Extended Abstract

Preference Optimization and Curriculum RLOO for Countdown Reasoning

Akshar Sarvesh

Motivation. Countdown arithmetic is a useful small-scale testbed for language-model reasoning because fluent chain-of-thought text is not enough: the model must satisfy exact symbolic constraints. Each prompt gives a target number and three or four input numbers, and a valid answer must use every input number exactly once to form an arithmetic expression that evaluates to the target. I study whether preference optimization and online reinforcement learning can improve a supervised Countdown model, and whether a simple difficulty curriculum improves RLOO.

Method. The training pipeline has three stages. First, supervised fine-tuning (SFT) provides the initialization and base model. Second, Identity Preference Optimization (IPO) trains from chosen/rejected response pairs without fitting an explicit reward model. Third, online RLOO samples groups of responses per prompt, scores them with the Countdown verifier, and updates the policy using a leave-one-out reward baseline. The final extension adds curriculum sampling to RLOO. The primary curriculum uses number count as a proxy for difficulty: three-number prompts are treated as easier than four-number prompts. I evaluate both a base staged curriculum and a mixed curriculum that includes a fixed fraction of harder examples during the easy stage. I also test an empirical pass@16-difficulty curriculum, where examples are marked easy if a prior RLOO baseline solves them in 16 samples and hard otherwise.

Implementation. I implemented IPO response-only log-probability computation, the RLOO leave-one-out update, and curriculum-controlled RLOO sampling. A key implementation finding was that the milestone RLOO run was missing a stabilization step; adding it moved the base RLOO result from roughly IPO-level performance to the stronger non-curriculum RLOO baseline used for curriculum experiments. The verifier reward is 1.0 for a valid expression, 0.1 for a formatted but invalid expression, and 0.0 when answer tags are missing.

Results. On 50 held-out Countdown prompts with 16 samples per prompt, SFT reaches average sampled reward 0.347 and IPO improves this to 0.414. RLOO reaches 0.617 average reward and pass@1 of 0.640. Number-count curriculum improves RLOO further: base curriculum reaches average reward 0.635 and pass@16 0.820, while mixed curriculum gives the best average reward, 0.652, and the highest response-level correct fraction, 0.636. The pass@16 empirical difficulty curriculum is useful but not dominant: base pass@16 curriculum reaches 0.641 average reward, while mixed pass@16 curriculum falls to 0.604.

Discussion and Conclusion. The strongest result is that a simple structural curriculum can improve verifier-driven RLOO without changing the model, reward, or optimizer. The pass@16 ablation shows that empirical difficulty labels are not automatically better than simple task-structure labels: they can improve some aspects of behavior, but they require careful calibration. Overall, the project supports the view that online RL with a symbolic verifier can substantially improve arithmetic correctness, but small implementation choices in the RLOO update and curriculum sampling strongly affect the reliability/correctness tradeoff.

Preference Optimization and Curriculum RLOO for Countdown Reasoning

Akshar Sarvesh
Department of Computer Science
Stanford University
asarvesh@stanford.edu

Abstract

I study preference optimization and online reinforcement learning for the Countdown arithmetic reasoning task. Countdown prompts require a model to output an arithmetic expression that uses each provided number exactly once and evaluates to a target. Starting from a supervised model, I implement IPO preference optimization, RLOO, and a curriculum extension to RLOO. IPO improves average sampled reward from 0.347 to 0.414. RLOO improves to 0.617, and number-count curriculum RLOO improves further: the best mixed curriculum reaches average reward 0.652 and pass@16 0.820. I also test pass@16-based empirical difficulty labels as a curriculum signal. This ablation improves over vanilla RLOO in some metrics but does not beat the simpler number-count curriculum, suggesting that curriculum labels based on task structure can be more robust than noisy online difficulty estimates in short training runs.

1 Introduction

Large language models can produce convincing reasoning traces while still failing exact symbolic constraints. The Countdown arithmetic task isolates this issue in a compact environment. A prompt specifies a target value and a set of numbers, and the model must produce an expression in `<answer>` tags that uses each number exactly once and evaluates to the target. For example, for target 98 and numbers $[44, 19, 35]$, one correct answer is `<answer>44 + 35 + 19</answer>`. This task is small enough to score with a deterministic verifier, but difficult enough to expose failures in arithmetic, formatting, and constraint satisfaction.

This project asks three questions. First, does offline preference optimization improve Countdown reasoning beyond supervised fine-tuning? Second, does online RLOO with verifier rewards improve beyond offline preference optimization? Third, can curriculum sampling improve RLOO by ordering examples from easier to harder prompts?

The final system starts from a public SFT Countdown checkpoint, trains an IPO preference model, and then performs RLOO with a symbolic reward function. During implementation, I found that the RLOO objective was sensitive to large importance weights between the sampled policy and the updated policy. Clipping those weights to $[0.5, 2.0]$ substantially improved stability and performance. I then used this RLOO implementation as the base for curriculum experiments.

The main extension is a curriculum over training prompts. The primary curriculum uses a simple structural proxy for difficulty: examples with three numbers are easier than examples with four numbers, since the search space of possible expressions is smaller. I evaluate both a staged base curriculum and a mixed curriculum. I also test a pass@16 empirical difficulty curriculum, where a prior RLOO baseline samples 16 responses for training prompts and marks examples as easy if any response solves the prompt. This ablation is useful scientifically because it tests whether a model-specific difficulty signal beats the simple number-count proxy.

The results support the value of online RL and curriculum, but also show that more elaborate difficulty estimation is not automatically better. IPO improves average reward from 0.347 to 0.414, RLOO improves it to 0.617, and the best number-count curriculum improves it to 0.652. The pass@16 base curriculum reaches 0.641, but the pass@16 mixed curriculum drops to 0.604. The best final method is therefore the mixed number-count curriculum, while pass@16 difficulty is best treated as a diagnostic ablation and future-work direction.

2 Related Work

Learning from human preferences. RLHF trains language models using feedback-derived objectives and is a standard recipe for aligning language models with desired behavior (Ouyang et al., 2022). Direct Preference Optimization (DPO) showed that pairwise preferences can be used to directly optimize a policy without fitting an explicit reward model (Rafailov et al., 2023). IPO gives a related preference objective with a squared target margin and a theoretical framing of preference learning (Azar et al., 2024). This project uses IPO as the offline preference-optimization stage before online RL.

Online RL for language models. Policy-gradient methods for language-model post-training often sample completions from the current model, score them, and update the model under a reference-model regularizer. RLOO uses multiple responses for the same prompt and computes a leave-one-out baseline from the other responses in the group. This is attractive for Countdown because the reward is deterministic and cheap relative to human feedback: no learned reward model is needed.

Curriculum learning. Curriculum learning proposes that training can benefit from presenting examples in a meaningful order, often easier examples before harder ones (Bengio et al., 2009). Countdown gives a natural difficulty signal: using four input numbers creates a larger expression search space than using three input numbers. The extension in this report tests whether this simple curriculum signal improves RLOO. It also compares this structural proxy against pass@16 empirical difficulty labels computed from a trained RLOO baseline.

3 Task and Evaluation

The dataset is `asingh15/countdown_tasks_3to4`, which contains Countdown examples with either three or four input numbers. Each example includes a prompt and ground-truth metadata used by the verifier. The model response is scored by extracting the expression inside `<answer>` tags and checking three properties:

1. the answer tag exists;
2. the expression uses each provided number exactly once;
3. the expression evaluates to the target.

The reward is 1.0 if all conditions hold, 0.1 if the answer is formatted but invalid or incorrect, and 0.0 if no answer is extracted.

Evaluation uses 50 held-out prompts with 16 sampled responses per prompt. I report average sampled reward, pass@k for $k \in \{1, 2, 4, 8, 16\}$, and a score decomposition over all 800 sampled responses: exact correct, formatted but wrong, and missing answer. The pass@k metric is the fraction of prompts for which at least one of the first k samples receives reward 1.0.

4 Methods

4.1 Supervised Fine-Tuning Baseline

The project starts from a supervised Countdown checkpoint. SFT trains the model to imitate correct reasoning and answer formatting. It is useful as an initialization because the model learns the task format, including the need to produce a final expression inside `<answer>` tags. However, SFT alone does not directly optimize the verifier reward, and many samples remain formatted but arithmetically invalid.

4.2 IPO Preference Optimization

IPO trains from a preference dataset of prompt, chosen response, and rejected response triples. For a prompt x, chosen response y_w, rejected response y_l, policy π_θ, and frozen reference model π_{ref}, I compute response-only sequence log probabilities and form

$$h_\theta = (\log \pi_\theta(y_w \mid x) - \log \pi_\theta(y_l \mid x)) - (\log \pi_{\text{ref}}(y_w \mid x) - \log \pi_{\text{ref}}(y_l \mid x)) .$$

The IPO loss is

$$\mathcal{L}_{\text{IPO}} = \left(h_\theta - \frac{1}{2\beta} \right)^2 .$$

Prompt tokens are masked out so that only response tokens contribute to the log probabilities. The IPO run was initialized from the SFT checkpoint, trained on `countdown_tasks_3to4-dpo`, and used learning rate 5×10^{-6}, $\beta = 0.1$, one epoch, and effective batch size 64.

4.3 RLOO with Clipped Importance Weights

RLOO samples $G = 8$ responses per prompt. Each response receives a verifier reward r_i. For each response in the group, the leave-one-out baseline is

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

and the advantage is

$$A_i = r_i - b_i.$$

This baseline compares a response against the other responses from the same prompt, reducing variance while keeping the signal local to each prompt's difficulty.

The policy-gradient term uses response-only sequence log probabilities. Because the model is updated after samples are generated, the update uses an importance ratio between the current policy and the sampling policy:

$$\rho_i = \exp\left(\log \pi_\theta(y_i \mid x_i) - \log \pi_{\text{sample}}(y_i \mid x_i)\right).$$

In the milestone implementation, I had not yet included this clipping step, and base RLOO stayed near the IPO result, around 0.4 average reward. After adding clipping, the base RLOO score immediately jumped to roughly 0.6 average reward. The RLOO implementation therefore clips the ratio:

$$\tilde{\rho}_i = \text{clip}(\rho_i, 0.5, 2.0).$$

The policy objective is proportional to $-\tilde{\rho}_i A_i \log \pi_\theta(y_i \mid x_i)$, with entropy regularization and a KL penalty to the frozen reference model. The selected RLOO run was initialized from IPO and used learning rate 10^{-5}, batch size 128 prompts, 100 updates, entropy coefficient 0.001, KL coefficient 0.001, and temperature 1.0.

4.4 Number-Count Curriculum Extension

The primary curriculum uses number count as a difficulty proxy. The dataset contains three-number and four-number Countdown problems. Three-number examples are treated as easy because they require combining fewer operands and have a smaller expression search space. Four-number examples are treated as hard.

I evaluate two number-count curriculum schedules:

- **Base curriculum:** early batches sample from easy three-number examples, then switch to the full distribution after 50 steps.
- **Mixed curriculum:** early batches still emphasize easy examples but include a minimum hard-example fraction of 20%, avoiding a complete distribution shift at the curriculum boundary.

Both schedules use the same RLOO update, model initialization, optimizer, reward function, and evaluation. Thus, the ablation isolates the training-data schedule.

Table 1: Evaluation on 50 held-out Countdown prompts with 16 samples per prompt. Pass@16 difficulty curricula are included as diagnostic ablations.

Method	Avg.	pass@1	pass@2	pass@4	pass@8	pass@16
SFT	0.347	0.260	0.480	0.560	0.720	0.760
IPO	0.414	0.260	0.460	0.660	0.720	0.740
RLOO	0.617	**0.640**	0.700	0.720	0.760	0.760
Number-count base curriculum	0.635	**0.640**	**0.740**	**0.780**	0.780	**0.820**
Number-count mixed curriculum	**0.652**	0.620	0.680	0.760	**0.800**	**0.820**
Pass@16 base curriculum	0.641	**0.640**	0.700	0.760	0.780	0.780
Pass@16 mixed curriculum	0.604	0.560	0.640	0.660	0.740	0.740

4.5 Pass@16 Difficulty Curriculum

I also test a more empirical difficulty signal. I used the RLOO baseline to sample 16 responses for 20,000 training examples. If any of the 16 responses solved the prompt, I labeled the example easy; otherwise I labeled it hard. This creates a model-specific pass@16 difficulty dataset. I then reran both the base and mixed curriculum schedules using these labels instead of number count.

This ablation tests a plausible hypothesis: a curriculum should focus on examples the current model cannot solve, not merely examples with more numbers. However, it also introduces noise. Pass@16 labels depend on sampling temperature, the particular baseline checkpoint, and the 20,000-example subset. The results in Section 6 show that this empirical signal is useful but not clearly superior to the simpler structural proxy.

5 Experimental Setup

All training runs use the Countdown 3-to-4-number dataset and the same verifier. IPO is initialized from the SFT checkpoint. RLOO and curriculum RLOO are initialized from the IPO checkpoint. RLOO runs use group size 8, batch size 128 prompts, 100 update steps, learning rate 10^{-5}, entropy coefficient 0.001, KL coefficient 0.001, AdamW weight decay 10^{-4}, and importance-weight clipping to $[0.5, 2.0]$. Checkpoints are evaluated with 16 sampled responses per prompt.

The main reported evaluation set contains 50 held-out prompts. This is small, so results should be interpreted as project-scale evidence rather than a definitive benchmark. To reduce ambiguity, I report multiple metrics rather than only average reward.

6 Results

6.1 Main Quantitative Results

Table 1 summarizes the main evaluation. IPO improves average reward over SFT, but the largest gain comes from online RLOO. Number-count curriculum improves RLOO further, with mixed curriculum achieving the best average sampled reward.

The strongest single result is number-count mixed curriculum: it reaches average reward 0.652, compared with 0.617 for RLOO and 0.414 for IPO. Number-count base curriculum is also strong, tying the best pass@16 value at 0.820 and giving the best pass@2/pass@4 in this evaluation.

6.2 Score Decomposition

Table 2 decomposes sampled responses into exact-correct, formatted-but-wrong, and missing-answer categories. This reveals a tradeoff that average reward alone hides. Number-count mixed curriculum has the best exact-correct fraction, 0.636, but also a higher missing-answer rate than number-count base curriculum. Base curriculum is more formatting-reliable, with missing-answer rate 0.084.

The decomposition also shows that missing-answer rate and average reward capture different failure modes. Number-count base curriculum and pass@16 mixed curriculum have the lowest missing-answer rates, suggesting that they preserve answer-tag formatting more reliably. However, average

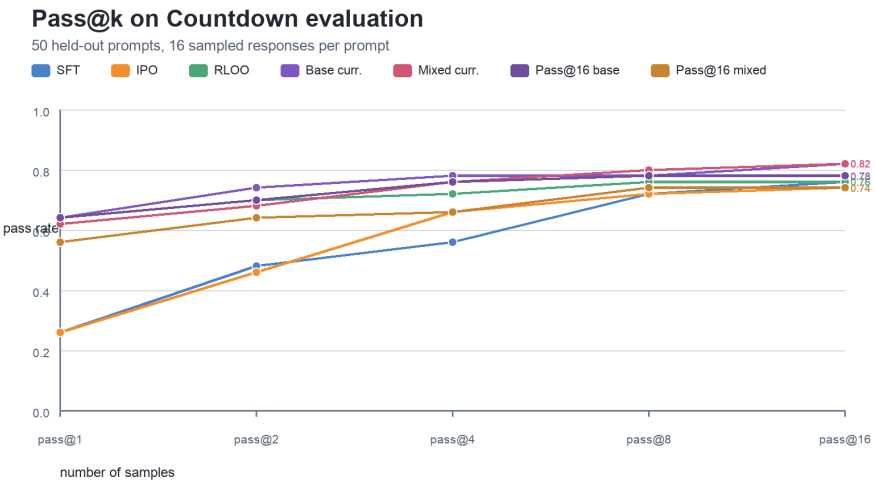

Figure 1: Pass@k comparison for the main SFT, IPO, RLOO, and number-count curriculum results.

Table 2: Response-level score decomposition across 800 sampled responses per model.

Method	Correct	Format wrong	Missing	Avg.
SFT	0.285	0.621	0.094	0.347
IPO	0.367	0.469	0.164	0.414
RLOO	0.595	0.223	0.182	0.617
Number-count base curriculum	0.604	0.312	0.084	0.635
Number-count mixed curriculum	**0.636**	0.160	0.204	**0.652**
Pass@16 base curriculum	0.626	**0.144**	0.230	0.641
Pass@16 mixed curriculum	0.569	0.350	0.081	0.604

reward is dominated by exact correctness because correct responses receive reward 1.0 while formatted but wrong responses receive only 0.1. Thus, number-count mixed curriculum and pass@16 base curriculum can score higher despite more missing answers: when they do produce parseable answers, they solve the arithmetic constraints more often. This suggests that curriculum changes both formatting reliability and symbolic correctness, rather than improving all error types uniformly.

6.3 Pass@16 Difficulty Ablation

The pass@16 curriculum results are useful experimentation even though they are not the final winning method. Pass@16 base curriculum reaches average reward 0.641, which is above RLOO's 0.617 and close to number-count base curriculum's 0.635. It also has a strong exact-correct fraction of 0.626. This suggests that empirical difficulty labels can produce a meaningful training signal.

However, pass@16 curricula do not dominate the simpler number-count curricula. Pass@16 base has lower pass@16 than number-count base, and pass@16 mixed falls below RLOO on average reward. One plausible explanation is that pass@16 labels create a brittle partition: examples solved at least once in 16 stochastic samples may still be unreliable, while examples never solved may be too hard for a short 100-step curriculum. In contrast, number count gives a smooth and task-structural proxy for search complexity. The result is a useful negative finding: model-specific empirical difficulty is promising, but needs calibration, larger precompute coverage, or a softer continuous difficulty score.

6.4 Qualitative Example

The following example illustrates the kind of behavior rewarded by the verifier. The response includes reasoning text, but the crucial component is the extracted final expression.

$$\text{Target} = 98, \quad \text{Numbers} = [44, 19, 35].$$

```
<answer>44 + 35 + 19</answer>
```

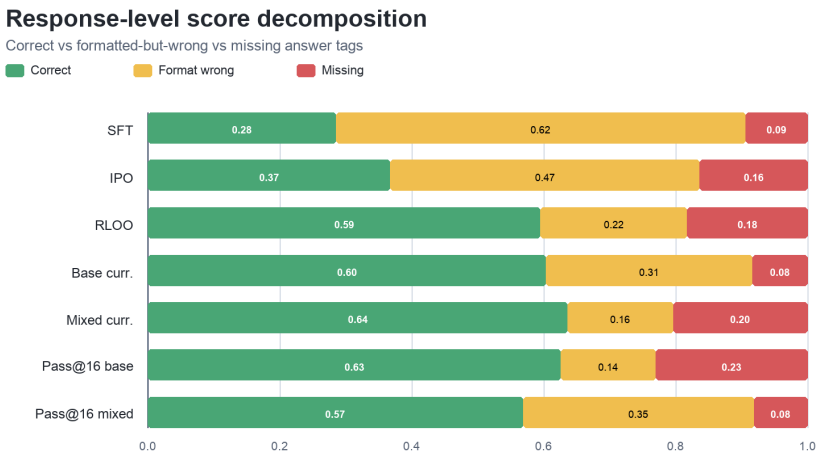

Figure 2: Score decomposition for the main SFT, IPO, RLOO, and number-count curriculum runs.

The verifier extracts the expression, checks that the numbers 44, 35, and 19 are each used once, and evaluates the expression to 98. This receives reward 1.0. By contrast, an answer that omits tags receives 0.0, and a tagged but arithmetically wrong answer receives 0.1.

7 Discussion

The results support three conclusions. First, online verifier-based RL is much more effective than offline preference optimization alone for this task. IPO improves average reward over SFT, but RLOO produces the largest increase in first-sample correctness. This makes sense because Countdown has a deterministic reward function aligned with the evaluation metric.

Second, implementation stability matters. The RLOO result only emerged after constraining the importance ratio. Without clipping, large ratios can produce high-variance updates and destabilize training. This is an important practical lesson for short-horizon class-project RL runs: a theoretically valid update can still be numerically brittle when sequence-level probabilities are used.

Third, curriculum changes the reliability/correctness tradeoff. Number-count mixed curriculum gives the best average reward and correctness fraction, while number-count base curriculum has better formatting reliability and ties the best pass@16. This suggests that curriculum is not simply a monotonic improvement; it shifts which kinds of errors the model makes. The pass@16 ablation reinforces this point. Empirical difficulty labels are attractive, but in this setup they did not beat the simpler number-count split.

8 Limitations and Future Work

The main limitation is evaluation scale. The reported test set has 50 prompts and 16 samples per prompt. This is enough to compare project variants, but a stronger final claim would require more held-out prompts and repeated seeds. Second, the curriculum is simple: number count is a coarse proxy, and pass@16 difficulty is binary. A better approach could use a continuous difficulty score, such as empirical solve rate over 16 samples, reward distribution entropy, or verifier margin. Third, missing-answer behavior remains a concern. Mixed curriculum increases correctness but can increase missing tags, suggesting that a format regularizer or SFT mixing term may help.

Future work should test larger evaluation sets, repeat curriculum runs across seeds, and explore hybrid curricula that combine number count with empirical solve rate. Another natural extension is adaptive curriculum: recompute difficulty during training instead of using a fixed precomputed pass@16 file.

9 Conclusion

This project implemented IPO, RLOO, and curriculum RLOO for Countdown arithmetic reasoning. IPO improved over SFT, RLOO improved substantially over IPO, and number-count curriculum improved RLOO further. The best result was mixed number-count curriculum with average reward 0.652 and pass@16 0.820. Pass@16 empirical difficulty labels provided a useful ablation but did not beat the simpler structural curriculum. The main takeaway is that verifier-based online RL can strongly improve exact arithmetic correctness, but curriculum design and update stabilization are central to making the improvement reliable.

10 Team Contributions

This was a solo project.

Changes from Proposal. The final project follows the original plan of studying preference optimization and curriculum learning for three- and four-number Countdown tasks. The main addition beyond the proposal is the pass@16 empirical-difficulty ablation, which compares a model-specific curriculum signal against the simpler number-count curriculum.

References

Mohammad Gheshlaghi Azar, Zhaohan Daniel Guo, Bilal Piot, Remi Munos, and Mohammad G. Azar. 2024. A General Theoretical Paradigm to Understand Learning from Human Preferences. In *Proceedings of The 27th International Conference on Artificial Intelligence and Statistics*.

Yoshua Bengio, Jerome Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum Learning. In *Proceedings of the 26th International Conference on Machine Learning*.

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]

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2023. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290 [cs.LG]