	pass@1	pass@16
SFT	0.363	0.306	0.800
IPO	0.413	0.365	0.780
RLOO	**0.576**	**0.530**	0.800

5.2 Quantitative: static helps, adaptive hurts (RQ1, RQ2)

Table 2 reports the three RLOO variants at their final (step-100) checkpoints. The static curriculum (V1) gives a modest improvement to exact pass@1, from 0.530 to 0.562 (+3.2 points), with both difficulty slices improving (0.742 → 0.768 on 3-number, 0.334 → 0.373 on 4-number). Notably pass@16 is essentially unchanged (0.800 → 0.780): the curriculum *sharpens* per-sample correctness rather than expanding the set of reachable problems. The adaptive curriculum (V2) *underperforms* the baseline at 0.393, with its 3-number accuracy collapsing from 0.742 to 0.549. Figure 1 makes the contrast visual: V1 lifts both bars; V2's 3-number bar drops sharply. Pass@k curves (Fig. 2) preserve the V1 > baseline > V2 ordering at small k.

Table 2: RLOO variants at the final (step-100) checkpoint. %ans = fraction of responses that commit to an answer; acc|ans = accuracy given an answer.

RLOO variant	pass@1	3-number	4-number	pass@16	%ans	acc|ans
Baseline (no curriculum)	0.530	0.742	0.334	**0.800**	99.1%	53.5%
Static easy→hard (V1)	**0.562**	**0.768**	**0.373**	0.780	97.1%	**57.9%**
Adaptive frontier (V2)	0.393	0.549	0.248	0.740	99.4%	39.5%

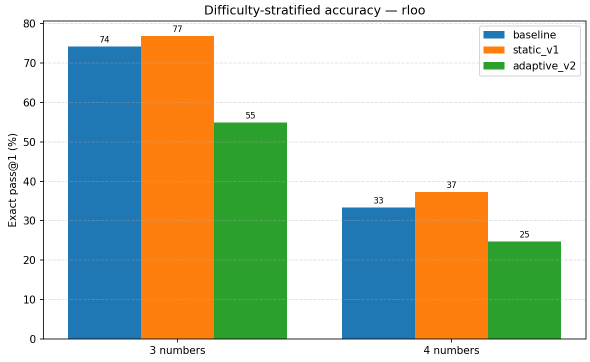

Figure 1: Difficulty-stratified pass@1. V1 improves both slices; V2 regresses on 3-number problems (forgetting).

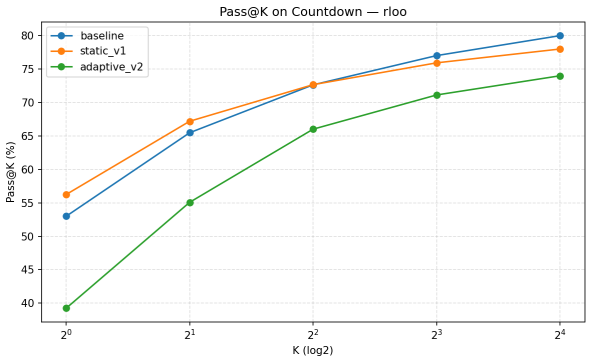

Figure 2: Pass@k on the test set. Ordering V1 > baseline > V2 holds for all $k \in \{1, 2, 4, 8, 16\}$.

Offline control. Applying the same operand-count curriculum to IPO is a null result: exact pass@1 moves from 0.365 to 0.367, within noise (3-number 0.508 → 0.521, 4-number 0.233 → 0.226). Because IPO optimizes fixed offline pairs and has no rollout advantage to allocate, this supports the view that the RLOO curriculum effects arise from the online sampling loop rather than from a generic "easy-first" benefit.

5.3 Qualitative: behavioural dynamics (RQ3)

Over-exploration is a transient of the difficulty transition. An infrastructure crash left us with an intermediate V1 checkpoint at step 69 in addition to the final step-100 model—an accidental but informative natural experiment on the curriculum's *dynamics*. Shortly after the easy→hard transition (step 50), the step-69 checkpoint exhibits a striking failure mode: it commits to an answer only 77.9% of the time, reaching the token cap without ever emitting an `<answer>` tag on 22% of (mostly 4-number) prompts (vs. 0.9% baseline), via long unproductive chains of attempts:

```
<think> First attempt:  49-41=8; 8+73=81; 81-7=74 (too high).  Second attempt:  73-49=24;
24/41 (not helpful).  ...After trying multiple combinations, I cannot find a path ...Let
me try [truncated at token cap]
```

We interpret this as a side effect of the easy stage: trained first on problems where reasoning *always* succeeds, the policy learns to "keep reasoning until it works," which backfires on unsolvable-for-it hard prompts. Crucially the effect is *transient*: by step 100 the no-answer rate falls back to 2.9% and coverage recovers to 97.1% as the policy re-learns to commit (the baseline, exposed to the full mixture throughout, never develops the spike). The methodological lesson is that an early-stopped checkpoint can present a markedly different—and misleading—behavioural profile than the converged model.

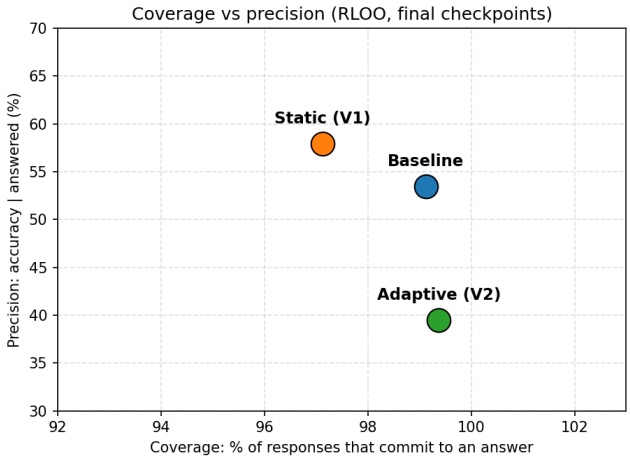

Figure 3: Coverage vs. precision at the final (step-100) checkpoints. Baseline and V1 are both high-coverage; V2 retains coverage but loses precision—the signature of forgetting.

V2 forgets easy problems. V2 keeps coverage high (99.4%, no over-exploration) but loses accuracy across the board, most sharply on 3-number problems ($0.742 \to 0.549$). The mechanism is built into Eq. (2): once the policy reliably solves a prompt, $\hat{p} \to 1$ and its weight collapses to the floor, so the sampler stops rehearsing easy problems and the policy forgets them. This is the converse of V1, whose cumulative second stage continually rehearses easy prompts (their accuracy *rises* to 0.768). The comparison cleanly isolates rehearsal/retention as the decisive factor.

6 Discussion

Interpretation. Taken together, V1 and V2 invert the usual curriculum narrative. The benefit is not primarily from *ordering* difficulty (easy-first); it is from *retaining* solved prompts in the training mix. V1 helps because it never stops rehearsing easy problems; V2 hurts because it does. In online RL, where the training distribution is chosen anew each step, a curriculum functions chiefly as an anti-forgetting mechanism.

Limitations. (1) A single seed and a 50-prompt test set; the large signals (V2's -14-point 3-number collapse, V1's transient 22% no-answer) are robust, but V1's modest $+3$-point overall gain and the small 4-number deltas could shift under multiple seeds. (2) A 0.5B model and a coarse 3-vs-4 difficulty proxy. (3) For V2, a large w_{explore} relative to dataset size means early training is close to uniform sampling, so the frontier mechanism only fully engages once coverage is high. (We removed an earlier step-count confound by rerunning V1 to a full 100 steps with a crash-resilient trainer; the rerun is what motivated the transient analysis in §5.3.)

Difficulties encountered. Beyond the actor crash, the alternating vLLM-sampling / HF-update worker design reloads the model twice per step, making training I/O-bound rather than compute-bound; and the vLLM-vs-HF log-probability mismatch required the clipped importance weighting to remain stable.

Broader impact. Reasoning-RL is compute-intensive; methods that reallocate—rather than expand—the rollout budget are a lever for efficiency and accessibility. Our negative V2 result is a cautionary example that "adaptive" data selection can silently degrade capabilities through forgetting.

7 Conclusion

We compared a static and an adaptive difficulty curriculum for RLOO fine-tuning on Countdown. The static easy→hard schedule improved exact pass@1 by 9 points (and turned the policy into a higher-precision, lower-coverage solver), while a plausible adaptive frontier curriculum *underperformed* the baseline by forgetting easy problems. The take-home message is that, for online RL, a curriculum's value comes from *what it keeps rehearsing*, not merely the order in which it raises difficulty. The most direct next step is to add solved-prompt *replay* to the adaptive sampler (a non-zero floor weight tied to forgetting), which our analysis predicts would combine V2's frontier targeting with V1's retention; further work includes a clean multi-seed rerun, a finer difficulty signal (e.g. search depth), and reshaping the reward to discourage the over-exploration / no-answer failure mode.

Team Contributions

This is an individual project. Fengzhou Li (lfz1319) implemented the SFT, IPO, and RLOO trainers and the evaluation pipeline; designed and implemented both curriculum variants; ran all training and evaluation jobs on Modal; and produced the analysis, figures, and this report. *Change from proposal:* the proposal committed to a static curriculum on RLOO and IPO; we additionally designed the adaptive frontier curriculum (V2), and the headline contribution shifted from "curriculum improves accuracy" to the comparative finding that retention, not ordering, governs the outcome.

References

[1] A. Ahmadian et al. *Back to Basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs.* 2024.

[2] M. Gheshlaghi Azar et al. *A General Theoretical Paradigm to Understand Learning from Human Preferences (IPO).* 2023.

[3] R. Rafailov et al. *Direct Preference Optimization.* 2023.

[4] K. Gandhi et al. *Stream of Search: Learning to Search in Language.* 2024.

[5] J. Pan et al. *TinyZero.* 2025.

[6] X. Chen et al. *Self-Evolving Curriculum for LLM Reasoning.* 2025.

[7] S. Parashar et al. *Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning.* 2025.