Extended Abstract

Motivation Online reinforcement learning (RL) elicits reasoning in large language models (LLMs), but it breaks down for sub-billion-parameter models on combinatorial tasks. This is due to the sparse-reward problem: when a problem exceeds the model's current ability, stochastic exploration almost never produces a correct answer. Since reward is near-zero, the policy gradient vanishes. We study this on the Countdown benchmark, where the model must combine 3–4 integers with arithmetic operations to reach a target. We focus on curriculum reinforcement learning (CRL) as a principled remedy to keep the model at the edge of its competence by raising difficulty only as it improves. Prior CRL methods target larger models and open-ended benchmarks that rely on static step schedules or bandit statistics. We ask whether a mastery-gated curriculum, driven purely by the model's live reward signal and requiring no oracle data, can improve sample efficiency for a `Qwen-2.5-0.5B` policy.

When a problem's computational state space exceeds a small model's immediate reasoning capacity, stochastic exploration fails to hit valid terminal tokens. This results in zero-reward signal that causes gradient updates to collapse. While a baseline agent can learn syntax constraints from Supervised Fine-Tuning (SFT), scaling exploration directly into sparse 4-operand search spaces is sample-inefficient. This paper explores difficulty scheduling as a scaffolding tool to understand whether dynamically matching environment complexity to an agent's real-time competency can alleviate the sparse reward exploration trap.

Method We train through a three-phase pipeline:

- Supervised fine-tuning (SFT) to learn the reasoning format and basic arithmetic.
- Identity Pairwise Optimization (IPO) to perform offline preference alignment over contrastive (correct, incorrect) solution pairs to warm-start the policy.
- Reinforce Leave-One-Out (RLOO) to run online policy-gradient optimization. The agent computes each response's advantage against the mean reward of the other samples in its group, with importance-weight clipping and a KL penalty against the SFT reference for stability.

On top of RLOO we add a competency-gated curriculum that partitions the data into three difficulty tiers (3-operand; 4-operand high-density; 4-operand sparse). An exponential moving average (EMA) of batch reward feeds a sigmoid gate, $P(\text{Tier}_{n+1}) = \sigma(\kappa(R_{\text{avg}} - \tau))$, that probabilistically samples the next tier. The anchor tier advances permanently only when the EMA clears a threshold for a minimum dwell time, after which the EMA resets.

Implementation The system is built on a Ray-based architecture with separate vLLM sampling and policy-update workers, trained on a single H100 GPU via Modal. We ran four curriculum configurations. Three early runs each failed in a distinct way, revealing a failure-mode taxonomy:

- Policy tethering, where an overly strong KL penalty (0.05) and an overly high threshold ($\tau = 0.7$) freeze the policy below the gate.
- Cold-start EMA inflation, where initializing the EMA to the first (high) batch reward of a warm-started policy cascades it straight to the hardest tier and collapses training.
- Probe contamination and a noisy advantage estimator, where probe-batch rewards pollute the competency EMA and a group size of 2 makes the leave-one-out baseline a single noisy sample.

The corrected run addresses all three: EMA initialized to zero, EMA updated only on anchor (non-probe) batches, group size raised to 8, τ lowered to 0.45, and training extended to 500 steps.

Results Each pipeline phase improves single-shot accuracy (also known as `Pass@1`): SFT 33.10% to IPO 37.00% to plain RLOO 46.75% `Pass@1`. There is also a corresponding decline in `Pass@100`: 88% to 78% to 70%. The three failed curriculum runs scored 29.62% (Naive), 0.00% (Ablation), and 36.12% (Hybrid). The Fixed run is the first to exhibit genuine difficulty progression: its dominant

training tier advances from Tier 1 to Tier 2 by step 59 and to Tier 3 by step 172, spending 55% of training (275 of 500 steps) on the hardest tier, all with a stable KL trajectory and no collapse. It reaches 42.62% `Pass@1` / 64.00% `Pass@100` — the best curriculum result by a wide margin. However, this is still 4.13 points below plain RLOO on `Pass@1`.

Discussion The fixed curriculum (`Pass@1` 42.62%, `Pass@100` 64.00%) underperforms plain RLOO (`Pass@1` 46.75%, `Pass@100` 70.00%). However, we treat this as inconclusive rather than a clean negative result. The baseline was trained under an unmatched configuration at 100 steps, batch size 16, entropy 0.001, KL 0.001, as opposed to the curriculum's 500 steps, batch size 4, entropy 0.01, KL 0.01. The curriculum used five times more gradient updates and still lost, so it did not demonstrate the hypothesized sample-efficiency gain. A matched-budget baseline would be the key missing experiment. We do observe a robust secondary finding: a novel temporal form of the exploration–exploitation tradeoff. Mainly, across all methods `Pass@1` rises as `Pass@100` falls, and within the fixed run alone the `step-400` checkpoint matches RLOO's `Pass@100` (70%) while the `step-500` checkpoint trades that diversity for higher `Pass@1`.

Conclusion We present a competency-gated curriculum for small-LLM reasoning and, through systematic ablation, a taxonomy of three independent, individually fatal implementation failures that prior curriculum-RL work does not surface. Correcting all three yields the first functioning curriculum in our experiments, but it does not beat plain RLOO. Because the baseline was unmatched, we cannot attribute the gap cleanly. Therefore, this paper does not offer any sample-efficiency claim. Instead, the lasting contribution is the failure-mode taxonomy. A matched-budget RLOO baseline is the experiment needed to settle whether the curriculum helps.

Adaptive Difficulty Scheduling: A Competency-Gated Curriculum for Small-LLM Reasoning

Donna Choi
Department of Computer Science
Stanford University
dchoi1@stanford.edu

Abstract

We investigate whether a competency-gated curriculum can improve the sample efficiency in reinforcement learning (RL) fine-tuning for mathematical reasoning in a sub-billion parameter language model. We start from a Supervised Fine-Tuning (SFT) baseline and apply Identity Pairwise Optimization (IPO) for preference alignment and Reinforce Leave-One-Out (RLOO) for online policy optimization on the Countdown arithmetic benchmark using `Qwen-2.5-0.5B`. We introduce a three-tier difficulty curriculum partitioned by operand count and solution density, with transitions governed by a sigmoid gating function over an exponential moving average (EMA) of batch rewards. Through a series of ablations, we identify three concrete failure modes: cold-start EMA initialization, probe-batch reward contamination, and insufficient group size. Each of these individually prevents the curriculum from behaving as intended. After correcting all three, our fixed curriculum exhibits intended difficulty progression, with the dominant training tier advancing from Tier 1 to Tier 2 around step 59 and to Tier 3 by step 172, spending 55% of training on the hardest tier. Yet it only reaches 42.62% `Pass@1`, below the 46.75% of a plain RLOO baseline trained for far fewer steps under an unmatched configuration. Within our experiments the curriculum did not demonstrate a sample-efficiency benefit, and a matched-budget baseline remains the key missing experiment. There is a secondary finding: the exploration–exploitation tradeoff exhibits as a temporal phenomenon across checkpoints under curriculum pacing, which is not a behavior seen in fixed-distribution RLOO.

1 Introduction

Reinforcement learning from verifiable rewards is a powerful paradigm for eliciting complex reasoning in large language models. Frameworks such as `DeepSeek-R1` Guo et al. (2025) demonstrate that RL can induce chain-of-thought behavior at scale. However, applying online RL directly to small (sub-billion parameter) models presents an obstacle: the sparse reward problem. When the model's current capability is insufficient to stochastically discover valid solutions, the policy gradient receives a near-zero signal and causes training to stall.

The Countdown benchmark is one such challenge. Given a set of 3 or 4 random integers and a target integer, the model must construct an arithmetic expression using basic operations ($+$, $-$, $\times$, $\div$) that evaluates to the target. As problem complexity scales from 3-operand to 4-operand tasks that have sparse solution spaces, the probability of random exploration finding a valid answer drops.

Curriculum reinforcement learning (CRL) is a principled remedy that proposes structuring the training distribution so that the model always operates at the boundary of its competence so that it receives non-zero gradients throughout training. Prior works such as Easy-to-Hard (E2H) Parashar et al. (2026) and Self-Evolving Curriculum (SEC) Chen et al. (2025) validate this direction for larger

models on open-ended benchmarks. However, these methods rely on static step-count schedules or continuous advantage-based bandit mechanisms that are not well-suited to the Countdown domain or small models.

We propose and evaluate an adaptive, mastery-gated curriculum that partitions the Countdown training data into three tiers based on operand count and target complexity. The tier transitions will be probabilistically gated using a sigmoid function over an EMA reward signal. Our training pipeline is structured in three phases: SFT for syntax initialization, IPO for offline preference alignment, and RLOO with curriculum for online policy optimization. Through ablations, we diagnose the mechanisms by which curriculum-naive implementations fail, present a corrected design that achieves the intended tier progression, and characterize why the curriculum does not outperform plain RLOO under our comparison. Our findings offer actionable design guidelines for practitioners applying CRL to small LLMs on combinatorial reasoning tasks.

2 Related Work

2.1 RL for LLM Reasoning

Post-training RL has proven highly effective for reasoning tasks. `DeepSeek-R1` Guo et al. (2025) used large-scale GRPO with rule-based verifiers to elicit step-by-step reasoning from a 671B model and then distilled the resulting capabilities into smaller architectures. Without such distillation, applying pure online RL to small models from scratch typically leads to gradient stalling from sparse rewards, motivating our curriculum approach.

2.2 Curriculum Reinforcement Learning

The Easy-to-Hard Reasoner Parashar et al. (2026) demonstrated that a static curriculum progressively shifting from simple to complex tasks significantly boosts reasoning in 1.5B–3B parameter LLMs. The Self-Evolving Curriculum Chen et al. (2025) replaced static schedules with a multi-armed bandit that dynamically adapts the training distribution. Both approaches improve sample efficiency but focus on open-ended math benchmarks like GSM8K (Grade School Math 8K), where the solution search space grows gradually. Countdown's combinatorial structure creates a sharper reward cliff.

2.3 Policy Gradient Variance Reduction

REINFORCE Leave-One-Out (RLOO) Kool et al. (2019) reduces policy gradient variance by computing per-response baselines from the mean reward of the remaining $K - 1$ group members. This approach is unbiased and scales naturally with group size, making it well-suited to online RL with small batch sizes. Our implementation extends RLOO with importance weight clipping and a KL divergence penalty against the SFT reference model.

3 Method

3.1 Task and Environment

We fine-tune the `Qwen-2.5-0.5B` on the `asingh15/countdown_tasks_3to4` dataset and evaluate correctness using a rule-based verifier that checks both syntactic validity and arithmetic equivalence. The reward is binary: 1 for a correct answer, 0 otherwise.

3.2 Phase 1: Supervised Fine-Tuning (SFT)

The baseline policy is defined by training on a dataset of verified Countdown solutions using token-level cross-entropy loss:

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

SFT trains the model to replicate the structured syntax of the reasoning format (`<think>` and `<answer>` tags) and to generate valid arithmetic derivations. The baseline achieves 33.10% `Pass@1` and 88.00% `Pass@100` on the held-out test set.

3.3 Phase 2: Identity Pairwise Optimization (IPO)

To break through the sparse reward bottleneck on single-shot generation, we apply IPO over a dataset of contrastive solution pairs $\mathcal{D}_{\text{pref}} = \{(x, y_w, y_l)\}$, where y_w denotes a mathematically valid derivation and y_l a hallucinated or incorrect one. The IPO objective is:

$$\mathcal{L}_{\text{IPO}}(\pi_\theta; \pi_{\text{ref}}) = \mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}} \left[\left(\log \frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \log \frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)} - \frac{1}{2\beta} \right)^2 \right]$$

Unlike DPO, the quadratic penalty prevents log-likelihood over-optimization against the reference model. IPO improves `Pass@1` to 37.00% while reducing `Pass@100` to 78.00%. This indicates that distributional entropy is reduced in exchange for higher single-shot confidence.

3.4 Phase 3: RLOO with Competency-Gated Curriculum

3.4.1 RLOO Policy Gradient

The IPO-warmed policy enters online RL training via RLOO. For each prompt x, the policy samples a groupof K responses $\{y_1, \ldots, y_K\}$ with corresponding rewards $\{r_1, \ldots, r_K\}$. The leave-one-out advantage for response i is:

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

The policy is updated using a clipped surrogate objective with importance weights $\rho_i = \exp(\log \pi_\theta(y_i) - \log \pi_{\text{old}}(y_i))$, clipped to $[0.5, 2.0]$ to prevent off-policy divergence:

$$\mathcal{L}_{\text{RLOO}} = -\frac{1}{N} \sum_{i=1}^{N} \text{clip}(\rho_i, 0.5, 2.0) \cdot A_i$$

We add a KL divergence penalty against the frozen SFT reference model to prevent reward hacking:

$$\mathcal{L} = \mathcal{L}_{\text{RLOO}} - \lambda_{\text{ent}} \cdot \mathcal{H}(\pi_\theta) + \lambda_{\text{KL}} \cdot D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}}).$$

3.4.2 Curriculum Design

The training dataset is partitioned into three tiers:

- **Tier 1 (3-operand)**: Tasks with exactly 3 operands. These anchor basic arithmetic syntax and provide a dense reward signal for the cold-start phase.
- **Tier 2 (4-operand, high solution density)**: 4-operand tasks where the target is low-magnitude or a common multiple of the operands, providing multiple valid solution paths.
- **Tier 3 (4-operand, sparse solution space)**: 4-operand tasks where the target is large or prime, requiring backtracking and multi-path reasoning.

3.4.3 Competency Gating

The active training tier is governed by an EMA of batch rewards:

$$R_{\text{avg}}^{(t)} = \alpha \cdot R_{\text{batch}}^{(t)} + (1 - \alpha) \cdot R_{\text{avg}}^{(t-1)}.$$

The probability of sampling the next tier for a given batch is:

$$P(\text{Tier}_{n+1}) = \sigma\left(\kappa \cdot (R_{\text{avg}} - \tau)\right) = \frac{1}{1 + e^{-\kappa(R_{\text{avg}} - \tau)}},$$

where τ is the competency threshold and κ controls transition sharpness. Permanent tier advancement requires both:

- $R_{\text{avg}} > \tau + \delta$ (where δ is a safety margin),
- A minimum dwell time of 50 anchor-tier steps.

The EMA is then reset so that competency on each new tier is measured independently. Because the gate is probabilistic, the model continues to sample adjacent tiers throughout training even as its center of mass advances.

4 Experimental Setup

All experiments use `Qwen-2.5-0.5B` as the base model, fine-tuned on the `asingh15/countdown_tasks_3to4` dataset. Training runs on a single H100 GPU. We evaluate on a fixed held-out test set of 50 targets with 100 generation attempts each, reporting `Pass@1` (mean accuracy over a single generation attempt) and `Pass@100` (proportion of targets for which at least one correct solution is found in 100 attempts).

Table 1 summarizes the hyperparameter configurations for all RLOO runs.

Configuration	τ	Clamp	KL coefficient	Group size	Steps
Plain RLOO (Baseline)	-	[0.5, 2.0]	0.001	8	100
Naive Curriculum	0.70	[0.8, 1.2]	0.05	2	250
Ablation Curriculum	0.55	[0.8, 1.2]	0.01	2	250
Hybrid Curriculum	0.55	[0.5, 2.0]	0.01	2	250
Fixed Curriculum	0.45	[0.5, 2.0]	0.01	8	500

Table 1: RLOO hyperparameter configurations. All curriculum runs share $lr = 10^{-5}$, batch size 4, entropy coefficient 0.01, $\kappa = 10.0$, $\alpha = 0.1$. The Plain RLOO baseline differs: batch size 16 and entropy coefficient 0.001 (with KL 0.001, as shown).

5 Results

5.1 Progressive Training Pipeline

Table 2 summarizes performance across all training phases.

Model	`Pass@1`	`Pass@100`
SFT Baseline	33.10%	88.00%
IPO	37.00%	78.00%
RLOO (no curriculum)	**46.75%**	**70.00%**
Naive Curriculum ($\tau = 0.7$, strict clamp, high KL)	29.62%	76.00%
Ablation Curriculum ($\tau = 0.55$, strict clamp, low KL)	0.00%	0.00%
Hybrid Curriculum ($\tau = 0.55$, loose clamp, low KL)	36.12%	74.00%
Fixed Curriculum ($\tau = 0.45$, gs=8, probe-aware EMA)	**42.62%**	**64.00%**

Table 2: `Pass@k` evaluation on the held-out test set (50 targets, 100 samples each).

Each phase of the pipeline (SFT to IPO to RLOO) improves single-shot precision over the prior stage. SFT establishes formatting syntax and basic arithmetic behavior. IPO boosts `Pass@1` by breaking through the sparse reward bottleneck on immediate rollouts at the expense of reduced exploratory diversity (`Pass@100` drops from 88% to 78%). RLOO further increases `Pass@1` to 46.75% by directly maximizing expected correctness, though `Pass@100` falls from 78% to 70% as the policy concentrates on high-confidence solution paths.

5.2 Curriculum Ablations

We evaluated three curriculum configurations prior to the corrected design. Table 3 summarizes the training dynamics of each.

Run	Tier 1 $\rightarrow$ 2	Tier 2 $\rightarrow$ 3	Steps on Tier 3	EMA Collapse	KL NAN
Naive	-	-	0	No (stagnant)	No
Ablation	Step 1	Step 2	248 (collapsed)	Yes (Step 56)	Yes
Hybrid	-	-	0	No (suppressed)	No
Fixed	Step 59	Step 172	275	No	No

Table 3: Curriculum progression dynamics across all runs.

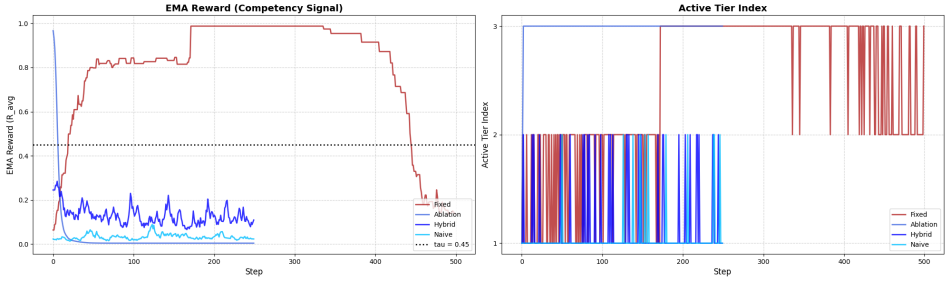

Figure 1: Curriculum training dynamics across all runs. The Ablation spikes and collapses; Naive, Hybrid remain pinned to Tier 1; only the Fixed run (Red) progresses through the tiers.

5.2.1 Naive ($\tau = 0.7$, strict clamp [0.8, 1.2], KL=0.05)

The high KL penalty acted as a strong tether that prevented any meaningful policy improvements. The model's EMA reward plateaued around 0.35, well below $\tau = 0.7$. This meant that the curriculum never advanced beyond Tier 1, spending 239 of the 250 training steps in Tier 1. The strict importance weight clamp further suppressed update magnitude. The result (29.62% `Pass@1`) fell below even the SFT baseline, effectively reducing the run to SFT with RL overhead.

5.2.2 Ablation ($\tau = 0.55$, strict clamp [0.8, 1.2], low KL)

This run collapsed. The EMA was initialized to the first observed batch reward (0.8875), which immediately satisfied both permanent advancement conditions. The model reached Tier 3 at step 2, received near-zero reward for the remaining 248 steps while the EMA decayed to zero. KL divergence tracking became `NaN` after step 56, indicating training instability. Both `Pass@1` and `Pass@100` were 0.00%.

5.2.3 Hybrid ($\tau = 0.55$, loose clamp [0.5, 2.0], KL=0.01)

With the looser clamp and lower KL coefficient, the policy explored more freely and KL grew steadily to 0.14 by step 250. However, probe batches drawn from Tier 2 (with near-zero reward) were folded into the same EMA used to decide whether to probe Tier 2. This prevented permanent advancement. The model spent 87% of training on Tier 1 and achieved 36.12% `Pass@1`, below the plain RLOO baseline.

5.3 Fixed Curriculum Run

The fixed run resolves all three identified failure modes simultaneously and is the first configuration to achieve the intended curriculum behavior. The EMA starts at zero and grows organically on Tier 1 anchor batches, reaching 0.60 at step 59, reaching above the $\tau + \delta = 0.5$ threshold after the required 50-step dwell and triggering a permanent advancement from Tier 1 to Tier 2. It continues to build on Tier 2, reaching 0.89 at step 172 and triggering a permanent advancement from Tier 2 to Tier 3. The model ultimately spends 51 steps on Tier 1, 174 on Tier 2, and 275 on Tier 3, with a stable KL trajectory growing from 0 to 0.18 and no training instabilities. Despite correct mechanistic behavior, the fixed curriculum achieves 42.62% `Pass@1` and 64.00% `Pass@100` at the final (step-500) checkpoint, both values below plain RLOO (46.75%, 70.00%).

We also observe a checkpoint-level tradeoff:

- Evaluating the `step-400` checkpoint (mid-Tier-3, EMA=0.69) yields 28.88% `Pass@1` and 70.00% `Pass@100`. This is the same `Pass@100` as plain RLOO but substantially lower `Pass@1`.

- The final checkpoint (EMA=0.27, after 100 additional steps of hard Tier-3 training) reverses this: `Pass@1` increases to 42.62% while `Pass@100` falls to 64.00%.

This within-run dynamic (high diversity early in hard-tier training, concentrating to higher precision as training continues) is an exploration-exploitation tradeoff that does not appear in plain RLOO, which operates on a fixed difficulty distribution throughout.

5.4 Exploration Tradeoff

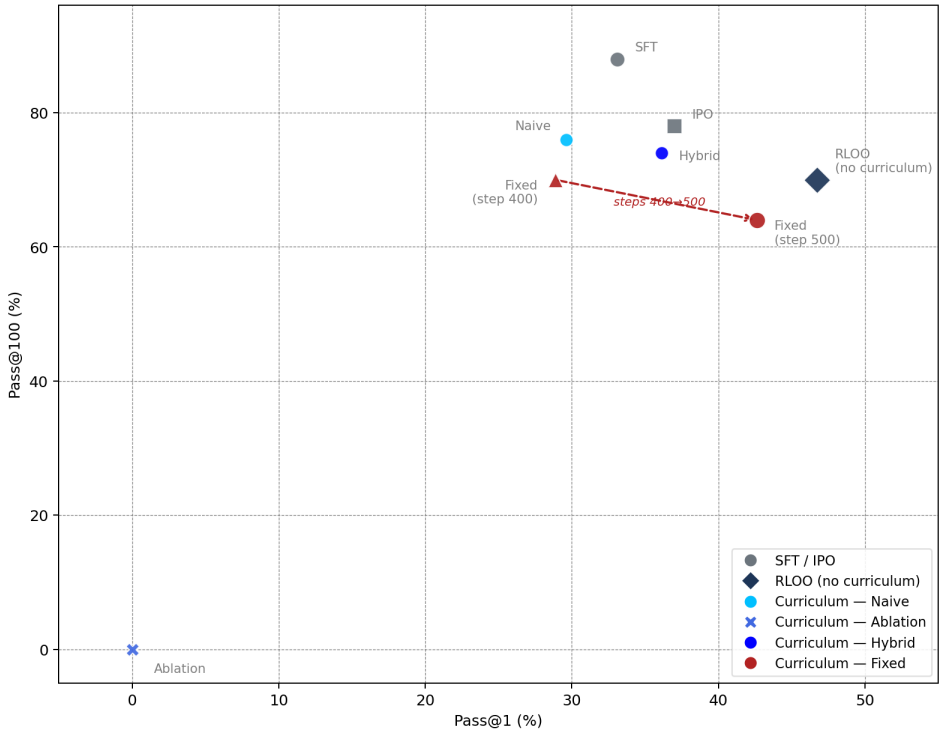

Figure 2: Exploration-Exploitation Tradeoff Across Methods.

A consistent inverse relationship between `Pass@1` and `Pass@100` appears across all methods and checkpoints. As shown in Table 2, every intervention that increases `Pass@1` reduces `Pass@100`: SFT (33.10%, 88.00%), IPO (37.00%, 78.00%), RLOO (46.75%, 70.00%). This tradeoff reflects the progressive concentration of the policy distribution under increasing exploitation pressure.

The curriculum runs occupy a distinct region of this tradeoff space. The non-collapsed curriculum runs exhibit higher `Pass@100` (Naive: 76%, Hybrid: 74%) than plain RLOO (70%). This is consistent with the fact that these models spent most of their training time on tractable Tier 1 problems and never concentrated the policy on hard 4-operand solution paths. The fixed curriculum inverts this pattern: its final checkpoint achieves the lowest `Pass@100` of any non-collapsed run (64%). This is because extensive Tier 3 training concentrates the policy more aggressively than unconstrained RLOO. Enforcing hard-tier training only after soft-tier mastery produces a more extreme form of exploitation than training on the mixed distribution from the start.

6 Discussion

6.1 Failure Mode Taxonomy

Our ablations collectively document three distinct failure modes for competency-gated curricula applied to warm-started LLM policies. These three failure modes are independent:

- The Naive run suffers from threshold miscalibration and clamp rigidity.
- The Ablation run suffers from cold-start inflation.
- The Hybrid suffers from probe contamination and group size.

Each individually prevents the curriculum from functioning, and all three must be addressed simultaneously.

6.1.1 Cold-start EMA inflation

When a warm-started policy has already internalized easier-tier syntax, initializing the EMA to the first batch reward proves to be dangerous. A post-IPO policy achieves high Tier 1 accuracy immediately, producing a high initial EMA that cascades through advancement gates before any genuine tier-level learning has occurred. The fix (initializing $R_{\text{avg}} = 0$ and bootstrapping from anchor batches) ensures the EMA reflects training-time competency rather than initialization-time capability.

6.1.2 Probe-batch EMA contamination

When stochastic probes of harder tiers return near-zero reward, folding them into the same EMA used to generate probe decisions creates a self-defeating negative feedback loop. Probe failures suppress the signal that would approve further probing. Restricting EMA updates to anchor-tier batches decouples the competency signal from sampling noise.

6.1.3 Insufficient group size for RLOO variance

With group size 2, each response's LOO baseline is a single other sample, which is a noisy estimator of expected reward. This noise destabilizes the advantage estimates sufficiently to prevent the policy from making progress even when curriculum mechanics are otherwise correct. Increasing the group size to 8 (32 total rollouts per update step) stabilizes the gradient signal and allows genuine tier advancement.

6.2 Comparison to Plain RLOO

The fixed curriculum reaches 42.62% `Pass@1`, below plain RLOO's 46.75%. Since the two runs were not trained under matched conditions, we caution against a strong reading of this gap. Plain RLOO ran for only 100 optimizer steps with batch size 16 and weak regularization (entropy 0.001, KL 0.001), whereas the fixed curriculum ran for 500 steps with batch size 4 and stronger regularization (entropy 0.01, KL 0.01). The curriculum thus used roughly five times more gradient updates and a comparable-or-larger number of total rollouts. Even then, it underperformed, which is the opposite of the sample-efficiency gain we hypothesized.

Because the comparison differs along at least four axes (step count, batch size, entropy, KL), we cannot cleanly attribute the gap to curriculum design versus the baseline's larger batch or stronger per-step signal. The single most important missing experiment is a matched-budget RLOO baseline trained with the curriculum's exact configuration (batch size 4, entropy 0.01, KL 0.01, 500 steps, no curriculum); only that isolates the curriculum's effect. What we can state with confidence is that the curriculum spends 225 of its 500 steps (45%) on Tiers 1–2 before Tier 3 dominates, and Tier 1 competency does not transfer directly to the 4-operand-heavy test set. However, whether matched-budget curriculum training would surpass RLOO remains an open question.

6.3 Limitations

This study is limited to a single model (`Qwen-2.5-0.5B`) and a single benchmark (Countdown). The three-tier curriculum was constructed manually based on operand count and target magnitude

heuristics. An automated difficulty estimator may have yielded a more principled and dynamic partitioning. The binary reward signal provides no gradient information for near-miss solutions. This may have limited curriculum effectiveness on Tier 3 tasks where the search space is maximally sparse. Finally, we evaluated only final and a single intermediate checkpoint. A full checkpoint sweep may have provided a more complete picture of within-run training dynamics.

7 Conclusion

We investigated competency-gated curriculum reinforcement learning for a sub-billion parameter LLM on the Countdown arithmetic benchmark. Our three-phase pipeline (SFT to IPO to RLOO) progressively improves `Pass@1` from 33.10% (SFT) to 46.75%. Our three curriculum ablations reveal that common implementation choices (warm EMA initialization, unconditional EMA updates, and small group size) are each individually sufficient to prevent the curriculum from functioning. The worst case caused complete training collapse (0% `Pass@1` and `Pass@100`). After correcting all three, the fixed curriculum achieved the first genuinely functional tier progression in our experiments (Tier 1 to Tier 2 at step 59, Tier 2 to Tier 3 at step 172), and reached 42.62% `Pass@1`. Though this was the best curriculum result, it was still 4.13 points below a plain RLOO baseline trained for far fewer steps under an unmatched configuration. We therefore cannot claim a sample-efficiency benefit: the curriculum required five times more gradient steps and still underperformed. The confounded baseline prevents a clean attribution of the gap, but the work provides contribution through the three-failure-mode taxonomy and the implementation guidelines that follow from it. Establishing whether the curriculum helps under a matched-budget protocol is a central question we leave to future work.

8 Team Contributions

- **Donna Choi** I am the sole contributor to this project.

Changes from Proposal Our proposal framed the project as a direct test of whether a competency-gated curriculum improves sample efficiency over a standard RLOO baseline. Two changes occurred.

- Our plain RLOO baseline was trained under a different configuration (fewer steps, larger batch, weaker regularization) than the curriculum runs, so the clean, matched sample-efficiency comparison we originally proposed remains incomplete and is identified as the key future experiment. This is also the reason why the title of the paper has changed deviated from the original proposal.
- The bulk of our effort shifted from the head-to-head curriculum-versus-baseline comparison to diagnosing why some of our naive curriculum implementations failed. The resulting three-failure-mode taxonomy (cold-start EMA inflation, probe contamination, insufficient group size) became the project's primary contribution.

We report `Pass@1` and `Pass@100` in place of the proposed exact-match and rule-adherence metrics.

References

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

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 REINFORCE Samples, Get a Baseline for Free! https://openreview.net/forum?id=r1lgTGL5DE

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