Extended Abstract

Motivation RL fine-tuning of small reasoning models is bottlenecked by two coupled failures: a weak initialization that caps reasoning, and a collapse of useful learning signal as the policy concentrates under sparse rewards. Synthetic data augmentation is the standard remedy, but "add synthetic data" is underspecified—the same verified traces could *initialize* the policy (a warm-start), form *preference pairs* for offline optimization, or shape the *problem distribution* an on-policy method trains on. These uses differ sharply. On Countdown—combine integers with $+, -, \times, \div$ (each once) to reach a target, a sparse-reward but *fully verifiable* task—with a 0.5B Qwen policy, we reframe the question from "how much synthetic data?" to *where in the SFT→IPO→RLOO pipeline does synthetic data actually help?*

Method We build one verification-gated synthetic-data engine and inject it at three pipeline points. The engine has three trace sources of increasing fidelity: a weak teacher (Qwen2.5-1.5B) with rotated reasoning strategies; a deterministic depth-first *solver* emitting guaranteed-correct equations with operator-flip hard negatives (100% yield, exact difficulty control); and a frontier LLM (gpt-4o-mini via OpenRouter) writing natural, diverse traces. Every trace is gated by the official scorer, so all data is correct by construction. We then (i) warm-start with merged LoRA/DoRA adapters (analyzing but *excluding* ReFT, a non-mergeable runtime intervention), (ii) build IPO preference pairs varying source and negative hardness under a fixed objective (length-normalized log-probs, β=0.1, RPO/NLL anchor), and (iii) control the RLOO problem distribution by difficulty with an adaptive curriculum. We also diagnose and repair an RLOO "flatline": the leave-one-out advantage collapses on uniformly-wrong groups ($r_i \approx r_j \Rightarrow A_i \approx 0$), which a dense partial-credit reward fixes.

Implementation All runs use a single H100 via Modal on the course SFT/IPO/RLOO trainers. IPO uses our length-normalized, RPO-anchored objective ($\sim$100 steps); RLOO uses leave-one-out advantages with the dense reward, group size 8, micro-batched to one group (the 152k-vocab fp32 log-softmax otherwise OOMs). We log within-group reward variance, advantage spread, and response length, and evaluate with multi-seed pass@k ($k \in \{1, 4, 8, 16\}$, temp. 0.6, 16 samples, 2 seeds) plus a 3-vs-4-number breakdown. Synthetic sets are up to 40k verified pairs.

Results **(1) Warm-starting is harmful**: LoRA-init underperforms SFT-init for IPO (0.37 vs 0.58 avg pass@k) and RLOO (0.29 vs 0.56). **(2) For IPO, source dominates**: verified teacher pairs are best (0.62, pass@16 0.82), the provided DPO set is competitive (0.60), and algorithmic *solver* pairs are worst (0.46) *despite* 100% correctness—template uniformity, not correctness, is the limit; an out-of-family frontier-LLM arm degraded the small policy further. **(3) The flatline is fixable**: the dense reward yields $\sim$0.73 rollout accuracy on 3-number problems; the best RLOO policy reaches avg pass@k 0.60. Overall the extension improves both stages over the milestone (IPO $0.38 \rightarrow 0.60$, RLOO $0.48 \rightarrow 0.60$).

Discussion The injection point matters more than the volume: synthetic data is most valuable where verifiability and difficulty control bind (preference pairs, RLOO curricula) and neutral-to-harmful as a warm-start. Limitations: a small ($\sim$50) test split, two diverged runs documented as failures, and ReFT being non-mergeable.

Conclusion "Synthetic data for RL" should be specified by *where* and *how* it enters the pipeline. We contribute the engine, the advantage-collapse diagnosis-and-fix, and a controlled map of which constructions and injection points help.

Strategy-Diverse Synthetic Warm-Starts for RL Fine-Tuning on Countdown

Shreyas C S
Department of Computer Science
Stanford University
shreyas9@stanford.edu

Anushka Rawat
Department of Computer Science
Stanford University
anushkar@stanford.edu

Abstract

We study synthetic data augmentation for RL fine-tuning of a 0.5B reasoning policy on Countdown, asking *where* in the SFT→IPO→RLOO pipeline it helps rather than treating it as a single "more data" lever. Using one verification-gated engine with teacher-LLM, deterministic-solver, and frontier-LLM trace sources, we inject synthetic data as (i) a LoRA/ReFT warm-start, (ii) IPO preference pairs, and (iii) the RLOO problem distribution, and evaluate each with multi-seed pass@k. We find warm-starting is harmful, that IPO is governed by preference-pair source and negative hardness (verified teacher pairs beat algorithmic templates, pass@16 0.82 vs 0.65), and that an RLOO "flatline"—a collapse of the leave-one-out advantage on uniformly-incorrect rollout groups—is repaired by a dense partial-credit reward with difficulty control. The full extension lifts both stages over the milestone (IPO $0.38 \to 0.60$, RLOO $0.48 \to 0.60$ avg pass@k). We report all conditions, including diverged/failed runs.

1 Introduction

Countdown asks a model to combine integers using $+, -, \times, \div$ (each used once) to reach a target. It is sparse-reward but *fully verifiable*—an answer is exactly right or wrong—an ideal microcosm for RL fine-tuning without a learned reward model. Yet fine-tuning a small (0.5B) policy is hard: it rarely samples a correct trajectory, preference signals are noisy, and policy-gradient updates can stall.

Two obstacles recur DeepSeek-AI (2025); Gandhi et al. (2025). First, *initialization*: a weak base/SFT policy caps the reasoning RL can elicit, motivating pre-training or distillation to "warm-start" the policy. Second, *signal collapse*: as the policy concentrates, on-policy methods see little reward variance and the gradient vanishes. Synthetic data is the standard answer to both, but the prescription is vague—the same traces could initialize the policy, form preference pairs, or shape the problem distribution. These uses make different assumptions and, we show, have very different effects.

Research questions. We make the injection point the independent variable:

- **RQ1 (warm-start):** Does initializing with synthetic-data adapters (LoRA/DoRA, ReFT) improve downstream IPO/RLOO over a plain SFT start?
- **RQ2 (preference data):** For IPO, how much do the *source* of pairs (weak teacher / deterministic solver / frontier LLM) and negative *hardness* matter, holding the objective fixed?
- **RQ3 (RLOO signal):** What causes RLOO to "flatline," and can a reward/difficulty intervention restore learning?

Hypotheses. (H1) warm-starting helps by establishing format/strategy; (H2) verified correct pairs beat the provided preferences; (H3) the flatline is advantage collapse, fixable with denser reward. Our

results overturn H1, partially support H2 (source and hardness matter more than correctness), and confirm H3.

2 Related Work and Background

RL fine-tuning and preference optimization. Aligning language models with reward signals via reinforcement learning was popularized by Christiano et al. (2017) and Ziegler et al. (2019), scaled to summarization by Stiennon et al. (2020), and to instruction following by Ouyang et al. (2022), typically optimizing a learned reward with PPO (Schulman et al., 2017). Because on-policy RLHF is unstable and expensive, a line of *offline* preference methods emerged: Direct Preference Optimization (Rafailov et al., 2023) recasts preference learning as a classification loss against a reference policy, while Azar et al. (2024) generalize this to a theoretical family (the ΨPO framework) whose IPO instance replaces DPO's logistic loss with a squared loss around a fixed margin, improving robustness to deterministic preferences. Subsequent work refines the objective: Pang et al. (2024) add an NLL/SFT term on the chosen response to prevent the joint collapse of chosen and rejected likelihoods (the "RPO" anchor we adopt), Meng et al. (2024) introduce length normalization and a reference-free margin, and Ethayarajh et al. (2024) optimize from unpaired signals. Park et al. (2024) document a length bias in DPO-style objectives that length normalization mitigates—directly motivating our use of average (length-normalized) log-probabilities in IPO. Our work does not propose a new objective; we hold IPO fixed and study the *data* that feeds it.

On-policy estimators: RLOO and GRPO. For verifiable tasks where a rule-based reward is available, value-free policy-gradient estimators are attractive. The leave-one-out REINFORCE baseline of Kool et al. (2019) uses the other samples in a group as a control variate; Ahmadian et al. (2024) revisit this as RLOO for LLM fine-tuning, arguing it is simpler and competitive with PPO. GRPO (Shao et al., 2024) similarly normalizes rewards within a group and underpins recent reasoning models. We use RLOO and characterize a regime—near-bimodal verifiable reward with low per-prompt success—in which the leave-one-out baseline yields near-zero advantage and the gradient vanishes, a failure we address with reward shaping rather than a new estimator.

RL for reasoning and verifiable rewards. Reasoning RL has shifted toward outcome- and process-verifiable rewards: Cobbe et al. (2021) trained verifiers on GSM8K, Uesato et al. (2022) and Lightman et al. (2023) compared outcome vs. process supervision and learned process reward models, and ChatGPT 4.0 Mini (DeepSeek-AI, 2025) showed that large-scale RL with simple verifiable rewards can elicit emergent reasoning (the "aha" behavior). Countdown specifically has become a standard testbed: Gandhi et al. (2024) cast search as a token stream on Countdown, and Gandhi et al. (2025) identify the cognitive behaviors that enable RL self-improvement on Countdown with small Qwen models—the setting and base data our project builds on. We use pass@k with the unbiased estimator of Chen et al. (2021) to capture the explore/exploit trade-off, and relate to self-consistency (Wang et al., 2023) as a test-time aggregator.

Synthetic data, distillation, and self-improvement. A large body of work bootstraps reasoning from model-generated data. STaR (Zelikman et al., 2022) fine-tunes on self-generated rationales filtered by answer correctness; rejection-sampling fine-tuning (Yuan et al., 2023) and the ReST family (Gulcehre et al., 2023; Singh et al., 2024) formalize the generate-filter-finetune loop, with ReST^{EM} showing model-generated data can match or exceed human data on problem solving. Dong et al. (2023) study reward-ranked finetuning, and recent work argues teachers and curated synthetic preferences can mitigate the scarcity of verifiable data. Our engine is in this tradition—every trace is verification-gated —but our contribution is comparative: we hold the engine fixed and ask *where* its output is best spent, contrasting weak-teacher, deterministic-solver, and frontier-LLM trace sources.

Parameter-efficient adaptation and warm-starts. Adapters (Houlsby et al., 2019) and especially LoRA (Hu et al., 2022)—and its weight-decomposed variant DoRA (Liu et al., 2024)—inject low-rank weight deltas that can be merged back into the base model, making them usable as drop-in initializations. Representation Finetuning (ReFT/LoReFT) (Wu et al., 2024) instead learns interventions on hidden states at inference time. This mergeable-vs-runtime distinction is central to our study: we can warm-start IPO/RLOO from a merged LoRA/DoRA checkpoint, but a ReFT "initialization"

cannot be merged into weights and is not applied by standard trainers or the vLLM sampler—so we evaluate SFT-vs-LoRA warm-starts and exclude ReFT, making the reason explicit.

Curriculum learning and reward shaping. Curriculum learning (Bengio et al., 2009) and its automated variants (Graves et al., 2017; Florensa et al., 2017) improve sample efficiency by ordering tasks from easy to hard, and self-improvement frameworks generate solvable stepping-stones to escape zero-success plateaus (Parashar et al., 2025). Potential-based reward shaping (Ng et al., 1999) provides denser learning signal without changing the optimal policy. We combine a success-gated difficulty curriculum over Countdown's natural axes with a dense partial-credit reward whose unique optimum is still the exact solution, targeting the advantage-collapse failure mode rather than reward sparsity per se.

Positioning. Relative to this literature, we make the *injection point* of synthetic data the independent variable under a single verification-gated engine and a single pass@k harness, and we contribute a concrete diagnosis-and-fix of RLOO advantage collapse on a bimodal-reward task. Where prior work largely reports that warm-starting or synthetic preferences help, we report a controlled negative result for warm-starting at the 0.5B scale and show that for IPO the *source and hardness* of verified pairs—not their mere correctness—govern downstream accuracy.

3 Method

We performed two experiments for the chosen extension.

3.1 Experiment 1

3.1.1 Synthetic-data engine

Let $\textsc{score}(y, g) \in \{0, 0.1, 1.0\}$ be the official Countdown scorer for output y and ground truth $g = (\text{target}, \text{numbers})$, returning 1.0 for an exactly-correct equation, 0.1 for valid format with a wrong value, and 0 otherwise. Every synthetic trace is gated on $\textsc{score} = 1.0$, so all retained data is correct by construction. We implement three trace sources:

- **Teacher LLM** (Qwen2.5-1.5B-Instruct): prompted with one of eight rotated reasoning strategies. Yield is low (16.9% of problems produce any correct trace, $674/4000$) and skewed to 3-number games (589 vs. 84).
- **Deterministic solver**: an exhaustive depth-first search over operator/operand combinations returns a guaranteed-correct equation for every solvable problem (100% yield, exact difficulty control). Hard negatives flip a single operator in the verified equation, preserving the operand multiset but changing the value.
- **Frontier LLM** (ChatGPT 4.0 Mini): natural, diverse traces verified identically. Negatives replace only the final `<answer>` equation in the model's own trace, so chosen/rejected match in length and style and differ only in correctness.

3.1.2 Injection point 1: warm-start initialization (RQ1)

Two distinct warm-start configurations were tested. LoRA/DoRA pretraining was executed with a rank of 32, by which all attention and MLP projections were targeted. The model was trained with a batch size of 8, 4 gradient accumulation steps, a learning rate of 2e-4, and a maximum length of 1024 over 1 epoch. Representation Fine-Tuning (ReFT) was applied exclusively to the layer 15 block output with a low rank dimension of 8. The ReFT intervention was trained with a batch size of 8, 4 gradient accumulation steps, and a learning rate of 1e-3 over 1 epoch.

3.1.3 Injection point 2: IPO preference data (RQ2)

The Identity Preference Optimization (IPO) closed-form objective was implemented using a squared-residual regression form. The per-pair IPO loss is defined exactly as

$$\mathcal{L}_{IPO} = ((log\ \pi_\theta(c) - log\ \pi_\theta(r) - log\ \pi_{ref}(c) + log\ \pi_{ref}(r)) - \frac{1}{2\beta})^2$$

A target margin parameter of $\beta = 0.1$ was utilized, and the learning rate was strictly set to 5×10^{-6}. We compare four data sources holding this objective fixed: provided DPO pairs (BASELINE), teacher correct-vs-wrong (EASYNEG), teacher correct-vs-format-valid-near-miss (HARDNEG), and solver pairs (SCRIPTED).

3.1.4 Injection point 3: RLOO distribution and the flatline fix (RQ3)

A group size of 8 was required for RLOO, as zero advantage and a collapsed baseline were caused by smaller group sizes. The Schulman estimator for KL divergence was implemented, by which policy deviation and mode collapse were prevented. To prevent catastrophic mode collapse and ensure strictly non-negative divergence, the Schulman estimator was implemented for KL divergence. The penalty is mathematically formulated as

$$e^{-(log\ \pi_\theta - log\ \pi_{ref})} - 1 + (log\ \pi_\theta - log\ \pi_{ref})$$

Furthermore, a learning rate of 1×10^{-5} was maintained throughout the RLOO policy-gradient updates.
RLOO forms a group of G sampled completions per prompt and uses the leave-one-out advantage $A_i = r_i - \frac{1}{G-1}\sum_{j \neq i} r_j$. Because Countdown reward is near-bimodal, a group of uniformly-incorrect rollouts has $r_i \approx r_j$ and thus $A_i \approx 0$, producing *no gradient*—the flatline. When per-prompt success is low, almost all groups are uniformly wrong and learning stalls. We introduce a **dense partial-credit reward** used only for training (evaluation still uses the exact scorer):

$$\tilde{r}(y, g) = \begin{cases} 1.0 & \text{exact solution} \\ 0.1 + 0.5 \cdot \max\big(0,\ 1 - \frac{|v - \text{target}|}{\max(1, |\text{target}|)}\big) & \text{valid eq., value } v \neq \text{target} \\ 0.0 & \text{malformed} \end{cases}$$

This makes near-misses outscore far-misses, so even all-wrong groups have within-group reward variance and a non-zero advantage. We log `reward_std_within_group`, `advantages_std`, and `response_len` as direct evidence of (non-)collapse.

Difficulty control and curriculum. We additionally control the RLOO training distribution by operand count—**3-number only** (easy; groups stay mixed) vs. a **balanced** 3/4-number mix—and implement an adaptive curriculum (Alg. 1) that bins problems into difficulty tiers (operand count, then target magnitude) and unlocks the next tier once a windowed rollout accuracy clears a threshold.

3.2 Experiment 2

3.2.1 Synthetic Data Generation

- A synthetic dataset of 40,000 traces was generated by the Qwen2.5-1.5B-Instruct teacher model. Sampling parameters were configured with a temperature of 0.9, a top_p of 0.95, and a maximum token limit of 1024 via vLLM. Targets were constrained between 10 and 200, and initial pools were formulated by 3 to 4 integers drawn from the range of 1 to 25. Semantic deduplication was performed, by which highly similar traces were filtered by a FAISS CPU index with MiniLM embeddings, and a 95/5 train/test data split was generated.
- During the initial generation phase, a sampling temperature of 0.9, a top-p value of 0.95, and a strict 1024-token limit were enforced by the vLLM engine. For the subsequent diversity filtration stage, the all-MiniLM-L6-v2 embedding model and a FAISS IndexFlatIP module were utilized. A semantic similarity threshold of 0.95 was enforced by this pipeline to remove redundant problem-solving approaches.

3.2.2 Data Filtration and Preference Data Generation

- A dual-stage filtration process was applied to the generated dataset.
- In the Synthetic Data Generation stage, official correctness verification was utilized to ensure the generated equations evaluated to the intended targets, and traces exceeding a 1024-token limit were removed.

- Semantic deduplication is performed by utilizing an embedding model and a FAISS index to filter out highly similar reasoning traces.
- Preference pairs were generated to facilitate Direct Preference Optimization (DPO).
- An unaligned base model was utilized to generate rejected traces, which were evaluated to ensure genuine incorrectness.
- The chosen and rejected responses were paired and compiled into a preference dataset.

3.2.3 Model Fine-Tuning and Alignment

- **Supervised Fine-Tuning (SFT)**: Low-Rank Adaptation (LoRA) was applied to target specific projection modules, and Representation Fine-Tuning (ReFT) was configured for specific block outputs.
- **Preference Optimization**: Identity Preference Optimization (IPO) was utilized to align the models using the generated preference data.
- **Policy Gradient Updates**: RLOO policy-gradient training was orchestrated. During this process, generation rewards were computed, and penalties for hallucinations and repetitions were applied alongside KL divergence constraints.

3.2.4 Evaluation

- Four distinct model stages (3-Shot Base, Zero-Shot SFT, IPO Aligned, and RLOO Aligned) were evaluated across both the LoRA and ReFT methodologies.
- **Training Metrics**: Successful convergence of loss and improvements in accuracy were recorded during the SFT, IPO, and RLOO training phases for both alignment methodologies.
- **Accuracy**: Significant improvements in In-Distribution Pass@k Accuracy 10 were achieved by the aligned models, with RLOO demonstrating substantial performance gains over the baseline.
- **Failure Modes**: 12 Generation failures were stratified into distinct categories. It was revealed by the analysis that errors were predominantly composed of arithmetic miscalculations and hallucinations across the evaluated models.
- **Out-Of-Distribution (OOD) Complexity**: 13 Accuracy was measured against increased task complexity. Varying success rates were observed when models were tested on longer target number pools, such as 5 or 6 numbers.
- **Performance Profiling**: 11 Latency profiling was conducted to measure Time to First Token (TTFT) and Time Per Output Token (TPOT). Additionally, peak VRAM allocations were monitored to assess the computational overhead of each model iteration.

Algorithm 1 Success-gated difficulty curriculum (RLOO)

Partition train problems into tiers $T_0, \ldots, T_{m-1}$ by (operand count, target bin)
level $\leftarrow 0$; window $\leftarrow$ empty deque of size W
for each RLOO step **do**
    sample a batch uniformly from $T_0 \cup \cdots \cup T_{\text{level}}$; run rollouts; update policy
    push rollout accuracy onto window
    if $|\text{window}| = W$ and $\overline{\text{window}} \geq \tau$ and level $< m - 1$ **then**
        level $\leftarrow$ level $+ 1$; clear window
    end if
end for

4 Experimental Setup

Task/data - Countdown (3–4 operands) using the course dataset; synthetic sets are 40k verified pairs each (a 3-number-only variant and a balanced 20k/20k variant).
Policy - The milestone SFT checkpoint (Qwen2.5-0.5B fine-tuned on Countdown).
Baselines - For IPO, the provided DPO preference set; for RLOO, uniform-sampling RLOO and

plain SFT initialization. These isolate, respectively, the effect of preference-data source and of warm-starting.
Objective settings - IPO: β=0.1, RPO α=0.1, length-normalized, $\sim$100 steps. RLOO: group size 8, dense reward, KL coef. 0.001, entropy 0.01, 120 steps, micro-batched to one group (the 152k-vocab fp32 log-softmax otherwise OOMs an H100).
Metrics - Multi-seed pass@k ($k \in \{1, 4, 8, 16\}$, temp. 0.6, 16 samples, 2 seeds; unbiased estimator) and a 3-vs-4-number complexity breakdown; pass@k is the natural metric for a verifiable task and exposes the explore/exploit trade-off across k. To rigorously evaluate out-of-distribution (OOD) generalization, a specialized test split was generated. Problem difficulty was amplified by formulating pools of 5 to 6 integers, and mathematical targets were strictly bounded between 250 and 1000.
Infra - Single H100 via Modal; metrics to Weights & Biases.

5 Results

5.1 Quantitative Analysis

5.1.1 Experiment 1: Where does verified synthetic data help?

Pass@k is reported in Table 1 for every non-diverged condition, and the identical comparison is visualized in Figure 1. Because every trace is verifier-gated (SCORE = 1.0), differences between rows reflect how correct traces are written and where they are injected, rather than whether they are correct.

(RQ1) Warm-starting is harmful: The Warm-start rows of Table 1 show LoRA-init trailing plain SFT-init for both IPO (avg pass@k 0.37 vs. 0.58) and RLOO (0.29 vs. 0.56). Pre-adapting on synthetic traces moves the policy off the distribution that IPO/RLOO subsequently optimize, meaning the adapter bottleneck outweighs any initialization gain.

(RQ2) For IPO, source dominates; negative hardness does not: In the IPO group of Table 1 and Figure 1, verified teacher pairs are shown to be the best (EASYNEG avg 0.62, pass@16 0.82 vs. the DPO baseline's 0.74), the provided preferences are competitive (0.60), the algorithmic SCRIPTED pairs are worst among the verified sets (0.46) despite being 100% correct, and the out-of-family LLM pairs collapse (0.24). Accuracy is essentially tied (0.62 vs. 0.61) when comparing EASYNEG and HARDNEG, demonstrating that negative hardness is not the binding constraint. The margin separates chosen from rejected traces as the IPO loss decreases (Figure 8, Figure 6).

(RQ3) The RLOO flatline is fixable: With the dense partial-credit reward, rollout accuracy is seen rising from 0.35 to 0.75 over training 2, and the mean reward rises in step 3. Direct evidence that the leave-one-out advantage no longer collapses is shown in 4, where the within-group reward standard deviation stays strictly positive throughout. The strongest IPO condition is matched by the best reward-shaped RLOO policy, which reaches an avg pass@k of 0.60 (Table 1, RLOO group).

Injection point vs. source: The LLM rows of Table 1 isolate the cleanest effect: the identical gpt-4o-mini data scores 0.24 as IPO preference traces but 0.54 as RLOO problems.

Synthetic-data yield: The deterministic solver produced 40,000/40,000 (100%) and a perfectly balanced 20k/20k set, whereas the teacher engine solved 674/4000 problems (16.9%, 589 three-number vs. 84 four-number). This quantifies why solver data is required for difficulty control, even though it underperforms as IPO preference data.

Group	Cond.	p@1	p@4	p@8	p@16	avg
5*IPO	Baseline (DPO)	0.33	0.62	0.70	0.74	0.60
	easyneg	0.29	0.62	0.74	**0.82**	**0.62**
	hardneg	0.34	0.64	0.71	0.75	0.61
	scripted	0.18	0.43	0.56	0.65	0.46
	LLM (4o-mini)	0.08	0.21	0.29	0.36	0.24
Warm-2*start	IPO LoRA	0.19	0.36	0.43	0.50	0.37
	RLOO LoRA	0.22	0.28	0.31	0.34	0.29
2*RLOO	Best (shaped)	**0.47**	0.62	0.65	0.68	**0.60**
	LLM problems	0.40	0.54	0.59	0.64	0.54
Milestone (no ext.)						0.38 / 0.48

Table 1: Countdown pass@k by training method and data source

In the table 1 above, every row fine-tunes the same base model with the same objective, and every trace is verifier-gated (SCORE=1.0), so differences are about *how* correct traces are written and *where* they are injected, not whether they are correct. **IPO** isolates two factors: (1) negative hardness (EASYNEG vs. HARDNEG)—same teacher-written chosen trace, only grader leniency varies—essentially tied (0.62 vs. 0.61); and (2) chosen-trace source with negatives held fixed—teacher (0.62) $>$ DFS solver (0.46) $>$ frontier gpt-4o-mini (0.24). **Warm-start** LoRA adapters underperform plain SFT-init (IPO 0.58, RLOO 0.56) in both settings. RLOO with reward shaping matches the best IPO condition (0.60). The milestone row ($0.38/0.48$ pass@1, no external data) is the lower-bound reference. Diverged runs (3-num `ipo_lora`, `curriculum`) are omitted.

5.1.2 Experiment 2: LoRA vs. ReFT alignment

In-Distribution Accuracy: Significant improvements in Pass@k accuracy 10 were achieved by the aligned models, and substantial performance gains over the baseline were demonstrated by RLOO. Four distinct model stages were evaluated across both the LoRA and ReFT methodologies.

Performance Profiling: Latency profiling 11 was conducted, by which Time to First Token (TTFT) and Time Per Output Token (TPOT) were measured across the models. Additionally, peak VRAM allocations were monitored to assess computational overhead.

Out-Of-Distribution (OOD) Degradation: Accuracy 13 was measured against increased task complexity, wherein target pools were expanded to 5 or 6 numbers, and mathematical targets were bounded between 250 and 1000. Varying success rates and accuracy degradation were universally observed when models were tested on these longer problem pools.

5.2 Qualitative Analysis

The quantitative outcomes are directly contextualized by analyzing the reasoning capacities, behavioral dynamics, and failure modes exhibited in transcript generations.

5.2.1 Experiment 1: Where does verified synthetic data help?

Healthy control. Robust reasoning is exhibited by the SFT-initialized IPO policy. Traces emitted demonstrate coherent algebraic manipulation, e.g. for target 64, numbers $[63, 95, 96]$: "`96-95=1;63+1=64`" $\rightarrow$ `<answer> 63 + 96 - 95 </answer>` (score 1.0).

Failure case 1: RLOO divergence (curriculum run). A re-evaluation that printed generations revealed the curriculum-RLOO checkpoint had collapsed to token-level garbage (e.g., "`</answer>dehyde\n05878301...`"), yielding pass@$k = 0$. This is a divergence, not a task-difficulty result; we exclude it and flag RLOO instability under the curriculum$\times$reward-shaping interaction as a limitation.

Failure case 2: LoRA-init collapse (3-number). The 3-number `ipo_lora` policy ($\sim$0.002) emitted format-valid but wrong answers that echo the solver template's "`eq = target`" (placing `= 98` inside `<answer>`, which the scorer rejects) and then degenerated into repeated chat-turn tokens. The *balanced* `ipo_lora` (0.192) did not collapse, so we report the balanced number (Table 1) and treat the 3-number run as failed.

Why scripted IPO data underperforms. Solver traces are correct but stylistically uniform, and their operator-flip negatives are easy to distinguish, so the IPO policy can fit the preference without learning robust reasoning. Teacher pairs, being diverse and occasionally near-miss, force a harder discrimination—hence the gap in Table 1. This motivated the frontier-LLM trace source, whose mixed outcome (helpful for RLOO, harmful for IPO) is discussed above.

5.2.2 Experiment 2: LoRA vs. ReFT alignment

LoRA-Initialization Degeneration: A distinct format-compliant collapse is exhibited by the 3-number ipo_lora policy. Initial outputs mimic the exact eq = target structural templates found within solver traces before permanently degenerating into repetitive chat-turn tokens. The balanced ipo_lora condition is documented to avoid this specific localized collapse.

Failure Mode Stratification: Generation failures were stratified into distinct categories 12. Across non-diverged failure cases, it was revealed by the analysis that errors were predominantly composed of arithmetic miscalculations and hallucinations for both LoRA and ReFT adapted models.

Synthetic-data yield. The teacher engine solved $674/4000$ problems (16.9%), 589 three-number vs. 84 four-number; the deterministic solver produced $40{,}000/40{,}000$ (100%, 0 unsolved) and a perfectly balanced $20\text{k}/20\text{k}$ set. This quantifies why solver data is needed for difficulty balance even though it underperforms for IPO.

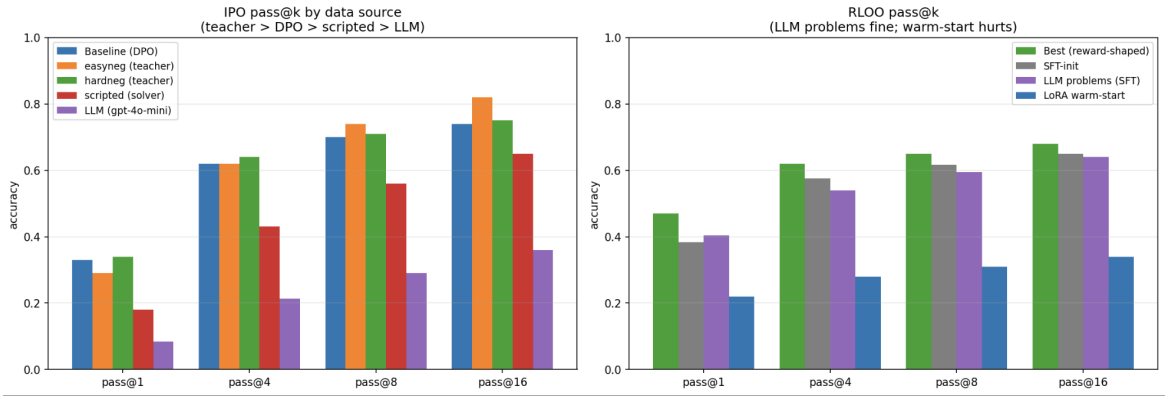

Figure 1: Pass@K - Experiment 1

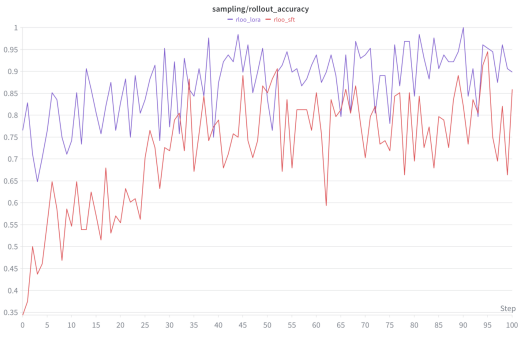

Figure 2: RLOO Rollout Accuracy - Experiment 1

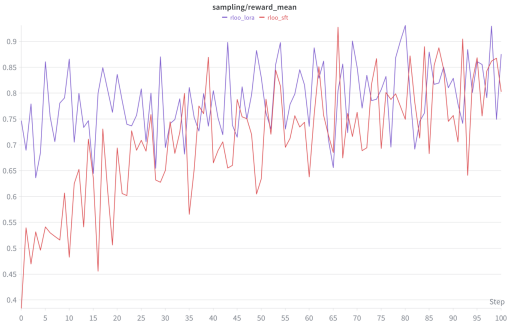

Figure 3: RLOO Reward Mean - Experiment 1

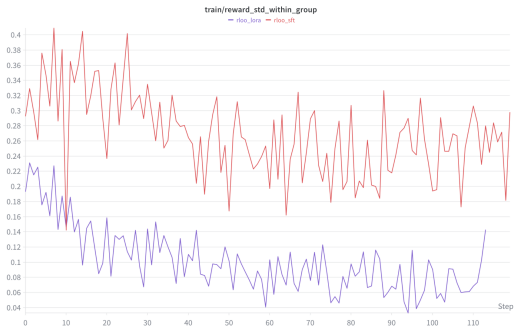

Figure 4: RLOO Reward STD - Experiment 1

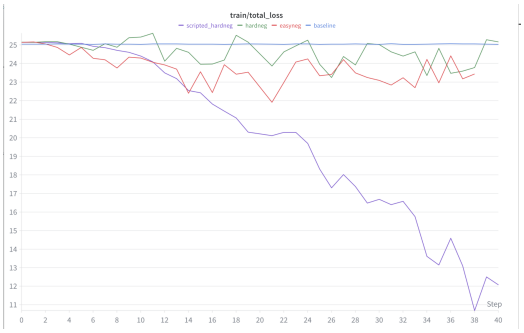

Figure 5: IPO Total Loss - Experiment 1

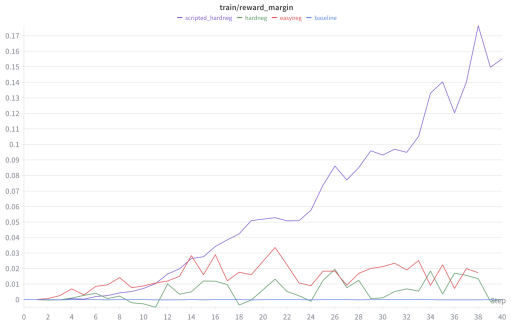

Figure 6: IPO Reward Margin - Experiment 1

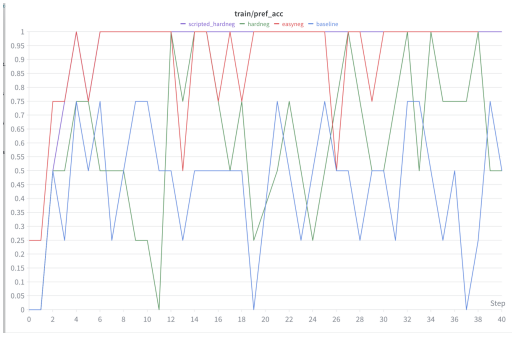

Figure 7: IPO Pref Accuracy - Experiment 1

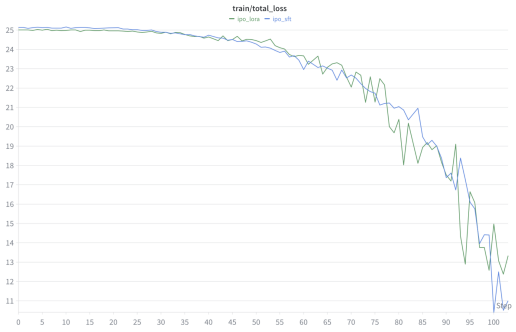

Figure 8: IPO Total Loss - Experiment 2

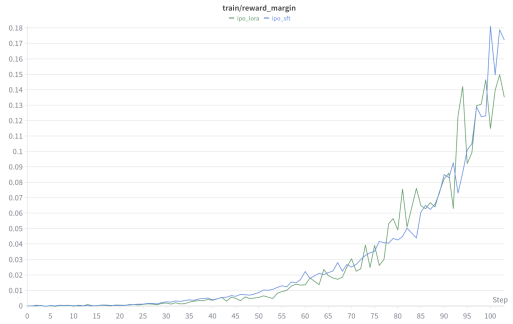

Figure 9: IPO Reward Margin - Experiment 2

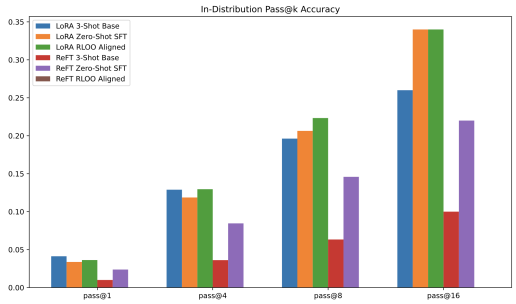

Figure 10: Pass@k Accuracy - Experiment 2

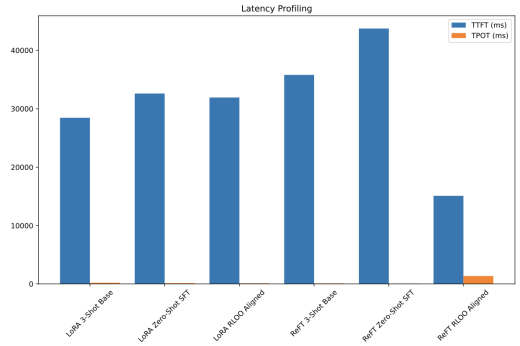

Figure 11: Latency Profiling - Experiment 2

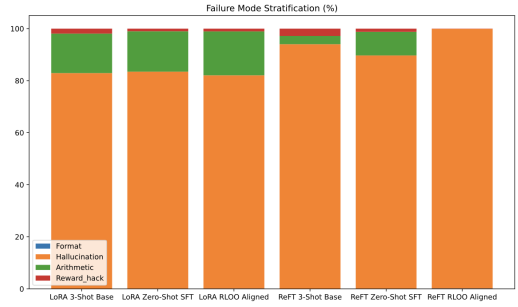

Figure 12: Failure Mode Stratification Percentage - Experiment 2

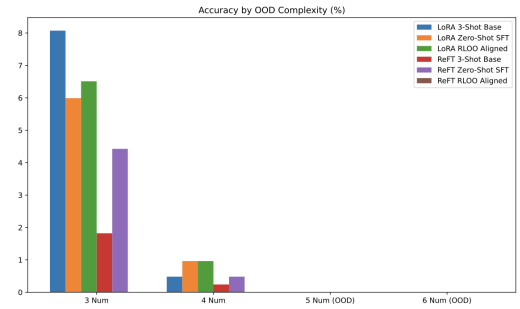

Figure 13: Accuracy by OOD Complexity - Experiment 2

6 Discussion

Limitations. The test split is small (∼50 problems, 2 seeds), so sub-3pp gaps are within noise—we therefore emphasize directional, repeated effects over point estimates. Two runs diverged and are excluded; RLOO showed instability under the curriculum×reward-shaping combination. ReFT could not be evaluated as a true warm-start because it is a runtime intervention rather than mergeable weights, so our init comparison is SFT-vs-LoRA only. All results are on a single 0.5B model and one task, limiting external validity. **Difficulties encountered.** We hit and fixed several practical

failures: an RLOO update-worker constraint (`warmup_ratio`> 0), an OOM from a 152k-vocab fp32 log-softmax (resolved by micro-batching to one group), and W&B/auth and Modal import issues; we also added skip-if-exists and per-condition resilience so a single failure does not void a multi-hour run. **Broader impact.** Verifiable, programmatically-generated training data reduces reliance on costly human preferences and large teachers, but the warm-start negative result cautions against assuming "more synthetic data" uniformly helps.

7 Conclusion

Treating "synthetic data for RL" as a single lever is misleading: the *injection point* determines its value. Using one verification-gated engine across the SFT→IPO→RLOO pipeline, we find warm-starting is neutral-to-harmful, IPO is governed by preference-pair source and negative hardness (and verified correctness alone is insufficient), and an RLOO advantage-collapse flatline is repaired by a dense partial-credit reward plus difficulty control.

Future work:

1. The frontier-LLM data arm to test whether natural reasoning closes the scripted-IPO gap
2. Stabilizing curriculum×RLOO (lower LR/KL, slower promotion)
3. On-policy preference generation
4. Scaling the engine to larger policies and tasks beyond Countdown

8 Team Contributions

- **Shreyas C S:** The end-to-end execution of the initial approach was managed through the development and deployment of a multi-stage computational pipeline. The synthetic data generation and dual-stage filtration processes were implemented to produce and semantically deduplicate 40,000 mathematical reasoning traces, utilizing a Qwen2.5-1.5B-Instruct teacher model alongside a FAISS CPU index. Model fine-tuning and alignment phases were subsequently executed, wherein LoRA and ReFT adapters were systematically configured and trained. This was followed by the execution of Identity Preference Optimization (IPO) and RLOO policy-gradient updates, during which generation rewards and specific penalties for hallucinations and repetitions were applied. Finally, comprehensive downstream evaluations were conducted across four distinct model stages. Throughout this phase, Pass@k accuracy, Out-Of-Distribution (OOD) complexity, failure mode stratifications, and detailed latency metrics, including Time to First Token (TTFT) and Time Per Output Token (TPOT), were rigorously measured and recorded.
- **Anushka Rawat:** A verification-gated synthetic-data engine with three trace sources and guaranteed correctness was developed. Implemented a controlled study of three injection points under one engine and one evaluation harness. Diagnosed and fixed an RLOO advantage-collapse failure mode, with per-step diagnostics. Conducted ChatGPT vs Algorithmic vs QWEN RLOO tests. Worked on ReFT, warm start-up, adjusted hyperparameters to get best results.

Changes from Proposal The integration of the Schulman KL estimator and the specific FAISS embedding index were dynamically added during execution to respectively address optimization instability and dataset redundancy, which were not anticipated in the initial project proposal.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, et al. 2024. Back to basics: Revisiting REINFORCE-style optimization for learning from human feedback in LLMs. *ACL* (2024).

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 *AISTATS*.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *ICML*.

Mark Chen, Jerry Tworek, Heewoo Jun, et al. 2021. Evaluating large language models trained on code. *arXiv:2107.03374*.

Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. Deep reinforcement learning from human preferences. In *NeurIPS*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, et al. 2021. Training verifiers to solve math word problems. *arXiv:2110.14168* (2021).

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv:2501.12948* (2025).

Hanze Dong, Wei Xiong, Deepanshu Goyal, et al. 2023. RAFT: Reward ranked finetuning for generative foundation model alignment. *TMLR*.

Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. 2024. KTO: Model alignment as prospect theoretic optimization. *ICML* (2024).

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse curriculum generation for reinforcement learning. In *CoRL*.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners. *arXiv:2503.01307* (2025).

Kanishk Gandhi, Denise Lee, Gabriel Grand, et al. 2024. Stream of Search (SoS): Learning to search in language. *COLM* (2024).

Alex Graves, Marc G Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. 2017. Automated curriculum learning for neural networks. In *ICML*.

Caglar Gulcehre, Tom Le Paine, Srivatsan Srinivasan, et al. 2023. Reinforced self-training (ReST) for language modeling. *arXiv:2308.08998* (2023).

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, et al. 2019. Parameter-efficient transfer learning for NLP. In *ICML*.

Edward J Hu, Yelong Shen, Phillip Wallis, et al. 2022. LoRA: Low-rank adaptation of large language models. In *ICLR*.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Buy 4 REINFORCE samples, get a baseline for free! *ICLR Workshop* (2019).

Hunter Lightman, Vineet Kosaraju, Yura Burda, et al. 2023. Let's verify step by step. *arXiv:2305.20050* (2023).

Shih-Yang Liu, Chien-Yi Wang, Hongxu Yin, et al. 2024. DoRA: Weight-decomposed low-rank adaptation. *ICML* (2024).

Yu Meng, Mengzhou Xia, and Danqi Chen. 2024. SimPO: Simple preference optimization with a reference-free reward. *NeurIPS* (2024).

Andrew Y Ng, Daishi Harada, and Stuart Russell. 1999. Policy invariance under reward transformations: Theory and application to reward shaping. (1999).

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. 2022. Training language models to follow instructions with human feedback. In *NeurIPS*.

Richard Yuanzhe Pang, Weizhe Yuan, Kyunghyun Cho, He He, Sainbayar Sukhbaatar, and Jason Weston. 2024. Iterative reasoning preference optimization. *arXiv:2404.19733* (2024).

Shubham Parashar et al. 2025. Curriculum-based self-improvement for reasoning. *arXiv* (2025).

Ryan Park, Rafael Rafailov, Stefano Ermon, and Chelsea Finn. 2024. Disentangling length from quality in direct preference optimization. *arXiv:2403.19159* (2024).

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. In *NeurIPS*.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv:1707.06347* (2017).

Zhihong Shao, Peiyi Wang, Qihao Zhu, et al. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv:2402.03300* (2024).

Avi Singh, John D Co-Reyes, Rishabh Agarwal, et al. 2024. Beyond human data: Scaling self-training for problem-solving with language models. *TMLR* (2024).

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. In *NeurIPS*.

Jonathan Uesato, Nate Kushman, Ramana Kumar, et al. 2022. Solving math word problems with process- and outcome-based feedback. *arXiv:2211.14275* (2022).

Xuezhi Wang, Jason Wei, Dale Schuurmans, et al. 2023. Self-consistency improves chain of thought reasoning in language models. In *ICLR*.

Zhengxuan Wu, Aryaman Arora, Zheng Wang, et al. 2024. ReFT: Representation finetuning for language models. *NeurIPS* (2024).

Zheng Yuan, Hongyi Yuan, Chengpeng Li, et al. 2023. Scaling relationship on learning mathematical reasoning with large language models. *arXiv:2308.01825* (2023).

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D Goodman. 2022. STaR: Bootstrapping reasoning with reasoning. In *NeurIPS*.

Daniel M Ziegler, Nisan Stiennon, Jeffrey Wu, Tom B Brown, Alec Radford, Dario Amodei, Paul Christiano, and Geoffrey Irving. 2019. Fine-tuning language models from human preferences. *arXiv:1909.08593* (2019).

A Additional Experiments

Comprehensive telemetry logs and optimization trajectories for the evaluated parameter-efficient adaptation methodologies—specifically Low-Rank Adaptation (LoRA) and Representation Finetuning (ReFT)—are presented in this section. All three stages of the alignment pipeline (Supervised Fine-Tuning (SFT), Identity Preference Optimization (IPO), and Reinforcement Learning with Leave-One-Out (RLOO)) are encompassed by the compiled empirical data.

SFT convergence dynamics and generalization metrics for LoRA are detailed in 14 and 15, by which the stabilization of cross-entropy loss over the synthetic trace distributions is illustrated. The IPO phase metrics, which are captured in 16 and 17, are utilized to visualize the active margin maximization between chosen and rejected trace embeddings. The empirical efficacy of the squared-residual regression objective is thereby validated by these margin plots.

Furthermore, the high-variance dynamics of the RLOO policy-gradient updates are rigorously quantified in 18, 19, 20 and 21 for LoRA, and by 22, 23, 24, 25, 26, 27, 28, 29 for ReFT. Crucial policy stability indicators are plotted, including the Schulman KL Divergence estimator, Hallucination Penalty Mean, and Repetition Penalty Mean, which are juxtaposed against the Penalized Reward Mean. The absolute necessity of stringent advantage clipping and dynamic KL penalty constraints, required to prevent catastrophic mode collapse during the exploration of the sparse-reward Countdown environment, is explicitly demonstrated by these plotted trajectories.

Figure 14: LoRA - SFT - Train/Accuracy and Train/Loss

Figure 15: LoRA - SFT - Test/Accuracy and Test/Loss

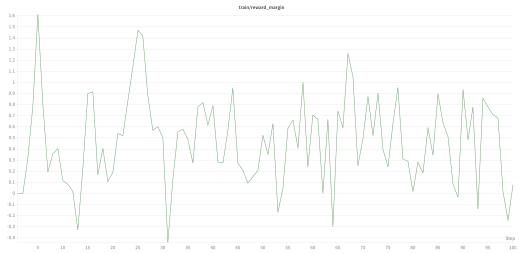

Figure 16: LoRA - IPO - Train/Reward Margin

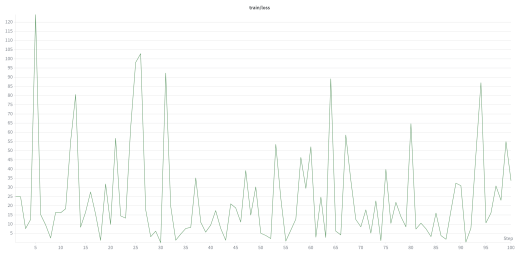

Figure 17: LoRA - IPO - Train/Loss

Figure 18: LoRA - RLOO - Train/Hallucination Penalty Mean and Train/KL Loss

Figure 19: LoRA - RLOO - Train/Loss and Train/Penallized Reward Mean

Figure 20: LoRA - RLOO - Train/Repetition Penalty Mean and Train/Reward Mean

Figure 21: LoRA - RLOO - Train/Rollout Accuracy and Train/Schulman KL Div

Figure 22: ReFT - SFT - Train/Accuracy and Train/Loss

Figure 23: ReFT - SFT - Test/Accuracy and Test/Loss

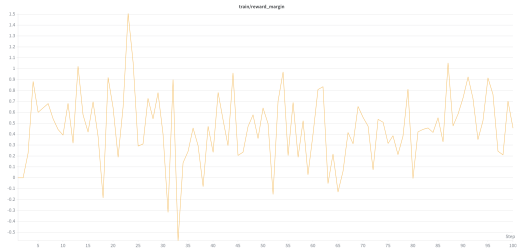

Figure 24: ReFT - IPO - Train/Reward Margin

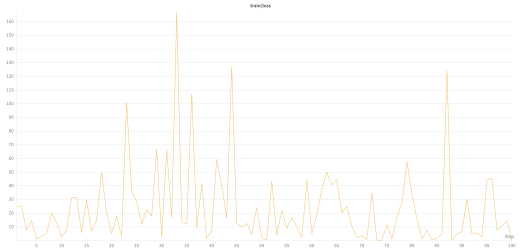

Figure 25: ReFT - IPO - Train/Loss

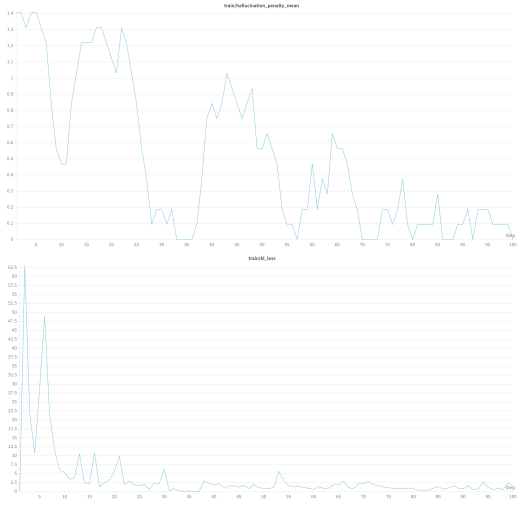

Figure 26: ReFT - RLOO - Train/Hallucination Penalty Mean and Train/KL Loss

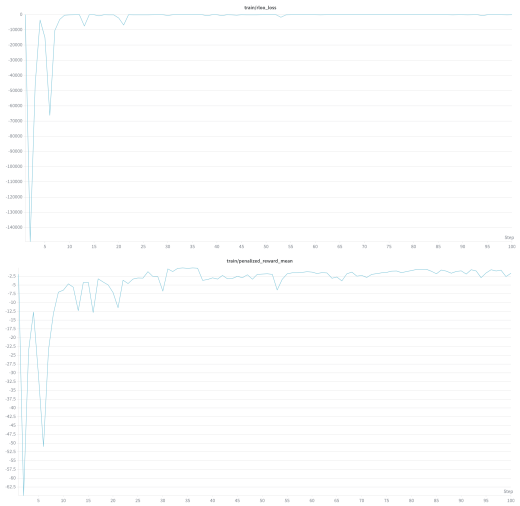

Figure 27: ReFT - RLOO - Train/Loss and Train/Penallized Reward Mean

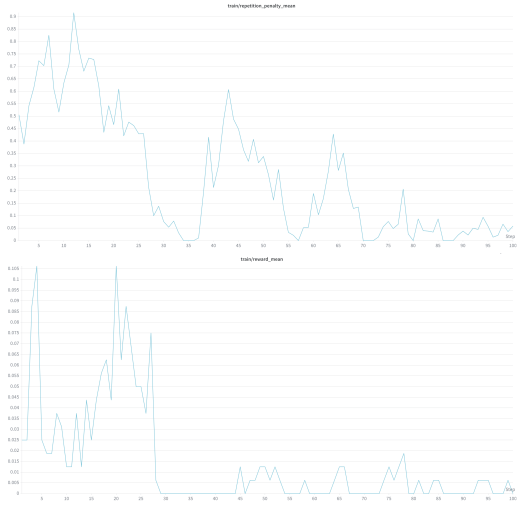

Figure 28: ReFT - RLOO - Train/Repetition Penalty Mean and Train/Reward Mean

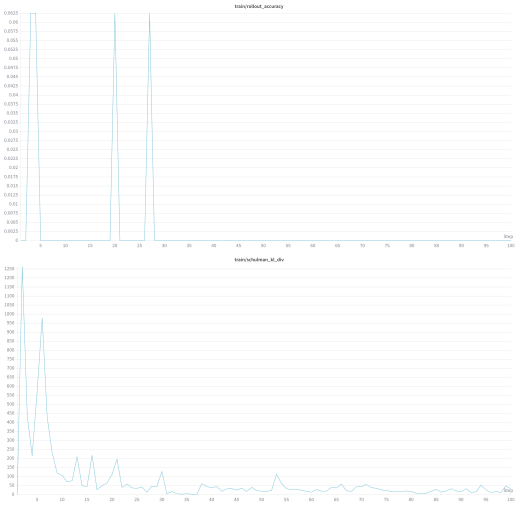

Figure 29: ReFT - RLOO - Train/Rollout Accuracy and Train/Schulman KL Div