Extended Abstract

Motivation During milestone RLOO training on the Countdown task, we noticed that mean policy entropy decreased while training reward rose. As the policy progressively sharpened, pass@k revealed a substantial gap between greedy performance (pass@1) and multi-sample performance (pass@16), suggesting that diverse completions remain important even when mean reward improves. Recent work argues that only a small fraction of high-entropy tokens act as "forking points" for reasoning diversity, while uniform or global entropy pressure may be inefficient. Our extension asks: who should receive an extra entropy signal during RLOO, and should that signal be gated by the importance weights that RLOO already computes?

Method We introduce `entropy_mode=advantage`, which adds token-level entropy shaping on the RLOO advantage. For each response token, we compute policy entropy H_t. We select the top 20% highest-entropy tokens per sequence and add a detached bonus αH_t to the broadcast RLOO advantage at those positions. The policy-gradient loss becomes token-level, weighted by per-sequence importance weights. We gate the entropy bonus using the log importance ratio between the current and sampling policies: the bonus is active when $|\log w| > \log(1+\tau)$, where τ is `iw_gate_threshold`. We optionally retain the milestone global term alongside advantage-mode shaping. Relative to milestone RLOO and entropy shaping on other estimators, we provide an implementation of selective entropy *advantage* shaping with IW-gated ablations, evaluated on Countdown pass@k. Our main empirical contribution is ablative: ungated selective shaping improves pass@16, while hybrid global entropy improves pass@1: a tradeoff not visible from milestone training alone.

Implementation and Results We evaluate on the Countdown arithmetic task: given numbers and a target, the model must produce a valid equation using each number exactly once. Our primary baseline is milestone RLOO (global entropy only). Extension ablations vary (i) IW gate threshold, (ii) entropy advantage scale α, and (iii) global entropy $\lambda = 0.001$ inclusion (hybrid run). All non-baseline extension runs use selective advantage shaping with top-20% token selection; $\lambda = 0$ except in the hybrid run. We report pass@k on 50 held-out Countdown prompts with k up to 16 samples per prompt. Extension training matches milestone RLOO where possible: details itemized in paper.

Run	IW gate τ	pass@1	pass@16
RLOO baseline (global ent.)	—	49.6%	72.0%
+ adv., loose IW	0.1	41.5%	66.0%
+ adv., strict IW	1.0	39.5%	72.0%
+ adv., no IW gate	-0.99	48.0%	**74.0%**
+ adv. + global ent.	-0.99	**52.0%**	72.0%

Discussion Our ablations suggest that where entropy pressure is applied matters more than whether it is tied to importance-weights. Ungated selective shaping improves pass@16 by 2 percentage points over milestone RLOO, consistent with the view that a small fraction of high-entropy tokens act as branching points for diverse reasoning trajectories. Strict IW gating ($\tau = 1.0$) suppresses the bonus on roughly half of token positions and yields the weakest pass@1, refuting our proposal hypothesis that gating would focus exploration on "genuine" off-policy updates. Notably, mean training entropy still declines under all variants; improved pass@k therefore likely reflects *targeted* exploration pressure on high-entropy positions rather than sustained higher terminal entropy Yue et al. (2025). In terms of limitations, results are based on 50 evaluation prompts, a single training seed, fixed α and top-20% token selection without hyperparameter sweeps, and 100 RLOO steps on a 0.5B-parameter model. We did not implement adaptive entropy schedules from our original proposal.

Conclusion Selective entropy advantage shaping is a lightweight modification to milestone RLOO that can improve multi-sample Countdown performance when the bonus is applied consistently to high-entropy tokens. IW-based gating did not help at strict thresholds and should be treated as a negative result rather than a default design choice. Future work should sweep intermediate gate thresholds, adaptive α, and larger pass@k evaluations to test whether these tradeoffs hold at scale.

Selective Entropy Shaping For RLOO: When Importance Weight Gating Hurts Exploration

Syed Ashal Ali
Department of Computer Science
Stanford University
ashal@stanford.edu

Abstract

Online RL fine-tuning can improve language-model reasoning, but methods such as RLOO may also reduce policy entropy and over-concentrate on a narrow set of solution strategies. This project studies entropy collapse in the Countdown arithmetic task and proposes token-selective entropy shaping for RLOO. Instead of applying a global entropy bonus to every token, we add an entropy bonus to the RLOO advantage only for the top 20% highest-entropy response tokens, which are treated as reasoning "forks," and evaluate whether importance-weight deviation should gate this bonus. Using the milestone evaluation protocol, the ungated top-20% entropy-advantage method achieves the best pass@16 result, improving from 72.0% to 74.0%, while a hybrid selective-plus-global entropy method achieves the best pass@1 result at 52.0%. In contrast, strict importance-weight gating activates the bonus on fewer token positions and weakens performance, suggesting that gating can suppress useful exploration. These results suggest that selective entropy shaping can modestly improve multi-sample reasoning performance, but its effectiveness depends strongly on how and when the entropy bonus is applied.

1 Introduction

Reasoning-focused language models have increasingly relied on reinforcement learning with verifiable rewards (RLVR), where model outputs are scored by automatic checkers rather than human preference labels. This paradigm is especially natural for mathematical and code reasoning, since solutions can often be verified directly. For example, DeepSeek-R1 demonstrates that reinforcement learning can substantially improve reasoning behavior when correctness can be automatically rewarded, making RLVR an important framework for studying post-training in language models Guo et al. (2025). In this project, we study RLVR through the Countdown arithmetic task, where a model receives a set of numbers and a target value and must generate a valid arithmetic expression that reaches the target.

The default pipeline uses REINFORCE Leave-One-Out (RLOO) as the online RL stage after supervised fine-tuning and preference optimization. Ahmadian et al. show that, despite the popularity of PPO, carefully implemented REINFORCE-style objectives can be competitive while avoiding some of PPO's computational and tuning complexity Ahmadian et al. (2024). This makes RLOO a useful setting for studying lightweight modifications to online RL: the method samples multiple completions per prompt, scores them using a verifier, subtracts a leave-one-out baseline to reduce variance, and updates the model toward higher-reward trajectories.

However, optimizing verifier reward can create a tension between exploitation and exploration. RLVR improves performance by shifting probability mass toward trajectories that receive high reward, but this can also narrow the model's output distribution. Yue et al. argue that RL-trained reasoning models may not always acquire fundamentally new reasoning abilities; instead, RL often makes already-existing correct paths easier to sample, improving small-k pass@k while potentially reducing

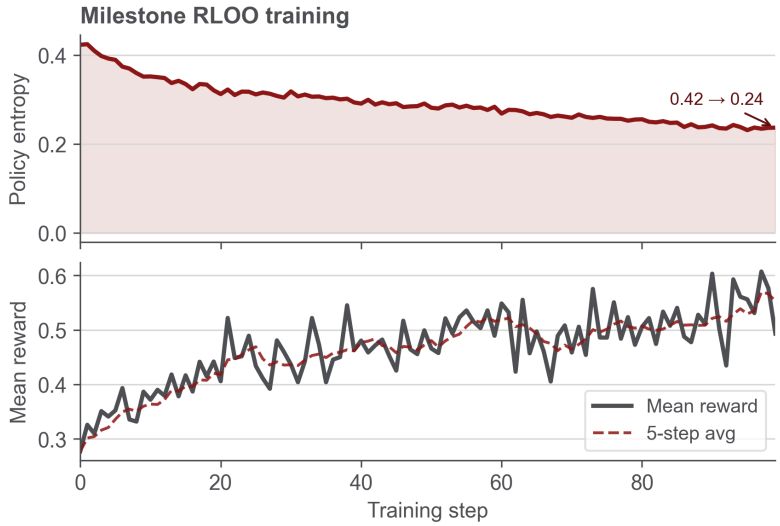

Figure 1: Policy entropy collapses from 0.42 to 0.24 as mean reward increases. This figure highlights the motivation behind the project.

the broader reasoning boundary visible at larger k Yue et al. (2025). This issue is directly relevant to Countdown. Many Countdown prompts can be solved through multiple arithmetic paths, and the evaluation uses pass@k, so performance depends not only on whether the most likely response is correct but also on whether the model maintains enough diversity across samples for at least one trajectory to succeed.

This motivates studying entropy during RLOO training. Entropy measures how spread out the policy is over next-token choices, so falling entropy indicates that the policy is becoming sharper and less exploratory. In my milestone RLOO run, mean policy entropy decreased while training reward increased, suggesting that the model was becoming more confident as it learned to exploit rewarded solution patterns. This is not necessarily bad for pass@1, but it may be harmful for pass@16 if the model collapses toward a limited set of reasoning strategies. Therefore, the goal of this extension is not simply to maximize reward faster, but to ask whether RLOO can preserve useful exploration while still improving correctness.

This project investigates token-selective entropy shaping for RLOO. Instead of applying a global entropy bonus to every response token, we add an entropy bonus to the RLOO advantage only at the top 20% highest-entropy response tokens, treating these positions as likely reasoning "forks." We also test whether the bonus should be gated by RLOO importance weights, since these weights measure how much the current policy likelihood differs from the sampling policy likelihood. The central research question is: who should receive an entropy bonus during RLOO fine-tuning for Countdown? More specifically, we investigate whether targeting high-entropy tokens improves pass@k, and whether importance-weight gating helps focus or instead suppresses the exploration signal.

2 Related Work

A central concern for verifier-based RL is that reward optimization can make the policy narrower over time. Yue et al. study whether RLVR truly expands a model's reasoning capacity and find that RL often improves small-k pass@k by biasing the model toward already-existing correct trajectories, rather than generating fundamentally new reasoning patterns Yue et al. (2025). Their analysis suggests that RL-trained models may become better at sampling correct paths efficiently while also reducing the broader diversity of possible reasoning paths.

Cheng et al. directly connect entropy to exploration in language-model reasoning. They show that high-entropy regions are correlated with exploratory reasoning behaviors, including pivotal logical transitions, reflective actions such as self-verification, and rare under-explored responses Cheng et al. (2025). Their method augments the advantage with a clipped, gradient-detached entropy term,

encouraging the model to continue exploring uncertain reasoning actions.
Wang et al. refine the entropy-exploration idea by arguing that only a minority of tokens carry most of the useful exploration signal. Their "80/20" framing suggests that roughly the top 20% highest-entropy tokens function as reasoning forks, while the remaining tokens contribute less to trajectory diversity Wang et al. (2025). This motivates the central design choice in my method: rather than applying entropy regularization globally, we select the top 20% highest-entropy response tokens within each sequence and apply the entropy bonus only at those positions.
Overall, prior work motivates three design principles: RLVR can improve sampling efficiency but may narrow the policy; entropy is a useful signal for exploratory reasoning; and entropy pressure should likely be token-aware rather than uniform. My extension combines these ideas in the RLOO setting by adding a detached entropy bonus to the advantage only for high-entropy response tokens, with optional gating based on RLOO importance weights. This differs from prior entropy-shaping work by focusing on RLOO specifically and by testing whether importance weights can decide when high-entropy tokens should receive the bonus.

3 Method

This project modifies only the online RLOO stage of the default post-training pipeline. The SFT and IPO stages are held fixed, and all extension variants are implemented as changes to the RLOO update rule. The goal is to test whether exploration pressure should be applied uniformly across the generated response, or selectively at token positions where the policy is most uncertain.

3.1 Selective Entropy Advantage Shaping

The main extension replaces uniform sequence-level entropy pressure with token-selective entropy advantage shaping. Formatting tokens and predictable arithmetic continuations may not need additional exploration pressure, while high-entropy positions may correspond to reasoning forks: points where the model chooses an operation, forms an intermediate value, or branches into a different solution strategy. For each response y_i, we compute the entropy $H_{i,t}$ at every response-token position. Then, we select the top fraction ρ of response tokens with the highest entropy within that sequence. In all main experiments, we set $\rho = 0.2$. This produces a binary mask

$$m_{i,t} = \begin{cases} 1, & \text{if } t \text{ is in the top } \rho \text{ highest-entropy response tokens of } y_i, \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Thus, the entropy bonus is applied only to the highest-entropy token positions in each sequence. Instead of adding entropy as a separate global regularizer, the extension adds entropy directly to the RLOO advantage at selected token positions. The shaped token-level advantage is

$$A_{i,t}^{\text{shape}} = A_i + \alpha \,\text{stopgrad}(H_{i,t})\, m_{i,t}\, g_i, \tag{2}$$

where α is the entropy advantage scale and g_i is an optional importance-weight gate.

Because the shaped advantage varies across token positions, the policy-gradient loss becomes token-level:

$$\mathcal{L}_{\text{shape}} = -\frac{\sum_i \sum_{t \in \mathcal{T}_i} w_i A_{i,t}^{\text{shape}} \log \pi_\theta(y_{i,t} \mid x, y_{i,<t})}{\sum_i |\mathcal{T}_i|}. \tag{3}$$

The same sequence-level importance weight w_i is broadcast across all response tokens in completion i. This preserves the original RLOO importance correction while allowing the advantage term to vary by token.

3.2 Importance-Weight Gating

The second component of the extension tests whether the selective entropy bonus should be gated by the importance weight. The intuition is that if the likelihood of a sampled completion under the current policy differs meaningfully from its likelihood under the sampling policy, then the rollout may correspond to a more consequential policy update direction. In that case, high-entropy tokens in that rollout may be especially useful places to encourage exploration.

Let

$$\ell_i = \log \pi_\theta(y_i \mid x) - \log \mu(y_i \mid x) \tag{4}$$

denote the log importance ratio. Given a threshold τ, the gate is defined as

$$g_i(\tau) = 1\left[|\ell_i| > \log(1+\tau)\right]. \tag{5}$$

When $g_i(\tau) = 1$, the selected high-entropy tokens in completion i receive the entropy advantage bonus. When $g_i(\tau) = 0$, the update reduces to the ordinary RLOO advantage for that completion. I evaluate several gate settings, including loose gating, strict gating, and an ungated variant where $g_i = 1$ for all completions.

3.3 Hybrid Selective and Global Entropy Variant

In addition to pure selective advantage shaping, we also test a hybrid variant that keeps the original global entropy term while adding selective entropy shaping to the advantage. The hybrid loss is

$$\mathcal{L}_{\text{hybrid}} = \mathcal{L}_{\text{shape}} - \lambda \bar{H} + \beta_{\text{KL}} \mathcal{L}_{\text{KL}}. \tag{6}$$

This variant separates two possible roles of entropy. The token-selective term encourages exploration specifically at high-entropy reasoning forks, while the global entropy term prevents the overall policy distribution from becoming too sharp. Comparing the hybrid method against pure selective shaping helps distinguish whether performance changes come from targeted exploration pressure or from simply increasing the total amount of entropy regularization.

3.4 KL Regularization

All variants optionally retain the KL penalty from the milestone RLOO implementation. The KL term constrains the updated policy relative to a frozen reference model, reducing the risk that verifier optimization causes the model to drift too far from its initial language-modeling behavior. In the implementation, the KL penalty is computed at the response-token level using the current policy and reference policy token log-probabilities, then averaged over response tokens.

3.5 Summary of Method Variants

The extension studies four RLOO variants. The first is the milestone baseline, which applies global entropy regularization at the sequence level. The second applies selective entropy advantage shaping with a loose importance-weight gate. The third applies the same shaping with a stricter gate. The fourth removes the importance-weight gate entirely, applying the bonus to the top 20% highest-entropy response tokens in every rollout. Finally, the hybrid variant combines ungated selective advantage shaping with the original global entropy term. Across these variants, the core question is whether useful exploration in RLOO is better encouraged globally across all tokens or selectively at high-entropy reasoning forks.

4 Experimental Setup

4.1 Task and Model

All experiments are conducted on the Countdown arithmetic reasoning task from the default project. The base model for the project is Qwen2.5-0.5B. The full course pipeline consists of supervised fine-tuning (SFT), preference optimization using IPO, and online RL using RLOO. In this extension, the SFT and IPO stages are held fixed. All experiments modify only the RLOO update worker. This isolates the effect of the entropy-shaping intervention and ensures that differences in performance are attributable to the modified RLOO objective rather than to changes in the earlier training stages.

4.2 Baseline

The primary baseline is the milestone RLOO checkpoint. This baseline uses verifier-based RL with leave-one-out advantages, sequence-level importance weighting, KL regularization to a reference policy, and a global entropy bonus. The baseline milestone checkpoint achieves $\text{pass@1} = 49.6\%$ and $\text{pass@16} = 72.0\%$. These values are used as the main comparison point for all extension runs.

4.3 Extension Variants

All extension runs use `entropy_mode=advantage`. In this mode, the RLOO worker computes token-level policy entropy for each response token, selects the top 20% highest-entropy response tokens in each sequence, and adds a detached entropy bonus to the RLOO advantage at those positions. The token-selection fraction is fixed as $\rho = 0.2$. Thus, every extension run applies selective entropy shaping only to the highest-entropy minority of response tokens, rather than to all tokens.
We evaluate four main extension variants. The first two variants test importance-weight gating. The gate is active when $|\log w_i| > \log(1 + \tau)$, where w_i is the sequence-level importance weight for rollout i and τ is the gate threshold. The loose-gate run uses $\tau = 0.1, \alpha = 0.01$. The strict-gate run uses $\tau = 1.0, \alpha = 0.02$. The third variant removes importance-weight gating by setting the gate to be always active. In implementation, this is obtained with an effectively always-on threshold $\tau = -0.99$ so that selected high-entropy tokens receive the entropy advantage bonus in every rollout. This run uses top-20% token selection but no global entropy term. The fourth variant is a hybrid method that also uses the ungated top-20% entropy advantage bonus, while retaining the milestone global entropy term with $\lambda = 0.001$. This hybrid run tests whether targeted token-level exploration and global entropy regularization provide complementary benefits.

Method	**Entropy Mode**	**IW Gate τ**	α	**Global Entropy λ**
RLOO baseline	loss	—	—	0.001
Advantage entropy, loose IW gate	advantage	0.1	0.01	0
Advantage entropy, strict IW gate	advantage	1.0	0.02	0
Advantage entropy, no IW gate	advantage	-0.99	0.02	0
Advantage entropy + global entropy	advantage	-0.99	0.02	0.001

Table 1: RLOO variants evaluated in the extension. All non-baseline methods use top-20% token-level entropy selection with $\rho = 0.2$. The no-gate and hybrid runs use an always-active gate.

4.4 Training Details

The RLOO update worker receives tokenized completions, attention masks, response-token masks, verifier rewards, and optionally sampling log-probabilities. The model computes token log-probabilities only over response tokens. Rewards are reshaped into groups, and for each rollout the leave-one-out baseline is computed from the rewards of the other completions in the same group. The resulting advantage is then used in the policy-gradient loss.

For extension runs, the worker computes token-level entropy from the current policy logits:

$$H_{i,t} = -\sum_{v \in \mathcal{V}} \pi_\theta(v \mid x_i, y_{i,<t}) \log \pi_\theta(v \mid x_i, y_{i,<t}). \tag{7}$$

The implementation then constructs a per-sequence high-entropy mask by selecting the top $\rho = 0.2$ fraction of response tokens according to $H_{i,t}$. The entropy bonus is detached before being added to the advantage:

$$A_{i,t}^{\text{shape}} = A_i + \alpha \,\text{stopgrad}(H_{i,t}) m_{i,t} g_i. \tag{8}$$

The shaped advantage is optimized with a token-level policy-gradient loss. Sequence-level importance weights are broadcast across response tokens. When sampling log-probabilities are available, the importance weight is computed as

$$w_i = \exp\left(\log \pi_\theta(y_i \mid x) - \log \mu(y_i \mid x)\right), \tag{9}$$

with clipping for numerical stability.

All variants use the same optimizer structure as the RLOO worker: AdamW, gradient clipping, and a constant learning-rate schedule. The worker supports gradient accumulation and saves both model and optimizer/scheduler checkpoints. The extension experiments are trained for 100 RLOO update steps on the 0.5B model. Unless otherwise stated, hyperparameters other than the entropy mode, entropy coefficient, advantage entropy scale, and IW gate threshold are held fixed across runs.

4.5 Evaluation

The main evaluation metric is pass@k on held-out Countdown prompts. For each prompt, the model samples k completions and the verifier checks whether at least one completion is correct. The reported evaluation uses $N = 50$ held-out prompts and up to $k = 16$ samples per prompt. We report both pass@1 and pass@16. Pass@1 measures single-sample correctness and reflects how likely the model is to produce a correct answer immediately. Pass@16 measures multi-sample correctness and is more sensitive to whether the model preserves enough diversity to find a valid equation across multiple attempts.
A method that sharpens the policy may improve pass@1 while failing to improve pass@16. Conversely, a method that preserves useful exploration may improve pass@16 even if pass@1 does not increase. Therefore, I treat pass@16 as the primary metric for testing whether selective entropy shaping improves multi-sample reasoning, while pass@1 is used to measure whether the intervention sacrifices single-sample accuracy.

4.6 Training Metrics

In addition to pass@k, we logged training metrics from the RLOO worker to diagnose how each entropy variant changes the update dynamics. The most important diagnostics for this extension are:

- **Mean policy entropy**, which measures whether the policy is becoming sharper during RL training.
- **Reward mean**, which measures whether verifier reward improves during training.
- **Importance weight mean**, which tracks the scale of the off-policy correction.
- **Entropy-bonus active fraction**, which measures how often the IW gate activates the selective entropy bonus.
- **Token entropy bonus fraction**, which verifies that only the intended top-20% response tokens receive the bonus.

These diagnostics are used to interpret the main pass@k results. In particular, the strict IW-gated run activates the entropy bonus on only about half of the token positions, while the ungated run keeps the selective entropy bonus active throughout training. This difference is used to explain why ungated selective shaping improves pass@16 more reliably than strict IW gating.

5 Results

Method	IW Gate τ	pass@1	pass@16
RLOO baseline (global entropy)	—	49.6%	72.0%
Advantage entropy, loose IW gate	0.1	41.5%	66.0%
Advantage entropy, strict IW gate	1.0	39.5%	72.0%
Advantage entropy, no IW gate	-0.99	48.0%	**74.0%**
Advantage entropy + global entropy	-0.99	**52.0%**	72.0%

Table 2: Countdown evaluation on 50 held-out prompts with 16 samples per prompt. All extension runs modify only the RLOO update. The ungated top-20% entropy-advantage method gives the best pass@16, while the hybrid method gives the best pass@1.

5.1 Quantitative Evaluation

In Figure 2, each curve shows the fraction of prompts solved when allowing up to k independent samples. Curves that rise faster and stay higher at large k indicate policies that produce more diverse correct solutions, not just a sharper single answer. The ungated advantage-entropy run leads at high k, while the hybrid run is strongest at low k, illustrating a sharpening–diversity tradeoff.

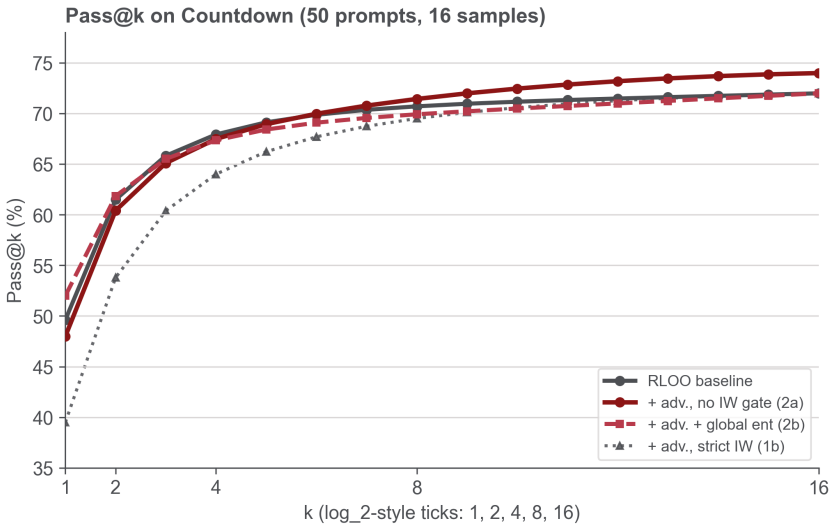

Figure 2: Pass@k plot of different variants.

5.2 Qualitative Analysis

The sampled generations reveal a useful distinction between the different entropy-shaping variants. The milestone RLOO model often produces plausible arithmetic search traces and, when it succeeds, tends to find simple additive or subtractive identities. However, its failures are frequently caused by repetition, self-correction loops, or answers that use only a subset of the provided numbers. This is visible in the raw evaluation outputs: for example, several completions repeatedly propose nearby arithmetic expressions, verify them, then continue generating additional attempts rather than stopping after a valid block. The same qualitative failure mode appears across the extension runs, suggesting that the main bottleneck is not only arithmetic ability but also controlled termination and consistency between the reasoning trace and the final answer.

The most informative comparison is between loose and tight weight gating. The loose gate used $\tau = 0.1$, with $\alpha = 0.01$. Since the gate was active for roughly 93% of eligible positions, it was close to an always-on entropy-advantage bonus. It showed more exploratory behavior, but this exploration is often unproductive: many responses cycle through repeated "final attempts," revisit the same incorrect arithmetic path, or emit multiple answer blocks. A representative failure repeatedly tries variants of the same arithmetic decomposition for a target and eventually apologizes or contradicts itself rather than converging to a clean final expression.

Comparing with the tightened IW gate ($\tau = 1.0$), the entropy-advantage scale was increased to $\alpha = 0.02$. This made the intervention more selective: the entropy bonus fired only when the rollout's importance weight deviated substantially from the sampling policy. These samples appeared less uniformly exploratory than with loose gating. More often than not, the samples reach partially correct reasoning, and the score distribution shifts toward partial-credit completions. This is consistent with the raw evaluation counts: loose gate has more zero-reward samples, while tighter has fewer zero-reward samples but more partial-credit samples. In other words, the stricter gate seems to reduce some completely off-track generations, but it does not reliably convert those attempts into fully valid Countdown equations.

Comparing the run with the tight gate to no gate helps explain why gating did not become the best design choice. The no gate run applies the top-20% entropy-token bonus without using the importance-weight gate, so high-entropy "fork" tokens receive shaping consistently. Its generations contain the same broad family of errors, but it produces more fully correct completions across the 16 samples. Qualitatively, this suggests that the useful exploration signal may come primarily from selecting uncertain token positions, not from additionally filtering by rollout-level importance weight. Since it is applied at the rollout level, the gate may suppress entropy shaping even when a particular token is a meaningful local reasoning fork. Conversely, the loose gate was so frequently active that it largely failed to test the intended selectivity. The tight gate moved in the right direction conceptually, but the threshold appears too strict for pass@k performance.

6 Discussion

The results suggest that selective entropy shaping is useful, but only when the entropy signal is applied consistently to genuinely uncertain token positions. The strongest pass@16 result comes from the ungated top-20% entropy-advantage run, which improves over the milestone RLOO baseline from 72.0% to 74.0%. This supports the core hypothesis that not all tokens require entropy pressure: arithmetic reasoning benefits more from preserving uncertainty at high-entropy decision points than from applying a uniform entropy bonus across the whole response.

At the same time, the importance-weight gate did not behave as expected. The loose gate was active for most eligible positions, so it behaved similarly to an always-on entropy bonus but without producing the same pass@16 gains as the ungated top-20% run. The stricter gate made the intervention more selective, but it reduced pass@1 and only matched the baseline at pass@16. This suggests that sequence-level importance-weight deviation may be too coarse a signal for deciding when to apply token-level exploration pressure. A rollout can have a small overall importance-weight deviation while still containing local high-entropy tokens that are important reasoning forks. Conversely, a large rollout-level deviation does not guarantee that every selected token in that rollout is useful for exploration.

The hybrid selective-plus-global entropy variant further highlights a tradeoff between single-sample accuracy and multi-sample diversity. It achieves the best pass@1 result, suggesting that retaining global entropy regularization can help stabilize or sharpen the model's most likely output. However, it does not improve pass@16 over the baseline, while the ungated selective method does. This indicates that the hybrid method may improve immediate correctness without increasing the diversity of successful samples. For Countdown, where many prompts benefit from multiple independent attempts, pass@16 is especially important, so the ungated selective method is the most promising variant for the extension's main goal.

There are several limitations. The evaluation uses only 50 held-out prompts and a single training seed, so the reported differences should be interpreted as preliminary rather than definitive. The runs also use fixed hyperparameters, including a fixed top-20% token fraction and fixed entropy-advantage scales. In addition, the qualitative samples show that many failures are caused by formatting, repeated answer blocks, or failure to stop after a correct answer. These errors are not directly solved by entropy shaping, so future work should combine token-selective exploration with stronger stopping or answer-format constraints.

7 Conclusion

In conclusion, this project explored whether entropy collapse in RLOO fine-tuning can be mitigated by applying entropy pressure selectively to high-entropy response tokens. We implemented token-level entropy advantage shaping for the Countdown task and compared loose IW gating, strict IW gating, no IW gating, and a hybrid selective-plus-global entropy method against the milestone RLOO baseline.

The main finding is that selective entropy shaping can modestly improve multi-sample reasoning performance when applied without importance-weight gating. The ungated top-20% entropy-advantage run achieves the best pass@16 score, improving from 72.0% to 74.0%. In contrast, importance-weight gating does not improve performance: the loose gate is nearly always active and therefore not meaningfully selective, while the strict gate suppresses too much of the useful exploration signal. The hybrid method achieves the best pass@1 score at 52.0%, but does not improve pass@16, suggesting that it improves single-sample accuracy more than multi-sample diversity.

Overall, these results support the idea that high-entropy tokens are useful targets for exploration in RLVR, but they do not support rollout-level IW gating as the best mechanism for deciding when to apply the entropy bonus. Future work should test intermediate gate thresholds, adaptive entropy coefficients, token-level gating rules, larger evaluation sets, and multiple random seeds. More broadly, this extension suggests that exploration in language-model RL should be treated as a token-level design problem rather than only as a sequence-level regularization problem.

8 Team Contributions

Syed Ashal Ali was the team member and produced the whole project. Huge credit to Shengqu Cai for being a great TA and advisor for the project!

Changes from Proposal Adaptive scheduling mentioned in the proposal was not implemented and remains as future work.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

Zhou Cheng et al. 2025. Reasoning with Exploration: An Entropy Perspective. *arXiv preprint arXiv:2506.14758* (2025). arXiv:2506.14758

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948* (2025).

Zichen Wang et al. 2025. Beyond the 80/20 Rule: High-Entropy Minority Tokens Drive Effective Reinforcement Learning for LLM Reasoning. *arXiv preprint arXiv:2506.01939* (2025). arXiv:2506.01939

Yang Yue et al. 2025. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? *arXiv preprint arXiv:2504.13837* (2025). arXiv:2504.13837