ADDRESSING PERFORMANCE SATURATION FOR LLM RL VIA PRECISE ENTROPY CURVE CONTROL

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has enabled complex reasoning abilities in large language models (LLMs). However, most RL algorithms suffer from performance saturation, preventing continued gains as RL training scales. This problem can be characterized by the collapse of entropy, a key diagnostic for exploration in RL. Current approaches focus on preventing entropy collapse through regularization or clipping. However, their resulting entropy curves often exhibit instability in the long term, which hinders performance gains. In this paper, we introduce *Entrocraft*, a simple rejection-sampling approach that realizes *user-customized entropy schedule* by biasing the advantage distributions. Entrocraft requires no objective regularization and is advantage-estimator-agnostic. Theoretically, we relate per-step entropy change to the advantage distribution under minimal assumptions. Our analysis explains the behavior of existing RL and entropy-preserving methods and enables a systematic study of entropy schedules. It shows that linear annealing, which starts high and decays to a slightly lower target, achieves strong performance. Empirically, Entrocraft addresses performance saturation, significantly improving generalization, output diversity, and long-term training. It enables a 4B model to outperform an 8B baseline, sustains improvement for up to 400K training samples, and significantly raises pass@K over the baseline.[1,2]

1 INTRODUCTION

Reinforcement learning (RL) has become the dominant approach for aligning large language models (LLMs) with human preference and realizing multi-step reasoning ability (Schulman et al., 2017; Bai et al., 2022; Ouyang et al., 2022). Despite these successes, many RL algorithms underperform with respect to the expected performance improvement. As training scales, performance saturates earlier than expected, resulting in the failure to translate additional data and compute to sustained improvements (Hou et al., 2024; Park et al., 2025; Beukman et al., 2026). A core reason behind this saturation is the collapse of the exploration–exploitation balance, where the LLM over-commits to a narrow region of solutions and stops exploring alternative reasoning trajectories (Cui et al., 2025; Yue et al., 2025; Li et al., 2026). Empirically, this phenomenon is well captured by entropy dynamics: the frequently observed entropy collapse corresponds to a shrinking exploration ability during RL.

Recent efforts have resulted in several entropy-preserving techniques to prevent entropy drop during RL. These techniques are based on loss regularization (Williams & Peng, 1991), clipping (Yu et al., 2025; Cui et al., 2025; Wang et al., 2026a), or positive-negative decoupling (Zhu et al., 2025; Yang et al., 2025b). While effectively increasing entropy, the entropy curves during RL training are still *coarsely* controlled. Entropy can drift too high after a few steps, which in turn makes RL unstable and thus hinders sustained performance gains. Furthermore, these methods typically control entropy indirectly through the loss or update rule, making it difficult to prescribe an explicit entropy schedule over long training horizons. These drawbacks are particularly severe in long-term RL training.

To address these challenges, we propose *Entrocraft*, a method for precise control over the entropy curve that allows entropy schedules to be user-customized. Fig. 1 summarizes our method, the entropy curve control technique, and experimentally demonstrated improvements. We begin with

[1]Code repository URL withheld for anonymous review.
[2]Interactive demo URL withheld for anonymous review.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

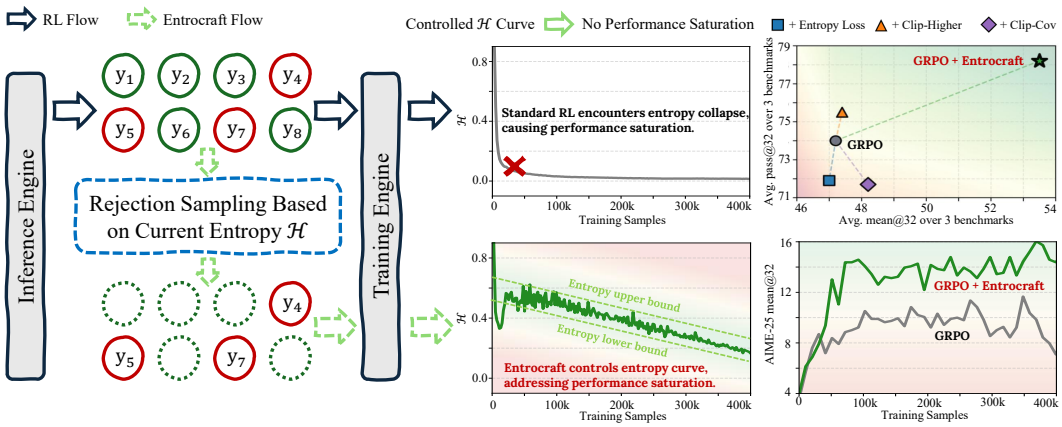


Figure 1: Overview of Entrocraft: Use of entropy-guided rejection sampling to filter rollouts against a target entropy, enabling precise control over the entropy curve throughout RL training. This control addresses performance saturation, a key obstacle to scaling RL. We show that a linear annealing schedule performs best, improving generalization, output diversity, and sustained training.

an LLM-oriented theoretical analysis of entropy change based on realistic policy assumptions. We highlight that entropy changes are negatively related to the advantage, and high model confidence amplifies such entropy changes. Based on our theoretical results, we design a simple rejection sampling technique to filter out positive/ negative-advantage rollout samples when entropy is lower/ higher than a threshold, biasing the advantage distribution towards the entropy-increasing/decreasing region. Since rejection sampling directly modifies the advantage distribution, it is able to move the entropy to target values within very few steps, enabling accurate crafting of entropy curves. Our method requires no entropy regularization and can be deployed as a drop-in to existing RL algorithms. Precise control opens a question that the field has not yet been studied thoroughly: *what entropy schedule is the best?* Comparing across schedule families, we find that a linear annealing schedule performs best.

The main contributions of this paper are as follows:

- We provide rigorous theoretical results on entropy changes grounded in realistic LLM-based policy assumptions. Entropy changes are negatively related to the advantages and high model confidence amplifies such changes.
- We introduce a lightweight controller based on rejection sampling for entropy schedules in LLM RL. Unlike entropy regularization, clipping, or decoupling methods, Entrocraft requires no additional regularization and is advantage-estimator-agnostic. Entrocraft can tailor the entropy curve to user-specified entropy schedules, which is key to addressing performance saturation.
- Extensive experiments demonstrate the effectiveness of Entrocraft in improving generalization (a 4B model surpasses an 8B baseline), increase in solution coverage (AIME-25 pass@32 is 50% higher than baseline), and extending the training horizon (sustaining improvement for up to 4× longer before plateauing as training scales).

2 Preliminaries

2.1 Reinforcement Learning for LLMs

In a standard policy-gradient RL framework like Group Relative Policy Optimization (GRPO) (Shao et al., 2024) or Group Sequence Policy Optimization (GSPO) (Zheng et al., 2025b), the language model (or actor) we aim to train is denoted as a $\boldsymbol{\theta}$-parameterized distribution (or policy) $\pi_{\boldsymbol{\theta}}$. The direct output of language models is a softmax distribution over the entire vocabulary $\mathbb{V}$, interpreted as next-token probabilities: $\boldsymbol{p}_t = \pi_{\boldsymbol{\theta}}(\cdot|x, y_{<t})$. Each new token y_t is drawn from $\boldsymbol{p}_t$.

Each RL step consists of rollout generation, advantage estimation, and policy update, allowing the model to explore different potential answers and learn from environment feedback. For a single question (or prompt) x, rollout generation samples a set of G responses $\{y_i\}_{i=1}^{G}$ from an old checkpoint $\pi_{\boldsymbol{\theta}_{\text{sampler}}}(\cdot|x)$. The following PPO-style objective is used by many recent RL algorithms:

$$\mathcal{J}(\boldsymbol{\theta}) = \frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|y_i|}\min\Big[r_I(t)\cdot\hat{A}(x,y_i), \text{clip}\left(r_I(t), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{A}(x,y_i)\Big], \tag{1}$$

where $r_I(t) = \frac{\pi_{\boldsymbol{\theta}}(y_{i,t}|x,y_{i,<t})}{\pi_{\boldsymbol{\theta}_{\text{sampler}}}(y_{i,t}|x,y_{i,<t})}$ is the importance sampling ratio, and $\hat{A}$ is the estimated advantage.

Our theoretical analysis is based on a simplified policy-gradient objective that does not consider clipping or importance sampling: $\mathcal{J}(\boldsymbol{\theta}) = \mathbb{E}_{y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}\hat{A}(x,y)$, and thus the per-step policy update is:

$$\Delta\boldsymbol{\theta} = \eta\cdot\nabla_{\boldsymbol{\theta}}\mathcal{J}(\boldsymbol{\theta}) = \eta\cdot\mathbb{E}_{y\sim\pi_{\boldsymbol{\theta}}(\cdot|x)}[\hat{A}(x,y)\cdot\nabla_{\boldsymbol{\theta}}\log\pi_{\boldsymbol{\theta}}(y|x)], \tag{2}$$

where η is the learning rate.

2.2 Entropy of LLMs

The predictive entropy of LLMs provides a principled measurement of model uncertainty and serves as an indicator of response diversity and exploration capability. For a single question x and answer y, the aggregated entropy is computed as: $\mathcal{H} = -\frac{1}{|y|}\sum_{t=1}^{|y|}\sum_{i=1}^{|\mathbb{V}|}p_{t,i}\log p_{t,i}$. The expected entropy, averaged over all prompts in a batch and their corresponding rollout samples, serves as an indicator of how LLMs' exploration capability evolves during RL. This evolution is known as entropy dynamics (Ren & Sutherland, 2025; Wang et al., 2026a). In this paper, we primarily study entropy change during RL updates:

$$\Delta\mathcal{H} = \mathcal{H}(\boldsymbol{p}+\delta\boldsymbol{p}) - \mathcal{H}(\boldsymbol{p}), \tag{3}$$

to enable accurate and per-step entropy control.

3 Theoretical Analysis: How Entropy Evolves during LLM RL

This section presents theoretical results on entropy changes during RL training. We use these results to interpret the entropy dynamics of existing RL algorithms, particularly in long-running scenarios. Our analysis extends prior work (Liu, 2025; Cui et al., 2025; Yang et al., 2025b; Wang et al., 2026a; Shen, 2025) to a more realistic setting that does not require the actor to follow a tabular softmax policy[3]. The resulting bounds are direct and easy to interpret, avoiding the complicated covariance and expectation terms that appear in prior analyses (Liu, 2025; Wang et al., 2026a).

3.1 From Advantages to Entropy Changes

Our analysis addresses two fundamental questions: (i) What is the sign of entropy change $\Delta\mathcal{H}$, and (ii) what is its magnitude? These questions help us predict entropy change at each RL step, revealing how advantages affect entropy dynamics. To obtain exact analytical results, we make minimal assumptions about the actor policy and advantage distribution, requiring that the learning rate be sufficiently low, as stated in Assumption 1.

Assumption 1 (Proximity of Policy Updates) *We assume the learning rate η in Eq. (2) is small enough that the Taylor expansion approximation of policy probability updates holds (i.e., $\|\delta\boldsymbol{p}\|_2^2 \ll \|\delta\boldsymbol{p}\|_1$). This is a standard assumption in continuous optimization, and is satisfied in practice by modern adaptive optimizers like Adam (Kingma & Ba, 2014) with typical learning rates (e.g., $10^{-6}\le\eta\le10^{-4}$).*

Theorem 1 (Token-Level Entropy Change in LLMs) *Consider a single policy-gradient update step of the form in Eq. (2). Let p_k be the probability that token k is sampled during rollout generation.*

[3]The tabular softmax policy assumes $\boldsymbol{\theta} = \boldsymbol{z}$, where logits are model parameters. However, in realistic LLM settings, the logits $\boldsymbol{z}$ are the functions of model parameters $\boldsymbol{\theta}$, and even a simple MLP module would make this assumption invalid.

Then, up to a first-order approximation, the sign entropy change $\Delta\mathcal{H}$ (Eq. (3)) triggered by token k is opposite to that of its estimated advantage $\hat{A}_k$:

$$\hat{A}_k \cdot \Delta\mathcal{H} \leq 0, \quad \textit{whenever} \quad p_k > \prod_{i \in \mathbb{V}\setminus\{k\}} p_i^{-\frac{\delta p_i}{\delta p_k}},$$

where δp_i is the probability change at this RL step.

Theorem 2 (Sequence-Level Entropy Change in LLMs) *Under Assumption 1 and Appendix B.1, consider a single policy-gradient update step of the form in Eq. (2), and assume that all tokens share the same outcome reward. Let $p_{t,i} = \pi_{\boldsymbol{\theta}}(y_t = i|x, y_{<t})$ be the probability that i is sampled as the t-th token in the sequence. Up to a first-order approximation, the sign of the entropy change $\Delta\mathcal{H}$ (Eq. (3)) triggered by response y is opposite to that of the estimated advantage $\hat{A}(x, y)$:*

$$\hat{A}(x, y) \cdot \Delta\mathcal{H} \leq 0, \quad \textit{whenever} \quad \pi_{\boldsymbol{\theta}}(y|x) \geq \prod_{t=1}^{|y|} \prod_{i \in \mathbb{V}\setminus\{y_t\}} p_{t,i}^{-\frac{\delta p_{t,i}}{\delta p_{t,y_t}}},$$

where $\delta p_{t,i}$ is the probability change at this RL step.

We provide theoretical guaranties in Theorem 1 and 2 for token-level entropy and sequence-level entropy[4] and outline their proofs in Appendix B. Intuitively, both theorems state that entropy changes are negatively related to the advantage, provided the probability of rollout samples is high enough to be above a baseline constant:

$$\text{Entropy Change} \propto -\text{Adv.} \times (\text{Log Likelihood of Rollouts} - \text{Output Space Baseline}), \tag{4}$$

where the *output space baseline* is: $-\sum_{t=1}^{|y|} \sum_{i \in \mathbb{V}\setminus\{y_t\}} \frac{\delta \pi_{\boldsymbol{\theta}}(y_t = i|x, y_{<t})}{\delta \pi_{\boldsymbol{\theta}}(y_t|x, y_{<t})} \log \pi_{\boldsymbol{\theta}}(y_t = i|x, y_{<t})$.

Theorem 2 suggests that positive-advantage rollout samples lead to an entropy drop if the model confidence $\pi_{\boldsymbol{\theta}}(y|x)$ is above the output space baseline. We further provide empirical evidence to support this condition in Fig. 2a, where we compare the log likelihoods (confidence) and output space baselines of training `Qwen3-4B-Base` under positive (RAFT++ (Xiong et al., 2025)), zero-mean (GRPO (Shao et al., 2024)), and negative (NSR (Zhu et al., 2025)) advantage estimators. Fig. 2a shows that the log likelihood is significantly higher than the output space baseline in all cases, verifying the condition for $\hat{A}(x, y) \cdot \Delta\mathcal{H} \leq 0$ to hold.

Entropy collapse/ explosion in RL is a predictable consequence of advantage-weighted updates. Our results show that positive-advantage updates tend to reduce entropy, while negative-advantage updates tend to increase it. As a result, entropy collapse becomes the default, once training is dominated by positive advantages. This explanation also justifies the "accuracy-entropy tradeoff" (Cui et al., 2025) in standard RL algorithms, where accuracy increase leads to negative entropy changes. However, the theoretical results also suggest that entropy changes are not directly related to the model performance. It is possible to maintain entropy while still improving rewards if the algorithms selectively choose which advantage regions contribute to the policy gradients.

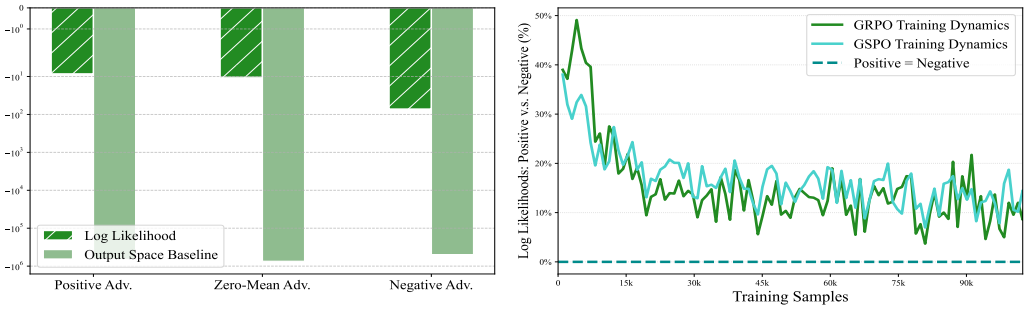
Takeaway for Entropy Analysis

We prove the relationship between advantages and entropy changes for LLM-based policies. Entropy changes are negatively related to the advantage, and high model confidence will amplify such entropy changes.

3.2 Interpreting the Entropy Dynamics of Existing Methods

The entropy dynamics of RL algorithms are important indicators of training stability and exploration-exploitation balance. Our theoretical results reveal a clear relationship between entropy change and advantage, explaining the entropy behavior of existing RL algorithms and their performance limitations. We discuss why existing RL algorithms exhibit specific entropy dynamics in the following discussion.

[4]NOTE: These theoretical results are based on the entropy computed from the learner policy $\pi_{\boldsymbol{\theta}}$



(a) All advantage estimators lead to sufficiently high confidence (b) Positive-sample confidence is consistently higher

Figure 2: Empirical justification for entropy analysis. (a) All advantage estimators lead to sufficiently large log likelihoods on average, so that the model confidence condition in Theorem 2 always holds. (b) The log likelihoods of positive-advantage samples are always larger than those of negative samples, justifying why GRPO/GSPO encounters entropy collapse even with normalized advantages.

Standard RL Algorithms. Our theoretical results imply a categorization of existing RL algorithms that do not explicitly consider entropy. There are three types of algorithms based on their advantage statistics: (i) In positive-advantage RL like RAFT (Dong et al., 2023) and RAFT++ (Xiong et al., 2025), most RL steps lead to entropy drop; (ii) in negative-advantage RL like NSR (Zhu et al., 2025), most RL steps lead to entropy increase; (iii) in zero-mean-advantage RL like GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025b), empirical results show that the entropy still tends to decrease. We interpret this phenomenon by comparing the training dynamics of `Qwen3-4B-Base`, and find that this is due to the overconfidence of positive samples, as shown in Fig. 2b. Models are consistently more confident in the positive samples, allowing the negative entropy changes to dominate the training dynamics.

Clipping. Many recent efforts leverage the clipping technique to address entropy collapse, including DAPO (Yu et al., 2025), ADAPO (Petrenko et al., 2026), Clip$_{\mathcal{B}/\mathcal{V}}$ (Wang et al., 2026a), and Clip-Cov (Cui et al., 2025). The mechanism behind clipping is the removal of high-advantage and/or high-confidence tokens, which biases the advantage distribution toward 0-mean. Our theory explains why this works: it reduces expected $|\Delta\mathcal{H}|$ and thereby alleviates entropy drop.

Positive-Negative Decoupled RL. Recent studies also propose decoupled objectives for positive (correct) and negative (incorrect) rollout samples, respectively, inspired by the empirical finding that negative-only RL increases entropy (Zhu et al., 2025). This approach is well explained by our theoretical framework, as it explicitly enforces the sign of advantages. For example, W-Reinforce (Zhu et al., 2025) modifies the coefficients of positive RL: $\mathcal{J}_{\text{W-Reinforce}} = \lambda \cdot \mathcal{J}_{\text{pos}} - \mathcal{J}_{\text{neg}}$, to weaken the entropy drop triggered by the positive objective; EntroPIC (Yang et al., 2025b) further makes the coefficients adjustable: $\mathcal{J}_{\text{EntroPIC}} = (1+\alpha(\mathcal{H})) \cdot \mathcal{J}_{\text{pos}} - (1-\alpha(\mathcal{H})) \cdot \mathcal{J}_{\text{neg}}$, and eventually converges to a targeted entropy value.

4 Methodology

In this section, we introduce our entropy-control framework (Entrocraft), which builds upon a simple rejection-sampling filter. We begin with rejection sampling in rollout generation (Section 4.1), and then introduce the dynamic rejection sampling filter for entropy control (Section 4.2). Finally, we discuss our insights on entropy curve annealing, highlighting that the shape entropy curves in RL can be tuned just like learning-rate schedules (Section 4.3).

4.1 Rejection Sampling as a Simple Entropy Controller

Our theoretical results suggest that entropy change is not directly tied to model performance. Entropy can remain stable or even increase while training accuracy improves, as long as the positive-advantage

rollout samples are filtered out and no longer contribute to the policy gradients. This behavior can be realized by rejection sampling.

Our key observation is that entropy collapse/ explosion is a consequence of uncontrolled gradient updates. From the theoretical results in Section 3.1, the subset of rollouts contributing to the gradient determines whether an update is entropy-decreasing or entropy-increasing. The sign of $\Delta\mathcal{H}$ can be controlled by selecting which rollouts enter the policy gradient. Therefore, rather than developing new RL objectives or adding an auxiliary entropy loss, we find that a simple rejection-sampling filter at rollout generation suffices to precisely control entropy changes.

For example, to increase entropy, we apply rejection sampling to retain only the negative subset: $\mathcal{S}_x = \{y_i | \hat{A}(x, y_i) < 0\}$, and the RL training objective becomes:

$$\mathcal{J}_{\text{rej}}(\boldsymbol{\theta}) = \frac{1}{|\mathcal{S}_x|} \sum_{y \in \mathcal{S}_x} \sum_{t=1}^{|y|} \min\Big[r_I(t) \cdot \hat{A}(x, y), \text{clip}\left(r_I(t), 1 - \epsilon_{\text{low}}, 1 + \epsilon_{\text{high}}\right) \hat{A}(x, y)\Big], \tag{5}$$

with the only difference from the standard RL objective (Eq. (1)) highlighted in color. Rejection sampling provides a simple, objective-agnostic entropy control knob, retaining the strengths of existing RL algorithms while eliminating the risk of entropy collapse or explosion. Since it directly modifies the advantage distribution, the filter is responsive enough to move entropy to target values within a few steps, enabling accurate crafting of entropy curves shown later in Section 4.3. The cost is comparable to or lower than standard RL, as only accepted samples contribute to the gradient computation. This can also be monitored by effective rollout batch sizes, as shown in Appendix C.3. We also include a comparison with token-level rejection sampling implementation in Appendix C.8.

4.2 Stabilizing and Crafting Entropy Dynamics

Entropy dynamics have been used to monitor the training stability of RL (Zheng et al., 2025a). In a stable training run, the entropy curve should be within a reasonable range, neither low enough to trigger performance saturation (Cui et al., 2025; Lu et al., 2024; Kim et al., 2026), nor high enough to cause numerical overflow (Zheng et al., 2025a).

To realize stable entropy dynamics, we apply the rejection sampling filter to dynamically encourage or discourage the exploration of LLMs. The acceptance probability of rejection sampling depends on the current batch entropy $\overline{\mathcal{H}}$ against a target range $(h_{\text{low}}, h_{\text{high}})$, in which we use an *entropy out-of-range indicator*: $m = \mathbb{I}(\overline{\mathcal{H}} > h_{\text{high}}) - \mathbb{I}(\overline{\mathcal{H}} < h_{\text{low}})$ to measure the direction of entropy drift. When entropy is too low, the filter rejects most high-advantage rollouts while retaining lower- and negative-advantage ones. When entropy is too high, the filter retains positive-advantage rollouts and rejects most negative samples, steering RL updates toward entropy reduction. The full procedure is given in Algorithm 1.

Entrocraft provides a plug-and-play entropy control framework, applicable to all policy-gradient methods. It treats the entropy curve as a controllable training hyperparameter in the same spirit as a learning-rate schedule, making the training dynamics of RL stable and customizable.

4.3 Precise Entropy Curve Control and Annealing Schedules

The Long-Term RL Challenge. A growing body of work has shown that RL tends to sharpen the base policy around existing solutions rather than discover new ones (Karan & Du, 2025; Zhang et al., 2025; Yue et al., 2025; Zhao et al., 2025), a behavior consistent with entropy collapse observed empirically. Once the policy becomes slightly more confident in a small subset of correct solutions, such solutions will be sampled more often, which further increases their likelihood. The problem is exacerbated in long-term RL. As training rewards rise, the advantage distribution becomes increasingly imbalanced and heavy-tailed, leaving fewer negative-advantage samples to counteract the drift. By Theorems 1 and 2, these positive-advantage and high-likelihood solutions are mostly entropy-decreasing. The self-reinforcing feedback loop would lead to entropy collapse just within a few steps.

A Constant Entropy Target Is Not Enough. This fragility motivated us to stress-test Entrocraft under long-term RL training. As demonstrated in Appendix C.6, Entrocraft with a slightly higher

Algorithm 1: Entrocraft

Inputs: Question x, original rollout samples $\{y_i\}_{i=1}^{G}$, current policy $\pi_{\boldsymbol{\theta}}$, advantage estimator $\hat{A}$, and the entropy range $(h_{\text{low}}, h_{\text{high}})$.
Outputs: The actual rollout samples used for the RL update $\mathcal{S}_x$.

$\mathcal{S}_x \leftarrow \emptyset$;
Compute the current entropy $\overline{\mathcal{H}}$;
$m \leftarrow \mathbb{I}(\overline{\mathcal{H}} > h_{\text{high}}) - \mathbb{I}(\overline{\mathcal{H}} < h_{\text{low}})$; `/* entropy out-of-range indicator */`

for $i = 1 \,..\, G$ **do**
  if $m \cdot \hat{A}(x, y_i) \geq 0$ **then**
    $\mathcal{S}_x \leftarrow \mathcal{S}_x \cup \{y_i\}$; `/* rejection sampling */`
  end
end
Return $\mathcal{S}_x$.

constant entropy target would become unstable and fluctuate a lot, eventually. We attribute this instability to the imbalance of rollouts, which makes the negative samples so scarce in the long term that Entrocraft's entropy-increasing steps (rejecting all positive samples) rely on very few samples.

Curve Control with Annealing Schedules. To address this, we propose to anneal the entropy curves as training proceeds. For example, we set the initial entropy target to be around 0.6, and gradually lower this target toward 0.2 during RL training. This can stabilize the training dynamics as it reduces the unstable entropy-increasing steps in the later phase of RL. We compare different annealing schemes in Section C.7, finding that the simple linear-decaying entropy curve achieves the best performance. This annealing design converts entropy in RL from a passive training diagnostic into a controllable hyperparameter, extending the toolkit for tuning RL performance to any policy-gradient method.

5 Experiments

We present empirical results to demonstrate the effectiveness of the proposed Entrocraft algorithm. Specifically, we present a comprehensive benchmark comparison in Section 5.2, elaborate on the entropy curve annealing schemes in Section C.7, and discuss the case of long-term RL in Section 5.3.

5.1 Settings

Data and Models. Experiments described in this section focus on math reasoning tasks, using Numina-Math (Beeching et al., 2024) (440K questions in total) as the training set. We hold out a 100K subset for general RL experiments, and the full-size dataset is used for long-term RL experiments. We primarily demonstrate the RL results using `Qwen3-4B-Base` (Yang et al., 2025a), as well as a comparison with larger models (`Qwen3-8B-Base` and `Qwen3-14B-Base`) and models from different model families (`Llama-3.1-8B-Instruct`) (Grattafiori et al., 2024).

RL Algorithms and Baselines. We primarily use the proposed Entrocraft algorithms to augment GRPO (Shao et al., 2024) and GSPO (Zheng et al., 2025b). In comparison with Entrocraft, we also implement other entropy-preserving methods on top of these RL algorithms, including loss-regularization (entropy loss), clipping (Clip-Higher (Yu et al., 2025) and Clip-Cov (Cui et al., 2025)), and positive-negative decoupled RL (W-Reinforce (Zhu et al., 2025) and EntroPIC (Yang et al., 2025b)). The implementation follows the standard verl framework (Sheng et al., 2025). Other training details and hyper-parameters are summarized in Appendix C.1.

Evaluation. The evaluation scheme consists of AMC-23 (Knovel Engineering, 2025), and AIME-24/25/26 (Sun et al., 2025). Following prior work (Xiong et al., 2025; Zhu et al., 2025; Yang et al., 2025b), we randomly sample 32 answers per question, with temperature set to 0.6. Due to space constraints, we show results from the full AIME experiments in Appendix C.4.

Table 1: Overview of entropy-preserving baselines. We compare the methods along three properties: (i) Can the method reach a target entropy value? (ii) Can it control entropy curves? (iii) Does it apply to any policy-gradient method?

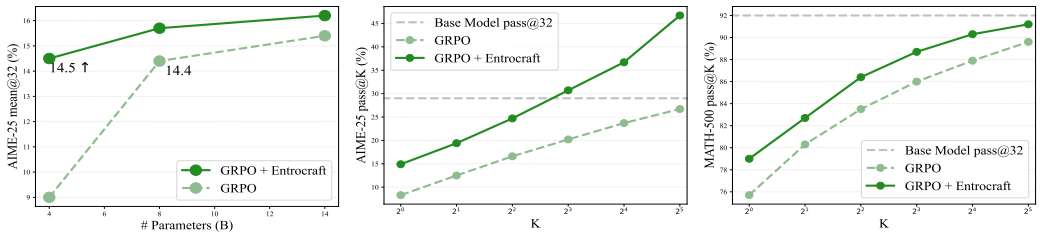
Method	Reach Target Entropy	Control Entropy Curve	Algorithm-Agnostic
Entropy Loss			✓
Clip-Higher (Yu et al., 2025)			✓
Clip-Cov (Cui et al., 2025)			✓
W-Reinforce (Zhu et al., 2025)			
EntroPIC (Yang et al., 2025b)	✓		
Entrocraft (ours)	✓	✓	✓

Table 2: Evaluations on math reasoning benchmarks. Entrocraft consistently improves the final performance of RL algorithms, better than other entropy-preserving methods. The best scores are in **bold**, and the second-best scores are marked with underlines.

Method	MATH-500		AMC-23		AIME-25		Avg. Score	
	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32
GRPO	75.3	89.6	57.4	92.5	9.0	26.7	47.2	69.6
+ Entropy Loss	76.0	89.8	55.0	92.5	9.9	33.3	47.0	71.9
+ Clip-Higher	75.8	91.4	55.5	95.0	10.8	40.0	47.4	75.5
+ Clip-Cov	76.8	91.8	57.1	90.0	10.8	33.3	48.2	71.7
+ Entrocraft	**79.0**	**93.0**	**65.0**	95.0	14.5	**46.7**	**52.8**	**78.2**
GSPO	75.6	90.2	57.7	92.5	8.9	33.3	47.4	72.0
+ Entropy Loss	74.7	90.6	55.9	**97.5**	11.9	33.3	47.5	73.8
+ Clip-Higher	75.5	90.4	54.7	95.0	10.0	40.0	46.7	75.1
+ Clip-Cov	75.3	91.2	56.3	92.5	10.0	40.0	47.2	74.6
+ Entrocraft	78.7	92.2	56.1	95.0	**14.7**	40.0	49.8	75.7
W-Reinforce	74.9	90.8	53.5	90.0	11.3	40.0	46.6	73.6
EntroPIC	73.7	90.2	55.3	92.5	10.9	43.3	46.6	75.3

5.2 Benchmark Evaluation

We first conduct general RL experiments and evaluate the final checkpoints on math reasoning benchmarks, as shown in Table 2 and Fig. 3. We demonstrate that Entrocraft outperforms other baselines under both mean@32 and pass@32 settings. Fig. 3a highlights that Entrocaraft can enable `Qwen3-4B-Base` to outperform `Qwen3-8B-Base` trained using standard GRPO. The pass@K experiments in Fig. 3b-3c also demonstrate that Entrocraft successfully prevents the actor from collapsing to just a few solutions, which benefits inference-time scaling.

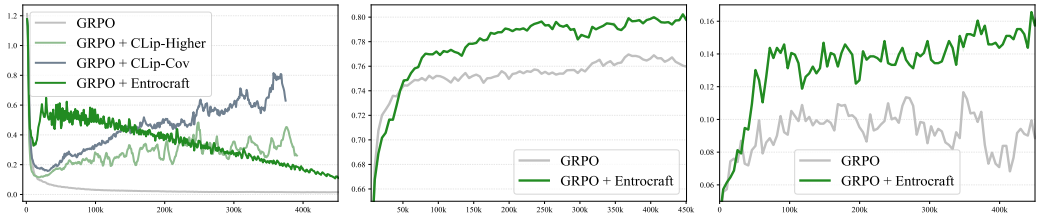


(a) Model Size v.s. AIME-25 (b) pass@K curve on AIME-25 (c) pass@K curve on MATH-500

Figure 3: Effectiveness of Entrocraft across model size and inference cost. (a) A 4B model can outperform an 8B model with Entrocraft; (b-c) Entrocraft improves inference-time scaling, with pass@K growing faster than under standard GRPO.

5.3 LONG-TERM RL

Finally, we demonstrate continual improvement in long-term RL powered by Entrocraft. As shown in Fig. 4 and Table 3, standard RL algorithms like GRPO (Shao et al., 2024) improve smoothly through the first 100K training samples. However, we observe only minimal sustained gains thereafter, a phenomenon known as performance saturation (Cui et al., 2025; Lu et al., 2024; Kim et al., 2026). In contrast, entropy-preserving methods alleviate this saturation and achieve better final performance. Among all entropy-preserving methods, Entrocraft achieves the highest scores at 400K training samples on both benchmarks. It leads at all four reported AIME-25 stages and at three of the four MATH-500 stages. We attribute this to its precise entropy control, which prevents entropy from drifting and avoids entropy instability common in uncontrolled entropy-preserving methods (Yu et al., 2025; Cui et al., 2025). As a concrete illustration, Clip-Cov (Cui et al., 2025) suffers a performance drop after 300K training samples, caused by entropy explosion (Fig. 4a) that destabilizes training in the later phase. We also show the long-term training curves with different entropy curves in Appendix C.7.



(a) Entropy (b) MATH-500 mean@32 (c) AIME-25 mean@32

Figure 4: Long-term performance comparison. Entrocraft accurately controls the entropy dynamics to be linearly decaying, and thus prevents the performance saturation and significantly improves over GRPO. Here, x-axis corresponds to the number of samples used for training.

Table 3: Long-term RL comparison at the first 100-400K training samples. GRPO suffers from performance saturation after 100K samples due to entropy collapse, while entropy-preserving methods overcome this saturation. Among all, Entrocraft is the most stable and highest-performing one.

Method	**MATH-500** mean@32				**AIME-25** mean@32			
	100K	200K	300K	400K	100K	200K	300K	400K
GRPO	75.2	75.4	75.6	76.6	9.9	10.1	9.6	9.4
+ Clip-Higher	76.0	77.6	78.4	78.8	9.6	12.3	12.8	13.6
+ Clip-Cov	76.5	78.1	**79.7**	77.5	10.9	12.5	14.2	15.0
+ Entrocraft	**76.9**	**78.9**	79.3	**79.9**	**13.1**	**14.2**	**14.8**	**16.2**

6 CONCLUSION

This paper introduces Entrocraft, a precise entropy-control method that addresses entropy collapse and the consequent performance saturation in RL. We first provide an LLM-oriented theoretical analysis of what drives entropy change, then design a rejection-sampling-based method that accurately controls entropy by biasing the advantage distribution. Experiments demonstrate that Entrocraft enables highly customizable entropy curves and consistently outperforms existing entropy-preserving methods across benchmarks. Entrocraft integrates as a drop-in to any policy-gradient method, enabling continual improvement on more data without saturation.

Ethics and Broader Impact Statement

This paper does not involve human-subject data collection, personally identifiable information, or deployment in safety-critical settings. Potential risks include enabling stronger reasoning models that may also be misused in broader downstream applications. However, the paper primarily contributes a training-stability technique rather than a new capability domain. We document datasets, implementation details, hyperparameters, and compute requirements, supporting external scrutiny while discouraging inappropriate use.

AI Use Disclosure

We used generative AI tools to polish manuscript text and generate plots. The authors take responsibility for the final text, figures, numerical results, and scientific claims.

References

Milton Abramowitz and Irene A Stegun. *Handbook of mathematical functions with formulas, graphs, and mathematical tables*, volume 55. US Government printing office, 1948.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, et al. Training a helpful and harmless assistant with reinforcement learning from human feedback. *arXiv preprint arXiv:2204.05862*, 2022.

Edward Beeching, Shengyi Costa Huang, Albert Jiang, Jia Li, Benjamin Lipkin, Zihan Qina, Kashif Rasul, Ziju Shen, Roman Soletskyi, and Lewis Tunstall. Numinamath 7b cot. https://huggingface.co/AI-MO/NuminaMath-7B-CoT, 2024.

Michael Beukman, Khimya Khetarpal, Zeyu Zheng, Will Dabney, Jakob Foerster, Michael Dennis, and Clare Lyle. Preventing learning stagnation in ppo by scaling to 1 million parallel environments. *arXiv preprint arXiv:2603.06009*, 2026.

Kun Chen, Peng Shi, Fanfan Liu, Haibo Qiu, Zhixiong Zeng, Siqi Yang, and Wenji Mao. Flexible entropy control in rlvr with a gradient-preserving perspective. *arXiv preprint arXiv:2602.09782*, 2026.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment. *Transactions on Machine Learning Research*, 2023, 2023.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. In *Neural Information Processing Systems*. Curran Associates, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Zhenyu Hou, Pengfan Du, Yilin Niu, Zhengxiao Du, Aohan Zeng, Xiao Liu, Minlie Huang, Hongning Wang, Jie Tang, and Yuxiao Dong. Does rlhf scale? exploring the impacts from data, model, and method. *arXiv preprint arXiv:2412.06000*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Leslie Pack Kaelbling, Michael L Littman, and Andrew W Moore. Reinforcement learning: A survey. *Journal of artificial intelligence research*, 4:237–285, 1996.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*, 2025.

Minwu Kim, Safal Shrestha, and Keith Ross. Training reasoning models on saturated problems via failure-prefix conditioning. *arXiv preprint arXiv:2601.20829*, 2026.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Knovel Engineering. AMC-23 Dataset. https://huggingface.co/datasets/knoveleng/AMC-23, 2025. Hugging Face dataset.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Pengyi Li, Elizaveta Goncharova, Andrey Kuznetsov, and Ivan Oseledets. Back to basics: Revisiting exploration in reinforcement learning for llm reasoning via generative probabilities. *arXiv preprint arXiv:2602.05281*, 2026.

Jiacai Liu. How rl policy entropy converges during updates. Blog, 2025. URL https://zhuanlan.zhihu.com/p/28476703733.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *Advances in Neural Information Processing Systems*, 38:17998–18031, 2026.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025.

TaiMing Lu, Lingfeng Shen, Xinyu Yang, Weiting Tan, Beidi Chen, and Huaxiu Yao. It takes two: On the seamlessness between reward and policy model in rlhf. In *ICML Workshop on Foundation Models in the Wild*, 2024.

Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous methods for deep reinforcement learning. In *International conference on machine learning*, pp. 1928–1937. PmLR, 2016.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Seohong Park, Kevin Frans, Deepinder Mann, Benjamin Eysenbach, Aviral Kumar, and Sergey Levine. Horizon reduction makes rl scalable. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Aleksei Petrenko, Ben Lipkin, Kevin Chen, Erik Wijmans, Marco Francis Cusumano-Towner, Raja Giryes, and Philipp Kraehenbuehl. Entropy-preserving reinforcement learning. In *The Fourteenth International Conference on Learning Representations*, 2026.

Yi Ren and Danica J Sutherland. Learning dynamics of llm finetuning. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Han Shen. On entropy control in llm-rl algorithms. *arXiv preprint arXiv:2509.03493*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.

Vaishnavi Shrivastava, Ahmed Hassan Awadallah, Vidhisha Balachandran, Shivam Garg, Harkirat Behl, and Dimitris Papailiopoulos. Sample more to think less: Group filtered policy optimization for concise reasoning. In *First Workshop on Foundations of Reasoning in Language Models*, 2025.

Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Lei Fang, Zheng Liu, Zhongyuan Wang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Richard S Sutton, Andrew G Barto, et al. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge, 1998.

Shumin Wang, Yuexiang Xie, Wenhao Zhang, Yuchang Sun, Yanxi Chen, Yaliang Li, and Yanyong Zhang. On the entropy dynamics in reinforcement fine-tuning of large language models. *arXiv preprint arXiv:2602.03392*, 2026a.

Xingjin Wang, Hao Sun, Lu Wang, Linjing Li, and Daniel Dajun Zeng. Entropy scheduling in reinforcement learning for large language models. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 4239–4251, 2026b.

Lilian Weng. Reward hacking in reinforcement learning. *lilianweng.github.io*, Nov 2024. URL https://lilianweng.github.io/posts/2024-11-28-reward-hacking/.

Ronald J Williams and Jing Peng. Function optimization using connectionist reinforcement learning algorithms. *Connection Science*, 3(3):241–268, 1991.

Wei Xiong, Jiarui Yao, Yuhui Xu, Bo Pang, Lei Wang, Doyen Sahoo, Junnan Li, Nan Jiang, Tong Zhang, Caiming Xiong, et al. A minimalist approach to llm reasoning: from rejection sampling to reinforce. *arXiv preprint arXiv:2504.11343*, 2025.

Wujiang Xu, Wentian Zhao, Zhenting Wang, Yu-Jhe Li, Can Jin, Mingyu Jin, Kai Mei, Kun Wan, and Dimitris N Metaxas. Epo: Entropy-regularized policy optimization for llm agents reinforcement learning. *arXiv preprint arXiv:2509.22576*, 2025.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Kai Yang, Xin Xu, Yangkun Chen, Weijie Liu, Jiafei Lyu, Zichuan Lin, Deheng Ye, and Saiyong Yang. Entropic: Towards stable long-term training of llms via entropy stabilization with proportional-integral control. *arXiv preprint arXiv:2511.15248*, 2025b.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Jiazheng Zhang, Ziche Fu, Junrui Shen, Yunbin Zhao, Yunke Zhang, Zhiheng Xi, Long Ma, Chenxin An, Zhihao Zhang, Shichun Liu, et al. Entropy polarity in reinforcement fine-tuning: Direction, asymmetry, and control. *arXiv preprint arXiv:2605.11775*, 2026a.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Xiaoyun Zhang, Xiaojian Yuan, Di Huang, Wang You, Chen Hu, Jingqing Ruan, Kejiang Chen, and Xing Hu. Revisiting entropy regularization: Adaptive coefficient unlocks its potential for llm reinforcement learning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 18005–18020, 2026b.

R Zhao, A Meterez, SM Kakade, C Pehlevan, S Jelassi, and E Malach. Echo chamber: Rl post-training amplifies behaviors learned in pretraining. In *The Conference on Language Modeling (COLM)*, 2025.

Yanli Zhao, Andrew Gu, Rohan Varma, Liang Luo, Chien-Chin Huang, Min Xu, Less Wright, Hamid Shojanazeri, Myle Ott, Sam Shleifer, et al. Pytorch fsdp: Experiences on scaling fully sharded data parallel. *Proceedings of the VLDB Endowment*, 16(12):3848–3860, 2023.

Chujie Zheng, Kai Dang, Bowen Yu, Mingze Li, Huiqiang Jiang, Junrong Lin, Yuqiong Liu, Hao Lin, Chencan Wu, Feng Hu, et al. Stabilizing reinforcement learning with llms: Formulation and practices. *arXiv preprint arXiv:2512.01374*, 2025a.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025b.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. The surprising effectiveness of negative reinforcement in llm reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

A Discussion

A.1 Related Works

Reinforcement Learning with Verifiable Rewards Reinforcement Learning (Kaelbling et al., 1996) has become a major approach in the post-training of LLMs, in which LLMs generate a set of rollout trajectories (exploration) and then reinforce the rewarded trajectories while punishing the less-rewarded ones (exploitation). Defining the reward scores for LLMs' trajectories is a critical challenge. Reinforcement Learning from human feedback (RLHF) (Ouyang et al., 2022) separately trains a reward model (RM) as the proxy of human preference. However, RM-based reward has been shown to suffer from significant reward hacking problems (Weng, 2024), which can limit generalization to new questions. Post-training of LLMs has increasingly focused on tasks that exhibit long, complex trajectories with simple verification, including math reasoning (Cobbe et al., 2021) and code generation (Chen et al., 2021; Jain et al., 2025). For example, o1 (Jaech et al., 2024) and R1 (Guo et al., 2025) use RL with verifiable rewards to elicit complex reasoning capabilities, and GRPO (Shao et al., 2024) has become a widely used RL algorithm for training reasoning models. Recent works propose many variants of GRPO to mitigate its drawbacks, including length inflation (Dr. GRPO (Liu et al., 2025) and GFPO (Shrivastava et al., 2025)) and training instability (GSPO (Zheng et al., 2025b) and Zheng et al. (2025a)). ProRL (Liu et al., 2026) further investigates prolonged RL training using KL-divergence control and reference-policy resetting, reporting continued improvements across reasoning tasks. A key challenge remains the scalability of RL algorithms: with more data and compute, under what conditions will RL algorithms continually improve model performance?

Entropy Dynamics of Reinforcement Learning Entropy has been used as a key indicator for models' exploration during RL (Williams & Peng, 1991; Mnih et al., 2016; Haarnoja et al., 2017), which is important for achieving strong performance (Sutton et al., 1998). Entropy collapse is frequently observed in RL algorithms, trading exploration potential for higher rewards (Cui et al., 2025; Karan & Du, 2025; Zhang et al., 2025). However, recent theoretical analyses relate entropy dynamics to advantage-weighted policy updates and identify conditions under which performance can improve without an entropy drop (Liu, 2025; Cui et al., 2025; Yang et al., 2025b; Wang et al., 2026a; Shen, 2025). Further, Petrenko et al. (2026) analyze entropy dynamics under parameterized policies and develop explicit entropy-control mechanisms. Empirically, techniques used to avoid entropy collapse include entropy regularization (including adaptive variants such as AEnt (Shen, 2025) and AER (Zhang et al., 2026b)), clipping (Clip-Higher (Yu et al., 2025), Clip-Cov (Cui et al., 2025), $\text{Clip}_{\mathcal{B}/\mathcal{V}}$ (Wang et al., 2026a), and ADAPO (Petrenko et al., 2026)), and advantage modification (W-Reinforce (Zhu et al., 2025), EntroPIC (Yang et al., 2025b), REPO (Petrenko et al., 2026), and polarity-aware weighting in PAPO (Zhang et al., 2026a)). Existing methods also explicitly control entropy targets and schedules: EntroPIC uses proportional-integral feedback to adjust positive and negative sample weights, Entropy Scheduling (ES) (Wang et al., 2026b) adjusts clipping thresholds to track prescribed entropy schedules, and Chen et al. (2026) investigate dynamic clipping for multi-stage and oscillatory entropy trajectories. Entrocraft implements entropy feedback through rejection sampling of rollouts according to advantage sign, and studies fixed and annealed target schedules for sustained RL training.

A.2 Limitation and Future Directions

This paper focuses on the stability and continual improvement of RL algorithms, and has validated the method's effectiveness on standard math reasoning tasks with dense models. However, in more challenging settings like multi-turn RL and mixture-of-expert (MoE) models, the entropy instability is more catastrophic (Xu et al., 2025; Zheng et al., 2025a) due to the more sparse solution space and the variable policies. The current method design is optimized for single-turn math reasoning task. We plan to extend the entropy analysis and entropy-control algorithm to such settings.

B PROOFS

B.1 ASSUMPTIONS

For $\hat{A}(x, y) \neq 0$, we assume that, for every token t:

$$\hat{A}(x, y) \cdot \delta p_{t,y_t} > 0 \quad \text{and} \quad b_t := \log p_{t,y_t} + \sum_{i \in \mathbb{V} \setminus \{y_t\}} \frac{\delta p_{t,i}}{\delta p_{t,y_t}} \log p_{t,i} \geq 0, \tag{6}$$

with $b_t > 0$ for at least one token. These assumptions imply that logit changes are in the same direction as the corresponding advantage, and that the confidence of selected tokens is sufficiently high (higher than the unselected tokens).

B.2 PROOF OF THEOREM 1

Token-Level Entropy Change of LLMs. Consider a single RL update step. Let p_k be the probability that token k is sampled during rollout generation. The sign of the entropy change triggered by token k is bounded by the sign of the advantage score:

$$\hat{A}_k \cdot \Delta\mathcal{H} \leq 0, \ \text{ whenever } \ p_k > \prod_{i \in \mathbb{V} \setminus \{k\}} p_i^{-\frac{\delta p_i}{\delta p_k}},$$

where δp_i is the probability change at this RL step.

Proof:

Let $\boldsymbol{p}$ be the probability vector before the RL update and $\delta\boldsymbol{p}$ be the update. The entropy change can be approximated by Taylor expansion (Abramowitz & Stegun, 1948):

$$\begin{aligned}
\Delta\mathcal{H} &= \mathcal{H}(\boldsymbol{p} + \delta\boldsymbol{p}) - \mathcal{H}(\boldsymbol{p}) \\
&= \sum_{i \in \mathbb{V}} \frac{\partial \mathcal{H}}{\partial p_i} \delta p_i + O(\|\delta\boldsymbol{p}\|_2^2) \\
&= -\sum_{i \in \mathbb{V}} (1 + \log p_i) \cdot \delta p_i + O(\|\delta\boldsymbol{p}\|_2^2) \\
&\overset{\text{(a)}}{=} -\sum_{i \in \mathbb{V}} \delta p_i \log p_i + O(\|\delta\boldsymbol{p}\|_2^2).
\end{aligned} \tag{7}$$

Here, (a) is due to the constraint on output probabilities: $\sum_{i \in \mathbb{V}} p_i \equiv 1$. Then, we identify the condition for entropy to decrease (i.e., $\Delta\mathcal{H} < 0$). This occurs when:

$$\sum_{i \in \mathbb{V}} \delta p_i \log p_i > 0. \tag{8}$$

For token k with positive advantage $\hat{A}_k > 0$, the RL update increases its probability: $\delta p_k > 0$. By the probability conservation constraint $\sum_{i \in \mathbb{V}} \delta p_i \equiv 0$, we have $\sum_{i \in \mathbb{V} \setminus \{k\}} \delta p_i = -\delta p_k < 0$. Then, the entropy change can be rewritten as:

$$\Delta\mathcal{H} = -\delta p_k \log p_k - \sum_{i \in \mathbb{V} \setminus \{k\}} \delta p_i \log p_i + O(\|\delta\boldsymbol{p}\|_2^2). \tag{9}$$

Since $\delta p_i = \delta p_k \frac{\delta p_i}{\delta p_k}$ for $i \neq k$, we obtain:

$$\begin{aligned}\Delta\mathcal{H} &= -\delta p_k \left(\log p_k + \sum_{i \in \mathbb{V}\setminus\{k\}} \frac{\delta p_i}{\delta p_k} \log p_i \right) + O(\|\delta \boldsymbol{p}\|_2^2) \\ &= -\delta p_k \left(\log p_k - \log \prod_{i \in \mathbb{V}\setminus\{k\}} p_i^{-\frac{\delta p_i}{\delta p_k}} \right) + O(\|\delta \boldsymbol{p}\|_2^2).\end{aligned} \tag{10}$$

Entropy decreases ($\Delta\mathcal{H} < 0$) when the term in parentheses is positive: $\log p_k > \log \prod_{i \in \mathbb{V}\setminus\{k\}} p_i^{-\frac{\delta p_i}{\delta p_k}}$. Symmetrically, for tokens with a negative advantage, entropy increases. Therefore, $\hat{A}_k \cdot \Delta\mathcal{H} \leq 0$ holds with high probability.

□

B.3 Proof of Theorem 2

Sequence-Level Entropy Change of LLMs. Consider a single RL update step and assume all tokens share the same outcome reward. Let $p_{t,i} = \pi_{\boldsymbol{\theta}}(y_t = i | x, y_{<t})$ be the probability that i is sampled as the t-th token in the sequence. The sign of the entropy change triggered by response y is bounded by the sign of the advantage score:

$$\hat{A}(x, y) \cdot \Delta\mathcal{H} \leq 0, \quad \text{whenever } \pi_{\boldsymbol{\theta}}(y|x) \geq \prod_{t=1}^{|y|} \prod_{i \in \mathbb{V}\setminus\{y_t\}} p_{t,i}^{-\frac{\delta p_{t,i}}{\delta p_{t,y_t}}},$$

where $\delta p_{t,i}$ is the probability change at this RL step.

Proof:

Similar to the token-level entropy, by Taylor expansion (Abramowitz & Stegun, 1948), the sequence-level entropy change can be expressed as:

$$\begin{aligned}\Delta\mathcal{H} &= \sum_{t=1}^{|y|} \sum_{i \in \mathbb{V}} \frac{\partial \mathcal{H}}{\partial p_{t,i}} \cdot \delta p_{t,i} + O\left(\sum_{t=1}^{|y|} \|\delta \boldsymbol{p}_t\|_2^2 \right) \\ &= -\frac{1}{|y|} \sum_{t=1}^{|y|} \sum_{i \in \mathbb{V}} (1 + \log p_{t,i}) \cdot \delta p_{t,i} + O\left(\sum_{t=1}^{|y|} \|\delta \boldsymbol{p}_t\|_2^2 \right) \\ &\overset{\text{(b)}}{=} -\frac{1}{|y|} \sum_{t=1}^{|y|} \sum_{i \in \mathbb{V}} \delta p_{t,i} \cdot \log p_{t,i} + O\left(\sum_{t=1}^{|y|} \|\delta \boldsymbol{p}_t\|_2^2 \right) \\ &= -\frac{1}{|y|} \sum_{t=1}^{|y|} \delta p_{t,y_t} \left(\log p_{t,y_t} + \sum_{i \in \mathbb{V}\setminus\{y_t\}} \frac{\delta p_{t,i}}{\delta p_{t,y_t}} \cdot \log p_{t,i} \right) + O\left(\sum_{t=1}^{|y|} \|\delta \boldsymbol{p}_t\|_2^2 \right).\end{aligned} \tag{11}$$

Here, (b) is due to the constraint on output probabilities: $\sum_{i \in \mathbb{V}} p_{t,i} \equiv 1$. Now assume the estimated advantage is positive: $\hat{A}(x, y) > 0$. The corresponding probability changes for the entire response are then all positive: $\min_t \delta p_{t,y_t} > 0$. Given Appendix B.1, we can simplify the above expression via:

$$\overline{\delta p_y} = \frac{\sum_{t=1}^{|y|} \delta p_{t,y_t} b_t}{\sum_{t=1}^{|y|} b_t}, \tag{12}$$

which is a convex weighted average of the sampled-token probability changes. Since $\delta p_{t,y_t} > 0$ for every token under our token-alignment assumption, we obtain $\overline{\delta p_y} > 0$.

Then, the entropy change can then be simplified as:

$$\begin{aligned}\Delta\mathcal{H} &\approx -\frac{1}{|y|}\cdot\overline{\delta p_y}\cdot\left(\log\prod_{t=1}^{|y|}p_{t,y_t}-\log\prod_{t=1}^{|y|}\prod_{i\in\mathbb{V}\setminus\{y_t\}}p_{t,i}^{-\frac{\delta p_{t,i}}{\delta p_{t,y_t}}}\right)\\ &= -\frac{1}{|y|}\cdot\overline{\delta p_y}\cdot\left(\log\pi_{\boldsymbol{\theta}}(y|x)-\log\prod_{t=1}^{|y|}\prod_{i\in\mathbb{V}\setminus\{y_t\}}p_{t,i}^{-\frac{\delta p_{t,i}}{\delta p_{t,y_t}}}\right),\end{aligned} \tag{13}$$

showing that entropy change is negatively related to advantage when the likelihood of the generated response exceeds a threshold: $\pi_{\boldsymbol{\theta}}(y|x) > \prod_{t=1}^{|y|}\prod_{i\in\mathbb{V}\setminus\{y_t\}}p_{t,i}^{-\frac{\delta p_{t,i}}{\delta p_{t,y_t}}}$. The same derivation applies to the negative advantage case ($A(x,\hat{y})<0$). Therefore, $\hat{A}(x,y)\cdot\Delta\mathcal{H}\leq 0$ holds when the rollout sample likelihood is sufficiently high.

□

C Additional Experiment Details

C.1 Implementation Details

The implementation is based on the verl framework (Sheng et al., 2025), which uses vLLM (Kwon et al., 2023) for inference and FSDP2 (Zhao et al., 2023) for training. All experiments are conducted on a node of 8 NVIDIA H100 GPUs. We summarize the core hyperparameters in Table 4. The average training time for `Qwen3-4B-Base` is around 10K training samples per hour. We also list the hyperparameters used by compared baselines in Table 5.

Table 4: Core hyperparameters for training implementation. The origin of each hyperparameter is listed, and all empty-source hyperparameters are determined by our experiments.

Hyperparameter	**Value**	**Source**
`train_batch_size`	1024	verl examples
`ppo_mini_batch_size`	8×32	verl examples
`max_context_length`	1024 + 3072	Xiong et al. (2025)
`rollout.n`	8	
`optim.lr`	1e-6	verl examples
`kl_loss_coef`	1e-3	verl examples
`val_kwargs.n`	32	
`val_kwargs.temperature`	0.6	

C.2 Entropy Curves of Baselines

We show the full entropy curves of baselines in Fig. 5. Standard GRPO (Shao et al., 2024) exhibits entropy collapse. On top of GRPO, many entropy-preserving methods successfully alleviate the entropy-decreasing trend. However, they are not necessarily responsive enough to enable accurate entropy control and may cause the entropy curves to be unstable in the long term. In contrast, the proposed Entrocraft accurately stabilizes entropy at the target (0.8), making exploration controllable as a hyperparameter.

C.3 Effective Batch Size

To explicitly show the effect of Entrocraft on the number of rollout samples used for gradient update, we pick the steps that trigger rejection sampling, and show their effective batch sizes throughout the training in Fig 6. The RL training is GRPO + Entrocraft with initial `rollout.n=8`, and the entropy range setting is the linear decaying as used in Section C.7. The dropping effective batch sizes verify that Entrocraft requires less gradient computation than naive GRPO.

Table 5: Hyperparameters of Entrocraft and compared baselines.

Method	**Hyperparameter**	**Value**
Entropy loss	GRPO coefficient	0.003
	GSPO coefficient	0.003
Clip-Higher	$(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(0.2, 0.28)$
Clip-Cov	Token selection ratio	0.0002
	Covariance bounds	$[1.0, 5.0]$
	$(\epsilon_{\text{low}}, \epsilon_{\text{high}})$	$(1.0, 1.0)$
W-Reinforce	Positive / negative reward weights	$0.1/1.0$
EntroPIC	Target entropy	0.5
	Controller (a, b, s_0)	$(1.0, 0.01, 0)$
Entrocraft	Fixed entropy target	0.5
	GRPO, linear (100k subset)	$(0.6, 0.7) \rightarrow (0.4, 0.5)$
	GSPO, linear (100k subset)	$(0.6, 0.7) \rightarrow (0.2, 0.3)$
	GRPO, cosine (full dataset)	$(0.6, 0.7) \rightarrow (0.1, 0.2)$

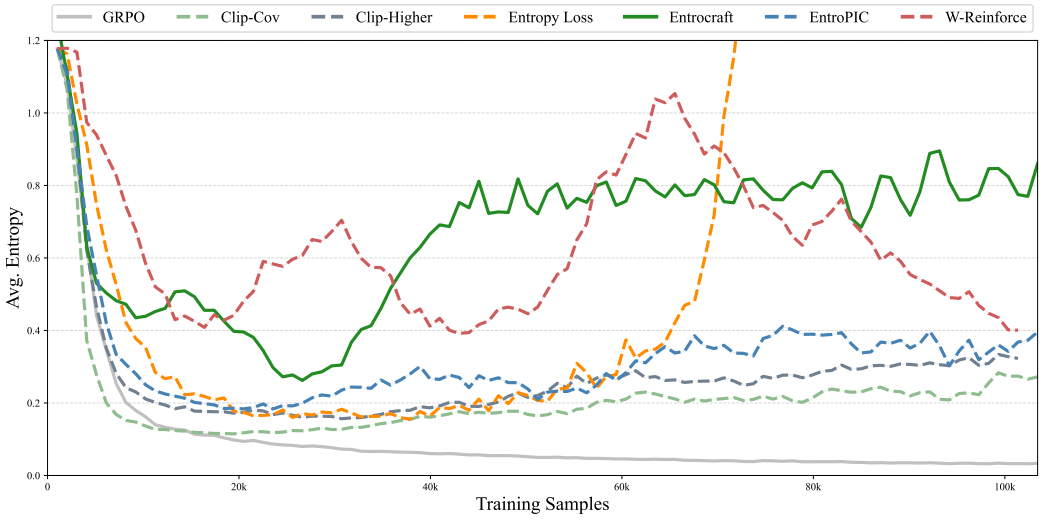


Figure 5: Entropy curve comparison across baselines. Other entropy-preserving methods may induce instability during long-term training or may not be sufficiently responsive. In contrast, the proposed Entrocraft effectively controls the entropy curve to be the customized shape.

C.4 Full Evaluation on AIME

In addition to the benchmark evaluations of Table 2, we list the full AIME results in Table 6. The conclusions on all AIME benchmarks are consistent with Table 2. Entrocraft outperforms all other entropy-preserving methods across diverse RL algorithms.

C.5 Results on Other Models

Aside from `Qwen3-4B-Base`, we also try Entrocraft on larger models and models from different model families, as shown in Table 7, 8, and 9. The results report consistent improvement over GRPO across different base models.



Figure 6: Effective batch sizes for the RL steps affected by Entrocraft. The negative samples become less as the model's performance improves.

Table 6: Full Evaluations on AIME. The proposed Entrocraft consistently improves the final performance of RL algorithms, better than other entropy-preserving methods. The best scores are in **bold**, and the second-best scores are marked with underlines.

Method	AIME-24		AIME-25		AIME-26	
	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32
GRPO	7.7	30.0	9.0	26.7	8.1	33.3
+ Entropy Loss	9.0	26.7	9.9	33.3	9.1	26.7
+ Clip-Higher	10.2	23.3	10.8	40.0	9.1	36.7
+ Clip-Cov	9.7	26.7	10.8	33.3	10.9	36.7
+ Entrocraft	**16.7**	**36.7**	**14.5**	**46.7**	**15.2**	**40.0**
GSPO	8.8	26.7	8.9	33.3	8.8	33.3
+ Entropy Loss	7.9	23.3	11.9	33.3	9.6	36.7
+ Clip-Higher	10.0	20.0	10.0	40.0	9.5	36.7
+ Clip-Cov	9.7	26.7	10.0	40.0	9.1	36.7
+ Entrocraft	10.2	26.7	14.7	40.0	13.3	36.7
W-Reinforce	8.8	26.7	11.3	40.0	9.3	26.7
EntroPIC	9.9	20.0	10.9	43.3	9.7	30.0

C.6 Failure Case: Maintaining High Entropy May Induce Instability in The Long Term

To discuss the boundary of entropy control methods, we use Entrocraft with different entropy targets and record their training dynamics in long-term RL. We show two distinct examples in Fig. 7. We find that an overly high entropy level introduces instability into RL training, and such instability will accumulate and become more catastrophic in the long term. This also validates our design of entropy curve annealing schemes.

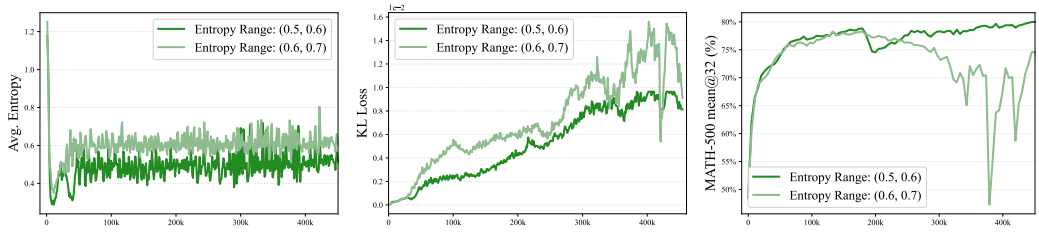


Figure 7: Failure case study. Overly high entropy level introduces instability into RL training, making the empirical performance fragile to subtle perturbations.

Table 7: Benchmark results on `Llama-3.1-8B-Instruct`. Entrocraft demonstrates consistent improvements over GRPO. The best scores are in **bold**.

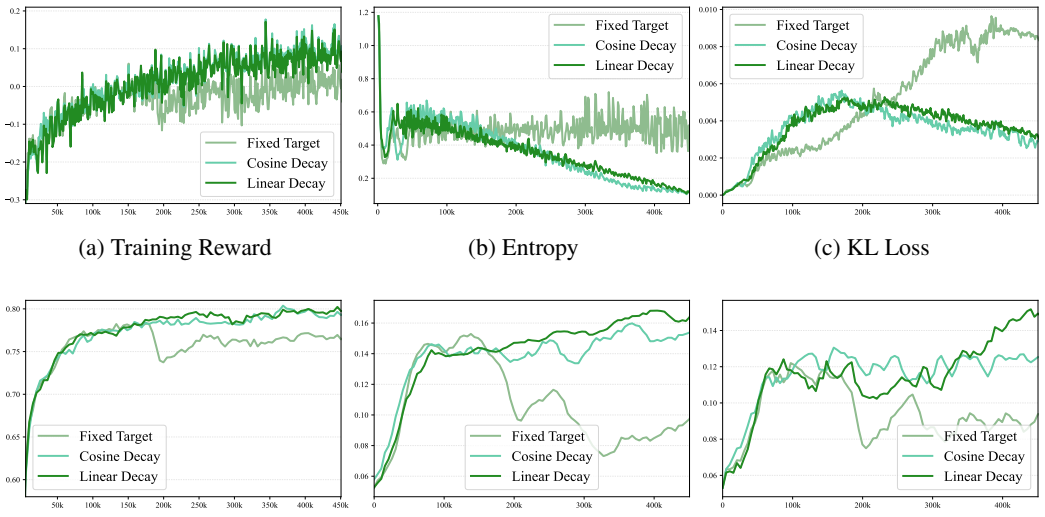
Method	MATH-500		AMC-23		AIME-25		Avg. Score	
	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32
GRPO	56.2	82.2	30.8	75.0	10.4	16.7	32.5	58.0
+ Entrocraft	**57.1**	**85.2**	**36.8**	**85.0**	**18.8**	**26.7**	**37.6**	**65.6**

Table 8: Benchmark results on `Qwen3-8B-Base`. Entrocraft demonstrates improvements over GRPO. The best scores are in **bold**.

Method	MATH-500		AMC-23		AIME-25		Avg. Score	
	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32
GRPO	77.2	91.8	59.7	92.5	**14.4**	43.3	50.0	75.9
+ Entrocraft	**78.4**	**94.0**	**63.5**	**97.5**	13.8	**46.7**	**51.9**	**79.4**

C.7 Crafting Entropy Curves

We demonstrate the entropy control capability of Entrocraft. The entropy control powered by the proposed entropy-guided rejection sampling filter is powerful enough that entropy curves can be crafted directly, much like learning-rate schedules. As empirical evidence of entropy annealing introduced in Section 4.3, we comprehensively compare three entropy annealing schemes (fixed target, linear decay, and cosine decay) in Fig. 8.[5] The fixed-target scheme, similar to previous entropy-control studies (Yang et al., 2025b; Petrenko et al., 2026), becomes unstable after the first 200k training samples. Its entropy curve fluctuates sharply, leading to a drop in performance. In contrast, decaying schemes eliminate the instability and sustain improvement even after 400k training samples.



(a) Training Reward (b) Entropy (c) KL Loss

(d) MATH-500 mean@32 (e) AIME-25 mean@32 (f) AIME-26 mean@32

Figure 8: Long-term training dynamics of three entropy annealing schemes implemented in Entrocraft. The fixed-target scheme becomes unstable due to rollout imbalance. Both linear and cosine decay schemes remain stable and sustain improvement, and linear decay is slightly better. x-axis: the number of samples used for training.

[5]We set the fixed entropy target to be 0.5. Both linear and cosine decaying use annealing entropy range schedules, starting at (0.6, 0.7) and ending at (0.1, 0.2)

Table 9: Benchmark results on `Qwen3-14B-Base`. Entrocraft demonstrates consistent improvements over GRPO. The best scores are in **bold**.

Method	MATH-500		AMC-23		AIME-25		Avg. Score	
	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32	mean@32	pass@32
GRPO	80.8	92.2	66.7	**97.5**	14.7	40.0	54.1	76.6
+ Entrocraft	**81.7**	**93.8**	**67.1**	**97.5**	**17.0**	**53.3**	**55.3**	**81.5**

C.8 Variant Method: Token-Level Rejection Sampling

The Entrocraft introduced in Section 4 is a instance-level rejection sampling algorithm. We also implement a token-level rejection sampling version, which uses each token's entropy and the outcome reward to decide whether to remove a token from the loss mask or not. The token-level variant is compared in Fig. 9. We keep the hyperparameters the same as standard Entrocraft, and only change the rejection sampling granularity. The results show that token-level rejection sampling is less effective in controlling the entropy curves, as it keeps the gradients from many tokens that should be excluded in the instance-level version, making the advantage distribution noisier.

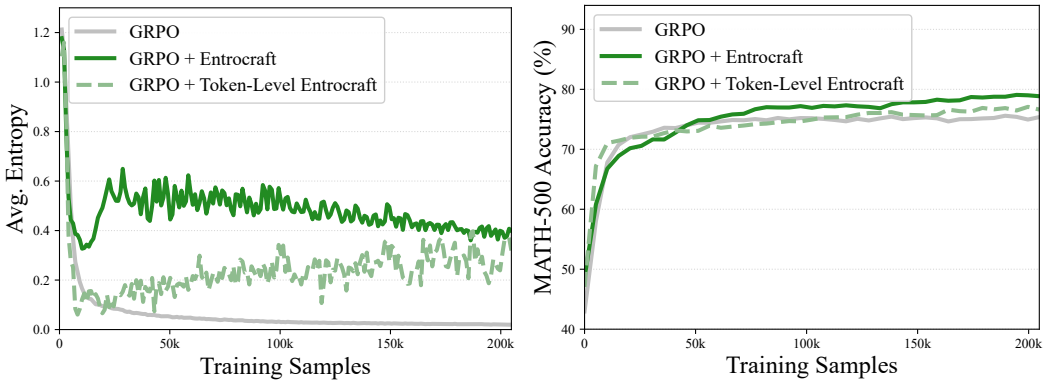


Figure 9: Comparison with token-level rejection sampling. Token-level rejection sampling is less effective in controlling the entropy curves, and improves less than the instance-level rejection sampling in terms of MATH-500 averaged accuracy.