

Extended Abstract

Motivation Reinforcement learning (RL) fine-tuning significantly improves reasoning performance of large language models (LLMs). In the context of problem solving applications (e.g., count down task), the model is required to solve problems with different difficulty levels. However, early in training, model fails on majority of hard reasoning tasks and this yields sparse rewards, making early-stage RL optimization challenging and inefficient. Moreover, hard reasoning tasks require more exploration (usually controlled by entropy regularization), but applying uniform entropy regularization across all difficulty levels is suboptimal. Specifically, it introduces unnecessary exploration noise on easy problems (largely deterministic) while providing insufficient exploration on hard problems where diverse solutions are most needed.

Method We propose two main enhancements that can help with the different difficulty-levels problem solving RL fine-tuning. First, improve sample efficiency by adaptively increasing curriculum complexity during the RL training process (**Adaptive curriculum scheduling**). This allows the model to warm up fast on easier examples to reach acceptable accuracy, then gradually transition toward more difficult reasoning tasks. Second, encourage more exploration for difficult examples by introducing **difficulty-conditioned entropy regularization** into the optimization objective function. The **novel** part is to **jointly** apply adaptive curriculum scheduling and difficulty-conditioned entropy regularization within a single training loop. To the best of our knowledge, joint algorithm has not been investigated in literature.

Implementation Our proposed enhancements are built based on the Reinforce Leave-One-Out (RLOO), to fine tune base model QWEN2.5-.5B on countdown task. In particular, **adaptive curriculum scheduling** starts training on easy examples to quickly establish a capable policy, then gradually blend in medium and hard examples based on predefined accuracy thresholds. A rollback mechanism temporarily reduces hard problems in case of collapses. The dataset is split into easy (3-number), medium (4-number, target ≤ 55), and hard (4-number, target > 55). We use smoothed accuracy over 30 steps as switching metric, with minimum number of steps per stage. To implement the **difficulty-conditioned entropy regularization**, RLOO objective function is regularized with a difficulty weighted entropy bonus $\lambda(d)$.

Results We compare against two flat-entropy RLOO baselines. The high entropy (1e-3) shows poor pass@1 reliability (medium: 38.2%, hard: 46.0%) but a strong pass@16 ceiling (medium: 70.3%, hard: 70.5%), while the low entropy (2e-4) improves reliability (medium: 43.6%, hard: 47.6%) but collapses the ceiling (medium: 61.2%, hard: 67.0%). Our **Combined difficulty-conditioned entropy + adaptive curriculum (with extended medium training stage)** achieves **best of both**: highest reliability (**medium: 44.7%, hard: 51.2%** at pass@1) while fully maintaining the ceiling (**medium: 69.9%, hard: 70.8%** at pass@16).

Discussion As depicted in different results, using a flat entropy for RLOO can either help with reliability (pass@1) or performance ceiling (pass@16) but not both at the same time, based on how aggressive entropy is selected. This is intuitive as high entropy (more exploration) can help introducing more solutions for difficult examples but may impact the reliability of easier examples. Consequently, having difficulty based entropy is essential and effectively combine the best of the two factors. Moreover, adaptive curriculum learning adds stability and benefits from extended medium region. Specifically, training dynamics show that curriculum runs are monotonically improving medium/hard accuracy, suggesting higher benefits with longer training, unlike non-curriculum runs which are very noisy.

Conclusion Difficulty-conditioned entropy with extended curriculum medium stage achieves the **best reliability (pass@1)** while fully **maintaining** the knowledge ceiling (**pass@16**), confirming the benefit of the combined technique. Specifically, different entropy selection adds flexibility for exploration based on difficulty while curriculum learning helps to stabilize training. The smooth curriculum training dynamics suggest further gains with longer training. Another useful lesson is extending the medium stage (or other stages) is very beneficial as long as accuracy is still improving. **Future work** to explore training beyond 100 steps and fine tuning for medium threshold. Also, optimizing the medium/hard splitting technique is worth more investigation.

Adaptive Curriculum Learning with Difficulty-Conditioned Entropy Regularization for RL Fine-Tuning of Large Language Models

Mahmoud Elgenedy
Department of Computer Science
Stanford University
melgened@stanford.edu

Abstract

Reinforcement learning (RL) fine-tuning has been shown to significantly improve reasoning capabilities of large language models (LLMs). However, during early training stages of hard tasks, base model rarely solves a problem and reward signal becomes very sparse, making early-stage RL optimization challenging and very inefficient. In addition, using a flat entropy regularization to encourage exploration for hard tasks is suboptimal. In particular, flat entropy introduces unnecessary exploration noise on easy problems (largely deterministic) while providing insufficient exploration on hard problems where diverse solutions are most needed. To solve such problems, we propose combining adaptive curriculum RL training with difficulty based entropy regularization to enhance Reinforce Leave-One-Out (RLOO) algorithm. Specifically, adaptively increasing curriculum complexity based on training accuracy allows the model to warm up fast on easier examples to reach acceptable accuracy, then gradually transition toward more difficult reasoning tasks. In addition, difficulty-conditioned entropy regularization encourages more exploration for difficult examples while avoids unnecessary exploration noise for easier ones. For the Countdown arithmetic reasoning, our proposed combined technique achieves up to 44.7% and 51.2% pass@1 on medium and hard difficulty questions respectively, outperforms best settings of flat entropy baselines, while preserving performance at pass@16 knowledge ceiling. In other words, baseline flat entropy tuning can either help with reliability or knowledge ceiling, while our proposed algorithm is shown to enhance reliability while maintaining knowledge ceiling. We also observe smoother training dynamics and better generalization with curriculum learning, especially with an extended intermediate difficulty stage. Our results highlight that complementing RL fine-tuning with adaptive curriculum and difficulty-based exploration yields more robust and efficient training.

1 Introduction

LLMs develop broad linguistic knowledge through pretraining on massive text data. However, pretraining alone is not enough to achieve high accuracy on task-specific applications. Pretraining teaches the model the statistical structure of language (predict next token), but does not teach how to reason through a problem or produce a correct answer of a question. To bridge this gap, post-training phase is required to fine-tune the model to task-relevant feedback which helps direct the model behavior toward the desired capability Zhang et al. (2025a); Jiang et al. (2026).

Two main methodologies dominate post-training, namely, Supervised Fine-Tuning (SFT) and Reinforcement Learning (RL). SFT maximizes the likelihood of tokens on curated demonstrations. However, since SFT is imitative, the model performance is limited by the quality of the training data

and may suffer overfitting. On the other hand, RL overcomes SFT limitation by optimizing a reward signal, allowing the model to discover new solutions through exploration rather than imitation Guo et al. (2025); Jaech et al. (2024). In practice, SFT is applied first to establish a base capable model, then RL is used to push beyond the demonstration ceiling. In other words, RL enables LLMs to improve reasoning through self-evolution with reduced reliance on human annotations Guo et al. (2025).

In the context of RL fine-tuning, two paradigms are investigated in literature dependent on how reward is defined. When ground-truth reward is hard to specify but responses can be ranked, preference-based methods are used. This includes traditional Proximal Policy Optimization PPO with an explicit reward model (RLHF) Ouyang et al. (2022), and implicit reward approaches such as Direct Preference Optimization DPO Rafailov et al. (2024) and Identity Preference Optimization IPO Azar et al. (2024). When reward is binary and verifiable, as in math or code reasoning, verifier-based methods such as Reinforce Leave-One-Out RLOO Ahmadian et al. (2024) apply policy gradients directly without any reward model. Figure 1 illustrates the full pipeline, from pretraining through SFT to the four main RL fine-tuning algorithms.

However, RL training for LLMs introduces several challenges. For instance, in arithmetic reasoning as tasks vary in difficulty, hard tasks where the base model struggles to solve (early in training stages), result in sparse reward signals that slow down training and prevent effective optimization. Moreover, to encourage exploration, using a flat entropy regularization regardless of task difficulty is inefficient, as hard tasks require more exploration while easy tasks are more deterministic and should avoid exploration noise. Our work builds on RLOO in the verifier-based setting, extending it with adaptive curriculum scheduling and difficulty-conditioned entropy regularization.

2 Related Work

Adaptive Curriculum RL for LLMs, where training examples are gradually shifted from easy to hard, is one of the promising ideas to enhance sample efficiency and improve optimization process. A variety of researches explored different ideas on the difficulty switching mechanism. For instance, Bercovich et al. (2025); Team et al. (2025), proposed simple switching based on number of training steps. Moreover, Parashar et al. (2025) proposed the Easy-to-Hard (E2H) learner, which employs a probabilistic scheduler that gradually shifts focus from easy to hard. More advanced approaches introduce automatic curriculum learning methods that concurrently learn a curriculum policy during RL fine-tuning, such as Self-Evolving Curriculum (SEC) Chen et al. (2025b). In addition, SOAR Sundaram et al. (2026), a self-improvement framework, proposes a teacher–student setup in which a teacher copy of the model generates synthetic problems for a student copy and is rewarded based on the student’s improvement on a small set of hard problems.

Another direction of enhancement, entropy regularization has been widely used to encourage exploration during RL fine-tuning of LLMs Cui et al. (2025); Wang et al. (2025a,b). However, applying uniform entropy regularization across all training examples encourages long-form reasoning regardless of task difficulty, which is suboptimal and introduces unnecessary noise for easy examples that are largely deterministic and do not require extensive exploration. To address this limitation, adaptive entropy regularization conditions the strength of exploration on the difficulty of training examples, hence balancing increased exploration for hard examples with stronger exploitation for easier ones. Several works have explored adaptive entropy regularization in the context of RL fine-tuning for LLMs, including Zhang et al. (2025b); Chen et al. (2025a); Luo et al. (2026).

In this proposal, for the application of countdown arithmetic reasoning Gandhi et al. (2024), we investigate the combination of adaptive curriculum learning together with difficulty-conditioned entropy regularization. To the best of our knowledge, this combined approach has not been previously studied in the context of RL fine-tuning for LLMs. The proposed method aims to improve sample efficiency by gradually increasing training difficulty, while jointly enabling harder examples to benefit from increased exploration through variable entropy difficulty-based regularization.

3 Method

Our method enhances base RLOO algorithm with two key components: an adaptive three-stage curriculum learning and difficulty-conditioned entropy regularization.

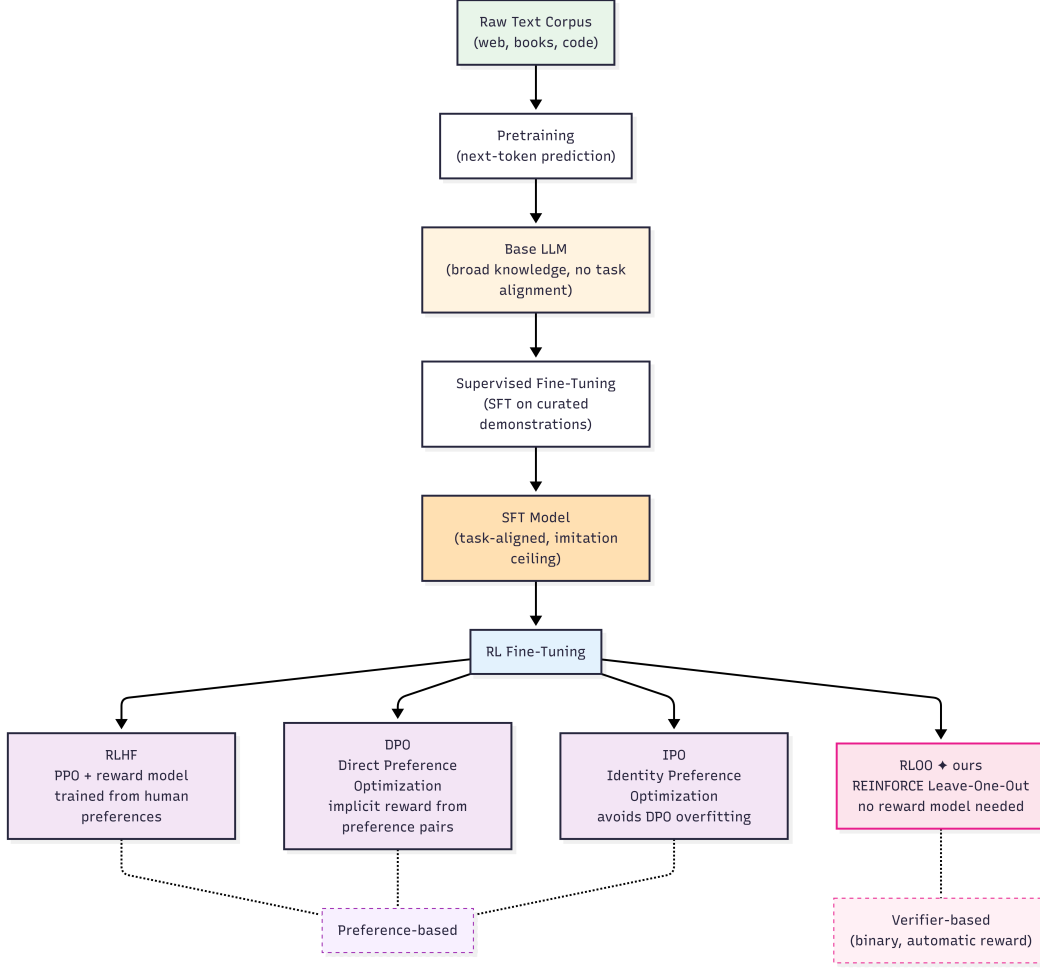


Figure 1: LLM Post-Training Pipeline

3.1 Base Model and Dataset

We use **Qwen2.5-0.5BTeam** (2024) as our base model, starting from a fine-tuned checkpoint via SFT on Countdown demonstrations asingh15 (2026), which provides a capable initialization for RL training. Our task is the **Countdown problem** asingh15 (2024), where the model is given a set of 3 or 4 numbers and required to reach a target value using basic arithmetic operations. The reward is 1 if the expression is correct, 0 otherwise.

3.2 Basic RLOO

In the basic RLOO algorithm Ahmadian et al. (2024), we sample k responses $y^{(1)}, \dots, y^{(k)} \sim \pi_{\theta}(\cdot | x)$ and compute the RLOO for each sample by subtracting the average reward of the other samples from its reward (a baseline to reduce variance of samples). This advantage is used to scale the policy gradient $\nabla \log \pi_{\theta}(y^{(i)} | x)$. When samples are computed from a different behavior policy μ , we correct for this mismatch using importance weights. The objective can be written as,

$$\mathcal{L}_{\text{PG}} = \frac{1}{k} \sum_{i=1}^k w(y^{(i)}, x) A_{\text{RLOO}}^{(i)} \nabla \log \pi_{\theta}(y^{(i)} | x), \quad y^{(1)}, \dots, y^{(k)} \stackrel{\text{i.i.d.}}{\sim} \pi_{\theta}(\cdot | x)$$

$$\text{where } A_{\text{RLOO}}^{(i)} = R(y^{(i)}, x) - \frac{1}{k-1} \sum_{j \neq i} R(y^{(j)}, x)$$

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

The final objective includes a KL (Kullback-Leibler) term to keep the policy close to a reference policy and an entropy term to encourage exploration.

$$\mathcal{L} = \mathcal{L}_{\text{PG}} + \lambda_{\text{KL}} \mathcal{L}_{\text{KL}} - \lambda_H \mathcal{L}_H$$

$$\mathcal{L}_{\text{KL}} = \frac{1}{k} \sum_{i=1}^k \left(\log \pi_\theta(y^{(i)} | x) - \log \pi_{\text{ref}}(y^{(i)} | x) \right)$$

$$\mathcal{L}_H = -\frac{1}{k} \sum_{i=1}^k \frac{1}{|y^{(i)}|} \sum_t \sum_v \pi_\theta(v | x, y^{(i)} < t) \log \pi_\theta(v | x, y_{<t}^{(i)})$$

Finally, the RLOO Roll-out accuracy is computed as,

$$Acc_{\text{rollout}} = \frac{1}{k} \sum_{i=1}^k \mathbf{1} \left[R(y^{(i)}, x) = 1 \right]$$

Training parameters are summarized in following Table,

Hyperparameter	Value
Base model	asingh15/qwen-sft-countdown-defaultproj
Optimizer	AdamW
Learning rate	1×10^{-5}
Gradient clipping	False
Training steps	100
Batch size	128
Gradient accumulation steps	128
Group size k	8
Entropy coefficient λ_H	0.001
KL coefficient λ_{KL}	0.001

Table 1: RLOO training hyperparameters.

3.3 Adaptive Curriculum Scheduling

We split the training dataset into three difficulty levels: **easy** (3-number problems), **medium** (4-number problems with target ≤ 55), and **hard** (4-number problems with target > 55), where 55 is the median target value across all 4-number problems, to have roughly balanced medium and hard pools. Moreover, training is divided into three stages by difficulty, initially (Stage 1), training uses only easy problems, allowing the model to quickly learn a capable baseline policy for easy tasks. Next, when easy-task smoothed accuracy reaches a predefined accuracy threshold of 70% (after at least 10 steps, measured over a 10-step window), Stage 2 starts with blending medium-difficulty examples. Finally, when medium smoothed accuracy exceeds threshold τ_2 (after at least 30 steps, measured over a 30-step smoothing window), Stage 3 starts mixing hard examples. Table 2 summarizes the curriculum stages and transition conditions.

Curriculum stage transitions are triggered by a rolling-window smoothed accuracy:

$$\tilde{a}_d(t) = \frac{1}{W} \sum_{i=t-W+1}^t a_d(i), \quad W = 30 \text{ steps}$$

We also implement a rollback mechanism when hard-task performance drops below a threshold after entering Stage 3, the curriculum temporarily reduces hard problems until performance recovers.

	Stage 1	Stage 2	Stage 3
Transition trigger	Start	Easy acc ≥ 0.7	Medium acc $\geq \tau_2$
Min steps in stage	0	10	30
Training mix	Easy (100%)	Easy (37%) + Med (63%)	Easy (12%) + Med (55%) + Hard (33%)
Observed span	Steps 0–17	Steps 17–47 or 17–82	Remainder
Entropy coef.	λ_{easy}	$\lambda_{\text{easy}}, \lambda_{\text{med}}$	$\lambda_{\text{easy}}, \lambda_{\text{med}}, \lambda_{\text{hard}}$

Default $\tau_2=0.3$: exits stage 2 at step 47; Extended mode (best) $\tau_2=0.4$: exits at step 82

Table 2: Adaptive three-stage curriculum schedule. Transition requires the smoothing window to be full before firing. The extended medium variant ($\tau_2=0.4$) keeps the model in stage 2 longer (until step 82), showing better medium-hard tasks performance.

Config	λ_{easy}	λ_{medium}	λ_{hard}
Flat high (baseline)	1×10^{-3}	1×10^{-3}	1×10^{-3}
Flat low (baseline)	2×10^{-4}	2×10^{-4}	2×10^{-4}
Diff ent equal med-hard (best)	5×10^{-5}	2×10^{-4}	2×10^{-4}
Diff ent high (Aggressive)	5×10^{-5}	2×10^{-3}	5×10^{-3}

Table 3: Entropy coefficient configurations. The equal-med-hard config combined with curriculum is our best-performing setting.

3.4 Difficulty-Conditioned Entropy Regularization

For a prompt of difficulty d , we apply a variable entropy bonus weight $\lambda(d)$ which depends on difficulty d . The regularized RL objective can be written as,

$$\mathcal{L} = \mathcal{L}_{\text{RLOO}} - \lambda(d) \mathbb{E}_{y \sim \pi_\theta} \left[\sum_t -\log \pi_\theta(y_t | y_{<t}, x) \right]$$

where $\mathcal{L}_{\text{RLOO}}$ is the standard RLOO policy gradient loss, and the expectation term computes the average entropy of the policy’s token distribution at each generation step, conditioned on the prompt and all previously generated tokens. The regularization term in loss equation is always negative, since the two negative signs will collapse, and $\log(\pi)$ is a logarithm of probability ($l < 1$), which is always negative. Hence, minimizing the total loss would push that regularization to be more negative (more uncertainty \Rightarrow more exploration). The higher λ factor will encourage the training to push that uncertainty higher for more difficult examples. Table 3 summarizes most useful explored entropy settings for both flat and variable entropy coefficients.

3.5 Combined Adaptive Curriculum and Difficulty-Conditioned Entropy Regularization

The key technical contribution we propose is to develop and experiment a single training framework where prompt difficulty can control both data sampling (gradually increase difficulty) and entropy based exploration. Figure 2 elaborates the upgraded RLOO framework versus base RLOO.

4 Experimental Setup

We test the independent contribution of each component through four configurations shown in table 4. All runs use the same base model, dataset, and hyperparameters. Only curriculum and entropy settings vary.

We further ablate **entropy magnitude** (conservative vs. aggressive) to identify boundaries where policy may collapse under aggressive entropy regularization, as shown in Table 3. In addition, we test different settings for **curriculum threshold** ($\tau_2=0.3$ vs. 0.4) to understand effect of extended regions transitions, as detailed in Table 2.

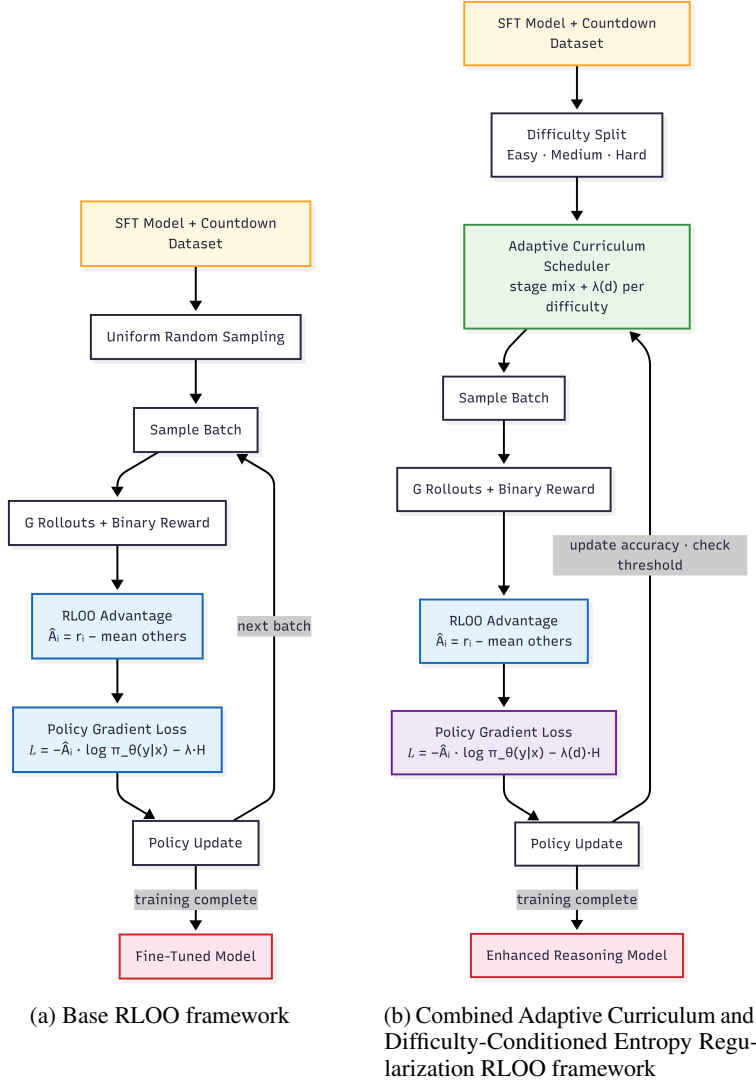


Figure 2: Comparison of base RLOO and our proposed extension.

Run	Curriculum	Diff. Entropy	Focus
Baseline	×	×	Reference: uniform sampling, flat entropy
Curriculum only	✓	×	Focus on effect of difficulty progression
Diff. entropy only	×	✓	Isolate effect of per-difficulty exploration scaling
Combined	✓	✓	Test the combined performance

Table 4: Summary of ablation study cases.

5 Results

5.1 Performance Metrics

We evaluate our method using two main metrics, the $\text{Pass}@k$ and the smoothed training accuracy, to capture both final task performance and training dynamics. In addition, we monitor KL divergence and importance weights as diagnostic signals to track training stability and policy drift throughout the RL fine-tuning.

Method	Medium		Hard	
	Pass@1	Pass@16	Pass@1	Pass@16
Flat High λ (10^{-3})	38.2%	70.3%	46.0%	70.5%
Flat Low λ (2×10^{-4})	43.6%	61.2%	47.6%	67.0%
Diff-Entropy Only (equal-med-hard)	42.3%	69.1%	48.5%	68.2%
Combined (Ours, $\tau_2=0.4$)	44.7%	69.9%	51.2%	70.8%

Table 5: Pass@k evaluation on medium and hard difficulty subsets. Our combined method achieves the best reliability (pass@1) while fully maintaining the knowledge ceiling (pass@16), resolving the trade-off present in flat-entropy baselines.

- **Pass@k** ($k \in \{1, 2, 4, 8, 16\}$): the fraction of test problems solved within k independent samples, computed separately for easy, medium, and hard sub-tasks. Pass@1 measures reliability; pass@16 measures knowledge ceiling.
- **Smoothed training accuracy per difficulty**: rolling-window average reward over the last $W=30$ steps for each difficulty level $d \in \{\text{easy, medium, hard}\}$, used both as a curriculum trigger signal and as a measure to compare learning dynamics across runs. Note that some figures are measured with smaller window (e.g., $W = 5$) to better understand the training dynamics.
- **KL divergence**: tracks how far the learned policy deviates from the reference (SFT) model throughout training, used to monitor training stability and make sure the policy did not drift much from the base model.
- **Importance weights**: the ratio $\pi_\theta(y|x)/\pi_{\text{ref}}(y|x)$ between the current policy and the reference, to detect distributional collapse. The large importance weights indicate that the policy is generating responses very different from the SFT distribution.

5.2 Quantitative Evaluation

Table 5 shows the main results of the pass@k. The key finding is that flat entropy tuning suffers a trade-off between reliability (pass@1) and knowledge ceiling (pass@16), whereas **our method avoids this trade-off** (achieving the highest pass@1 while fully maintaining the pass@16 ceiling),

- Flat high λ (10^{-3}): strong pass@16 ceiling (medium: 70.3%, hard: 70.5%) but poor pass@1 reliability (medium: 38.2%, hard: 46.0%) due to over-exploration.
- Flat low λ (2×10^{-4}): improved pass@1 reliability (medium: 43.6%, hard: 47.6%) but **collapsed ceiling** (medium: 61.2%, hard: 67.0%) due to under-exploration.
- **Our combined method**: achieves the best pass@1 on medium (44.7%) and hard (51.2%), while fully maintaining the knowledge ceiling (medium: 69.9%, hard: 70.8%), which is comparable to the high-entropy baseline at pass@16.

5.3 Qualitative Analysis

Training Dynamics: Figure 3 shows smoothed training accuracy (rolling window $W=5$) on medium and hard problems across four configurations: (1) Baseline flat low ($\lambda=2 \times 10^{-4}$, uniform sampling), (2) Diff-entropy only, no curriculum ($\lambda_{\text{easy}}=5 \times 10^{-5}$, $\lambda_{\text{med}}=\lambda_{\text{hard}}=2 \times 10^{-4}$), (3) Curriculum $\tau_2=0.3$ (same entropy, adaptive switching at medium accuracy ≥ 0.3), and (4) Curriculum $\tau_2=0.4$ extended medium (same entropy, switching delayed until medium accuracy ≥ 0.4).

Without curriculum, the baseline and diff-entropy-only runs show high-variance fluctuations throughout training on both medium and hard tasks. However, the curriculum runs produce clearly smoother training dynamics, confirming that gradually increasing difficulty (curriculum learning) reduces training noise. Note that for hard accuracy, curriculum runs only start after entering stage 3 (the $\tau_2=0.3$ run from step 47 and the $\tau_2=0.4$ run from step 82), meaning the comparison window is shorter and the noise-reduction benefit may be less visible for the $\tau_2=0.4$ run.

Entropy Ablation: Figure 4 shows pass@k curves for five entropy configurations without curriculum. For the flat entropy baseline curves, we can clearly see the crossing of the high- λ (10^{-3}) and low- λ

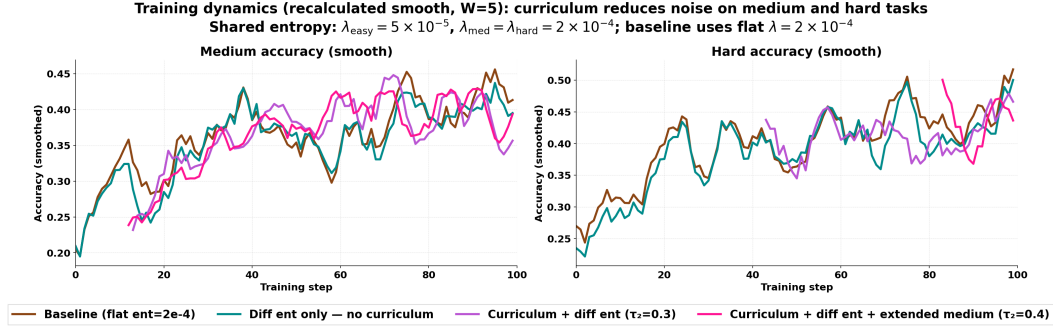


Figure 3: Smoothed training accuracy (rolling window $W=5$) on medium and hard problems for four configurations.

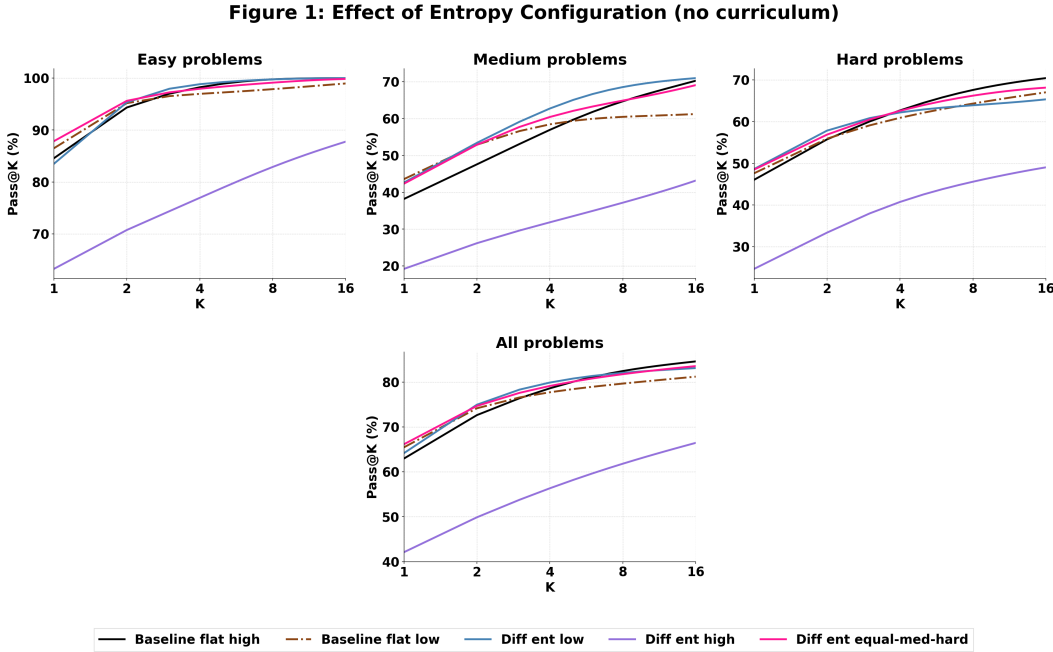


Figure 4: Pass@ k curves for five entropy configurations without curriculum, evaluated on all, easy, medium, and hard problems.

(2×10^{-4}) curves, which highlights the trade-off, high entropy achieves a stronger pass@16 ceiling but poor pass@1 reliability, while low entropy improves pass@1 but collapses the ceiling. Our diff-entropy equal-mid-hard configuration ($\lambda_{\text{easy}}=5 \times 10^{-5}$, $\lambda_{\text{med}}=\lambda_{\text{hard}}=2 \times 10^{-4}$) begins above the low- λ baseline at pass@1 and ends above the high- λ baseline at pass@16, confirming that difficulty-conditioned entropy resolves the trade-off even without curriculum. Note that aggressive entropy settings ($\lambda_{\text{hard}}=5 \times 10^{-3}$) cause policy collapse as shown in the graphs.

Curriculum Impact: Figure 5 shows pass@ k curves comparing the curriculum ablation. Adding curriculum to the diff-entropy configuration yields the best pass@1 across all runs while fully maintaining the pass@16 ceiling. The diff-entropy-only run (no curriculum) has already been shown to improve over both flat baselines, but the combined method pushes pass@1 further. Specifically, medium: 44.7% vs 42.3%, hard: 51.2% vs 48.5%, without degrading the knowledge ceiling (pass@16 medium: 69.9% vs 69.1%, hard: 70.8% vs 68.2%). This confirms that curriculum learning is complementing diff-entropy. Specifically, entropy adjusts exploration quality, while curriculum stabilizes training with progressive difficulty presented to the policy.

Extended Medium Stage: Figure 6 compares the two medium curriculum threshold settings on training dynamics. Raising τ_2 from 0.3 to 0.4 keeps the model in stage 2 until step 82 versus step 47,

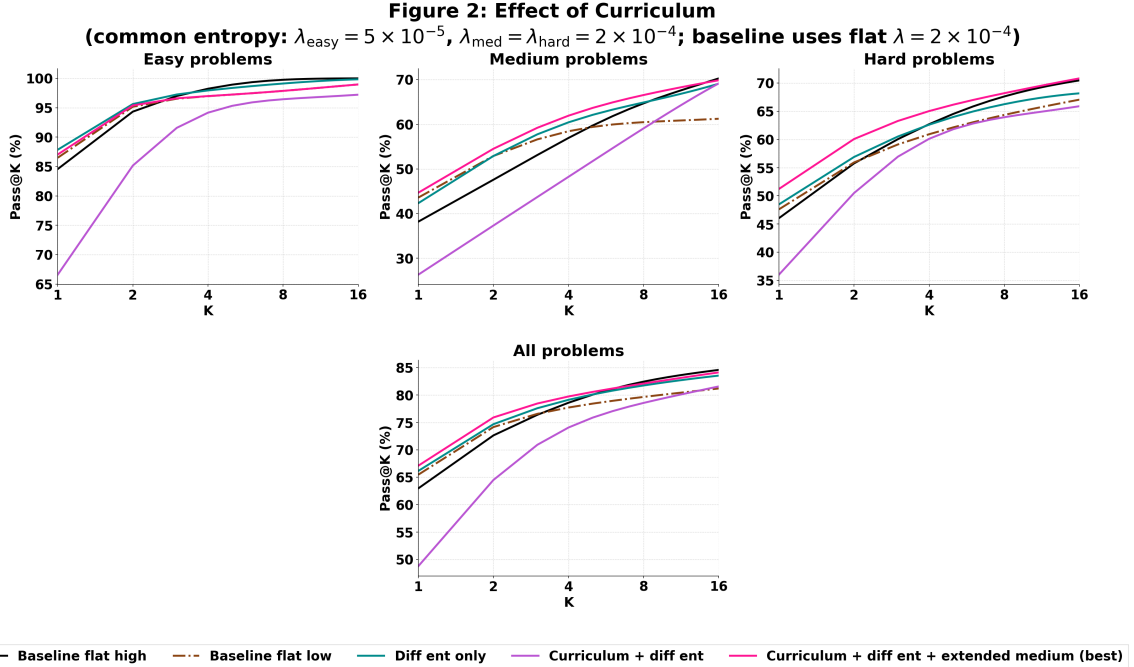


Figure 5: Pass@ k curves for the curriculum ablation, evaluated on all, easy, medium, and hard problems.

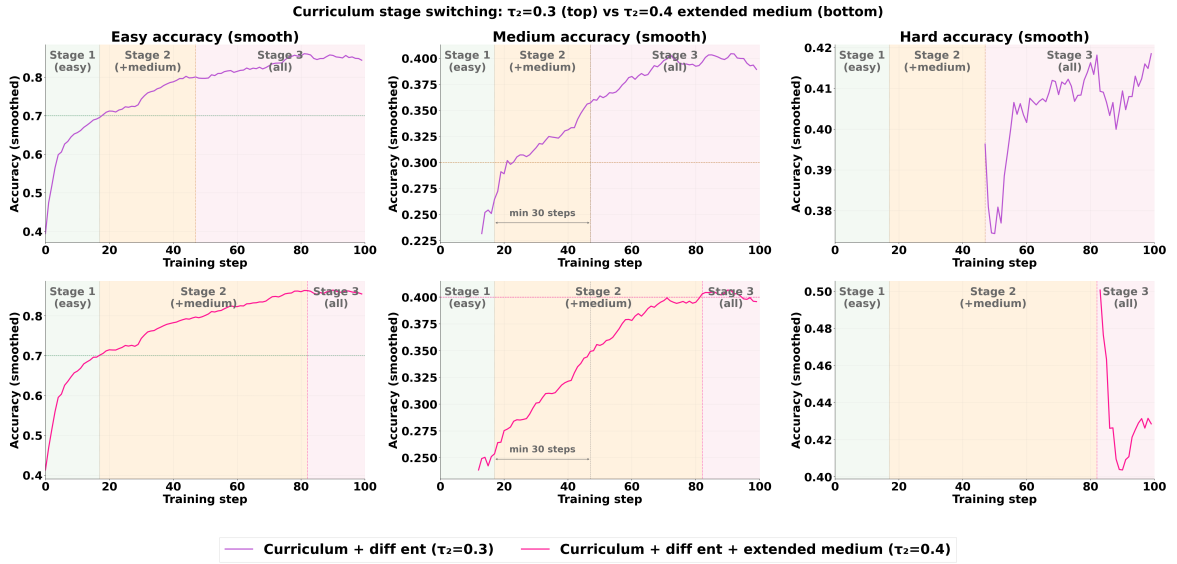


Figure 6: Curriculum stage switching for standard $\tau_2=0.3$ (top row) versus extended $\tau_2=0.4$ medium (bottom row) across easy, medium, and hard accuracy.

exposing it to medium problems for longer before adding hard examples. This yields stronger final accuracy on both medium and hard tasks, as reflected in the pass@ k results in figure 5. This reveals a key curriculum design principle: **if accuracy is still clearly improving, it is better to wait before advancing to harder problems rather than switching on a fixed threshold.**

6 Discussion

The results validate that combining curriculum learning with difficulty-conditioned entropy yields better RL fine-tuning than either technique alone. The difficulty conditioning of the entropy provides flexibility of exploration based on difficulty, which helps to avoid the trade-off between reliability and ceiling that flat entropy tuning could not tolerate. The curriculum component provides stable, monotonic learning progress that results in additional gain for medium and hard problems.

In addition, the curriculum training dynamics do not show saturation until step 100, suggesting that the benefit of curriculum will scale with longer training. Another observation that running aggressive entropy ($\lambda_{\text{hard}}=5\times 10^{-3}$) without curriculum causes policy to almost collapse, with pass@1 dropping to $\approx 1\%$ (Figure 4). However, examining the same aggressive coefficients with curriculum learning recovers to 58.7% pass@1, indicating that curriculum acts as a stabilizer for aggressive entropy settings.

Moreover, extending the medium stage reveals an important curriculum design principle, if accuracy is still clearly improving, it is better to wait before advancing to harder problems rather than switching early on a fixed threshold. This suggests that accuracy-based thresholds are more effective than fixed step-count switching alone, and that further tuning of the threshold parameters τ_1 and τ_2 is likely to yield additional gains.

Another point worth investigation is the medium-hard difficulty partitioning. Having a deeper look at figures 4 and 5, we notice that Pass@k on medium problems is consistently lower than on hard. Two hypothesis for this discrepancy: (a) **Inaccurate difficulty split**, as the median target value (55) may not accurately reflect arithmetic difficulty. For instance, smaller targets may impose more limited solution possibilities, making them harder to solve. (b) **Training sequence bias**, as medium problems are introduced in stage 2 when the policy is still relatively weak, while hard problems only blends in stage 3 after medium training has already enhanced the policy. Moreover, stage 3 is the final training phase, meaning the policy is most optimized at the point hard problems are evaluated, possibly overfitting the training distribution of the hard problems. Hence, investigating these two hypotheses is an interesting direction for future work, for instance, by evaluating a curriculum that introduces medium and hard problems simultaneously, or if there is a way to estimate the solution-space and use it for splitting instead of target value.

7 Conclusion

We proposed an RL fine-tuning approach that extends RLOO with combined adaptive curriculum learning and difficulty-conditioned entropy regularization to address sparse rewards and suboptimal exploration in hard reasoning tasks. Our key finding is that flat entropy tuning struggles with a trade-off between reliability (pass@1) and knowledge ceiling (pass@16) whereas our proposed method resolves by conditioning exploration on difficulty and stabilize training through progressive difficulty increase during training process. The combined approach achieves the best pass@1 on medium and hard sub-tasks while preserving the knowledge ceiling, with smoother training dynamics. Future work to evaluate beyond 100 training steps, refine the difficulty partitioning, explore the sensitivity of the region length (τ_2) and blend ratios, possibly with more advanced automated adaptive threshold tuning.

8 Team Contributions

This is a single-author project. The main contributions include,

- **Combined framework:** the first study, to our knowledge, to jointly apply adaptive curriculum scheduling and difficulty-conditioned entropy regularization within a single RLOO training loop. We investigated all necessary parts and design choices of the algorithm including curriculum splitting, switching methods and blend ratios.
- **Empirical analysis:** systematic ablation across entropy settings and curriculum thresholds on the Countdown reasoning task, yielding a best reliability (pass@1) while fully maintaining the knowledge ceiling (pass@16).

9 Acknowledgment

- I am so grateful to the course instructor, project mentor, and TA team for their great support and feedback throughout this project.
- Parts of the implementation, debugging, and visualization code were developed with the assistance of Claude Code (Anthropic).

Changes from Proposal The project mostly followed the initial proposal. However, based on experiments and findings, the study mainly focused on following: (1) the reliability-ceiling trade-off of the baseline flat entropy tuning, which discovered through analyzing different experimental results, and it is a key experimental contribution of the paper. (2) investigating and fine tuning the stages transitions. Specifically, extended Stage 2 (medium \Rightarrow hard) variant was tested after observing that medium accuracy was still improving at the default transition point, (3) impact of low versus aggressive entropy settings for both flat and variable entropy configurations was tested to identify exploration boundaries, (4) the difficulty split counter-intuitive observation (hard performance is better than medium), which worth more investigation as future work.

References

- Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 12248–12267.
- asingh15. 2024. Countdown Tasks 3-to-4. https://huggingface.co/datasets/asingh15/countdown_tasks_3to4. Hugging Face Datasets.
- asingh15. 2026. Qwen SFT Countdown (default project). <https://huggingface.co/asingh15/qwen-sft-countdown-defaultproj>. Hugging Face Model Repository.
- Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, Mark Rowland, Michal Valko, and Daniele Calandriello. 2024. A general theoretical paradigm to understand learning from human preferences. In *International Conference on Artificial Intelligence and Statistics*. PMLR, 4447–4455.
- Akhiad Bercovich, Itay Levy, Izik Golan, Mohammad Dabbah, Ran El-Yaniv, Omri Puny, Ido Galil, Zach Moshe, Tomer Ronen, Najeeb Nabwani, et al. 2025. Llama-nemotron: Efficient reasoning models. *arXiv preprint arXiv:2505.00949* (2025).
- Shuang Chen, Yue Guo, Yimeng Ye, Shijue Huang, Wenbo Hu, Haoxi Li, Manyuan Zhang, Jiayu Chen, Song Guo, and Nanyun Peng. 2025a. Ares: Multimodal adaptive reasoning via difficulty-aware token-level entropy shaping. *arXiv preprint arXiv:2510.08457* (2025).
- Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamaloo. 2025b. Self-evolving curriculum for llm reasoning. *arXiv preprint arXiv:2505.14970* (2025).
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. 2025. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617* (2025).
- Kanishk Gandhi, Denise Lee, Gabriel Grand, Muxin Liu, Winson Cheng, Archit Sharma, and Noah D. Goodman. 2024. Stream of Search (SoS): Learning to Search in Language. *arXiv:2404.03683 [cs.LG]* <https://arxiv.org/abs/2404.03683>
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948* (2025).
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720* (2024).

- Haitao Jiang, Wenbo Zhang, Jiarui Yao, Hengrui Cai, Sheng Wang, and Rui Song. 2026. Supervised Fine-Tuning versus Reinforcement Learning: A Study of Post-Training Methods for Large Language Models. *arXiv:2603.13985 [cs.AI]* <https://arxiv.org/abs/2603.13985>
- Qin-Wen Luo, Sheng Ren, Xiang Chen, Rui Liu, Jun Fang, Naiqiang Tan, and Sheng-Jun Huang. 2026. Compress the Easy, Explore the Hard: Difficulty-Aware Entropy Regularization for Efficient LLM Reasoning. *arXiv preprint arXiv:2602.22642* (2026).
- Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. *arXiv:2203.02155 [cs.CL]*
- Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, et al. 2025. Curriculum reinforcement learning from easy to hard tasks improves LLM reasoning. *arXiv preprint arXiv:2506.06632* (2025).
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. *arXiv:2305.18290 [cs.LG]* <https://arxiv.org/abs/2305.18290>
- 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 preprint arXiv:2601.18778* (2026).
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599* (2025).
- Qwen Team. 2024. Qwen2.5: A Party of Foundation Models. <https://qwenlm.github.io/blog/qwen2.5/>
- Chen Wang, Zhaochun Li, Jionghao Bai, Yuzhi Zhang, Shisheng Cui, Zhou Zhao, and Yue Wang. 2025a. Arbitrary entropy policy optimization: Entropy is controllable in reinforcement finetuning. *arXiv e-prints* (2025), *arXiv-2510*.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. 2025b. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939* (2025).
- Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2025a. Instruction Tuning for Large Language Models: A Survey. *arXiv:2308.10792 [cs.CL]* <https://arxiv.org/abs/2308.10792>
- Xiaoyun Zhang, Xiaojian Yuan, Di Huang, Wang You, Chen Hu, Jingqing Ruan, Kejiang Chen, and Xing Hu. 2025b. Rediscovering entropy regularization: Adaptive coefficient unlocks its potential for llm reinforcement learning. *arXiv preprint arXiv:2510.10959* (2025).

A Additional Experiments

A.1 Large Training Runs (200 steps)

We have few trials to increase the training from 100 steps to 200 steps, but since they are not solid enough, we keep them here in this section. As shown in figure 7 the combined mode trained for 200 steps shows a clear $\text{pass}@K$ improvement over the 200-step flat-entropy baseline, particularly on medium and hard problems. For example, $\text{pass}@1$ on medium increases from $\sim 43\%$ (baseline, 200 steps) to $\sim 50\%$ (combined, 200 steps), with the gap maintained across all K values. However, the training dynamics shown in figure 8 raise a reliability concern. In particular, after 100 steps we notice some divergence in the importance weights whereas by step 200 the importance weights have dropped to 0.78 (versus ~ 1.0 at step 100), indicating large drift compared to base model. This

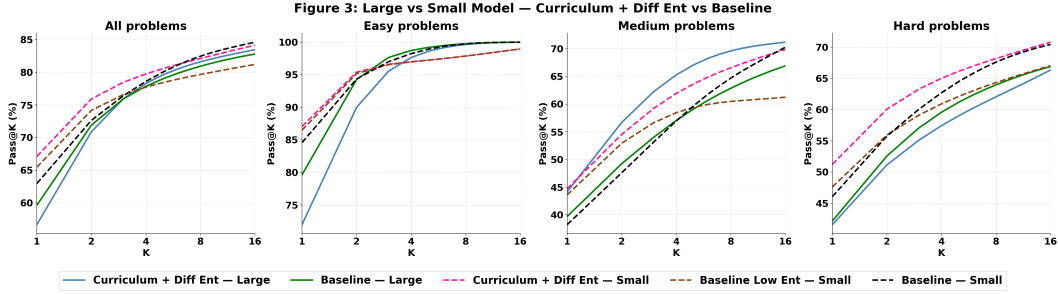


Figure 7: Pass@ k curves for the long runs (200 steps) versus small runs (100 steps), evaluated on all, easy, medium, and hard problems.

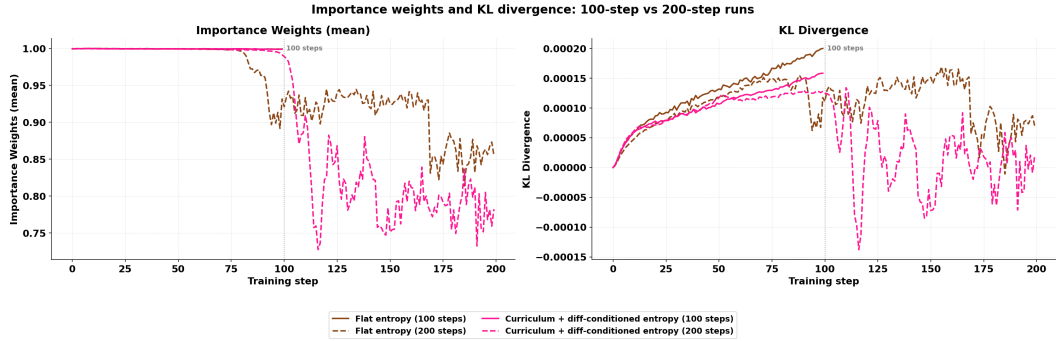


Figure 8: Importance weights (mean) and KL divergence over training for flat-entropy and combined curriculum + difficulty-conditioned entropy runs at 100 and 200 steps.

suggests the gains at 200 steps may be partly unstable as the policy has moved far enough from its reference. Hence, tuning is needed, possibly by introducing a clipping threshold on importance weights, or increasing the KL coefficient β to constrain drift.

A.2 Sample Rollout

In the following table 6, we show a summary of sample rollout of combined mode at best settings.

B Implementation Details

B.1 Code git repository

https://github.com/melgenedy/rloo_difficulty_adaptive.

B.2 Wandb all experiments

https://wandb.ai/melgenedy/rloo_experiments.

Problem: Numbers {83, 78, 1, 39}, target = 82

<think>

Let me try to find a way to get to 82 using these numbers.

First attempt:

$$83 - 78 = 5$$

$$5 + 1 = 6$$

$$6 + 39 = 45$$

Not helping.

Second attempt:

$$78 - 83 = -5$$

$$-5 + 39 = 34$$

$$83 - 34 = 49$$

Not working.

Third attempt:

$$83 - 78 = 5$$

$$5 * 39 = 195$$

$$195 - 1 = 194$$

Not working.

Fourth attempt:

$$78 / 39 = 2$$

$$83 - 2 = 81$$

$$81 + 1 = 82$$

This works!

</think>

<answer> $(83 - (78 / 39)) + 1$ </answer>

<think>

Let me verify:

$$78 / 39 = 2$$

$$83 - 2 = 81$$

$$81 + 1 = 82$$

Yes, this reaches our target of 82.

</think>

Reward: 1.0 (correct) **Verification:** $78 \div 39 = 2$, $83 - 2 = 81$, $81 + 1 = 82$ ✓

Table 6: Full rollout from the combined mode on a hard 4-number Countdown.