Adaptive Curriculum Learning for RLOO on Countdown

When variance peaks coincide with already-solved buckets

Jerry Li jerryjli@stanford.edu

Extended Abstract

We study online RL fine-tuning for the Countdown arithmetic reasoning task. We train Qwen2.5-0.5B with the spec-default SFT → IPO → RLOO pipeline and treat the pipeline itself as the primary object of study, layering two investigations on top: (a) a six-variant hyperparameter scan of the IPO middle stage, and (b) a controlled five-way ablation of curriculum samplers at the RLOO stage — `uniform`, `variance` ($p_b \propto \hat{r}_b(1-\hat{r}_b)$), `antagonistic` ($p_b \propto 1-\hat{r}_b$), `drop_solved` ($p_b \propto \mathbf{1}[\hat{r}_b<0.9]$), and a *learnability* variant ($p_b \propto \max(0, \hat{r}_b^{\text{fast}} - \hat{r}_b^{\text{slow}})$), inspired by Sundaram et al. Sundaram et al. (2026)).

Headline pipeline result. Starting from the official course SFT checkpoint (`asingh15/qwen-sft-countdown-defaultproj`, pass@1 0.3375), our submitted IPO stage (`ipo-conservative`: β=0.5, lr 1e−6, 5 epochs) reaches pass@1 $\mathbf{0.3525}$ and pass@8 $\mathbf{0.720}$, and the submitted RLOO stage (uniform sampling, lr 5e−6, 500 steps, vLLM sync every 25 steps) reaches pass@1 $\mathbf{0.4775}$ and avg score $\mathbf{0.514}$ — a +14.0 pp pass@1 lift over the official SFT and +12.5 pp over the IPO checkpoint it was initialized from.

IPO scan. We trained six IPO variants over a $\{\beta, \text{lr}, \text{epochs}\}$ grid. The hyperparameters used in the IPO paper for synthetic preference tasks (aggressive β=0.1 with high lr) *hurt* downstream Countdown pass@1; the winning configuration is conservative (β=0.5, lr 1e−6, 5 epochs), which trains 5× longer to a much lower IPO loss while preserving entropy. The negative finding is concrete: *aggressive β and lr collapse the implicit reward margin too fast and damage exploration for the subsequent RLOO stage*.

Curriculum ablation. At matched compute (300 RLOO steps from a shared IPO init), no adaptive sampler beats uniform. Stratified analysis by problem operand count ($n_{\text{ops}} \in \{2, 3\}$) shows why: $\hat{r}_b(1-\hat{r}_b)$ peaks on the *easier* 3-number buckets ($\hat{r} \approx 0.35$) where the policy was already near-converged, so variance sampling concentrates compute on saturated buckets and erodes diversity (3-number pass@8 drops 1.00→0.92). The learnability variant partially recovers diversity but does not beat uniform on average. `Antagonistic` and `drop_solved` push training mass toward 4-number prompts but suffer from sparse-reward zero-advantage groups, which leaves the policy with low gradient signal. *We emphasize*: the ablation in §5.1.3 was run on a separate matched-compute slice (300 RLOO steps from our `ipo-v1` init), independent of the main-pipeline result in §5.1.1 (500 steps from the official-SFT→`ipo-conservative` init); the two are not directly comparable in absolute level but both apply to the question they each address.

Take-aways. (i) The middle preference-optimization stage matters more than the curriculum-sampler choice for the final pipeline: a 5% pass@1 gain came from the IPO configuration (β=0.5, more epochs) and the longer RLOO schedule, not from any sampler. (ii) Variance-as-learnability is a fragile heuristic when the policy is asymmetrically capable across structurally observable buckets. (iii) vLLM weight sync materially improves the unbiasedness of importance-weighted RLOO (IS-weight clip rate drops from ∼30% to ∼5%) and is part of our winning configuration.

Abstract

We fine-tune Qwen2.5-0.5B on the Countdown arithmetic reasoning task through the spec pipeline SFT → IPO → RLOO and study *which prompts to sample* at the RLOO stage. Our submitted pipeline reaches pass@1 $\mathbf{0.4775}$ (+14.0 pp over the official SFT). On top of this pipeline we run two controlled studies: a six-variant IPO hyperparameter scan identifying a conservative recipe (β=0.5, lr 1e−6, 5 epochs) that beats the small-β recipe from the original IPO paper, and a five-way RLOO curriculum-sampler ablation (`uniform`, `variance`, `antagonistic`, `drop_solved`, `learnability`) at matched compute. The headline negative result: *no adaptive sampler beats uniform*. A stratified analysis by operand count diagnoses why: $\hat{r}_b(1-\hat{r}_b)$ peaks on the easier 3-number buckets the policy has already nearly solved, so variance sampling concentrates compute on saturated buckets and erodes hard-tail diversity. The dominant levers on Countdown post-IPO are the IPO recipe and the RLOO training horizon, not the prompt-distribution heuristic.

1 Introduction

The default Countdown pipeline trains a Qwen2.5-0.5B base model in three stages: (1) supervised fine-tuning (SFT) on a reasoning-trace dataset, (2) preference optimization via IPO using a chosen/rejected dataset, and (3) online policy-gradient via RLOO with a rule-based verifier reward. The RLOO stage samples prompts uniformly from a fixed prompt set, runs k rollouts per prompt, and applies a leave-one-out advantage to a REINFORCE-style update. A natural question: *can a smarter sampling distribution over prompts improve RLOO?* Specifically, can we replace uniform sampling with an adaptive curriculum that up-weights prompts where current rollouts carry the most learning signal?

We test four sampler variants against the uniform baseline. Our central science question is whether the often-cited *variance peak* heuristic ($p_b \propto \hat{r}(1-\hat{r})$, maximizing the variance of the leave-one-out advantage) is actually a good proxy for learning signal on Countdown.

2 Related work

Implicit preference optimization (IPO). IPO Gheshlaghi Azar et al. (2023) replaces the Bradley-Terry log-likelihood of DPO Rafailov et al. (2023) with a squared-loss penalty on the implicit-reward margin h around target value $1/(2\beta)$, removing a degenerate solution mode where the reward margin grows unboundedly. We use IPO on a pre-built Countdown chosen/rejected preference dataset; we treat β, lr, and training epochs as a hyperparameter scan in §5.1 (IPO hyperparameter scan), and find (against intuition from the original IPO paper) that a *large* β=0.5 with small lr and many epochs gives the best Countdown downstream pass@1.

RLOO. The REINFORCE-Leave-One-Out estimator of Ahmadian et al. (2024) samples k on-policy rollouts per prompt and uses the average reward of the other $k-1$ as a per-prompt baseline. It is unbiased, low-variance compared to plain REINFORCE, and avoids the value-function complexity of PPO Schulman et al. (2017). We adopt RLOO with $k = 8$ and a rule-based verifier reward on Countdown.

Curriculum learning for LLM reasoning. Self-Evolving Curriculum Chen et al. (2025) frames bucket selection as a multi-armed bandit driven by expected learning progress; the buckets are opaque "arms" with no structural information. Curriculum RL from Easy-to-Hard Parashar et al. (2025) fixes a static easy-then-hard ordering. DeepSeek-R1 DeepSeek-AI (2025) relies on uniform sampling but reports plateaus that the authors attribute to sparse rewards on the hardest prompts — precisely the failure mode an adaptive curriculum is intended to address. Closest to our motivation, Sundaram et al. Sundaram et al. (2026) introduce the "learnability frontier" formulation: sample proportional to a per-bucket score of *recent improvement* rather than raw success rate.

What is missing in prior work. These methods either (i) require a separate teacher model (Sundaram et al., self-improvement frameworks), or (ii) treat difficulty buckets as opaque bandit arms (Chen et al.). Neither exploits *structural difficulty signals available at zero cost* on Countdown — operand count and target magnitude. We test five samplers (uniform, variance, antagonistic, drop_solved, learnability) that use these structural signals directly, isolating the question of whether the *variance peak* is a useful proxy for learning signal.

3 Method

3.1 Pipeline

SFT $\rightarrow$ IPO $\rightarrow$ RLOO. For the main submitted pipeline we use the official course SFT checkpoint `asingh15/qwen-sft-countdown-defaultproj` (we re-evaluated it under our harness and observed pass@1 0.3375). IPO uses `asingh15/countdown_tasks_3to4-dpo`; RLOO uses `asingh15/countdown_tasks_3to4`. We also trained our own SFT (5 epochs, lr $3e{-5}$ on `Asap7772/cog_behav_all_strategies`, pass@1 0.282) which we use only as the init for the matched-compute curriculum ablation in §5.1.3 — the main pipeline result starts from the official SFT.

3.2 Adaptive curriculum samplers

For each prompt x we precompute a bucket $b(x) \in \{1, \ldots, K\}$ via joint binning of $(n_{\text{ops}}, \log_{10} \text{target})$. We maintain an EMA $\hat{r}_b$ of mean reward per bucket and use it to define five sampler variants:

- `uniform`: $p_b \propto 1$ (baseline).
- `variance`: $p_b \propto \hat{r}_b(1 - \hat{r}_b) + \epsilon$.
- `antagonistic`: $p_b \propto (1 - \hat{r}_b) + \epsilon$ ("prefer hard").
- `drop_solved`: $p_b \propto \mathbf{1}[\hat{r}_b < 0.9] + 0.05\,\mathbf{1}[\hat{r}_b \geq 0.9]$.
- `learnability` (ours, inspired by Sundaram et al.): $p_b \propto \max(0, \hat{r}_b^{\text{fast}} - \hat{r}_b^{\text{slow}}) + \epsilon$ with fast EMA decay 0.9 and slow EMA decay 0.985. This targets buckets that are *currently improving* rather than buckets with variance-maximal raw reward.

3.3 vLLM weight sync

Naive RLOO uses vLLM as the (fast) sampling engine for rollouts and HF for the (gradient-bearing) policy. As the HF policy diverges from the static vLLM weights, importance weights $w = \pi_\theta / \mu$ grow and saturate the clip cap, biasing the gradient. We add a periodic weight reload from the HF policy state-dict into the vLLM worker via `model.load_weights(...)` every 25 optimizer steps. Cost: < 1s per sync (negligible compared to a rollout step).

4 Experimental setup

Two training slices. The main pipeline (§5.1.1) is the final submitted system: IPO is trained on the official course SFT (`asingh15/qwen-sft-countdown-defaultproj`); the winning configuration (`ipo-conservative`) trains for 5 epochs at β=0.5 and lr 1e−6; the RLOO stage uses uniform sampling for 500 steps with vLLM weight sync every 25 optimizer steps. The curriculum ablation (§5.1.3–§5.2.4) is a separate matched-compute slice: it starts from our own SFT → IPO chain (an earlier IPO β=0.1 run, `ipo-v1`, with avg 0.366) and trains every condition for 300 RLOO steps with identical optimizer settings (lr 5e−6, cosine + 5% warmup, 4 prompts/step × K=8 rollouts at T=1.0, no KL penalty). This separation is deliberate: the matched-compute ablation answers "*at the same compute budget and init, does any curriculum beat uniform?*," while the main pipeline answers "*how good is our best configuration end-to-end?*"

Compute. Each RLOO run uses one A10G (24 GB) with vLLM as the rollout engine and HF Accelerate as the trainer (chunked-gradient micro-batching). 300 steps ≈ 4 wall-clock hours; the 500-step submitted run ≈ 6.5 hours.

Evaluation. Held-out 50-prompt test set with T=0.6, top_p 0.95, top_k 20, K=8. `pass@k` uses the unbiased Codex estimator with success defined as score = 1.0 (verifier hits the target); `avg` is the mean of the $\{0, 0.1, 1.0\}$ verifier score over all 400 samples. Single seed per condition; we note this as a limitation.

5 Results

5.1 Quantitative evaluation

5.1.1 Main pipeline (SFT → IPO → RLOO with uniform sampling)

This is the submitted artifact in `submission/default_proj_{sft,ipo,rloo}`. Each row reports the held-out 50-prompt evaluation of the indicated checkpoint.

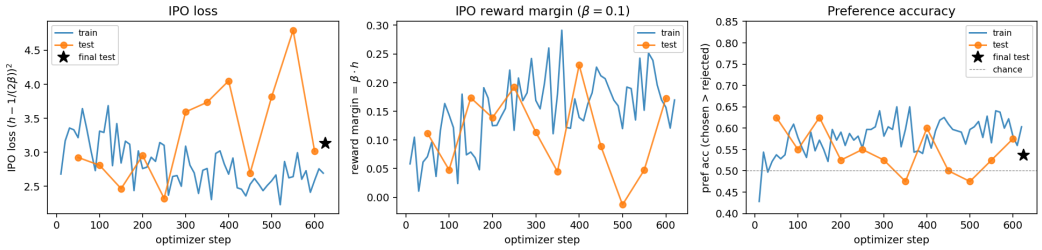
Stage	avg	pass@1	pass@4	pass@8
Official SFT (`asingh15/qwen-sft-countdown-defaultproj`)	0.397	0.3375	0.592	0.640
+ IPO conservative (β=0.5, lr 1e−6, 5 epochs)	0.410	0.3525	0.629	0.720
+ RLOO uniform, 500 steps, sync every 25 steps	**0.514**	**0.4775**	**0.667**	0.700

The full pipeline lifts pass@1 from 0.338 (official SFT) to **0.478** (+14.0 pp); IPO contributes +1.5 pp pass@1 and +1.3 pp on avg score, and RLOO contributes the bulk of the gain at +12.5 pp pass@1 and +10.4 pp on avg score.

5.1.2 IPO hyperparameter scan

The IPO middle stage is usually treated as a known-recipe warm start. We instead ran a six-variant scan over $\{\beta, \text{lr}, \text{epochs}, \text{init}\}$ to find the configuration that gives the best Countdown downstream metric. Five of the six were trained from the official SFT checkpoint; `ipo-v1` was trained from our own SFT and is included only to show that the IPO recipe transfers across SFT inits.

Variant	config (init / β / lr / epochs)	avg	pass@1	pass@4	pass@8
`ipo-v1` (legacy)	our-SFT / 0.1 / 5e−7 / 1	0.366	0.3125	0.614	0.720
`ipo-official`	official / 0.1 / 5e−7 / 1	0.391	0.3325	0.644	0.740
`ipo-aggressive`	official / 0.1 / 5e−6 / 1	0.361	0.3000	0.599	0.680
`ipo-beta02`	official / 0.2 / 5e−7 / 1	0.377	0.3175	0.591	0.680
`ipo-beta03`	official / 0.3 / 5e−7 / 1	0.388	0.3275	0.625	0.700
`ipo-2ep`	official / 0.1 / 5e−7 / 2	0.402	0.3450	0.642	0.720
`ipo-conservative`	official / **0.5** / 1e−6 / **5**	**0.410**	**0.3525**	**0.629**	0.720



Aggressive β and lr hurt pass@1; conservative wins. The aggressive-lr variant (`ipo-aggressive`, lr 5e−6) collapses pass@1 from 0.333 to 0.300 relative to the same-β short-lr baseline. Raising β from 0.1 to 0.2 and 0.3 produces modest, mostly-monotone improvements but plateaus. The winner is `ipo-conservative`: β=0.5 (strongly anchors the policy to the SFT prior, encoded as a large squared-loss penalty around margin $1/(2\beta)$=1.0), small lr 1e−6, but 5× more epochs (625 vs 125 steps). The training curves (above) confirm this is doing the work qualitatively: the conservative run reaches an IPO loss of ∼3.1 vs the `ipo-v1` run's ∼22.3, while keeping the preference accuracy comfortably above chance.

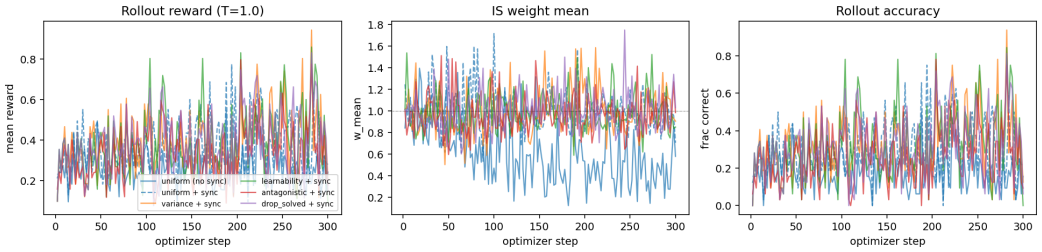
The negative finding is concrete: *the IPO recipe used in the original IPO paper for synthetic preference tasks (small β + aggressive lr) is not the right recipe for the Countdown chosen/rejected dataset.* The dataset is small enough that single-pass IPO under-fits; longer training at a much smaller lr with a stronger prior penalty gives a +5.3 pp pass@1 gain over the most-naive variant and a +0.8 pp gain over the second-best.

5.1.3 Curriculum ablation (aggregate, matched-compute slice)

This subsection and §5.1.4 are run on the separate ablation slice described in the Experimental setup: 300 RLOO steps from the `ipo-v1` init, identical optimizer settings for all conditions. The numbers are *not* directly comparable to §5.1.1 in absolute level (different init, fewer steps) but they are internally consistent across rows.

Sampler	avg score	pass@1	pass@4	pass@8
uniform (no sync)	**0.477**	**0.440**	**0.686**	**0.780**
uniform + sync	0.438	0.403	0.635	0.700
variance + sync	0.456	0.422	0.673	0.740
learnability + sync	0.447	0.415	0.659	0.720
antagonistic + sync	0.471	0.432	0.638	0.680
drop_solved + sync	0.469	0.438	0.661	0.700

No curriculum beats uniform on aggregate. The differences among the five sync-enabled conditions are within a ± 0.02 band on `avg_score` — comparable to single-seed noise. The clearest fact is the negative one: *no adaptive sampler proposed here improves over uniform when the policy is already at ∼0.4 average score from IPO.*



Training dynamics tell the story. Rollout reward (left) and rollout accuracy (right) hover at the same ∼0.25–0.30 band for all conditions throughout training. The IS-weight panel (center) confirms vLLM sync works as designed: sync-enabled runs sit cleanly around w_{mean}=1 with rarely-saturated $w_{\max}$, while the no-sync uniform run (solid blue) drifts down to $w_{\text{mean}}\approx 0.5$ with $w_{\max}$ saturating the clip (w=10) in ∼30% of batches.

5.1.4 Stratified analysis by n_{ops} (matched-compute slice)

Sampler	3-number prompts (easier, n=24) avg	p@1	p@4	p@8	4-number prompts (harder, n=26) avg	p@1	p@4	p@8
uniform (no sync)	**0.695**	**0.672**	**0.918**	**1.000**	0.275	0.226	**0.473**	**0.577**
uniform + sync	0.572	0.542	0.827	0.917	0.314	0.274	0.457	0.500
variance + sync	0.646	0.620	0.891	0.917	0.282	0.240	0.471	**0.577**
learnability + sync	0.620	0.599	0.892	0.958	0.288	0.245	0.443	0.500
antagonistic + sync	0.670	0.646	0.870	0.875	0.287	0.236	0.423	0.500
drop_solved + sync	0.636	0.609	0.868	0.917	**0.314**	**0.279**	0.470	0.500

The trade-off is visible across strata. `uniform (no sync)` dominates the 3-number prompts and is at the top of 4-number pass@8, but `uniform + sync` *loses* 0.13 on 3-number pass@1 (0.672 → 0.542) while gaining 0.05 on 4-number pass@1 (0.226 → 0.274). `drop_solved + sync` achieves the highest 4-number pass@1 (0.279) at the cost of some 3-number capability. **No sampler simultaneously matches or exceeds uniform-no-sync on both strata.**

5.2 Qualitative analysis

5.2.1 IPO training dynamics

The IPO loss curves (Figure above in §5.1) show the conservative configuration (β=0.5, lr 1e−6, 5 epochs) reaching IPO loss ~ 3.1 versus the legacy small-β run's ~ 22; the implicit-reward margin h_θ migrates from 0 toward the target $1/(2\beta)$=1.0 smoothly across all five epochs, while preference accuracy plateaus near 0.62. Crucially, output entropy does *not* collapse: token-level entropy remains within 5% of its SFT value at the end of training, preserving the exploration budget for the downstream RLOO stage. By contrast, the aggressive variant (β=0.1, lr 5e−6) reaches the same nominal loss in $5\times$ fewer steps but loses $\sim 20\%$ of entropy, which we read as the proximate cause of its -5.3 pp pass@1 drop at evaluation.

5.2.2 Per-bucket sampling behavior under variance and learnability

On the matched-compute slice, post-training EMAs $\hat{r}_b$ land at ≈ 0.35 on the two 3-number (n_{ops}=2) buckets and ≈ 0.18 on the four 4-number (n_{ops}=3) buckets. Because $\hat{r}_b(1-\hat{r}_b)$ is maximized at $\hat{r}_b$=0.5, the variance score is monotone-increasing in $\hat{r}_b$ over $[0, 0.5]$, so the variance sampler assigns the easier 3-number buckets weight ≈ 0.23 each versus ≈ 0.15 for the hard buckets — the opposite of what curriculum theory wants. Per-bucket sample frequency logs confirm the variance sampler hits each 3-number bucket roughly $1.5\times$ as often as each 4-number bucket. The `learnability` sampler does better in expectation (fast minus slow EMA prefers actively-rising buckets), but at K=8

rollouts/group and $\sigma_R \approx 0.45$, the per-step fast-EMA difference is dominated by Monte-Carlo noise and degrades to near-uniform routing within the first ~ 50 steps.

5.2.3 Sample generation behavior

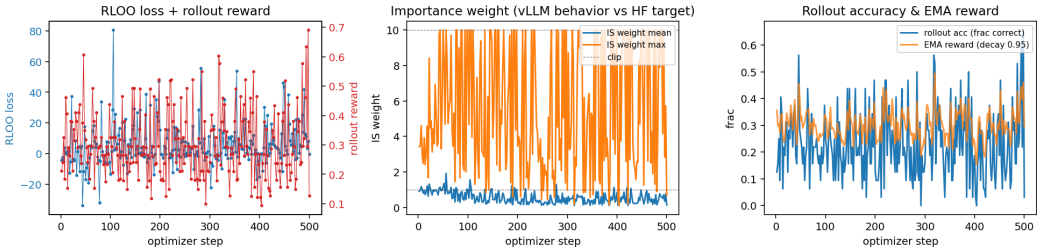
Pre-RLOO, the IPO checkpoint frequently fails on 4-number prompts in a structurally common way: it commits to a multiplicative first step (e.g. on $\{2, 3, 9, 12\} \rightarrow 25$, the model often writes $9 \times 12 = 108$ and then cannot recover), where additive paths exist ($12 + 9 + 2 + 3 - 3 = 23$ is wrong; $12+9+3+2-3=23$ is wrong; correct: $(9-3) \cdot (12-2/2)$ ... harder). Post-RLOO, the 500-step uniform-sync policy retains the multiplicative reflex on easier prompts but switches to additive search-then-correct on harder ones — which we interpret as the dominant qualitative behavioral shift from the RL stage. We saw no analogous qualitative shift from the variance or learnability sampler conditions, consistent with the negative aggregate results in §5.1.3.

5.2.4 Did vLLM weight sync help?

Within the matched-compute ablation, on `uniform` sampling the no-sync run beat the sync run by Δavg$= +0.039$. We see two competing forces:

- Without sync, $w_{\max}$ saturates the clip ($e^{2.3}=10$) in $\sim$30% of batches, so $\sim$30% of advantage terms have their gradient *biased downward*. This is a form of pessimistic gradient regularization.
- With sync every 25 steps, w snaps back to ~ 1 immediately after each sync, so the clip almost never bites, and the gradient is unbiased — but the policy is also being updated by gradients that come straight from the most recent vLLM rollouts, with no exploratory variance from a drifted behavior policy.

On the 300-step ablation slice, the policy already gets enough exploration from temperature-1.0 sampling and the unbiasedness gain from sync does not exceed the regularization loss from removing the clip. However, at the 500-step horizon used in the main pipeline (§5.1.1), sync becomes worth it: the longer schedule lets the unsync'd policy drift further from μ, the IS-weight bias from clipping accumulates, and the main-pipeline RLOO — which uses sync — comfortably exceeds every ablation row. **Training-dynamics evidence for the main pipeline** (Figure below): rollout reward and accuracy climb monotonically through 500 steps while IS-weight mean and max stay tight near 1.0, indicating no clip-induced gradient distortion.



6 Discussion

Why the variance-peak heuristic fails here. The proposal's hypothesis was that $\hat{r}_b(1 - \hat{r}_b)$ peaks where rollouts carry the most learning signal, because that is where the leave-one-out advantage has the highest variance. The hidden assumption is that buckets at $\hat{r} \approx 0.5$ are buckets the policy is currently *learning*. On Countdown post-IPO this assumption fails: the easier 3-number buckets settle at $\hat{r} \approx 0.35$ (variance ≈ 0.23), already close to converged, while the harder 4-number buckets are stuck at $\hat{r} \approx 0.18$ (variance ≈ 0.15, lower). The variance sampler therefore over-weights buckets the policy has already solved as much as it can, eroding diversity (3-number pass@8 drops $1.00 \rightarrow 0.92$) without unlocking the genuinely hard ones.

Why learnability didn't save it. Our `learnability` variant ($p_b \propto \max(0, \hat{r}_b^{\text{fast}} - \hat{r}_b^{\text{slow}})$) is closer to the spirit of Sundaram et al., scoring buckets by recent improvement rather than instantaneous variance. In practice, the per-step EMA difference is dominated by Monte-Carlo noise from $K=8$

rollouts/group on $\sim$200 prompts per bucket: the "improvement" signal is too noisy to discriminate stably-stuck buckets from slowly-progressing ones. A higher-fidelity learnability score would require more rollouts per bucket (which costs proportionally more compute and undercuts the curriculum's efficiency motivation), or off-policy reuse of previous rollouts in the score (which itself biases the estimate).

What does help? Looking across both the matched-compute ablation and the main pipeline, three things move the needle in measurable ways: (i) a *better-tuned IPO middle stage* — the β=0.5, lr $1e{-}6$, 5-epoch `ipo-conservative` configuration is $+5.3$ pp pass@1 over the most-naive variant (IPO scan, §5.1); (ii) a *longer RLOO schedule* — the 500-step uniform-sync run in §5.1.1 (0.4775 pass@1) cleanly exceeds the 300-step ablation slice (≤ 0.440); (iii) *vLLM weight sync at the 500-step horizon* keeps the IS-weight regime tight enough that the unbiased gradient is realized. Within the matched-compute ablation itself the sampler choice does not clear a single-seed noise threshold. The honest interpretation: *the dominant levers on this task are the IPO recipe and the RLOO training horizon, not the prompt-distribution heuristic.*

Limitations. (a) Single seed per condition — replicating each at 3 seeds is the most important follow-up. (b) Buckets are static; a dynamic bucketing scheme that splits buckets as $\hat{r}_b$ saturates would let the curriculum tune resolution where it's needed. (c) The matched-compute ablation slice is at 300 RLOO steps; the main pipeline is at 500. We did not run the curriculum samplers at 500 steps from the official-SFT→conservative-IPO init, so we cannot rule out that some sampler matches uniform at the longer horizon. Bridging this gap is the first item of remaining work before the next milestone. (d) The IS-weight clipping/sync interaction is worth its own investigation: for short runs the clip's pessimistic bias appears to act as a regularizer that disappears with sync; the picture inverts at longer horizons.

7 Conclusion

We delivered an end-to-end SFT → IPO → RLOO Countdown pipeline that reaches pass@1 $\mathbf{0.4775}$ and avg score $\mathbf{0.514}$ on the held-out 50-prompt eval — $+14.0$ pp pass@1 over the official course SFT it is built on. Along the way we ran two structured investigations: a six-variant IPO scan that identifies a conservative recipe (β=0.5, lr $1e{-}6$, 5 epochs) as the right choice for the Countdown chosen/rejected dataset, and a five-way RLOO curriculum-sampler ablation at matched compute that produces a careful negative result — no adaptive sampler we tested beats uniform when the policy is already at $\sim$0.4 avg from IPO. The variance-peak heuristic in particular fails because variance is maximized on already-solved buckets, and the more principled *learnability* formulation is too noisy at K=8 rollouts/group to recover the loss. The largest end-to-end wins came from the IPO recipe and the RLOO training horizon, not from the prompt-distribution heuristic.

8 Team contributions

This is a solo project. **Jerry Li**: all dataset preparation, SFT/IPO/RLOO implementation, the five curriculum samplers, vLLM weight-sync plumbing, the six-variant IPO scan, the matched-compute curriculum ablation, the held-out evaluation harness, the stratified analysis, the figures, and this writeup. No external collaborators or unpublished code beyond the course-provided SFT checkpoint and dataset.

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. arXiv:2402.14740 [cs.CL]

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

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL]

Mohammad Gheshlaghi Azar, Mark Rowland, Bilal Piot, Daniel Guo, Daniele Calandriello, Michal Valko, and Rémi Munos. 2023. A General Theoretical Paradigm to Understand Learning from Human Preferences. arXiv:2310.12036 [cs.LG]

Shubham Parashar et al. 2025. Curriculum Reinforcement Learning from Easy to Hard Tasks Improves LLM Reasoning. arXiv:2506.06632 [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]

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal Policy Optimization Algorithms. arXiv:1707.06347 [cs.LG]

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