Extended Abstract

Motivation Reinforcement learning fine-tuning of large language models is bottlenecked by the interaction of two problems. First, verifier rewards are sparse. When the policy cannot solve a problem, repeated rollouts return zero reward and provide no gradient signal. Second, problem difficulty varies widely, so uniform sampling spends much of a fixed compute budget on problems that are either already mastered or far beyond the model's reach. The most useful training signal comes from problems near the model's current competence frontier. We study whether difficulty-aware curriculum learning can concentrate RLOO training on this frontier and improve final performance on the Countdown arithmetic reasoning task.

Method We propose three curriculum variants on top of a RLOO fine-tuning baseline. First, a *fixed curriculum* stages training through easy, medium, and hard difficulty buckets at predetermined intervals. Second, an *offline adaptive curriculum* maintains a per-bucket exponential moving average of verifier reward and assigns sampling weight by proximity to a target competence level, with a uniform mixing term that keeps every bucket reachable. Third, an *online curriculum* estimates difficulty directly from rollout pass rates during training rather than from a precomputed scorer, partitioning observed prompts into dynamic buckets and reserving a fraction of each batch for unseen-prompt exploration. Difficulty is scored offline from structural problem features: operand count, solution depth, number of valid solutions, and operation complexity.

Implementation All runs begin from the same SFT checkpoint (`asingh15/qwen-sft-countdown-defaultproj`, Qwen 2.5 0.5B) and are trained for 100 steps on an NVIDIA H100 with batch size 128, $k{=}8$ rollouts per prompt, learning rate 10^{-5}, KL penalty 0.001, and entropy coefficient 0.001. We additionally conduct a granularity ablation varying the number of adaptive curriculum buckets from 3 to 16.

Results All curriculum methods outperform standard RLOO under the same training budget. The offline adaptive curriculum achieves the highest test rollout accuracy of 59.3%, compared to 53.9% for baseline RLOO and 57.5% for the fixed curriculum. The best online curriculum variant (exploration-heavy, 60% unseen-prompt sampling) reaches 58.9%, competitive with the offline adaptive approach despite learning difficulty estimates entirely from rollouts. Gains are largest on medium- and hard-difficulty problems: the adaptive curriculum improves hard-bucket accuracy from 12.5% to 20.7% and medium-bucket accuracy from 63.1% to 71.0%. The granularity ablation finds that the three-bucket configuration outperforms finer partitions (accuracy drops to 43.6% at 16 buckets), suggesting that noisy competence estimates from small per-bucket sample sizes undermine frontier identification.

Discussion Curriculum learning is most valuable precisely where sparse rewards are most problematic: on problems near but slightly beyond the model's current frontier. The offline adaptive curriculum's continuous reallocation of probability outperforms the fixed schedule's manual transitions, confirming that performance-conditioned progression matters. The online curriculum's exploration-heavy result shows that reliable difficulty signals can be obtained without a hand-crafted scorer, provided there is sufficient unseen-prompt coverage. Bucket granularity is also a practical hyperparameter. Broad groupings provide more stable competence estimates than fine-grained buckets.

Conclusion and Future Work Difficulty-aware curriculum learning consistently improves RLOO fine-tuning for Countdown. Adaptive curricula outperform both uniform sampling and fixed schedules, with the largest gains on harder problems where reward sparsity is most severe. Future work should evaluate online curricula under the full training budget, extend unseen-pool sampling to the offline setting, and explore per-prompt competence tracking as an alternative to bucket-level aggregation.

Adaptive Difficulty-Aware Curriculum Learning for RLOO

Darren Chan
Department of Computer Science
Stanford University
dchan04@stanford.edu

Jayna Huang
Department of Computer Science
Stanford University
jhuang23@stanford.edu

Sophie Zhang
Department of Computer Science
Stanford University
spzhang@stanford.edu

Abstract

Reinforcement learning (RL) fine-tuning for large language model reasoning tasks faces two related challenges: sparse verifier rewards and substantial variation in problem difficulty. First, when a model is unable to solve a problem, repeated rollouts with zero reward provide little useful learning signal; second, uniform sampling exposes the model to many problems that are either trivially easy or far beyond its current capabilities. We investigate whether curriculum learning can mitigate these challenge for Countdown arithmetic reasoning tasks by prioritizing problems according to difficulty. We introduce three curriculum variants: a fixed easy-to-hard curriculum, an offline adaptive curriculum that adjusts sampling probabilities using bucket-level success rates, and an online curriculum that estimates difficulty directly from rollout outcomes during training. Problems are grouped using structural difficulty features including operand count, solution depth, number of valid solutions, and operation complexity. Under a fixed training budget, all curriculum approaches outperform standard RLOO training. The offline adaptive curriculum achieves the highest test rollout accuracy of 59.3% compared to 53.9% for baseline RLOO, while the best online curriculum reaches 58.9%. Analysis by difficulty bucket shows that curriculum learning provides the largest gains on medium- and hard-difficulty problems, where sparse rewards are most severe. These results suggest that adaptive difficulty-aware curricula can improve sample efficiency and final performance in verifier-based RL fine-tuning, whether difficulty is estimated offline or learned online from rollout feedback.

1 Introduction

Large language models (LLMs) are increasingly fine-tuned using reinforcement learning (RL) to improve reasoning performance beyond supervised pretraining. Modern post-training pipelines often combine supervised fine-tuning (SFT), preference optimization methods such as IPO or DPO, and online policy-gradient methods such as PPO or RLOO. However, RL fine-tuning remains challenging because verifier-based rewards are often sparse: when a model cannot solve a problem, training may produce repeated low- or zero-reward rollouts, providing little useful learning signal.

This challenge is particularly relevant for Countdown arithmetic reasoning tasks, where problem difficulty varies substantially. Easier problems may require only simpler arithmetic, while harder problems demand longer reasoning chains. Under uniform sampling, the model spends training effort

on problems that are either already mastered or effectively unsolvable at its current capability level, reducing the quality of the reward signal.

Curriculum learning addresses this issue by prioritizing training examples according to difficulty. Easy-to-hard curricula have been shown to improve reasoning performance and sample efficiency, while more recent approaches dynamically adapt task selection throughout training. These results suggest that adaptive curricula may improve verifier-based RL fine-tuning on mathematical reasoning tasks such as Countdown.

In this project, we investigate whether difficulty-aware curriculum learning improves RLOO fine-tuning for Countdown. Building on the default CS224R RL pipeline, we explore both fixed and adaptive curriculum strategies. Our primary approach dynamically adjusts the sampling distribution according to the model's recent success rates, focusing training on problems near the model's current learning frontier—those that are neither fully mastered nor consistently unsolved. We additionally investigate an online curriculum that estimates difficulty directly from rollout outcomes during training rather than relying on a precomputed difficulty score.

We evaluate whether fixed, adaptive, and online curriculum learning improve sample efficiency and final Countdown performance relative to uniform sampling. We hypothesize that difficulty-aware sampling will produce more informative reward signals throughout training, leading to faster learning and higher test rollout accuracy under the same compute budget.

2 Related Work

Sparse rewards are a well-known challenge in RL fine-tuning for LLMs. DeepSeek-R1 Guo et al. (2025) notes that training can plateau when the model has near-zero success on a task, producing little useful gradient signal—a failure mode they term the "cold start" problem. Their solution relies on an SFT warm start rather than modifying how training examples are selected, leaving open the question of whether curriculum design itself can mitigate reward sparsity during online RL.

Curriculum learning offers one possible solution. Originally formalized by Bengio et al. (2009), curriculum learning improves optimization by presenting examples in a meaningful order, typically from easier to harder instances. The central idea is that models learn more effectively when training focuses on problems that are challenging but still within reach of their current capabilities.

Building on this intuition, Parashar et al. (2026) address this challenge with an easy-to-hard (E2H) curriculum that stages tasks from simple to complex. They report improved reasoning performance and sample efficiency on math and code tasks. However, E2H relies on a fixed schedule and predefined difficulty partitions, making it unable to adapt to the model's actual progress.

To address this limitation, Chen et al. (2025) propose Self-Evolving Curriculum (SEC), which treats difficulty categories as bandit arms and allocates training according to estimated learning gain. SEC improves generalization to harder problems but still requires meaningful difficulty bins and reward signals for curriculum selection.

More recently, Sundaram et al. (2026) introduce SOAR, a bilevel meta-RL framework in which a teacher model generates stepping-stone problems for a student model. While effective at addressing initially unsolvable tasks, SOAR is substantially more complex and computationally expensive than lightweight curriculum approaches.

Together, these works demonstrate that curriculum design can significantly influence RL fine-tuning outcomes. However, it remains unclear how a simple adaptive curriculum based on rolling success rates interacts with the RLOO objective on Countdown. Our work investigates this setting, focusing on a lightweight curriculum mechanism that requires neither a learned scheduler nor bilevel optimization.

3 Data

We evaluate on the *Countdown* arithmetic reasoning task. Each example consists of a set of numbers and a target value. The model must generate a valid arithmetic expression that reaches the target while using only the provided numbers. Difficulty varies substantially across problems: some can be solved with simple arithmetic, while others require longer reasoning chains involving multiple intermediate calculations.

Easier Example	Harder Example
Numbers: 1, 1, 30 **Target:** 30 **Solution:** $(1 \times 1) \times 30 = 30$	**Numbers:** 96, 92, 93, 90 **Target:** 88 **Solution:** $((90 \times 92) - 96)/93 = 88$

Table 1: Example Countdown problems illustrating variation in task difficulty.

We use the Countdown verifier reward from the CS224R RL fine-tuning framework. A rollout receives reward 1.0 if it contains a valid `<answer>...</answer>` span whose arithmetic expression uses exactly the provided numbers and evaluates to the target value. A reward of 0.1 is assigned if an `<answer>` span is present but the expression is incorrect. Rollouts that do not contain an `<answer>` span receive reward 0.0. This reward structure provides a small incentive for producing correctly formatted outputs while reserving the highest reward for fully correct solutions.

4 Methods

4.1 Difficulty Scoring

We partition Countdown problems into difficulty buckets using deterministic scoring functions based on structural properties of the equation solution. To evaluate whether a scoring function captures meaningful differences in problem complexity, we measure the pass rate of an SFT baseline across buckets: a useful difficulty metric should separate problems that the model solves reliably from those it struggles with, producing a general decline in accuracy as bucket difficulty increases.

We compare several candidate metrics, including operand count, solution depth, number of valid solutions, and operation complexity, as well as combined scores. After a small grid search, the weighted combination below provides the most consistent separation between easier and harder problems and is used throughout our curriculum experiments:

$$d_{\text{weighted}} = 1.0\,\tilde{d}_{\text{operands}} + 0.5\,\tilde{d}_{\text{depth}} + 2.0\,\tilde{d}_{\text{solutions}} + 1.0\,\tilde{d}_{\text{ops}}$$

where $\tilde{d}_{\text{operands}}$ denotes the number of input operands, $\tilde{d}_{\text{depth}}$ the minimum solution depth (fewest arithmetic operations required), $\tilde{d}_{\text{solutions}}$ the number of valid solutions, and $\tilde{d}_{\text{ops}}$ the operation complexity, computed using $c(+) = c(-) = 1$, $c(*) = 2$, and $c(/) = 3$.

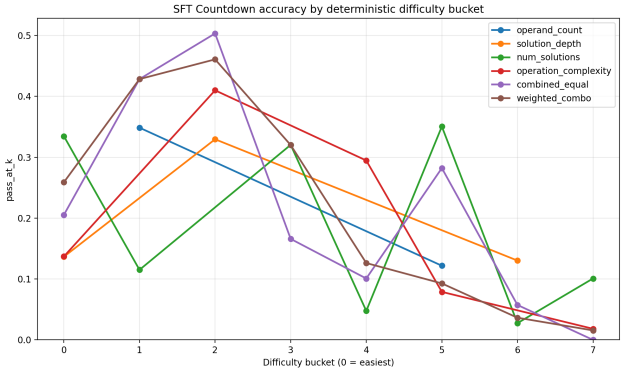

Figure 1: SFT Countdown accuracy across deterministic difficulty buckets for six scoring methods.

Figure 1 shows that the weighted score produces a strong overall relationship between bucket index and model performance. Easy and medium buckets achieve much higher pass rates than hard ones. Although performance is not perfectly monotonic, the weighted metric provides clearer separation than any individual component, making it a suitable basis for curriculum learning.

4.2 Fixed Curriculum Learning

We first investigate whether a fixed curriculum improves learning on the Countdown task. Using the scoring algorithm from above, we classify all of the training problems into easy, medium, and hard buckets. We then train on a fixed schedule–first introducing easy problems, then medium problems, and then hard problems at fixed intervals. The curriculum is cumulative: once a bucket is introduced, it remains in the sampling distribution for the remainder of training. Thus, easier examples continue to be sampled after harder examples are introduced. The training schedule and sampling distribution of problems can be seen in Figure 2. This allows for the model to learn on easier problems first before progressing to harder problems, and serves as a baseline for our adaptive-difficulty aware training process.

Unlike the adaptive curriculum described below, the fixed curriculum does not depend on observed rollout accuracy or reward. Its schedule and sampling distribution are specified before training, providing a controlled baseline for evaluating the effect of progressively introducing harder problems. However, because progression is not performance-dependent, the schedule may introduce harder examples before the model is ready or continue emphasizing easier examples after they are largely mastered. Figure 2 shows the fixed schedule and sampling distribution over difficulty buckets.

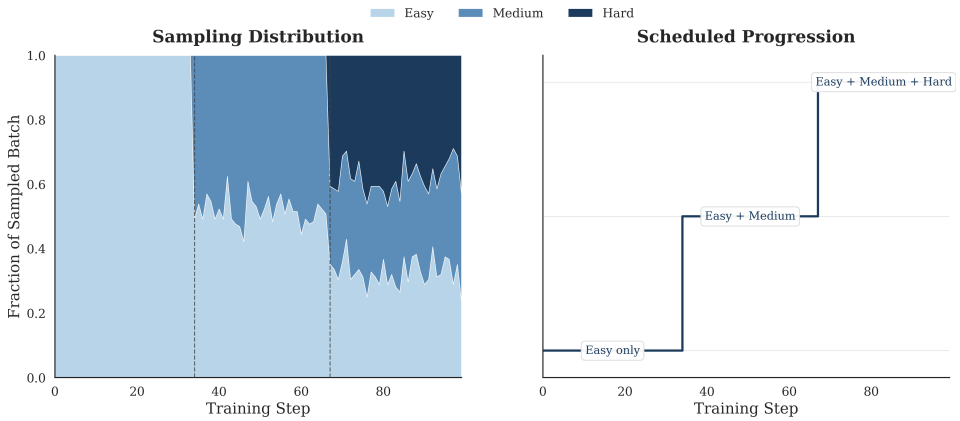

Figure 2: Fixed curriculum learning. Left: sampling distribution over difficulty buckets throughout training. Right: predefined progression from easy to medium to hard problems.

4.3 Adaptive Curriculum Learning

While the fixed curriculum relies on predetermined transitions between difficulty levels, our second approach explores adaptive curriculum learning. Rather than specifying when the curriculum should advance, adaptive sampling continuously tracks bucket-level competence using recent verifier rewards. For each difficulty bucket b, we maintain an exponential moving average (EMA) of verifier rewards,

$$\hat{r}_{t,b} = (1 - \alpha)\hat{r}_{t-1,b} + \alpha r_{t,b}, \tag{1}$$

where $r_{t,b}$ is the mean verifier reward observed from bucket b at training step t, $\hat{r}_{t,b}$ is the corresponding EMA estimate, and α controls the update rate. The EMA pass rate serves as a smoothed estimate of how well the current policy can solve problems from that bucket.

Sampling weights are then assigned according to proximity to the target competence level $\tau = 0.5$, giving the highest probability to buckets that are neither consistently solved nor consistently failed. Intuitively, these problems lie near the model's current learning frontier and therefore provide the most informative reward signal. Buckets that are already mastered contribute little new information, while buckets that remain far beyond the model's capabilities tend to produce sparse rewards and weak learning signals.

To prevent the curriculum from becoming overly narrow, a uniform mixing term ($\varepsilon = 0.2$) ensures that every bucket retains a nonzero probability of being sampled. This allows the model to periodically

revisit easier examples and continue exploring harder ones, reducing the risk of prematurely excluding potentially still useful training data.

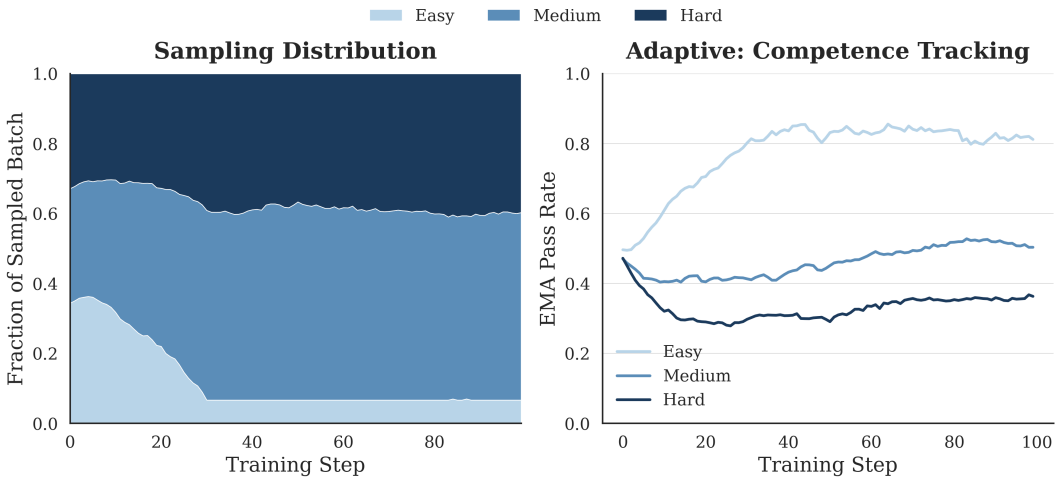

Figure 3: Adaptive curriculum learning. Left: sampling distribution over difficulty buckets throughout training. Right: EMA pass rates used to estimate bucket-level competence.

Figure 3 illustrates the resulting adaptive curriculum dynamics. The right panel shows the EMA pass rates used to estimate competence throughout training. Easy problems rapidly become solvable, with pass rates increasing from roughly 0.5 to above 0.8, indicating that the policy quickly masters this portion of the distribution. Medium- and hard-difficulty problems improve more gradually and remain closer to the target competence region, making them the most informative sources of training signal.

These competence estimates directly determine the sampling distribution shown in the left panel. As the pass rate on easy problems rises, their sampling probability steadily decreases. At the same time, probability mass shifts toward medium and hard buckets, which remain near the learning frontier for a larger portion of training. Rather than relying on manually specified curriculum stages, the adaptive approach automatically reallocates training effort as the model improves, focusing computation on the problems most likely to produce useful learning progress at each stage of optimization.

4.4 Online rollout-based curriculum

We additionally implemented an online curriculum learning variant for RLOO in which task difficulty is estimated from the policy's own rollouts during training rather than from a fixed offline difficulty scorer. For each sampled Countdown prompt, the current policy generates a group of responses, and each response is scored by the Countdown verifier. The prompt-level pass rate is computed as the mean verifier reward across the rollout group. This pass rate is then used to update an EMA estimate (using equation 1) of the model's current success rate on that prompt.

Prompts that have been observed at least once are sorted by their pass-rate EMA and partitioned into difficulty buckets. Low pass-rate prompts correspond to hard examples, high pass-rate prompts correspond to easy examples, and intermediate pass-rate prompts correspond to the model's current learning frontier. Sampling weights over observed buckets are computed using a frontier-weighting function that assigns higher probability to buckets whose average pass rate is near a target success rate. This encourages training on examples that are neither completely unsolved nor already mastered.

To avoid early curriculum collapse from sparse observations, unseen prompts are separated into their own exploration pool. After a short uniform warmup period, each training batch reserves a fixed fraction of prompts for unseen examples, while the remaining prompts are sampled from the observed difficulty buckets. This prevents unseen prompts from being treated as artificial frontier examples and keeps the curriculum grounded in actual rollout feedback. The sampling distribution and EMA pass rate over the training steps is shown for the exploration-heavy setting (the best performing variation) in Figure 4, and the same plots are shown in Figures 6 and 7 in the Appendix for the balanced and frontier approaches.

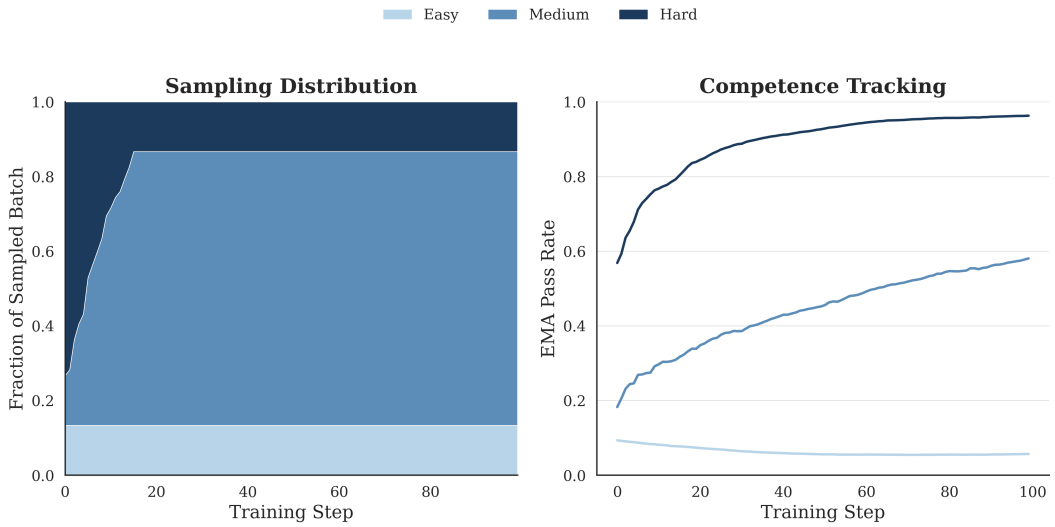

Figure 4: Online curriculum learning with exploration-heavy settings. Left: sampling distribution over difficulty buckets throughout training. Right: EMA pass rates used to estimate bucket-level competence.

5 Experimental Setup

All RLOO runs start from the same supervised Countdown checkpoint, use the same binary verifier reward, and are trained on the train set `asingh15/countdown_tasks_3to4`. Training is done on a single NVIDIA H100 with batch size 128 prompts per step, k=8 rollouts per prompt (1,024 sampled trajectories per optimizer step), 100 global steps (except for the online curriculum cases, see below), learning rate 10^{-5} with a constant schedule (no warmup), weight decay 10^{-4}, entropy coefficient 0.001, and KL penalty 0.001 to a frozen reference equal to the initial policy. Rollouts during training use temperature 1.0, top-p=1.0, and up to 1,024 new tokens per response (max_model_len=2048). Checkpoints are saved every 10 steps. Evaluation uses the held-out test split with 16 samples per prompt at temperature 0.6, top-p=0.95, and top-k=20.

5.1 Fixed curriculum

We use a fixed 3-phase schedule that cumulatively unlock buckets 0, then 0,1, then 0,1,2 with uniform random sampling at the one-third marks during the 100-step training process. Bucket labels came from equal quantile splits from the weighted difficulty score.

5.2 Adaptive curriculum

The adaptive curriculum uses difficulty buckets formed from equal-quantile splits of the weighted difficulty score. At each training step, we update a per-bucket EMA of verifier reward following Equation 1 with rate $\alpha = 0.1$. Sampling weight peaks at the target competence level $\tau = 0.5$ and tapers across the frontier band $[0.05, 0.8]$, so that buckets whose EMA pass rate falls near τ receive the most probability while buckets the model almost never solves still retain a small but nonzero share. A uniform mixing term $\varepsilon = 0.2$ is added to the resulting distribution to guarantee every bucket remains reachable throughout training.

5.3 Online curriculum

We evaluated three online adaptive RLOO variant, which differed in the amount of unseen-prompt exploration and in the target pass rate used to define the learning frontier.

- Balanced: 5 warmup steps, EMA $\alpha = 0.3$, uniform bucket mixing 0.5, and reserved 40% of each post-warmup batch for unseen problems. The frontier target was a 50% pass rate

with band $0.2, 0.8$. This variant was designed to balance exploration with adaptive frontier sampling.

- Exploration-heavy: 8 warmup steps, EMA $\alpha = 0.3$, increased unseen-problems sampling to 60%, and 50% target pass rate. This variant tested whether online curriculum performance was limited by insufficient problem coverage.
- Harder frontier: 5 warmup steps, MA $\alpha = 0.5$, 40% unseen exploration, and pass rate decreased to 35% pass rate with band $0.05, 0.7$. This tested whether focusing on less frequently solved but still learnable problems improved training.

Because of network errors, only the exploration variant completed all 100 training steps, the other two only completed 70 steps.

6 Results

6.1 Overview

Quantitative Evaluation. Table 6.1 shows test rollout accuracy across four training conditions on the Countdown task. For each test prompt, this is computed from 16 sampled rollouts as the fraction of rollouts receiving a verifier score of 1.0, then averaged across prompts.

Method	Test rollout accuracy
SFT Initialization	0.304
Baseline RLOO	0.539
Fixed curriculum	0.575
Adaptive curriculum	**0.593**
Balanced online curriculum	0.549
Exploration online curriculum	0.589
Harder frontier online curriculum	0.526

Table 2: Test rollout accuracy across four training conditions on Countdown.

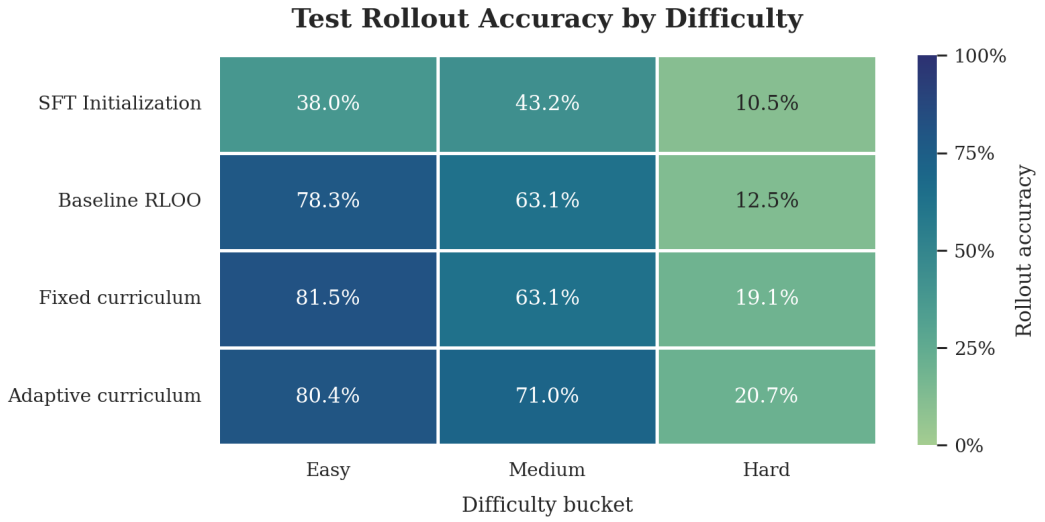

Figure 5: Test rollout accuracy stratified by problem difficulty using the weighted difficulty metric.

Figure 5 showcases performance by difficulty bucket. While all RL methods significantly improve over the SFT baseline, the gains are not distributed uniformly across problem difficulties. Easy problems are largely solved by all RL variants, with rollout accuracies near 80%, suggesting that these examples are quickly mastered during training. In contrast, medium- and hard-difficulty

problems remain substantially more challenging and therefore provide a clearer measure of curriculum effectiveness.

Both curriculum approaches outperform uniform RLOO on the hardest buckets. Relative to the baseline RLOO policy, the fixed curriculum improves hard-problem accuracy from 12.5% to 19.1%, while the adaptive curriculum further increases performance to 20.7%. Improvements are also observed on medium-difficulty problems, where adaptive curriculum learning reaches 71.0% accuracy compared to 63.1% for both baseline RLOO and the fixed curriculum.

Qualitative Analysis. These results suggest that curriculum learning is most valuable precisely where sparse rewards are most problematic. By concentrating training on problems near the model's capability frontier rather than sampling uniformly, the curriculum methods obtain stronger learning signals and can thus improve performance on more difficult Countdown instances. The adaptive approach performs best overall, indicating that dynamically tracking competence and adjusting the sampling distribution is more effective than relying on a fixed progression through difficulty levels.

6.2 Adaptive Curriculum Granularity Ablation

Quantitative Evaluation. To evaluate the sensitivity of the adaptive curriculum to the granularity of the difficulty partition, we conducted an ablation study varying the number of difficulty buckets used by the adaptive sampler. The original adaptive curriculum used three difficulty buckets, which we treat as the baseline configuration. We then trained otherwise identical adaptive RLOO models using 5, 8, 10, and 16 buckets. All runs used the same SFT initialization, RLOO hyperparameters, dataset, difficulty metric, and evaluation protocol; only the quantile bucketing of the precomputed difficulty scores was changed. Table 3 reports test rollout accuracy for training across several bucket counts.

Number of Buckets	Test Rollout Accuracy
3	**0.593**
5	0.575
8	0.494
10	0.531
16	0.436

Table 3: Test rollout accuracy for adaptive curriculum learning with different numbers of difficulty buckets.

Qualitative Analysis. The results indicate that adaptive curriculum performance is sensitive to bucket granularity. The original three-bucket configuration achieves the highest test rollout accuracy, while progressively finer partitions generally reduce performance. One possible explanation is that increasing the number of buckets fragments the training data and produces noisier estimates of bucket-level competence, making it more difficult for the adaptive sampler to accurately identify the model's learning frontier. These findings suggest that a small number of broad difficulty groups provides a more reliable signal for adaptive curriculum learning than highly fine-grained partitions.

6.3 Online Curriculum Learning

Quantitative Evaluation. Among the three online curriculum variants, the exploration-heavy online curriculum performed best, reaching a test rollout accuracy of 0.589. This was close to the offline adaptive curriculum result of 0.593 and exceeded both baseline RLOO and the balanced online curriculum. The balanced online curriculum achieved 0.549, improving over baseline RLOO initialization but underperforming the fixed and adaptive offline curriculum methods. The harder-frontier online curriculum performed worst among the online variants, reaching 0.526.

Qualitative Analysis. These results suggest that broader prompt coverage was important for online curriculum learning. The exploration-heavy run reserved the largest fraction of each batch for unseen prompts, which likely helped the learner collect more reliable online difficulty estimates. In contrast, the harder-frontier variant may have focused too aggressively on prompts with lower pass rates, reducing the usefulness or stability of the reward signal during RLOO updates.

7 Discussion

Our results demonstrate that curriculum design can substantially influence the effectiveness of verifier-based RL fine-tuning. Both fixed and adaptive curricula outperform standard RLOO, supporting the hypothesis that uniform sampling is inefficient when problem difficulty varies and rewards are sparse.

By concentrating training on problems that are challenging but still learnable, curriculum learning provides more informative reward signals and improves performance.

The strongest results come from the offline adaptive curriculum, which dynamically focuses training effort according to the model's current competence. Unlike the fixed curriculum, which relies on manually chosen transition points, the adaptive curriculum continuously shifts probability mass toward problems near the model's current learning frontier. This leads to the largest gains on medium- and hard-difficulty Countdown instances, which seems to suggest that adaptive task selection is particularly valuable in regions where sparse rewards would otherwise slow learning.

The granularity ablation highlights an important practical consideration. Increasing the number of difficulty buckets consistently reduced performance, with the three-bucket configuration producing the best results. Finer partitions likely produce noisier competence estimates and fewer training examples per bucket, making it more difficult to accurately identify the model's learning frontier. These findings suggest that robust curriculum signals may be more important than highly precise difficulty estimates.

The online curriculum results show that useful difficulty estimates can be obtained directly from rollout outcomes without requiring a handcrafted difficulty scorer, but also highlight the importance of exploration. The exploration-heavy variant achieved performance comparable to the offline adaptive curriculum, reaching a test rollout accuracy of 0.589 versus 0.593. Initially, we treated unseen problems as having a default pass-rate estimate of 0.5, which effectively classified them as frontier examples before the model had attempted them. However, because most problems remained unseen early in training, these default estimates dominated the bucket statistics and prevented the sampling distribution from adapting meaningfully. After introducing a separate unseen pool for exploration, we observed meaningful shifts in the sampling distribution over the course of training. Combined with the strong performance of the exploration-heavy variant, these results suggest that exploration is crucial in an online difficulty-learning setting.

Finally, for all online curriculum approaches, problems classified as hard are sampled at a lower rate compared to the adaptive curriculum using difficulty buckets. This indicates 1) that using an online difficulty assignment allows the model to perform nearly as good while exposed to fewer "difficult" examples, and 2) that the online curriculum learner would likely perform even better when the number of training steps increases and allows more time to learn harder problems.

8 Conclusion and Future Work

Sparse rewards remain a major challenge in RL fine-tuning for reasoning tasks. When rollouts consistently fail, learning signals vanish and optimization can stall. This work demonstrates that difficulty-aware curriculum learning can partially address this problem by concentrating training on problems near the model's current capability frontier.

Across Countdown experiments, both fixed and adaptive curricula outperform standard RLOO training. The offline adaptive curriculum achieves the highest overall performance, improving test rollout accuracy from 53.9% to 59.3%, while the best online curriculum reaches 58.9% despite a shorter training run. Curriculum-based methods produce the largest improvements on medium- and hard-difficulty problems, where sparse rewards are most severe. We additionally find that adaptive curricula are sensitive to difficulty partitioning, with a small number of broad buckets outperforming more fine-grained alternatives.

For future work, the online curriculum could be evaluated under the full training budget and with more extensive hyperparameter tuning to better understand the tradeoff between exploration and frontier-focused sampling. Additionally, the unseen pool sampling could also be employed by the offline curriculum approach. Looking futher, difficulty estimation could move beyond bucket-level tracking to maintain competence estimates for individual prompts, allowing fine-grained curriculum adaptation. Alternatively, learned difficulty estimators or teacher models could replace hand-engineered difficulty scores entirely.

9 Team Contributions

- **Group Member 1:** Darren implemented and tested IPO, RLOO, and Fixed Curriculum Learning.
- **Group Member 2:** Jayna implemented and tested SFT, difficulty scoring, and granularity ablation work.
- **Group Member 3:** Sophie implemented and tested Adaptive Curriculum Learning.

All group members contributed to the writing of the midterm reports, poster, and project report.

Changes from Proposal. We expanded the project to include a fixed easy-to-hard curriculum as a baseline and an online curriculum that samples directly from continuous difficulty scores without discretizing problems into buckets. We also conducted ablation studies on curriculum granularity to better understand how difficulty partitioning affects adaptive curriculum performance. Further, we expanded the difficulty-based curricula to include three variants of an online curriculum learning approach that used own rollouts to assess difficulty rather than an offline scorer.

References

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. doi:10.1145/1553374.1553380

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. 2025. Self-Evolving Curriculum for LLM Reasoning. arXiv:2505.14970 [cs.AI] https://arxiv.org/abs/2505.14970

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. 2025. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 645, 8081 (2025), 633–638. doi:10.1038/s41586-025-09422-z

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. 2026. Curriculum Reinforcement

Learning from Easy to Hard Tasks Improves LLM Reasoning. arXiv:2506.06632 [cs.LG] https://arxiv.org/abs/2506.06632

Shobhita Sundaram, John Quan, Ariel Kwiatkowski, Kartik Ahuja, Yann Ollivier, and Julia Kempe. 2026. Teaching Models to Teach Themselves: Reasoning at the Edge of Learnability. arXiv:2601.18778 [cs.LG] https://arxiv.org/abs/2601.18778

10 Appendix

10.1 Training the Balanced and Frontier Online Curricula

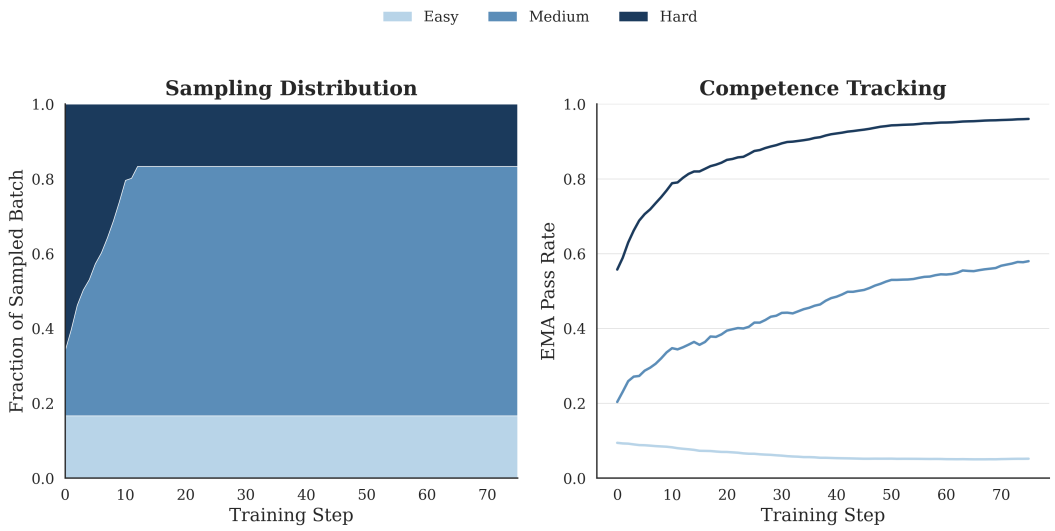

Figure 6: Balanced online curriculum. Left: sampling distribution over online difficulty buckets. Right: EMA pass rates for each bucket.

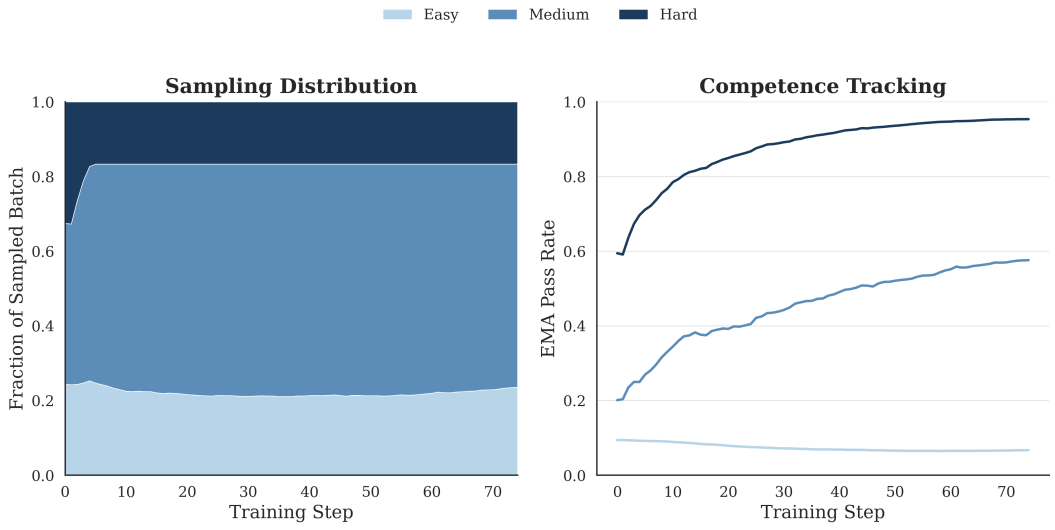

Figure 7: Harder-frontier online curriculum. Left: sampling distribution when the target competence level is shifted toward lower pass rates. Right: EMA pass rates used for competence estimation.