Extended Abstract

Motivation Reinforcement Learning with Leave-One-Out (RLOO) has emerged as an effective approach for post-training language models using verifier rewards. However, standard RLOO samples prompts uniformly from the training distribution despite large variation in prompt difficulty. Easy prompts are often already solved by the policy, while difficult prompts are rarely solved, causing leave-one-out advantages to collapse and reducing the usefulness of collected rollouts. We investigate whether curriculum-based prompt selection can improve sample efficiency for Countdown arithmetic reasoning by focusing training on prompts near the model's current competence boundary.

Method We introduce an adaptive competence-boundary curriculum for RLOO. Using an SFT-initialized model, prompts are assigned to difficulty buckets based on offline success rates. During training, we maintain running estimates of bucket-level success probabilities and compute sampling weights using a Gaussian function centered at 50% success. Buckets near the competence boundary are sampled more frequently, while consistently solved or consistently failed buckets receive lower probability. We compare this adaptive curriculum against standard uniform sampling and a fixed easy-to-hard curriculum while keeping the RLOO objective, optimizer, and reward function unchanged.

Implementation Experiments were conducted on the Countdown arithmetic reasoning task using the CS224R default project pipeline. Starting from the Qwen2.5-0.5B base model, we completed the default project milestones by implementing SFT, IPO, and RLOO. For our extension, we used the SFT-trained model as a baseline and introduced curriculum-based prompt sampling during RLOO training. In v1, difficulty buckets were constructed from approximately 10k directly evaluated prompts with heuristic assignment for remaining prompts, while v2 increased the number of directly evaluated prompts to 40k and incorporated multi-seed evaluation. We also introduced importance-weight clamping by clipping log-probability ratios to $[-20, 20]$ and capping importance weights at 100 to improve training stability.

Results Across the default project milestones, pass@1 improved steadily from 0.286 after SFT to 0.366 after IPO, ultimately reaching 0.526 following RLOO training. In the v1 experiments for our extension, adaptive curriculum learning appeared to give substantial gains, reaching a peak pass@1 of 0.629 compared to 0.476 for uniform sampling and 0.533 for the fixed curriculum. However, all methods exhibited performance collapse during extended training, and as single-seed runs these gains could not be separated from seed variance. Importance-weight clamping eliminated this instability and enabled more reliable, multi-seed evaluation. Under the stabilized v2 setting, the three samplers were statistically indistinguishable: adaptive reached a mean peak pass@1 of 0.570 compared to 0.568 for the fixed curriculum and 0.549 for uniform sampling, yielding an 0.021 spread smaller than the seed-to-seed variation within a single curriculum condition (up to 0.052). The same overlap held across pass@2, pass@4, pass@8, and pass@16.

Discussion Our results suggest that competence-boundary curricula, using adaptive sampling under our bucket-difficulty discretization and Gaussian-weight prompt-selection strategies, yield no observable improvements in RLOO sample efficiency over a uniform baseline on this task, as differences between experimental curriculum conditions remain within seed noise. The intervention that resolved the estimator stability challenge was clamped importance weighting, which reliably prevented policy collapse that otherwise precluded extended training with RLOO. Future work includes dynamic bucket reassignment during training, broader hyperparameter sweeps for importance-weight clipping, and evaluating competence-boundary curricula on other reasoning domains such as code generation and theorem proving.

Conclusion We presented an adaptive competence-boundary curriculum for RLOO that prioritizes prompts whose estimated success rates are near 50%. Under matched budgets and multi-seed evaluation, we do not observe evidence that curriculum-based prompt selection improves over uniform sampling or fixed difficulty-progression beyond seed noise. While adaptive curricula are a theoretically promising approach to yield improvements in sample efficiency during verifier-based training, our results indicate that further experimentation is necessary to substantiate this hypothesis.

Adaptive Competence-Boundary Curricula for Countdown with RLOO

Karishma Aggarwal
Center for Global and Online Education (CGOE)
Stanford University
aggk@stanford.edu

Sheng-Yong Niu
Department of Biomedical Data Science
Stanford University
syniu@stanford.edu

Daniel Schreck
Department of Computer Science
Stanford University
dschreck@stanford.edu

Abstract

Reinforcement Learning with Leave-One-Out (RLOO) has recently emerged as a simple and effective approach for post-training language models using verifier rewards. However, standard RLOO training samples prompts uniformly from the training distribution, despite large variation in prompt difficulty. As a result, many rollouts provide little learning signal: easy prompts are already solved by the policy, while difficult prompts produce uniformly unsuccessful rollouts. In both cases, leave-one-out advantages collapse, reducing the effectiveness of policy updates.

We investigate whether curriculum-based prompt selection can improve sample efficiency in Countdown arithmetic reasoning. To do so, we implement an adaptive competence-boundary curriculum that groups prompts into difficulty buckets based on offline SFT success rates and dynamically estimates bucket-level success during training. Prompt sampling probabilities are then assigned using a Gaussian weighting function centered at a 50% success rate, encouraging training on prompts near the policy's competence boundary. This results in a computationally-inexpensive modification to the sampling strategy alone, leaving the policy objective, KL regularization, importance weighting, and optimization procedure unchanged.

Under matched training budgets, model architectures, and optimization settings, we find that adaptive curriculum sampling does not improve over uniform sampling beyond seed noise: across our experimental curriculum conditions, the peak pass@1 gap (0.021) is smaller than the seed-to-seed spread within a single sampler (up to 0.052). While adaptive prompt selection at the policy's competence boundary was hypothesized to improve sample efficiency, further experimentation is necessary to substantiate this hypothesis.

1 Introduction

Reinforcement learning has become a central component of post-training pipelines for large language models. Recent approaches such as Reinforcement Learning with Leave-One-Out (RLOO) optimize language models directly against task-specific verifier rewards while avoiding the complexity of value-function estimation. In verifier-based settings, policy improvement depends heavily on the quality of sampled trajectories used during training.

A common assumption in RLOO training is that prompts should be sampled uniformly from the training distribution. However, prompt difficulty often varies substantially. In the Countdown arith-

metic reasoning task, an SFT-initialized model already solves many simple three-number problems, while more difficult four-number problems remain largely unsolved. Consequently, many sampled rollouts provide little useful learning signal. Easy prompts produce uniformly successful rollouts, causing leave-one-out advantages to collapse toward zero, while difficult prompts produce uniformly unsuccessful rollouts with the same effect. In both cases, expensive rollouts contribute minimally to policy improvement.

This observation motivates a curriculum-learning perspective. Educational theory suggests that learning is most effective near the boundary of current competence, often referred to as the Zone of Proximal Development (1). Analogously, we hypothesize that adaptively sampling RLOO prompts from buckets whose running success rate is near 50% (i.e. the competence boundary) improves peak pass@k across rollout budgets, relative to both uniform sampling and a fixed easy-to-hard schedule. Such prompts are expected to produce a mixture of successful and unsuccessful rollouts, yielding higher reward variance and stronger leave-one-out learning signals.

To test this hypothesis, we introduce an adaptive competence-boundary curriculum for Countdown reasoning. Prompts are partitioned into difficulty buckets using offline SFT success rates, and bucket-level success estimates are updated online throughout training. The sampler then preferentially selects prompts from buckets whose current success rate lies near 50%, using a Gaussian weighting function centered at the competence boundary. Importantly, our approach modifies only the prompt-selection mechanism and leaves the RLOO objective, importance weighting, KL regularization, and optimization procedure unchanged. We compare adaptive curriculum sampling against both standard uniform sampling and a fixed easy-to-hard curriculum under matched rollout budgets.

2 Related Work

Curriculum learning has long been studied as a mechanism for improving learning efficiency by controlling the order and difficulty of training examples. Recent work has extended these ideas to large language model (LLM) reasoning tasks, where training examples can vary substantially in complexity and usefulness for learning.

Parashar et al. (2) investigate curriculum reinforcement learning for LLM reasoning and demonstrate that organizing training examples from easier to harder tasks can improve reasoning performance. Their work shows that curriculum design can significantly affect learning dynamics, but relies on a predefined progression through task difficulty. Similarly, Chen et al. (3) propose a self-evolving curriculum framework in which the curriculum adapts throughout training. Their approach dynamically adjusts task selection based on model performance, highlighting the importance of matching training data to the model's current capabilities.

More recently, Sundaram et al. (4) argue that learning is most effective near the edge of a model's current abilities. Their framework encourages training on examples that are neither trivially solvable nor impossible, thereby maximizing the learning signal obtained from each training example. This idea is particularly relevant in reinforcement learning settings where successful and unsuccessful trajectories provide different levels of policy-gradient information.

Our work applies these ideas to Reinforcement Learning with Leave-One-Out (RLOO) on the Countdown reasoning task. Unlike prior curriculum-learning approaches that rely on predefined schedules, learned curriculum policies, or synthetic task generation, we modify only the prompt-selection mechanism while leaving the RLOO objective, importance weighting, KL regularization, and optimization procedure unchanged. We introduce a lightweight adaptive curriculum that tracks bucket-level success rates during training and preferentially samples prompts whose success probability is near 50%, corresponding to the policy's competence boundary. This allows us to study the effect of competence-boundary sampling in isolation under a fixed rollout budget.

3 Method

The default project pipeline consists of three stages: supervised fine-tuning (SFT), preference optimization using IPO, and online reinforcement learning using RLOO. Building upon this pipeline, we investigate whether curriculum-based prompt selection can improve the effectiveness of RLOO training on the Countdown reasoning task.

3.1 Supervised Fine-Tuning

Following the default project pipeline, we first warm-started the Qwen2.5-0.5B base model using supervised fine-tuning (SFT) on the Countdown reasoning task. Given a prompt x and reference completion y, the model was trained using the standard next-token prediction objective applied only to the completion tokens:

$$\mathcal{L}_{\text{SFT}} = -\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t})$$

Prompt tokens were masked from the loss computation so that optimization focused exclusively on learning the desired reasoning trajectory. This stage provides an initialization capable of solving a subset of Countdown problems and serves as the reference model for subsequent alignment stages.

3.2 Identity Preference Optimization

Following supervised fine-tuning, we further align the model using Identity Preference Optimization (IPO). IPO operates on preference pairs consisting of a preferred response y_w and a rejected response y_l for the same prompt.

Given a reference policy π_{ref} obtained from SFT, IPO optimizes the relative preference margin between chosen and rejected responses while remaining close to the reference model. Unlike reinforcement learning methods, IPO does not require online trajectory generation or reward modeling, making it a computationally efficient intermediate alignment stage.

This stage produces a stronger initialization for reinforcement learning by increasing the likelihood of high-quality reasoning trajectories relative to lower-quality alternatives.

3.3 Reinforcement Learning with Leave-One-Out

Our primary reinforcement learning baseline is Reinforcement Learning with Leave-One-Out (RLOO). During training, the policy generates K rollouts for each prompt. Each rollout is evaluated using the Countdown verifier, which assigns a reward based on formatting correctness and arithmetic validity.

To reduce gradient variance, RLOO computes a leave-one-out baseline for every rollout:

$$b_i = \frac{1}{K-1} \sum_{j \neq i} r_j,$$

where r_j denotes the verifier reward of rollout j. The corresponding advantage estimate is

$$A_i = r_i - b_i.$$

Policy updates are then performed using REINFORCE-style gradients weighted by these advantages.

Following the project specification, rollout generation is performed using vLLM while policy updates are computed using a Hugging Face model. To account for slight discrepancies between the behavior policy and target policy, we apply sequence-level importance weighting:

$$w(y, x) = \exp\left(\log \pi_\theta(y|x) - \log \mu(y|x)\right),$$

where μ denotes the behavior policy used during rollout generation. For numerical stability, importance weights are computed in log-space and clipped before optimization.

3.4 Curriculum-Based Prompt Sampling

While standard RLOO samples prompts uniformly from the training set, prompt difficulty varies substantially across Countdown problems. Easy prompts are frequently solved by the policy, while

difficult prompts often yield uniformly unsuccessful rollouts. In both cases, reward variance collapses and the resulting leave-one-out advantages provide limited learning signal.

To address this issue, we investigate curriculum-based prompt selection strategies that prioritize prompts near the model's current competence boundary (Figure 1).

3.4.1 Difficulty Estimation and Bucket Assignment

Prior to reinforcement learning, we estimate the difficulty of each training prompt using the supervised fine-tuned model. For every prompt, multiple completions are generated and evaluated using the Countdown verifier. The resulting success rate $s \in [0, 1]$ provides an estimate of prompt difficulty.

Following prior curriculum-learning work, prompts are partitioned into three difficulty levels:

$$\textbf{hard} : s < 0.3, \quad \textbf{medium} : 0.3 \leq s \leq 0.7, \quad \textbf{easy} : s > 0.7.$$

We further separate prompts based on Countdown problem size (3-number and 4-number tasks), producing six buckets: $[3_{\text{hard}}, 3_{\text{medium}}, 3_{\text{easy}}, 4_{\text{hard}}, 4_{\text{medium}}, 4_{\text{easy}}]$. Bucket assignments remain fixed throughout training.

3.4.2 Fixed Curriculum

As a curriculum baseline, we implement a fixed easy-to-hard schedule inspired by traditional curriculum learning. Buckets are ordered according to their initial SFT success rates, and sampling probabilities are manually scheduled to gradually shift training from easier prompts toward more difficult prompts.

This strategy modifies the training distribution while remaining independent of the policy's evolving performance.

3.4.3 Adaptive Competence-Boundary Curriculum

Our primary contribution is an adaptive curriculum that continuously adjusts prompt sampling based on the model's observed success rates.

For each bucket b, we maintain an estimate of the policy's success probability p_b. After each training iteration, bucket statistics are updated using an exponential moving average:

$$p_b \leftarrow (1 - \alpha)p_b + \alpha\hat{p}_b,$$

where $\hat{p}_b$ is the observed success rate among prompts sampled from bucket b during the current iteration.

To prioritize prompts near the competence boundary, we assign bucket weights using a Gaussian function centered at a success rate of 50%:

$$w_b = \exp\left(-\frac{(p_b - 0.5)^2}{2\sigma^2}\right)$$

Buckets whose success rates are close to 0.5 receive the highest weight, while buckets that are consistently solved or consistently failed receive lower weight (Appendix Figure 4). The weights are normalized to form a sampling distribution:

$$P(b) = \frac{w_b}{\sum_j w_j}.$$

During training, a bucket is first sampled according to $P(b)$, after which a prompt is selected uniformly from within the chosen bucket.

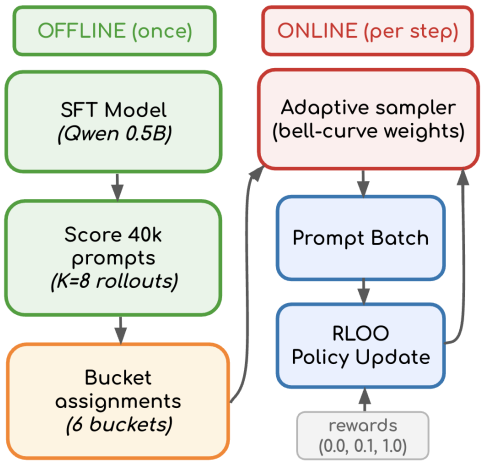


Figure 1: Curriculum-based prompt sampling. Offline, the SFT policy scores a 40k-prompt subset (K=8 rollouts per prompt) to assign six difficulty buckets. Online, an adaptive sampler weights buckets by a Gaussian centered at 50% success and draws each RLOO prompt batch near the competence boundary; per-bucket success estimates that drive the adaptive sampler are updated with an EMA after every batch.

Intuitively, prompts near the competence boundary produce a mixture of successful and unsuccessful rollouts, resulting in larger reward variance and stronger leave-one-out learning signals. By concentrating training on these prompts, the curriculum aims to improve the efficiency of RLOO without modifying the underlying reinforcement learning objective.

4 Experimental Setup

4.1 Task and Model

We evaluate curriculum-based prompt sampling on the Countdown arithmetic reasoning task using the default CS224R project pipeline. All experiments initialize from the supervised fine-tuned Qwen2.5-0.5B checkpoint (`asingh15/qwen-sft-countdown-defaultproj`) and train using the `countdown_tasks_3to4` dataset. Evaluation is performed on a held-out test set using `pass@`k metrics.

4.2 Sampling Conditions

We compare three prompt-sampling strategies:

- **Uniform**: standard RLOO prompt sampling.
- **Fixed**: easy-to-hard curriculum based on offline difficulty estimates.
- **Adaptive**: competence-boundary curriculum that prioritizes buckets with estimated success rates near 50%.

All other components of the training pipeline, including the RLOO objective, verifier reward, optimizer, KL regularization, and model initialization, are identical across conditions.

4.3 v1 Experiments

Our primary experiments (v1) use six buckets defined by Countdown problem size (3-number vs. 4-number) and SFT-estimated difficulty (easy, medium, hard). Difficulty labels are obtained by directly evaluating approximately 10k prompts and heuristically assigning labels to the remaining prompts.

Table 1: RLOO hyperparameters used in v1 experiments.

Parameter	Value
Training steps	300
Batch size	128
Group size	8
Rollouts / step	1024
Total rollouts	307,200
Learning rate	1×10^{-5}
KL coefficient	10^{-3}
Entropy coefficient	10^{-3}

Table 1 summarizes the training configuration used for all v1 runs.

We report results for two versions of v1:

- **v1-No-IW**: original poster results without using importance weighting.
- **v1-Clamped-IW**: identical setup with importance-weight clipping for improved training stability.

4.4 v2 Experiments

The v1 experiments were limited by sparse bucket labeling and single-seed evaluation. Only approximately 10k of the 490k training prompts were directly scored using the SFT model, with the remaining prompts assigned heuristic difficulty labels. In addition, each condition was evaluated using a single random seed.

To address these limitations, v2 performs dense offline difficulty estimation over a restricted subset of the training set (40k prompts, $K = 8$ rollouts per prompt) and evaluates multiple seeds per condition. The training horizon is also reduced from 300 to 150 steps to focus on the productive learning phase before policy collapse becomes dominant.

The goal of v2 is to validate whether the curriculum gains observed in v1 remain robust under higher-quality bucket assignments and reduced seed variance. Similar to v1, we report the results on two variants - with and without clamped importance weighting.

Table 2: Summary of experimental configurations.

Configuration	Description
v1-No-IW	10k prompts scored ($K = 8$), heuristic bucket assignment for remaining prompts, 300 training steps, 1 seed, no importance weighting
v1-Clamped-IW	Same as v1-No-IW, but with log-ratio clipping to $[-20, 20]$ and importance weights capped at 100
v2-No-IW	40k prompts scored ($K = 8$), curriculum samplers restricted to pre-scored prompts only, 150 training steps, 3 seeds, no importance weighting
v2-Clamped-IW	40k prompts scored ($K = 8$), curriculum samplers restricted to pre-scored prompts only, 150 training steps, 2 seeds, clamped importance weighting

5 Results

We evaluate curriculum-based prompt sampling under two experimental settings. The initial v1 experiments suggested that adaptive curriculum learning could substantially improve RLOO performance. However, these experiments relied on sparse bucket assignments and were sensitive to training instability. We therefore conducted a second experimental iteration (v2) using full-dataset bucket labeling, multiple random seeds, and clamped importance weighting to determine whether the observed gains were robust.

Across both iterations, two main findings emerge. First, importance-weight clamping substantially improves training stability and prevents the policy collapse observed in standard RLOO training. Second, once training is stabilized and difficulty estimates are made more reliable, curriculum-based prompt selection yields no observable improvement over uniform sampling beyond seed noise; the gap between sampling strategies is consistently smaller than the seed-to-seed variation within a single sampling strategy, in both the stabilized and the unstabilized regime.

5.1 Quantitative Evaluation

Figure 2 shows pass@1 as a function of cumulative rollouts across all four experimental configurations. In the original v1-No-IW setting, the adaptive curriculum achieved the strongest peak performance, reaching pass@1 of 0.629 compared to 0.533 for the fixed curriculum and 0.476 for uniform sampling. However, all three conditions eventually exhibited severe performance degradation later in training, making it difficult to determine whether the observed gains were due to curriculum learning or instability in the optimization procedure.

To improve stability, we introduced importance-weight clamping and re-ran the v1 experiments. While clamping successfully prevented collapse, the adaptive curriculum no longer consistently outperformed the baselines. Figure 3 shows that uniform sampling achieved comparable or stronger peak pass@k performance in several settings. We attribute this result to the limitations of the v1 bucket construction procedure. Only approximately 10k training prompts were directly scored using the SFT model, while the remaining prompts inherited heuristic labels based on median success rates within their number-count group. This coarse assignment likely introduced significant bucket noise and weakened the curriculum signal.

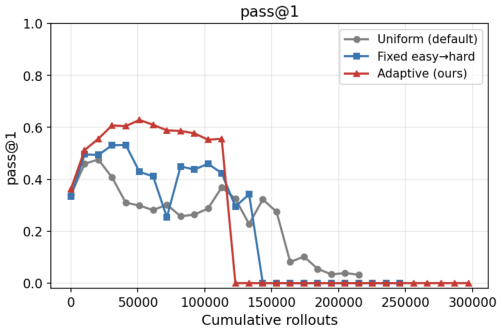
The v2 experiments addressed this limitation by increasing the number of directly evaluated prompts from 10k to 40k, restricting the sampler to draw only from these pre-scored prompts (rather than using heuristic-labeled data), and evaluating multiple random seeds. By reducing reliance on heuristic bucket assignment, v2 provides a more accurate estimate of prompt difficulty and therefore a cleaner test of the curriculum hypothesis.

Under this setting, the three sampling strategies are statistically indistinguishable, as shown in Figure 3. Adaptive sampling reached a mean peak pass@1 of 0.570, compared to 0.568 for the fixed curriculum and 0.549 for uniform sampling. Although adaptive is nominally highest, the 0.021 spread across samplers is smaller than the seed-to-seed spread within a single sampler (up to 0.052 for the uniform arm, $n = 2$ seeds), so the apparent ordering lies within noise. The same overlap holds for pass@2 through pass@16. We therefore find no evidence that competence-boundary sampling improves RLOO over uniform sampling on this task.

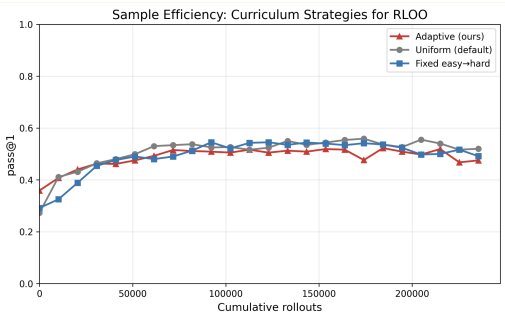
5.2 Qualitative Analysis

The adaptive curriculum successfully altered the distribution of prompts encountered during training. Rather than sampling uniformly across all buckets, the competence-boundary sampler concentrated probability mass on buckets whose estimated success rates remained near 50%, while reducing emphasis on prompts that were consistently solved or consistently failed. Yet despite this clear shift in the training distribution, peak performance did not improve over uniform sampling. This suggests that the leave-one-out baseline may already provide sufficient per-prompt variance reduction, so that reshaping which prompts are sampled adds little additional learning signal. Moreover, since the medium bucket spans the 0.3–0.7 success range (Appendix Fig. 5), the adaptive sampler primarily concentrates bucket weight here rather than on prompts specifically near the 50% boundary it aims to target.

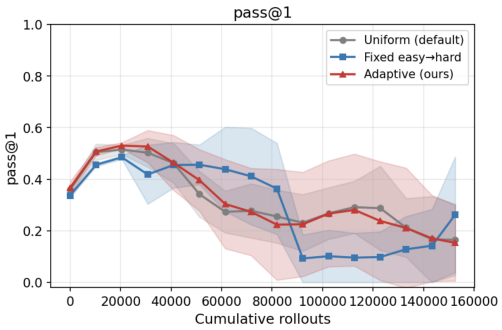
A second key observation concerns training stability. In the original RLOO implementation, large importance ratios occasionally produced unstable updates and led to policy collapse during extended training. Clipping the log-probability difference to $[-20, 20]$ and capping importance weights at 100 eliminated this collapse and produced stable learning curves across all curriculum conditions. While these conservative clipping thresholds improved stability, they may also limit policy improvement by suppressing large but potentially useful updates. Future work could explore alternative clipping ranges or adaptive clipping schedules to better balance stability and performance.



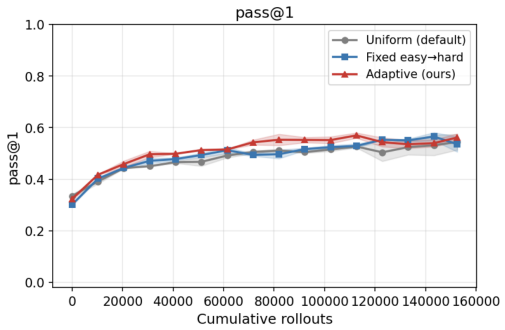
(a) v1-No-IW



(b) v1-Clamped-IW

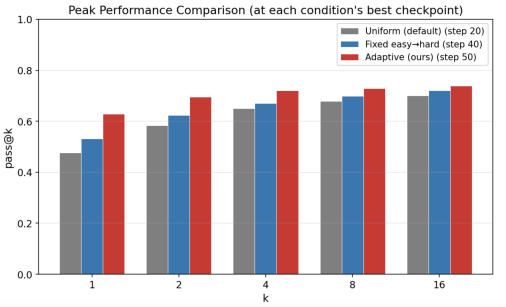


(c) v2-No-IW

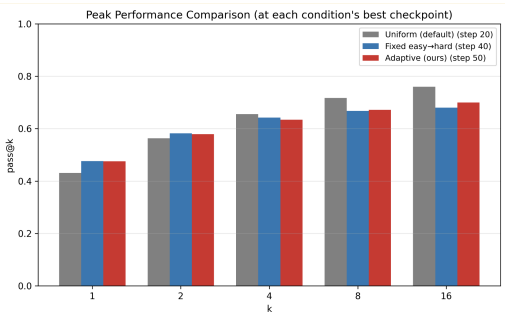


(d) v2-Clamped-IW

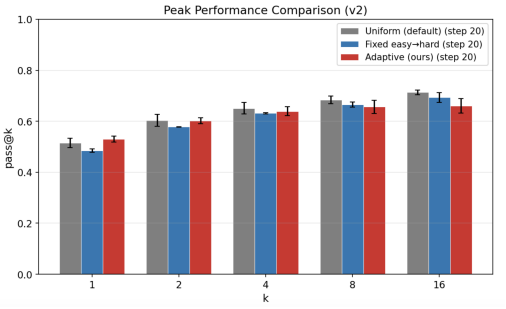
Figure 2: Pass@1 over cumulative rollouts across all experimental configurations. v1-No-IW shows the original curriculum result but suffers from instability; clamped-IW runs are more stable, and v2 uses full-dataset bucket assignments and multiple seeds.



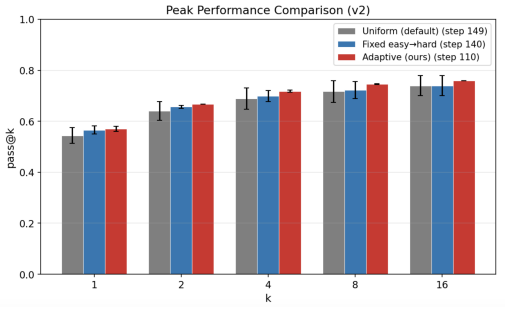
(a) v1-No-IW



(b) v1-Clamped-IW



(c) v2-No-IW



(d) v2-Clamped-IW

Figure 3: Peak pass@k comparison across all experimental configurations. The v2-Clamped-IW setup provides the cleanest comparison because it combines stable optimization with full-dataset bucket assignments.

6 Discussion

As the differences between sampling conditions remained within seed noise in both the unstable and stabilized regimes, our results suggest that competence-boundary curricula do not yield observable improvements in RLOO performance over a uniform baseline on this task, even with more accurate difficulty estimates.

We also find that importance-weight clamping is an effective mechanism for stabilizing RLOO training. By preventing extreme importance ratios, clamping eliminates the policy collapse observed in the original runs and enables more reliable evaluation of curriculum-learning strategies.

Finally, the marginal between-condition differences in v2 remain whether we deploy or omit the importance-weight estimator (i.e. the IW-clamped and no-IW runs). The larger differences in the original v1 experiments appeared in a single-seed experiment, and did not persist with multiple-seed runs regardless of clamping. We therefore attribute our v1 observations to seed variance rather than to a curriculum effect that stabilization removed; clamping's role is therefore interpreted to make extended training stable enough for a clean multi-seed comparison, rather than suppressing an underlying advantage for the adaptive condition.

7 Conclusion

In this work, we investigated curriculum-based prompt sampling as an extension to Reinforcement Learning with Leave-One-Out (RLOO) for the Countdown reasoning task. We introduced an adaptive competence-boundary curriculum that prioritizes prompts whose estimated success rates are close to 50%, with the goal of focusing training on examples that provide the strongest learning signal. We compared this approach against both uniform sampling and a fixed easy-to-hard curriculum under multiple experimental settings.

Under matched budgets and multi-seed evaluation, we do not observe evidence that curriculum-based prompt selection improves over uniform sampling or a fixed difficulty progression beyond seed noise. The intervention that resolved estimator stability challenges was clamped importance weighting, which reliably prevented policy collapse that otherwise precluded extended training with RLOO. While adaptive curricula remain a theoretically promising approach to improving sample efficiency in verifier-based training, our results indicate that further experimentation is necessary to substantiate this hypothesis.

Future work could explore more granular and dynamic bucketing strategies, including updating prompt difficulty assignments as the policy improves. Such experimentation may determine why the adaptive-sampling approach developed here did not yield performance and sample-efficiency improvements as hypothesized. It would also be valuable to further perform broader hyperparameter sweeps over both the curriculum and importance-weight clipping parameters, and evaluate whether competence-boundary curricula generalize beyond Countdown to other verifier-based reasoning tasks such as code generation, theorem proving, and scientific reasoning.

8 Team Contributions

- **Karishma Aggarwal:** Led the implementation and training of Reinforcement Learning with Leave-One-Out (RLOO), including rollout collection, policy-gradient updates, importance weighting, KL regularization, and training-stability analysis. She ran the v1 experiments with importance weighting, analyzed experimental results, contributed to the project poster, and led the writing and integration of the final report.
- **Sheng-Yong Niu:** Led the implementation of supervised fine-tuning (SFT), including completion-only loss masking, training and evaluation pipelines, and metrics logging. He led the curriculum-learning extension implementation, including difficulty bucketing, adaptive prompt sampling, and evaluation infrastructure. He ran the v1 experiments without importance weighting, analyzed results, and contributed to both the project poster and final report.
- **Daniel Schreck:** Led the implementation and training of Identity Preference Optimization (IPO), including the pairwise loss and metrics analysis. He ran the v2 multi-seed experiments

with importance weighting inclusion and omission, dense offline difficulty scoring, and sampler restriction to the prescored-only prompt subset. He contributed to the final report and led poster development.

Changes from Proposal: The adaptive sampler's target band was implemented as a smooth Gaussian weight centered at 50% success ($\sigma = 0.15$) rather than the hard 30–70% selection originally described. To separate curriculum effects from seed variance, we strengthened the evaluation with dense offline difficulty scoring (40k prompts, $K = 8$ rollouts each), restricted all samplers to this directly-scored pool, and evaluated with multiple random seeds. The optional self-play and cross-domain transfer directions were not investigated.

References

[1] Vygotsky, L. S. (1978). Mind in society: The development of higher psychological processes. *Harvard University Press*.

[2] Parashar, S., Gui, S., Li, X., Ling, H., Vemuri, S., Olson, B., Li, E., Zhang, Y., Caverlee, J., Kalathil, D., and Ji, S. (2025). Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. *arXiv preprint arXiv:2506.06632*.

[3] Chen, X., Lu, J., Kim, M., Zhang, D., Tang, J., Piché, A., Gontier, N., Bengio, Y., and Kamalloo, E. (2025). Self-evolving curriculum for LLM reasoning. *arXiv preprint arXiv:2505.14970*.

[4] Sundaram, S., Quan, J., Kwiatkowski, A., Ahuja, K., Ollivier, Y., and Kempe, J. (2026). Teaching models to teach themselves: Reasoning at the edge of learnability. *arXiv preprint arXiv:2601.18778*.

A Appendix

A.1 Additional Figures

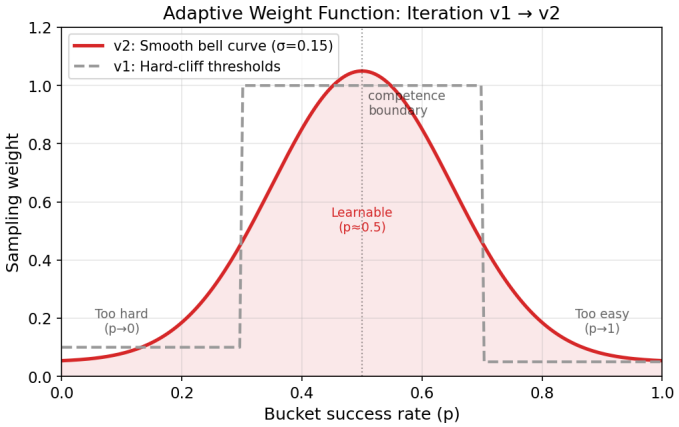


Figure 4: Adaptive competence-boundary sampling weight as a function of a bucket's running success rate p. Weight peaks at the competence boundary ($p \approx 0.5$) and falls off for buckets that are either consistently solved ($p \approx 1$) or failed ($p \approx 0$). The original-design hard-threshold scheme (dashed) was replaced in v1 by a smooth Gaussian ($\sigma = 0.15$, solid).

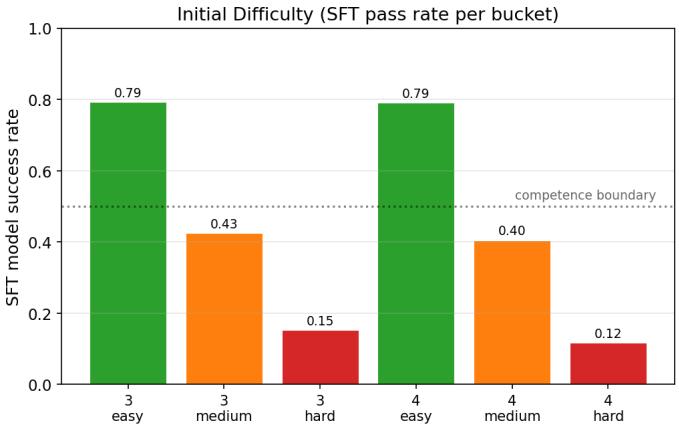


Figure 5: Initial SFT-model pass rate for each difficulty bucket in the v2 (40k directly-scored) prompt pool. Difficulty bins use fixed success-rate thresholds (hard < 0.3, medium 0.3–0.7, easy > 0.7). No bucket lies near the 0.5 competence boundary; the medium buckets (0.40–0.43) are closest, and span the $[0.3, 0.7]$ range. This coarse bucketing resolution near the boundary is among the likely reasons the observed curriculum signal is weak.