Curriculum Sampling for RLOO Fine-Tuning on Countdown

Catherine Zhang
czhang7@stanford.edu

Nora Menon
nmenon26@stanford.edu

Extended Abstract

Reinforcement Learning from Verifiable Rewards (RLVR) has emerged as a promising approach for improving language-model reasoning on tasks with objective correctness signals. Reinforcement Learning with Leave-One-Out (RLOO) is a computationally efficient RLVR algorithm that uses multiple sampled responses to estimate relative advantages without training a separate value function. However, standard RLOO samples prompts uniformly from the training distribution and therefore ignores differences in prompt difficulty and learning potential. In practice, some prompts are already mastered by the model, while others are so difficult that they provide little useful reward signal. This project investigates whether curriculum-based sampling can improve RLOO training by concentrating updates on prompts that are expected to be most informative.

We introduce two curriculum-learning extensions for RLOO on the Countdown arithmetic reasoning benchmark. The first, a *Fixed-Difficulty Curriculum*, partitions problems into 3-number and 4-number buckets and dynamically adjusts sampling probabilities using bucket-level success estimates. Two variants are explored: one that considers a prompt successful when at least one rollout is correct and a stricter version that requires two correct rollouts. The second, a *Reward-Adaptive Curriculum*, estimates prompt learnability directly from observed rewards using prompt-level exponential moving averages (EMAs). Prompts are categorized as too hard, learnable, or too easy, and sampling probabilities are adjusted to prioritize prompts near the model's current learning frontier. Unlike prior work that relies on predefined easy-to-hard schedules, our approach investigates whether reward signals generated during RL training can be used directly to guide curriculum selection.

All methods were implemented within the CS224R RLOO training framework and evaluated on the Countdown dataset (`asingh15/countdown_tasks_3to4`). Experiments used a Qwen 2.5 0.5B model initialized from the shared supervised fine-tuning checkpoint and trained for 100 RLOO steps with a learning rate of 1×10^{-5}, batch size 128, and rollout group size 8. Performance was measured using pass@k for $k \in \{1, 2, 4, 8, 16\}$. The uniform-sampling RLOO baseline achieved the strongest results, reaching pass@1 of 0.4300 and pass@16 of 0.6800. Fixed-Difficulty Curriculum variants achieved pass@1 scores of 0.3812 and 0.3762, while Reward-Adaptive Curriculum variants achieved pass@1 scores of 0.3312 and 0.3425. Difficulty-specific analysis revealed that curriculum sampling substantially altered learning behavior. Fixed-difficulty curricula largely preserved performance on harder 4-number problems while reducing performance on easier 3-number problems. Reward-adaptive curricula were associated with improved 3-number performance and substantially reduced 4-number performance, suggesting increased emphasis on easier learnable prompts.

Our results show that neither curriculum family outperformed the uniform RLOO baseline despite successfully modifying the distribution of training examples. These findings suggest a fundamental tradeoff between concentrating updates on potentially informative prompts and maintaining broad coverage of the training distribution. In the Countdown setting, broad exposure to both easy and difficult examples appears more valuable than aggressively targeting prompts estimated to be near the model's learning frontier. Several limitations should be noted, including the relatively small dataset, short training horizon, and limited evaluation set. Future work could explore curricula that explicitly balance learnability and coverage through minimum-exposure guarantees, uncertainty-aware exploration, or longer training schedules. Overall, this project demonstrates that reward-informed cur-

riculum sampling substantially changes learning dynamics, but that identifying informative prompts alone is insufficient to improve verifier-based reinforcement learning performance.

Abstract

Reinforcement Learning with Leave-One-Out (RLOO) has demonstrated strong performance on reasoning tasks with verifiable rewards, but standard implementations sample training prompts uniformly and therefore ignore differences in prompt difficulty and learning potential. We investigate whether curriculum-based sampling can improve RLOO fine-tuning by concentrating updates on prompts that provide the most informative reward signal. We introduce two curriculum-learning extensions for the Countdown arithmetic task. The first, a Fixed-Difficulty Curriculum, partitions prompts into 3-number and 4-number difficulty buckets and adjusts sampling probabilities using bucket-level success estimates. The second, a Reward-Adaptive Curriculum, dynamically categorizes prompts as too hard, learnable, or too easy using prompt-level success EMAs and prioritizes prompts near the model's current learning frontier. We evaluate both approaches against a uniform-sampling RLOO baseline. While the baseline achieves the strongest performance, reaching pass@1 of 0.4300 and pass@16 of 0.6800, curriculum variants achieve pass@1 scores between 0.3312 and 0.3812 and pass@16 scores between 0.6400 and 0.6600. Difficulty-specific analysis reveals that curriculum sampling substantially changes model behavior, often improving performance on easier problems while reducing performance on more challenging reasoning tasks. Our results suggest that, in the Countdown setting, maintaining broad coverage of the training distribution is more important than aggressively concentrating updates on prompts estimated to be informative. These findings highlight a fundamental tradeoff between curriculum targeting and training-data coverage in verifier-based RL fine-tuning with smaller datasets.

1 Introduction

Recent advances in reinforcement learning from verifiable rewards (RLVR) have demonstrated that language models can acquire stronger reasoning capabilities when trained using objective correctness signals rather than human preference judgments. Tasks such as mathematical reasoning, code generation, and symbolic problem solving naturally provide binary or scalar rewards that can be automatically verified, enabling large-scale reinforcement learning without expensive human annotation. Within this paradigm, Reinforcement Learning with Leave-One-Out (RLOO) has emerged as a computationally efficient policy-gradient method that leverages multiple sampled rollouts to estimate relative advantages and improve reasoning performance.

Despite its success, standard RLOO treats all training prompts equally by sampling uniformly from the training distribution. This assumption may be suboptimal. In practice, prompts vary substantially in difficulty and in the amount of useful learning signal they provide. Some prompts are already solved consistently by the model and may contribute little additional information. Others are so difficult that nearly all sampled rollouts receive zero reward, producing weak or noisy optimization signals. Intuitively, the most informative prompts may lie between these extremes: tasks that the model can sometimes solve but has not yet mastered.

This observation is closely related to curriculum learning, which seeks to improve training by controlling the order or frequency with which examples are presented to a model. Curriculum-based methods have been explored in supervised learning, reinforcement learning, and language model training, where they often aim to match training difficulty to the learner's current capabilities. However, relatively little work has examined how curriculum design interacts with verifier-based reinforcement learning and reward-driven optimization. In particular, it remains unclear whether concentrating updates on prompts that appear most informative can improve the efficiency of RLOO fine-tuning.

In this project, we investigate the hypothesis that language models learn most effectively near the *edge of learnability*: prompts that are neither consistently solved nor consistently failed. To test this idea, we introduce two curriculum-learning extensions for RLOO on the Countdown arithmetic

benchmark. The first, a *Fixed-Difficulty Curriculum*, partitions prompts according to structural difficulty, 3-number versus 4-number problems, and adjusts sampling probabilities using bucket-level success estimates. The second, a *Reward-Adaptive Curriculum*, estimates prompt difficulty directly from observed rewards and dynamically categorizes prompts as too hard, learnable, or too easy.

Our work addresses the following research questions:

1. Can curriculum-based sampling improve RLOO fine-tuning compared to uniform prompt sampling?
2. Does prioritizing prompts near the model's current learning frontier improve functional correctness on Countdown tasks?
3. How do fixed difficulty-based curricula compare to reward-adaptive curricula that estimate learnability directly from training rewards?

Across multiple curriculum variants, we find that neither fixed-difficulty nor reward-adaptive sampling outperforms a uniform RLOO baseline. However, the curricula substantially alter the distribution of learning across problem difficulties. Fixed-difficulty curricula trade performance on easier tasks for improved hard-task coverage, while reward-adaptive curricula increasingly concentrate training on easier learnable prompts and significantly degrade performance on difficult 4-number problems. These findings suggest that, for verifier-based reinforcement learning in Countdown, maintaining broad coverage of the training distribution may be more important than aggressively concentrating updates on prompts estimated to be most informative.

Rather than demonstrating a successful curriculum-learning strategy, our results reveal a fundamental tradeoff between targeted optimization and training-data coverage. Understanding this tradeoff is important for future RLVR systems that must balance learning efficiency with robust generalization.

2 Related Work

2.1 Reinforcement Learning from Verifiable Rewards

Recent advances in language model alignment have increasingly explored reinforcement learning from verifiable rewards (RLVR) as an alternative to preference-based approaches such as Reinforcement Learning from Human Feedback (RLHF). In RLVR settings, correctness can be determined automatically using a verifier, making the approach particularly well-suited for domains such as mathematics, coding, and symbolic reasoning. Unlike preference-based rewards, verifier rewards provide objective supervision and eliminate the need for large-scale human annotation.

Several recent reasoning-focused language model systems have demonstrated that reinforcement learning with verifiable rewards can substantially improve performance on tasks with well-defined correctness criteria. DeepSeek-R1, for example, shows that reasoning capabilities can be incentivized through rule-based reward signals [DeepSeek-AI et al., 2025]. Countdown arithmetic provides a particularly useful benchmark because solutions can be verified automatically and require multi-step symbolic reasoning. Our work operates within this RLVR framework and investigates whether training efficiency can be improved through curriculum-based sampling.

2.2 RLOO and Policy Optimization for Language Models

Reinforcement Learning with Leave-One-Out (RLOO) is a policy-gradient method designed to reduce the computational cost of reinforcement learning for language models while maintaining stable optimization. Instead of training a separate value function, RLOO estimates advantages using multiple sampled responses to the same prompt and compares each response against the average reward of the remaining responses. This approach avoids the complexity of actor-critic methods while providing a useful baseline for policy updates.

Prior work on RLOO primarily focuses on improving optimization stability, reward modeling, or overall alignment performance. In contrast, our work investigates the interaction between RLOO and the training data distribution itself. Rather than modifying the policy optimization objective, we explore whether selectively choosing which prompts are presented during training can improve learning efficiency.

2.3 Curriculum Learning and Adaptive Data Selection

Curriculum learning aims to improve training by presenting examples in a structured order, typically progressing from easier examples to more difficult ones. Since the seminal work of Bengio et al. [**?**], curriculum strategies have been applied across supervised learning, reinforcement learning, and large-scale language model training. Related approaches such as self-paced learning and learning-progress-based curricula attempt to adapt the training distribution based on the learner's current competence.

Recent work has applied curriculum learning to RL fine-tuning for language-model reasoning. Parashar et al. [Parashar et al., 2025] propose an easy-to-hard curriculum for reinforcement learning with verifiable rewards, showing that gradually scheduling tasks from easier to harder examples can improve reasoning performance. Chen et al. [Chen et al., 2025] propose a self-evolving curriculum that frames curriculum selection as a non-stationary multi-armed bandit problem and uses policy-gradient advantage as a proxy for learning gain. These works are closely related to our goal of improving RLVR through better data selection, but our approach uses a simpler mechanism based on verifier-measured rollout success rates.

A common intuition underlying these methods is that examples near the learner's current capability provide the most useful learning signal. Examples that are already mastered contribute little information, while examples that are consistently failed may be too difficult to drive meaningful improvement. Sundaram et al. [Sundaram et al., 2026] study this idea through reasoning at the "edge of learnability," where models improve through automatically generated stepping-stone problems. Our work builds on this intuition but does not generate new problems; instead, we investigate whether reward signals observed during RLOO training can be used to estimate prompt learnability and guide sampling toward existing Countdown prompts near the model's current learning frontier.

2.4 Positioning of Our Work

Unlike prior curriculum-learning approaches that rely solely on predefined difficulty measures or explicit easy-to-hard schedules [Parashar et al., 2025], we study both a fixed structural curriculum and a reward-adaptive curriculum within the same RLOO framework. The fixed-difficulty curriculum uses the number of integers in a Countdown puzzle as a coarse difficulty estimate, while the reward-adaptive curriculum directly estimates learnability using prompt-level reward statistics collected during training.

Our contribution is not a new policy optimization algorithm. Instead, we investigate a previously underexplored question in RLVR: whether reward-informed curriculum sampling can improve the efficiency of verifier-based reinforcement learning. Through a systematic comparison of fixed-difficulty and reward-adaptive curricula, we identify a tradeoff between concentrating updates on informative prompts and maintaining broad coverage of the training distribution.

3 Method

3.1 Baseline: Reinforcement Learning with Leave-One-Out

Our work builds upon Reinforcement Learning with Leave-One-Out (RLOO), a policy-gradient method designed for reinforcement learning from verifiable rewards. For each training prompt x_i, the policy generates a group of K responses:

$$\{y_{i,1}, y_{i,2}, \ldots, y_{i,K}\}.$$

Each response is evaluated by a verifier that assigns a scalar reward:

$$r_{i,k} \in \{0, 0.1, 1\},$$

where a reward of 1 indicates a correct Countdown solution, 0.1 indicates a valid but incorrect answer format, and 0 indicates that no valid solution was extracted.

RLOO estimates the advantage of each rollout by comparing its reward against the average reward of the remaining rollouts generated for the same prompt:

$$A_{i,k} = r_{i,k} - \frac{1}{K-1} \sum_{j \neq k} r_{i,j}.$$

Responses with above-average reward receive positive advantages, while responses with below-average reward receive negative advantages. The policy is then updated using these leave-one-out advantages together with importance-weighted policy gradients and KL regularization.

Standard RLOO samples prompts uniformly from the training dataset. Our extension modifies this sampling procedure while leaving the underlying policy optimization objective unchanged.

3.2 Motivation: Learning at the Edge of Learnability

The central hypothesis of this work is that not all prompts contribute equally useful learning signal. Intuitively, prompts fall into three categories:

- **Too easy**: the model already solves them consistently.
- **Too hard**: the model almost never solves them.
- **Learnable**: the model succeeds intermittently but has not yet mastered them.

For RLOO, the most informative gradients arise when rewards vary across rollouts. If every rollout receives zero reward or every rollout receives perfect reward, the resulting learning signal is weak. We therefore investigate whether concentrating training on prompts near the model's current capability can improve learning efficiency.

To test this idea, we implement two curriculum-learning strategies: a fixed-difficulty curriculum and a reward-adaptive curriculum.

3.3 Fixed-Difficulty Curriculum

The first curriculum uses a predefined notion of difficulty based on the structure of Countdown problems. Training examples are partitioned into two buckets:

$$d_i = \begin{cases} 3 & \text{if prompt } i \text{ contains three integers,} \\ 4 & \text{if prompt } i \text{ contains four integers.} \end{cases}$$

Here, 3-number problems are treated as easier and 4-number problems as harder.

For each bucket b, we maintain an exponential moving average (EMA) of success:

$$s_b^{(t)} = \alpha s_b^{(t-1)} + (1-\alpha)\hat{s}_b^{(t)},$$

where $\hat{s}_b^{(t)}$ is the observed success rate for prompts in bucket b and α is the EMA decay parameter.

The hard bucket becomes fully available once the easy bucket reaches a predefined mastery threshold:

$$s_{\text{easy}} \geq \tau_{\text{unlock}}.$$

Sampling probabilities are then adjusted dynamically using bucket-level EMAs. Buckets that appear less mastered receive higher sampling weight, encouraging the model to focus on areas where performance remains weak.

Fixed-Bucket v1. A prompt is considered successful if at least one rollout is correct:

$$\mathbf{1}\left\{\max_k r_{i,k} = 1\right\}.$$

Under this criterion, even a single correct rollout contributes positively to the bucket success estimate.

Fixed-Bucket v2. To reduce sensitivity to isolated lucky generations, we introduce a stricter success criterion. A prompt is considered successful only if at least two rollouts are correct:

$$\mathbf{1}\left\{\sum_{k=1}^{K} \mathbf{1}[r_{i,k} = 1] \geq 2\right\}.$$

This modification produces a more conservative estimate of competence and requires repeated success before a prompt contributes positively to the bucket EMA.

3.4 Reward-Adaptive Curriculum

While the fixed curriculum relies on a predefined difficulty measure, the reward-adaptive curriculum estimates difficulty directly from observed training rewards.

For each prompt i, we maintain a prompt-level success EMA:

$$\hat{s}_i^{(t)} = \alpha \hat{s}_i^{(t-1)} + (1-\alpha)\left(\frac{1}{K}\sum_{k=1}^{K} r_{i,k}\right).$$

Prompts are dynamically assigned to one of four categories:

- **Unknown**: insufficient observations.
- **Too Hard**: success rate below the learnable region.
- **Learnable**: intermediate success rate.
- **Too Easy**: success rate above the learnable region.

The curriculum then adjusts sampling probabilities according to prompt category.

Reward-Adaptive v1. The first reward-adaptive curriculum uses a broad learnability range:

$$0.1 \leq \hat{s}_i \leq 0.7.$$

Prompts within this region are considered learnable and receive the highest sampling weight. Too-hard and too-easy prompts receive reduced weight, while a small fraction of training examples are drawn uniformly to maintain exploration. Specifically, we sample 90% from "learnable" prompts and 10% uniformly.

Reward-Adaptive v2. The second reward-adaptive curriculum introduces several modifications intended to reduce noise:

1. Narrower learnable region:
$$0.2 \leq \hat{s}_i \leq 0.6.$$
2. Increased uniform exploration (samples uniformly 30% instead of 10%).
3. A 20-step uniform warmup phase.
4. Higher sampling weights for prompts outside the learnable region.

These changes were designed to reduce sensitivity to early noisy reward estimates and improve the stability of prompt classification.

3.5 Novelty Relative to Prior Work

Prior curriculum-learning methods typically rely on predefined difficulty measures or externally constructed training schedules. Our approach instead investigates whether reward signals generated during RLOO training can serve as a proxy for prompt learnability. By comparing fixed-difficulty and reward-adaptive curricula within the same reinforcement learning framework, we directly evaluate whether reward-informed data selection improves verifier-based language model fine-tuning.

4 Experimental Setup

4.1 Task and Dataset

We evaluate all methods on the Countdown arithmetic reasoning benchmark used throughout the CS224R default project. Each problem consists of a target integer and a set of allowed integers. The model must generate an arithmetic expression that reaches the target value while using each provided integer exactly once. Problems in the training and evaluation datasets contain either three or four integers, allowing us to naturally define difficulty buckets for curriculum learning.

We use the dataset `asingh15/countdown_tasks_3to4` for both training and evaluation. The training set is used for RLOO optimization, while performance is measured on a held-out test set containing 50 problems.

4.2 Verifier Reward

All methods use the same verifier-based reward function. Generated responses are parsed to extract a candidate arithmetic expression and evaluated automatically. Rewards are assigned as follows:

$$r = \begin{cases} 1.0 & \text{correct solution,} \\ 0.1 & \text{valid format but incorrect solution,} \\ 0.0 & \text{invalid or unparsable output.} \end{cases}$$

This reward structure provides an objective correctness signal without requiring human annotations.

4.3 Model and Training Configuration

All experiments begin from the same supervised fine-tuned checkpoint:

`asingh15/qwen-sft-countdown-defaultproj`.

We use the RLOO training framework provided in the course project. Unless otherwise specified, all curriculum variants share the same optimization settings as the baseline RLOO run.

Key hyperparameters include:

- Learning rate: 1×10^{-5}
- Batch size: 128
- Gradient accumulation steps: 128
- Group size: $K = 8$ rollouts per prompt
- Entropy coefficient: 0.001
- KL coefficient: 0.001
- Training steps: 100
- EMA decay: 0.95

For the fixed-difficulty curriculum, the hard bucket unlock threshold is set to:

$$\tau_{\text{unlock}} = 0.25.$$

Reward-adaptive variants additionally modify learnability thresholds, exploration rates, and warmup schedules as described above.

4.4 Baselines

We compare four classes of methods:

1. **RLOO Baseline**: Uniform prompt sampling throughout training.
2. **Fixed-Bucket Curriculum v1**: Difficulty-aware curriculum using a one-success criterion.
3. **Fixed-Bucket Curriculum v2**: Difficulty-aware curriculum using a two-success criterion.
4. **Reward-Adaptive Curricula**: Prompt-level learnability estimation using reward EMAs, evaluated under two parameter settings.

The baseline is intentionally simple and serves as a control condition for determining whether adaptive prompt selection improves learning.

4.5 Evaluation Metrics

Following the standard Countdown evaluation protocol, we measure functional correctness using pass@k for:

$$k \in \{1, 2, 4, 8, 16\}.$$

Pass@k measures whether at least one of the model's k sampled responses solves the problem correctly.

We report both overall pass@k and pass@1 broken down by problem difficulty, 3-number versus 4-number problems. The difficulty-specific analysis is particularly important because our curricula explicitly modify the distribution of easy and hard training examples. These metrics allow us to determine not only whether a curriculum improves overall performance, but also how it redistributes learning across different reasoning difficulties.

5 Results

5.1 Overall Quantitative Performance

The uniform RLOO baseline outperforms all curriculum variants across every value of k. Table 1 reports pass@k performance for the uniform RLOO baseline and all curriculum variants and Figure 1 shows the general trend.

Table 1: Overall Countdown test-set performance.

Method	Pass@1	Pass@2	Pass@4	Pass@8	Pass@16
RLOO Baseline	**0.4300**	**0.5512**	**0.6214**	**0.6582**	**0.6800**
Fixed Curriculum v1	0.3812	0.5042	0.5888	0.6370	0.6600
Fixed Curriculum v2	0.3762	0.4932	0.5723	0.6226	0.6600
Reward-Adaptive v1	0.3312	0.4680	0.5769	0.6359	0.6600
Reward-Adaptive v2	0.3425	0.4573	0.5424	0.6040	0.6400

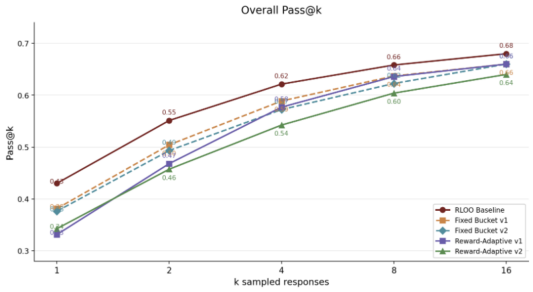

Figure 1: Overall pass@k performance.

Across all values of k, the uniform RLOO baseline achieves the strongest performance. Neither the fixed-difficulty nor reward-adaptive curricula surpass the baseline, suggesting that adaptive prompt selection alone is insufficient to improve performance in this setting.

However, a more nuanced pattern emerges when comparing pass@1 and pass@16. While curriculum methods substantially reduce pass@1 performance, their pass@16 scores remain relatively close to the baseline. For example, Reward-Adaptive v1 achieves pass@1 of only 0.3312 compared to 0.4300 for RLOO, yet reaches pass@16 of 0.6600 compared to 0.6800 for RLOO.

This gap suggests that the curriculum models often retain the ability to generate correct solutions, but are less likely to produce those solutions in their earliest samples. In other words, the curricula appear to affect the reliability with which correct reasoning trajectories are generated more than the model's underlying capability to solve the task

We further examine this pattern by breaking down pass@1 performance by problem difficulty, separating 3-number problems, 4-number problems, and overall accuracy in Figure 2. This difficulty-based view shows that the curriculum methods do not simply underperform uniformly. Instead, they redistribute performance across easier and harder problems, with some variants preserving or improving performance on 3-number problems while substantially reducing accuracy on 4-number problems. We analyze these difficulty-specific trends in more detail in the following sections.

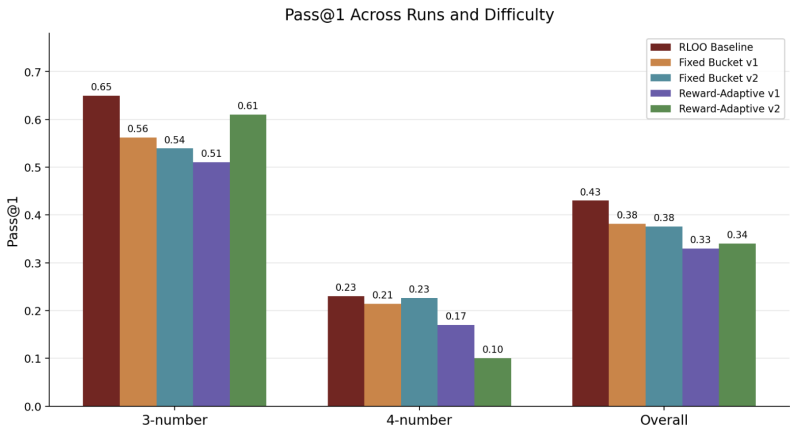

Figure 2: Pass@1 across 3-number, 4-number, and overall test problems.

5.2 Fixed-Difficulty Curriculum Analysis

The fixed-difficulty curriculum partitions Countdown problems into 3-number and 4-number buckets and dynamically adjusts sampling probabilities according to bucket-level success estimates.

The primary difference between the two variants is the definition of success used to update bucket EMAs. Fixed Curriculum v1 considers a prompt solved if at least one rollout is correct, whereas Fixed Curriculum v2 requires at least two correct rollouts. The second criterion therefore measures consistency rather than occasional success.

Difficulty-specific evaluation reveals an important tradeoff. For Fixed Curriculum v2:

$$\text{Pass@1}_{\text{3-number}} = 0.5391$$

and

$$\text{Pass@1}_{\text{4-number}} = 0.2260$$

compared to approximately

$$\text{Pass@1}_{\text{3-number}} \approx 0.65$$

and

$$\text{Pass@1} * \text{4-number} \approx 0.23$$

for the RLOO baseline.

Interestingly, performance on difficult 4-number problems is largely preserved, while performance on easier 3-number problems decreases. Training diagnostics indicate that the hard bucket unlocked very early in both variants, meaning the curriculum spent most of training in a mixed-difficulty regime rather than a prolonged easy-to-hard progression. As a result, the primary effect of the curriculum came from adaptive reweighting between difficulty buckets rather than delaying access to harder problems.

The stricter two-success criterion produced a more conservative estimate of competence by requiring repeated success before a bucket was considered mastered. Compared to v1, this reduced the influence of isolated correct generations and maintained greater training pressure on harder examples. Consequently, the curriculum allocated relatively more attention to 4-number problems, helping preserve hard-task performance while reducing performance on easier 3-number tasks.

Although neither fixed curriculum exceeds baseline performance, the results suggest that consistency-based competence estimates can meaningfully change how learning is distributed across problem difficulties. More broadly, the findings indicate that the fixed curriculum functioned primarily as an adaptive difficulty-weighting mechanism rather than a staged curriculum with a substantial unlock phase.

5.3 Reward-Adaptive Curriculum Analysis

The reward-adaptive curriculum estimates learnability directly from observed rewards rather than relying on predefined difficulty buckets. Prompts are categorized as too hard, learnable, or too easy based on a prompt-level success EMA, and sampling probabilities are adjusted accordingly.

The two reward-adaptive variants produce markedly different behavior. Reward-Adaptive v1 uses a broad learnable region and minimal exploration, while Reward-Adaptive v2 introduces a narrower learnable range, a 20-step warmup period, and increased uniform exploration.

The most striking result appears in the difficulty-specific breakdown:

Method	3-number	4-number
Reward-Adaptive v1	0.5104	0.1659
Reward-Adaptive v2	0.6068	0.0986

Reward-Adaptive v2 substantially improves performance on 3-number problems while dramatically reducing performance on 4-number problems. This indicates that the curriculum strongly altered the distribution of learning.

A plausible explanation is that many 4-number problems were classified as too hard under the stricter learnability thresholds introduced in v2. Because these prompts received lower sampling probability, the model received fewer opportunities to improve on difficult reasoning tasks. Meanwhile, easier prompts remained within the learnable region and continued to receive substantial training attention.

This interpretation is consistent with the original design motivation of the curriculum. The algorithm successfully identifies prompts with reliable reward signal, but doing so may inadvertently reduce exposure to difficult examples that are necessary for broader generalization.

5.4 Curriculum Learning versus Training Coverage

Across both curriculum families, a common pattern emerges. The curricula successfully changed which prompts received training attention, yet none improved overall performance relative to uniform sampling.

This suggests a tradeoff between two competing objectives:

1. Concentrating updates on prompts believed to be most informative.
2. Maintaining broad coverage of the training distribution.

The fixed-difficulty curricula shifted learning toward harder problems, while the reward-adaptive curricula increasingly concentrated training on prompts classified as learnable. In both cases, the resulting specialization appears to have reduced the benefits of broad exposure provided by uniform RLOO sampling.

Taken together, these findings indicate that identifying informative prompts is not sufficient to improve verifier-based reinforcement learning. Effective curricula must balance prompt informativeness against diversity and coverage, ensuring that difficult examples continue to receive enough training exposure to become learnable.

6 Discussion

Our original hypothesis was that prompts near the model's current learning frontier would provide the most informative reward signal for RLOO training. Under this view, curriculum sampling should improve learning efficiency by concentrating updates on prompts that are neither consistently solved nor consistently failed. While both curriculum families successfully altered the distribution of training examples, neither surpassed the uniform RLOO baseline.

One possible explanation is that the benefits of targeted sampling were outweighed by reductions in effective training coverage. Uniform RLOO continually exposes the model to the full problem distribution, including both easy and difficult examples. In contrast, curriculum methods intentionally

bias sampling toward selected subsets of prompts. While this can increase the density of informative updates, it also reduces exposure to other portions of the training distribution. Our results suggest that, in the Countdown setting, maintaining broad coverage may be more important than aggressively prioritizing prompts estimated to be informative.

The reward-adaptive curricula provide particularly strong evidence for this tradeoff. Tightening the learnable region and increasing curriculum selectivity improved performance on easier 3-number problems while dramatically reducing performance on more difficult 4-number problems. This indicates that the curriculum successfully focused training, but that the resulting specialization came at the expense of difficult-task generalization. More generally, these results highlight a challenge for adaptive data selection methods: examples that appear uninformative early in training may still be necessary for long-term improvement.

Several limitations should be acknowledged. First, all experiments were conducted on a relatively small Countdown benchmark with only 100 training steps, which may limit the extent to which curriculum benefits can emerge. Second, evaluation was performed on a 50-problem test set, introducing nontrivial variance into pass@k estimates. Third, we primarily analyzed performance outcomes rather than directly measuring prompt coverage, prompt reuse, or bucket occupancy over time. Such measurements would provide stronger evidence regarding the mechanisms responsible for the observed performance differences.

Future work could explore curricula that explicitly balance learnability and coverage. For example, adaptive sampling could be combined with coverage constraints, minimum exposure guarantees for difficult prompts, or uncertainty-aware exploration mechanisms. More broadly, investigating these ideas on larger reasoning benchmarks and longer training runs may help determine whether the tradeoffs observed here are specific to Countdown or reflect a more general property of verifier-based reinforcement learning.

7 Conclusion

In this work, we investigated whether reward-informed curriculum sampling could improve Reinforcement Learning with Leave-One-Out (RLOO) fine-tuning on the Countdown reasoning benchmark. We introduced two curriculum-learning approaches: a Fixed-Difficulty Curriculum based on problem structure and a Reward-Adaptive Curriculum based on prompt-level reward statistics. While both methods successfully altered the distribution of training examples and produced distinct learning behaviors, neither surpassed a uniform RLOO baseline.

Our results suggest that curriculum learning in verifier-based reinforcement learning involves a fundamental tradeoff between targeting informative examples and maintaining broad training coverage. Fixed-difficulty curricula shifted learning toward harder reasoning tasks but reduced performance on easier problems, while reward-adaptive curricula increasingly concentrated training on easier learnable prompts and substantially degraded performance on difficult 4-number tasks. Despite strong theoretical motivation, selectively prioritizing prompts near the model's estimated learning frontier did not translate into improved overall performance.

The primary takeaway from this work is that identifying informative prompts is not sufficient to improve reinforcement learning performance. In the Countdown setting, broad exposure to the training distribution appears to be more valuable than aggressively concentrating updates on a narrow subset of examples. These findings highlight the importance of balancing curriculum-guided optimization with diversity and coverage when designing future RLVR training systems.

8 Team Contributions

Both Catherine and Nora worked together to implement and train the RLOO baseline. Nora focused on implementing and adjusting the Fixed-Difficulty Curriculum while Catherine focused on implementing the Reward-Adaptive Curriculum. Both members of the team worked collaboratively to run different experiments, analyze the results, and produce the paper.

References

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Pich'e, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970*, 2025. URL https://arxiv.org/abs/2505.14970.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025. URL https://arxiv.org/abs/2501.12948.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025. URL https://arxiv.org/abs/2506.06632.

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. Teaching models to teach themselves: Reasoning at the edge of learnability. *arXiv preprint arXiv:2601.18778*, 2026. URL https://arxiv.org/abs/2601.18778.