Rethinking Entropy Interventions in RLVR: An Entropy Change Perspective

Zhezheng Hao[1*] **Hong Wang**[2*] **Haoyang Liu**[3] **Jian Luo**[3]
Jiarui Yu[2] **Hande Dong**[2†] **Qiang Lin**[2] **Can Wang**[1] **Jiawei Chen**[1†]
[1] Zhejiang University [2] Tencent [3] Independent Researcher
Emails: haozhezheng@outlook.com, donghd66@gmail.com, sleepyhunt@zju.edu.cn

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has become a key driver of reasoning improvements in large language models (LLMs), yet its training is often plagued by *entropy collapse* — a rapid decline in policy entropy that limits exploration and undermines training effectiveness. Recent approaches attempt to mitigate this issue via various heuristic entropy interventions, yet the mechanisms underlying their effect on entropy are not well understood. In this work, we present comprehensive theoretical and empirical analyses of entropy dynamics in RLVR, offering two main insights: (1) we derive an approximate expression for token-level entropy change at each update step, revealing four governing factors and providing a unified view of how existing methods influence entropy; (2) we show a fundamental limitation of recent approaches — they rely on heuristic adjustments to one or two of these factors, leaving other relevant factors unconsidered, thus inherently limiting their effectiveness. Guided by these analyses, we introduce STEER, a principled entropy-modulation method that adaptively reweighs tokens based on their estimated entropy change. Experiments across six math reasoning and three coding benchmarks show that STEER effectively mitigates entropy collapse and consistently outperforms state-of-the-art baselines.

1 Introduction

Large Language Models (LLMs) have recently demonstrated remarkable reasoning capabilities in complex tasks such as mathematics and coding (OpenAI, 2025; Google, 2025; Anthropic, 2025). A key technique driving these advances is Reinforcement Learning with Verifiable Rewards (RLVR) (Shao et al., 2024; Hu et al., 2025a; Hao et al., 2025; Ahmadian et al., 2024). RLVR employs policy-gradient algorithms (*e.g.,* PPO(Schulman et al., 2017) and GRPO(Shao et al., 2024)) with verifiable reward signals, providing an effective supervision mechanism for post-training. This paradigm has been instrumental in unlocking the post-training scaling of reasoning performance (Jaech et al., 2024; Shao et al., 2024; Liu et al., 2025a), thereby substantially advancing the reasoning capabilities of LLMs and laying the foundation for recent progress in advanced reasoning models.

Despite its empirical success, recent studies have identified a major pitfall of RLVR: *entropy collapse*, a rapid decline in policy entropy during training (Wu et al., 2025a; Song et al., 2025; Li et al., 2025b; Cui et al., 2025b). This notorious phenomenon severely impairs exploration, leading to increasingly homogeneous rollouts and limiting the model's ability to discover informative and potentially correct solutions. More critically, it fundamentally undermine the effectiveness of RLVR training. Particularly in advanced group-based algorithms like GRPO (Shao et al., 2024), the computed advantages become less discriminative, causing training to stagnate. This raises an important research question: *how can policy entropy be effectively modulated to preserve exploration?*

Several recent methods for addressing entropy collapse can be broadly classified into three categories: (1) *Clip-Higher*, which increases the upper bound of the importance-sampling ratio (Yu et al., 2025; Yang et al., 2025a); (2) *Positive-Reweighting*, which reduces the weight of positive samples with high generation probability (Zhu et al., 2025; He et al., 2025a); (3) *Entropy-Aware Advantage*, which assigns larger advantage to tokens with high entropy (Cheng et al., 2025; Tan and Pan, 2025; Wang et al., 2025d,c; Deng et al., 2025). While these strategies can alleviate entropy collapse to some extent, they remain largely heuristic, and the mechanisms through which they affect entropy are still

[*] Equal Contribution.
[†] Corresponding Authors

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 31105–31133

poorly understood. Consequently, their effectiveness is limited, and entropy remains only loosely controlled. These limitations call for a principled understanding of entropy dynamics in RLVR—one that not only explains the behavior of existing methods, but also guides the design of more effective entropy-control strategies.

To gain a comprehensive understanding of entropy dynamics in RLVR training, we conduct both theoretical and empirical analyses of entropy change under each update step. To obtain a more fine-grained view, our analysis focuses on token-level entropy change, rather than the global expectation commonly considered in prior work. Although (Cui et al., 2025b) offers an initial analysis of entropy, it relies on unrealistic assumptions that lead to imprecise estimates (*cf.* Section 3.1) and cannot explain the mechanisms underlying existing entropy-intervention methods. In contrast, our analysis yields two main insights: (1) We derive an approximate expression for token-level entropy change, showing that it is determined by four key factors—clipping strategy, advantage, token probability, and conditional entropy. We further clarify how existing methods influence these factors, thereby explaining their empirical success. (2) We identify key limitations of prior methods: their heuristic modulation are applied only to a small subset of tokens, neglecting others that may also experience severe entropy collapse; and they consider only partial relevant factors, which reduces their effectiveness and can even accelerate entropy decay in some cases.

Motivated by these insights, we introduce STEER (Stabilizing Token-level Entropy-changE via Reweighting), a simple yet principled entropy-modulation method grounded in our theoretical analysis. STEER directly translates the theoretically-estimated entropy variations into adaptive token-level weights. By down-weighting tokens with excessively large entropy change, it meticulously regulates per-step entropy dynamics, and steers the policy toward sustained exploration. We conduct extensive experiments on six mathematical reasoning benchmarks and three coding benchmarks, demonstrating that STEER consistently outperforms all 10 baselines by a substantial margin. Importantly, STEER also generalizes well across model scales (1.5B/7B/14B), model families (Qwen/Llama/Mistral), and RL algorithms (GRPO/RLOO/OPO), consistently maintaining stable entropy dynamics and achieving stronger performance.

In summary, our contributions are:

- We conduct comprehensive theoretical analyses of token-level entropy dynamics, revealing its key governing factors and explaining the mechanisms and limitations of recent entropy-intervention strategies.
- We propose STEER, a theoretically-grounded entropy-control method that adaptively down-weights tokens with large entropy changes.
- We conduct comprehensive experiments on six mathematical reasoning benchmarks and three coding benchmarks, demonstrating the effectiveness of STEER over strong baselines and its generalization across model scales, model families, and RL algorithms.

2 Preliminaries

2.1 RLVR Algorithms

Given a query q sampled from a dataset $\mathcal{D}$, let π_θ be the policy model and o be a sampled response. PPO (Schulman et al., 2017) optimizes the policy by maximizing the expected advantage and stabilizes the training process through the clipped surrogate. GRPO (Shao et al., 2024) removes the value model and instead samples a group of rollouts $\{o_i\}_{i=1}^{G}$ for query q, optimizing LLM policies using relative advantage scores computed within groups of samples:

$$A_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}, \tag{1}$$

where $R_i \in \{1, -1\}$ indicates whether the i-th response is correct and the advantage $A_{i,t}$ is shared across all tokens in the response. Following the token-level formulation (Yu et al., 2025), GRPO maximizes the following objective:

$$\mathcal{J}(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D}, \\ \{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}} \left[\frac{1}{L} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} \min \Big(r_{i,t} A_{i,t}, \text{clip}\left(r_{i,t}, 1-\varepsilon, 1+\varepsilon\right) A_{i,t} \Big) \right]. \tag{2}$$

where $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q, o_{i,<t})}$ denotes the importance sampling ratio and $L = \sum_{i=1}^{G} |o_i|$ denotes the sum of response lengths within a group. We omit the KL divergence term from the original GRPO formulation following (Shao et al., 2024). This

modification has been shown to yield better performance and encourage broader exploration (Chu et al., 2025; Hu et al., 2025b).

2.2 Policy Entropy of LLMs

Shannon entropy quantifies the uncertainty of a policy model's action selection given a state (Haarnoja et al., 2018). For LLMs, entropy can be quantified at each generation step. For each sampled response o_i and step t, the next-token entropy under policy π_θ is:

$$\mathcal{H}(q, o_{i,<t}) = -\mathbb{E}_{a\sim\pi_\theta(\cdot|q,o_{i,<t})}\left[\log \pi_\theta(a \mid q, o_{i,<t})\right]. \tag{3}$$

The (global) policy entropy measures the model's generation uncertainty over a query dataset, which can be estimated by averaging token-level entropy over sampled responses:

$$\mathcal{H}(\pi_\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D},\\ \{o_i\}\sim\pi_\theta(\cdot|q)}} \frac{1}{L}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\mathcal{H}(q, o_{i,<t}). \tag{4}$$

In practical, we may estimate $\mathcal{H}(\pi_\theta)$ using the training dataset and empirically observe that this estimation generalizes well to unseen data (*e.g.,* test sets, *cf.* Appendix D.1).

Maintaining adequate policy entropy is essential for balancing exploration and exploitation during RL training (Haarnoja et al., 2018; Ziebart et al., 2008). However, RLVR training often suffers from *entropy collapse*, in which policy entropy drops rapidly (Song et al., 2025; Li et al., 2025b; Cui et al., 2025b). This phenomenon is harmful in two respects: (1) It severely weakens exploration, limiting the model's ability to discover informative and potentially correct solutions. (2) It also undermines training effectiveness. Typically, for group-based methods such as GRPO(Shao et al., 2024), increasingly homogeneous rollouts yield relative advantage scores A_{it} that are less discriminative, causing training to stagnate.

3 Entropy-intervention Mechanism: An Entropy Change Perspective

In this section, we present a comprehensive theoretical and empirical analyses of entropy change. Our investigation is conducted at a fine-grained token level, which enables precise monitoring of entropy dynamics. Without loss of generality, this section primarily focuses on the representative GRPO algorithm for clarity of exposition. Nevertheless, **these theoretical insights generalize seamlessly to other advanced RLVR algorithms** (*e.g.,*, RLOO and OPO; *cf.* Appendix D.3).

3.1 Quantitative Analysis on Entropy Change

To quantitatively characterize the per-token entropy change after one update, we rewrite the GRPO policy gradient (Eq. (2)) as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D},\\ \{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}}\Bigg[\frac{1}{L}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \mathbb{I}_{\text{clip}}\, r_{i,t}A_{i,t}\nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\Bigg]. \tag{5}$$

where the clipping indicator $\mathbb{I}_{\text{clip}}$ is derived from the ratio clipping operation and is defined as:

$$\mathbb{I}_{\text{clip}} = \begin{cases} 0, & A_{i,t} > 0 \text{ and } r_{i,t} > 1+\varepsilon, \\ 0, & A_{i,t} < 0 \text{ and } r_{i,t} < 1-\varepsilon, \\ 1, & \text{otherwise.} \end{cases} \tag{6}$$

For notational convenience, we abbreviate $\pi_\theta(a \mid s)$ as π_θ and $A(s,a)$ as A, and denote the context (state) by $s \triangleq (q, o_{i,<t})$. Then we derive the following theorem on token-level entropy change.

Theorem 1. *For a logit-independent policy model π_θ, trained using GRPO with a learning rate η, the change of the token-level entropy on state s between two consecutive steps can be approximated as:*

$$\Omega(s) \triangleq -\frac{\eta}{L}\,\mathbb{E}_{\pi_\theta(\cdot|s)}\Big[\frac{\mathbb{I}_{\text{clip}}\,A}{\pi_{old}}\,\pi_\theta\big(1-\pi_\theta\big) \big(\log \pi_\theta + \mathcal{H}(s)\big)\Big]. \tag{7}$$

The proof is presented in Appendix G. The theorem is not specific to GRPO, and also hold for other algorithms (*cf.* Appendix D.3). Notably, in practice, since the learning rate η is typically small ($< 10^{-4}$), Theorem 1 delivers an accurate estimation of token-level entropy dynamics. We draw three important remarks:

Remark 1: While there is a prior work (Cui et al., 2025b) that also considers entropy change, it assumes a uniform entropy distribution across different queries within the same batch. This assumption is rarely attainable in practice, which incurs inaccurate approximation of the ground-truth entropy change. To quantify, we compare our entropy change estimation with their estimation (denoted as *Cov*) during a standard GRPO training process. Table 1 reports the Mean Squared Error

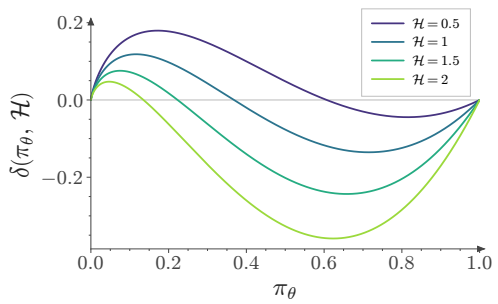


Figure 1: δ as a function of π_θ and $\mathcal{H}(s)$.

Model	Method	MSE↓	PCC↑	SRCC↑
Math-1.5B	Cov	5.37	−6e−5	+0.04
	Ours	**5e−4**	**+0.42**	**+0.65**
Qwen-7B	Cov	0.53	+0.05	+0.08
	Ours	**8e−4**	**+0.39**	**+0.72**
Math-7B	Cov	0.29	+0.03	+0.06
	Ours	**4e−4**	**+0.42**	**+0.61**

Table 1: Comparison of MSE, PCC and SRCC between covariance-based estimation (Cov) and ours.

(*MSE*), Pearson Correlation Coefficient (*PCC*), and Spearman's Rank Correlation Coefficient (*SRCC*) between ground-truth entropy change and estimation. Across all three metrics, $\Omega(s)$ delivers orders-of-magnitude lower MSE and substantially higher *PCC* and *SRCC* than *Cov*. More results are shown in Appendix D.2.

Remark 2: Theorem 1 implies token-level entropy change is jointly determined by multiple factors: ❶ the clip indicator $\mathbb{I}_{\text{clip}}$, which prevents the entropy change of the tokens with overly large or small importance sampling ratio; ❷ the advantage A and the old-policy term π_{old}, which act as weighting factors for entropy change; ❸ the token generation probability π_θ and ❹ the token entropy $\mathcal{H}(s)$ of current state. The contribution of π_θ and $\mathcal{H}(s)$ to entropy change can be expressed with the following function:

$$\delta(\pi_\theta, \mathcal{H}) \triangleq -\pi_\theta(1-\pi_\theta)\big[\log(\pi_\theta) + \mathcal{H}(s)\big]. \quad (8)$$

the impact of π_θ and $\mathcal{H}(s)$ on the function $\delta(\pi_\theta, \mathcal{H}(s))$ are illustrated in Figure 1.

Remark 3: We further investigate the direction of entropy change (increase or decrease), which is controlled by the joint effect of the advantage and token probability. Given that the function $\delta(\pi_\theta, \mathcal{H})$ takes negative values for high-probability tokens and positive values for low policy probability, we conceptualize this joint effect in terms of four qualitative quadrants in the (A, π_θ) space shown in Figure 2:

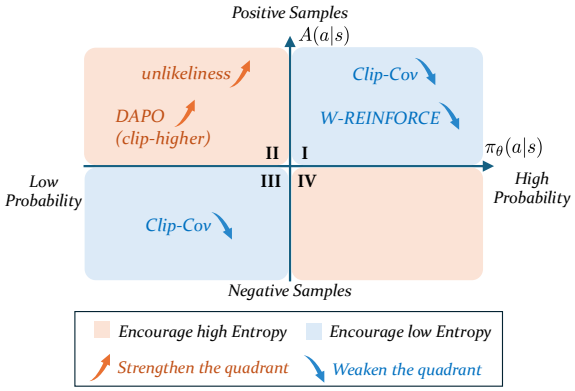


Figure 2: Four-quadrant view of entropy change direction in the advantage–probability plane. The red/blue arrows respectively strengthen/weaken the corresponding quadrant's entropy effect. Existing methods are mapped into the quadrants where they intervene.

Quadrant I: Exploitation (entropy decrease). For high-probability tokens in correct outputs ($A > 0, \delta < 0$), rewarding a well-learned behavior concentrates probability mass, thus *decreasing* entropy.

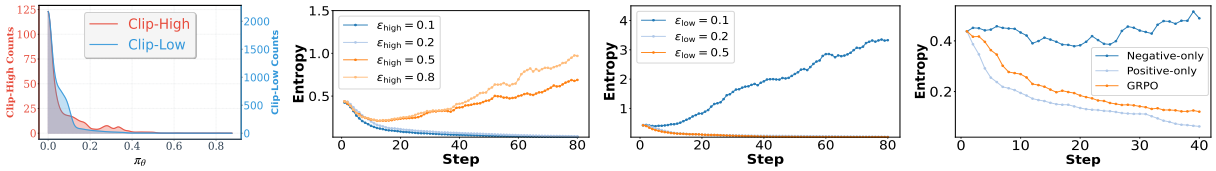
Quadrant II: Exploration (entropy increase). For low-probability tokens in correct outputs ($A > 0, \delta > 0$), rewarding a rare-but-correct behavior diversifies the policy, thereby *increasing* entropy.

Quadrant III: Suppression (entropy decrease). For low-probability tokens in incorrect outputs ($A < 0, \delta > 0$), promoting exploration of alternative responses and thereby *decreasing* entropy.

Quadrant IV: Error-Correction (entropy increase). For high-probability tokens in incorrect outputs ($A < 0, \delta < 0$), penalizing overconfident errors flattens the distribution to encourage seeking alternatives and tends to *increase* entropy.

We further perform empirical analyses by intervening in the samples on each quadrant. Our empirical observations are closely aligned with our theoretical findings. Readers may refer to Appendix D.4 for additional details.

In a standard RLVR process, these four quadrant-level dynamics co-exist as competing forces that shape the policy. The global policy entropy evolves from the combined effect of these competing updates. Consequently, entropy collapse can be understood as a state where the exploitation-driven, entropy-decreasing updates (Quadrants I and III) consistently overwhelm the exploration-driven, entropy-increasing updates (Quadrants II and IV).



(a) Clipping statistics. (b) Entropy effect of $\varepsilon_{\text{high}}$ (c) Entropy effect of ε_{low} (d) Positives/Negatives training

Figure 3: Empirical validation of entropy-change mechanisms via ratio clipping and PSR/NSR reweighting.

4 Analyses on Existing Entropy Intervention Methods

Building on the above theoretical findings, we conduct a comprehensive analysis of existing entropy-intervention techniques to reveal their underlying mechanisms and limitations.

Entropy Effect of Clip-Higher. Recent work (Yu et al., 2025) considers to decouple the lower and higher clipping bounds with $\varepsilon_{\text{high}}$ and ε_{low}, and demonstrate that increasing $\varepsilon_{\text{high}}$ can mitigate entropy collapse.

Mechanism: The underlying mechanism can be explained as follows: from the importance ratio $r = \frac{\pi_\theta}{\pi_{\text{old}}}$, the ratio is more likely to attain a large value when π_{old} is relatively small. Consequently, clipping is predominantly triggered for low-probability tokens, a phenomenon confirmed by our empirical observations in Figure 3a, which counts the clipping events in the first 10 steps of GRPO. In this case, the clipping via $\varepsilon_{\text{high}}$ acts as a filter that removes a considerable number of low-probability positive samples, which typically fall within Quadrant II. Increasing $\varepsilon_{\text{high}}$ therefore reduces the number of filtered instances, allowing more samples to contribute to entropy increase, thereby mitigating collapse. A similar reasoning applies to the role of ε_{low}: increasing ε_{low} filters fewer instances in Quadrant III which in turn exacerbates entropy collapse.

Empirical Evidence: We further validate these observations through empirical experiments. Figure 3b and Figure 3c illustrate the impact of varying $\varepsilon_{\text{high}}$ and ε_{low}. We observe that increasing $\varepsilon_{\text{high}}$ can mitigate or even reverses entropy collapse, while increasing ε_{low} intensifies collapse.

Limitation: Briefly, the effect of tuning clipping thresholds on entropy dynamics can be interpreted as reweighting tokens in Quadrant II and Quadrant III in Figure 2. However, such heuristic methods remain coarse-grained: DAPO, for example, seeks to affect entropy by controlling the updates of some tokens in Quadrant II, without explicitly controlling tokens in other quadrants.

Entropy Effect of Re-weighting Positives. Recent studies have shown that re-weighting positive samples can mitigate entropy collapse. For example, unlikeliness (He et al., 2025a) up-weighting tokens in Quadrant II and down-weighting tokens in Quadrant I, while W-REINFORCE (Zhu et al., 2025) down-weighting all positives in training.

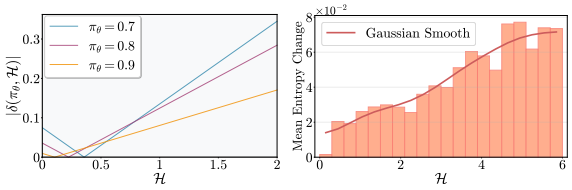
Mechanism: The effect of unlikeliness is clear from Figure 2: it strengthens the entropy-increasing quadrant and weakens entropy-decreasing quadrant, thereby increasing policy entropy. As for W-REINFORCE, the key insight is that token-level updates are dominated by high-probability tokens since they are more likely to be sampled in reasoning. Therefore, down-weighting all positives primarily weakens the entropy-decreasing contribution from Quadrant I, thereby increasing policy entropy.

Empirical Evidence: To validate this mechanism, we conduct positive-only and negative-only experiments in GRPO setting, where the policy is trained using only positive or only negative samples, respectively. The results, shown in Figure 3d, are consistent with the mechanism: training only on positive samples rapidly collapses policy entropy, while training only on negative samples sustains consistently high entropy.

Limitation: While these positive re-weighting methods improve global entropy, they only reweight a subset of tokens, leaving the remaining tokens still vulnerable to entropy collapse.

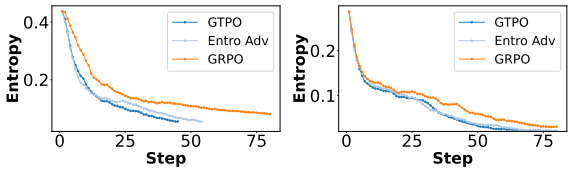
Entropy Effect of Entropy-aware Advantage. Several studies have proposed incorporating entropy-related terms into the advantage function to mitigate entropy collapse, such as Entro. Adv. (Cheng et al., 2025) and GTPO (Tan and Pan, 2025). Such methods typically assign larger advantages to tokens with higher entropy. However, our findings indicate that these methods are not universally effective; in fact, they can sometimes accelerate entropy collapse rather than prevent it.

Mechanism: As shown by $\delta(a|s)$ as a function of $\mathcal{H}(\pi_\theta \mid s)$ in Figure 4a, high-entropy tokens tend



(a) Theoretical correlation (b) Empirical behavior

Figure 4: Entropy and entropy change.



(a) On dataset DAPO-17k (b) On dataset Math

Figure 5: Entropy dynamics with entropy-aware advantage shaping on Qwen2.5-Math-7B backbone.

to induce larger changes in entropy. Consequently, assigning greater advantages to high-entropy tokens amplifies their influence. If these tokens tend to decrease entropy, their amplified contribution intensifies the collapse instead of mitigating it.

Empirical Evidence: We validate this mechanism by examining the entropy dynamics of these methods in Figure 5: compared to the standard GRPO baseline, Entro. Adv. and GTPO both exhibit faster entropy collapse when the policy enters an entropy-decreasing phase. When policy entropy starts to decline, these methods can even accelerate entropy collapse.

Limitation: This finding highlights a key flaw in these methods: rather than reliably encouraging exploration, they may instead aggravate entropy decline.

Limitations of Existing Methods. While the aforementioned entropy-intervention strategies can partly alleviate entropy collapse in practice, they remain largely heuristic and lack principled guidance. These strategies involve only qualitative adjustments to one or two factors in Eq. 7, leaving other relevant factors unconsidered and failing to capture their joint impacts on the entropy dynamics. As a result, there is a substantial gap between their interventions and the actual entropy evolution observed during training. Their effectiveness is compromised.

5 Stabilizing Token-level Entropy-change via Reweighting

The preceding analysis motivates us to develop a more fine-grained and theoretically-driven strategy, termed STEER. The key idea is to introduce an adaptive weight $\lambda(s)$ for each token in GRPO to regulate its entropy change.

Method	$\mathbb{I}_{\text{clip}}$	π_θ	A	$\mathcal{H}(s)$
DAPO	✓	✗	✗	✗
Unlikeliness	✗	✓	✓	✗
W-REINFORCE	✗	✗	✓	✗
Entropy Adv.	✗	✗	✓	✓
KL Reg.	✗	✓	✗	✗
Entropy Reg.	✗	✗	✗	✓
Edge-GRPO	✗	✗	✗	✓
Forking Tokens	✗	✗	✗	✓
Clip-Cov	✗	✓	✓	✗
STEER	✓	✓	✓	✓

Table 2: Factors considered by existing entropy-intervention methods and ours.

With STEER, the entropy change associated with each token is modulated from $\Omega(s)$ to $\lambda(s)\Omega(s)$, thus enabling fine-grained, per-token entropy modulation rather than only monitoring the global expectation as in prior methods.

The central question becomes how to design a suitable $\lambda(s)$. Naturally, $\lambda(s)$ should vary according to $\Omega(s)$ — larger entropy changes indicate a higher risk of entropy collapse and should be attenuated. To achieve this, we employ an exponential decay function:

$$\lambda(s) = \exp\left(-\alpha \frac{|\Omega(s)|}{\max_{s \in \mathcal{B}} |\Omega(s)|} \right), \quad (9)$$

where $\alpha > 0$ is the hyperparameter controlling the decay rate. This formulation satisfies two desired properties: (1) it is monotonically decreasing with respect to the normalized entropy change; (2) the introduction of the exponential ensures that all weights remain strictly positive. We apply the absolute value $|\Omega(s)|$ to measure the magnitude of entropy-change, since large entropy increases may also be sub-optimal. Controlling both increases and decreases within a stable range leads to more stable training dynamics. The normalization by the batch maximum ensures numerical stability across varying magnitudes of entropy change.

The hyperparameter α governs the slope of the decay curve. Equivalently, α determines the minimum attainable weight, $\lambda_{\min} = \exp(-\alpha)$. Thus, Eq.(9) can be understood as an inverse mapping of entropy changes into the range $[\lambda_{\min}, 1]$, with larger entropy changes corresponding to smaller weights. In practice, we find it more convenient to tune $\lambda_{\min}$ directly, rather than α, due to its better

Table 3: Benchmark results of different methods. We report avg@32 for AIME24, AIME25, and AMC23 and avg@1 for others. All results are presented as percentages.

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
Qwen2.5-Math-7B	13.8	5.3	44.6	39.6	9.9	13.8	21.2
Classical RLVR Methods							
GRPO	28.0	14.3	66.2	78.6	37.3	40.9	44.2
SimpleRL-Zoo	30.8	14.2	65.4	79.2	37.1	40.8	44.6
Eurus-PRIME	20.9	13.0	65.2	79.8	37.4	40.6	42.8
OPO	32.2	13.4	71.5	82.2	38.2	41.0	46.4
Entropy Intervention Methods							
GRPO w/ clip-high	31.7	12.8	66.8	79.0	38.6	39.3	44.7
GRPO w/ Entro. Loss	29.1	14.0	67.6	80.0	38.2	37.9	44.5
GRPO w/ Fork Tokens	31.9	14.3	65.5	79.2	37.1	40.9	44.8
W-REINFORCE	29.2	12.0	64.9	79.2	37.8	40.9	44.0
Entro. Adv.	27.5	13.5	70.2	79.6	36.8	42.8	45.1
Clip–Cov	32.5	12.9	68.4	78.0	40.8	41.3	45.7
KL–Cov	32.8	14.1	64.2	78.8	37.1	39.4	44.4
STEER (Ours)	**36.2**	**16.1**	**72.1**	**82.2**	**41.7**	**43.0**	**48.6**

interpretability in controlling the minimum modulation weight.

6 Experiments and Results

In this section, we evaluate STEER from the following perspectives. First, we compare STEER against strong baselines on six math reasoning benchmarks and three coding benchmarks. Second, we examine STEER's entropy control ability in RLVR training. Third, we test STEER's generalization across multiple model scales, families, and RL algorithms. Further results and detailed analyses are deferred to Appendix F.

6.1 Experimental Setup

Training. We follow the default training recipe of standard GRPO in verl (Sheng et al., 2025). For fair comparisons, we closely follow the experimental setup used in recent work(Yu et al., 2025; Wang et al., 2025b). To ensure the reliability, all results reported in the main experiments are averaged over two independent training.

For math reasoning task, we conduct experiments on three different models, including Qwen2.5-Math-1.5B/7B, Qwen2.5-14B (Yang et al., 2024a) and Llama-3.2-3B (Grattafiori et al., 2024a). Our training data is DAPO-Math-17k (Yu et al., 2025), containing only math problems with integer ground-truth answers.

For coding tasks, we conduct comparison experiments on three different models, including Qwen2.5-Coder-3B/7B/14B (Hui et al., 2024) and Mistral-7B (Jiang et al., 2023b). For code generation task, we adopt ArcherCodeR[1] for RLVR training. As for code editing task, since there is no large-scale open-sourced dataset is available, we adopt our in-house practical code editing benchmark for training.

Evaluation. For math reasoning task, we evaluate our models and baselines on six widely used mathematical reasoning benchmarks: AIME24, AIME25, AMC23, MATH-500, Minerva Math, and OlympiadBench. For code generation task, we adopt the widely used LiveCodeBench v5 (Jain et al., 2024) for the evaluation of code generation. We report avg@4 for code generation task following (Wang et al., 2025c). For code editing, we evaluate the model on both our internal held-out test split (3314 cases) and Zeta (Zed Industries, 2025). We report exact-match accuracy for measuring code edit task. More training and evaluation details of our method and baselines are listed in Appendix E.

Baselines. For a thorough comparison, we compare our method against 10 baselines, including standard GRPO (Shao et al., 2024), SimpleRL-Zoo (Zeng et al., 2025), Eurus-PRIME (Cui et al., 2025a), OPO (Hao et al., 2025), GRPO with clip-high (Yu et al., 2025), GRPO with entropy loss (Schulman et al., 2017), GRPO with Fork Tokens (Wang et al., 2025d), W-REINFORCE (Zhu et al., 2025), Entro. Adv. (Cheng et al., 2025), Clip-Cov and KL-Cov (Cui et al., 2025b). For all baselines, the default training hyperparameters

[1]Available at https://huggingface.co/datasets/Fate-Zero/ArcherCodeR-Dataset.

Table 4: Benchmark results of different methods on Qwen2.5-14B. We report avg@32 for AIME24, AIME25, and AMC23 and avg@1 for others. All results are presented as percentages.

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
Base	3.9	2.6	25.8	52.6	15.4	23.0	20.6
GRPO	17.2	13.2	66.3	80.6	38.0	42.2	42.9
OPO	17.8	12.6	68.2	78.6	37.7	42.6	42.9
Entro. Adv.	14.6	9.8	65.6	78.8	36.5	40.9	41.0
Clip-Cov	14.1	13.6	59.8	78.2	38.6	43.2	41.2
STEER	**19.3**	**14.3**	**70.3**	**81.4**	**39.4**	**46.7**	**45.2**

in RLVR are consistent with STEER, while the newly introduced hyperparameters follow are configured following the original implementations, respectively.

Dataset	Method	3B	7B	14B
Internal	GRPO	39.7	40.1	42.6
	STEER	**41.2**	**41.9**	**45.1**
Zeta	GRPO	17.4	22.0	22.3
	STEER	**19.3**	**24.0**	**24.1**
LCB-v5	GRPO	24.4	28.5	29.3
	STEER	**24.9**	**29.2**	**31.8**

Table 5: Performance on real-world coding tasks.

6.2 Main Results

Math Reasoning Tasks. The main results in math reasoning task are shown in Table 3. STEER outperforms classical RLVR baselines as well as existing entropy intervention baselines across all datasets. STEER improves average performance by 2.2 points over the second runner-up (OPO) and by 2.9 points over the third runner-up (Clip-Cov) across all baselines. The performance experiments on Qwen2.5-14B shown in Table 4 are compared with the top three competitors in Table 3 (i.e., OPO, Clip-Cov, and Entro. Adv.). STEER also consistently achieves the highest average performance, demonstrating its superior generalization in improving math reasoning abilities.

Beyond metric avg@32 and avg@1, the *pass@256/512/1024* results in math reasoning are shown in Figure 6. As observed, STEER delivers the best pass@256/512/1024 on both AIME24/25 and the highest average. By explicitly regulating entropy, STEER preserves sufficient exploration to discover more informative trajectories, thereby achieving better performance.

Coding Tasks. Beyond math reasoning, we evaluate our method on real-world code tasks. Table 5 compares STEER with GRPO on three Qwen2.5-coder models. We observe that STEER exceeds GRPO at least 1% in each model and test set. These results demonstrate the superior performance of

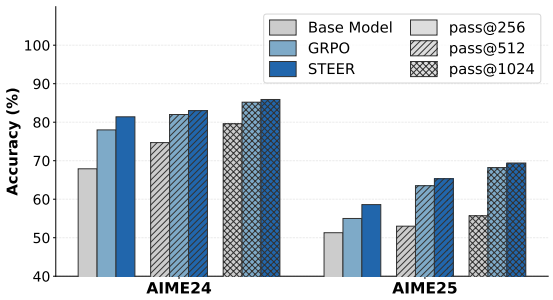


Figure 6: *Pass@256/512/1024* performance on AIME24 and AIME25.

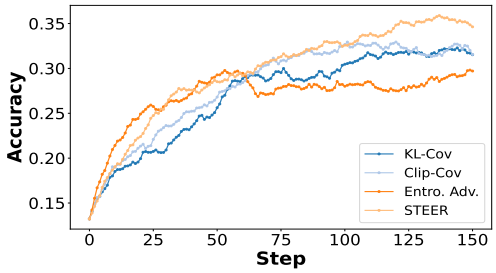


Figure 7: Test accuracy dynamics comparison.

STEER in various scenarios. More performance evaluation are detailed in Appendix F.

Training Dynamics. Figure 7 shows the curves of test accuracy during training. Notably, STEER exhibits a stable upward trajectory, ultimately achieving superior final performance compared to the baselines.

6.3 Empirical Results for Entropy Modulation

The strength of our method is not only reflected in its performance but also in its ability to regulate entropy across a wide range. We consider an extreme training setup with $\varepsilon_{\text{high}} = 5$ and $\varepsilon_{\text{low}} = 0.99$, where almost no ratio clipping is applied. In such scenarios, RL training is vulnerable due to unstable gradient updates under extreme clipping ratios. The results are shown in Figure 8. Most methods fail to maintain stable entropy: GRPO and Entro. Adv. tend toward entropy collapse; adding an Entropy Loss drives entropy up rapidly, leading to excessive uncertainty; and Clip-Cov cannot reliably control entropy. By contrast, STEER stabilizes after an initial decline and maintains steady entropy sub-

Table 6: Performance on test datasets in extreme scenarios.

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
GRPO	31.6	12.8	66.7	79.0	39.3	40.1	44.9
Entro. Adv.	34.8	13.4	64.3	77.6	37.6	39.9	44.6
Entro. Loss	32.7	14.7	71.3	79.0	36.8	41.4	46.0
Clip-Cov	30.4	14.0	72.3	79.6	37.1	41.7	45.8
STEER (Ours)	**36.1**	**16.0**	**76.3**	**80.5**	**39.5**	**42.3**	**48.5**

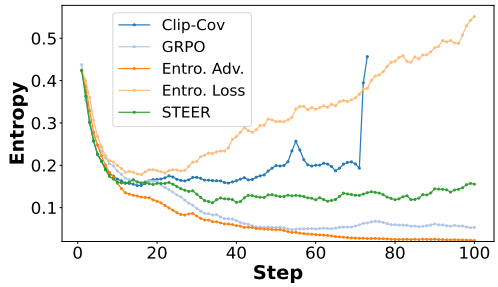


Figure 8: Entropy dynamics in extreme scenarios.

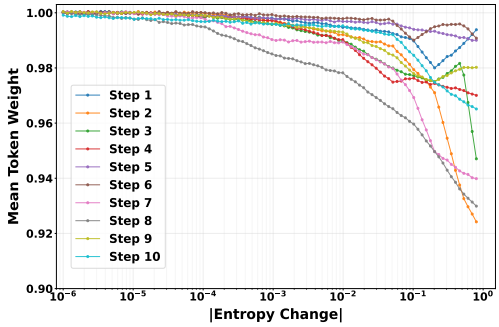


Figure 9: Relationship between mean token weight and entropy change across steps.

sequently. Besides, Figure 9 depicts the trend of average token weight as a function of the absolute token entropy change in the first 10 steps. When entropy changes are small, most weights remain near 1; only tokens with large entropy changes receive substantially reduced weights, indicating that STEER stabilizes training without impeding learning.

We test entropy intervention methods in uncontrolled training scenarios ($\varepsilon_{\text{high}} = 5$ and $\varepsilon_{\text{low}} = 0.99$), with test set accuracy shown in Table 6. It can be observed that, even in training scenarios where the clipping operation is almost completely removed, STEER maintains relatively stable performance compared to other entropy intervention methods and achieves the highest accuracy across all test sets.

6.4 Generalization Study

We add experiments on additional backbones, including Qwen2.5-Math-1.5B, Llama-3.2-3B-Instruct (Grattafiori et al., 2024b) for math reasoning and Mistral-7B-v0.3 (Jiang et al., 2023a) for code editing. For each task, we use the same STEER hyperparameter as in the main experiments, rather than re-tuned per backbone. The results are reported in Appendix F.2, and we find that STEER yields consistent performance improvements.

Besides, we evaluate STEER under RLOO and OPO algorithms with the default setting in verl implementations. The results are also shown in reported in Appendix F.2. This indicates the effectiveness of STEER can be adopted to other RLVR algorithms rather than specific to GRPO.

7 Conclusion

We conduct comprehensive theoretical and empirical analyses of entropy dynamics in RLVR. We first derive an accurate approximate expression for token-level entropy change at each update step, providing four key factors including clipping strategy, advantage, token probability, and conditional entropy. We then clarify how existing methods influence these factors, thus explaining their empirical success and potential limitations. Driven by our theoretical insights, we introduce STEER, a principled entropy-modulation method that adaptively reweighs tokens based on their estimated entropy change. Experiments show STEER outperforms existing strong baselines across multiple benchmarks.

8 Limitation

This study focuses on verifiable rewards, while RL training without verifiable rewards are not included.

9 Acknowledgements

This work is supported by the Starry Night Science Fund of Zhejiang University Shanghai Institute for Advanced Study (SN-ZJU-SIAS-001), and the National Natural Science Foundation of China (62476244,62372399). We also sincerely thank Tencent CodeBuddy (https://www.codebuddy.ai/), especially Yi Liu and the CodeBuddy Team, for their generous support and valuable assistance throughout this 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 (Volume 1: Long Papers)*, pages 12248–12267.

Shengnan An, Xunliang Cai, Xuezhi Cao, Xiaoyu Li, Yehao Lin, Junlin Liu, Xinxuan Lv, Dan Ma, Xuanlin Wang, Ziwen Wang, and 1 others. 2025. Amo-bench: Large language models still struggle in high school math competitions. *arXiv preprint arXiv:2510.26768*.

Anthropic. 2025. Introducing claude opus 4.5. https://www.anthropic.com/news/claude-opus-4-5.

Sikai Bai, Haoxi Li, Jie Zhang, Yongjiang Liu, and Song Guo. 2026. Ttvs: Boosting self-exploring reinforcement learning via test-time variational synthesis. *arXiv preprint arXiv:2604.08468*.

Yupeng Chang, Yi Chang, and Yuan Wu. 2026. BAloRA: Bias-alleviating low-rank adaptation to mitigate catastrophic inheritance in large language models. In *The Fourteenth International Conference on Learning Representations*.

Yupeng Chang, Chenlu Guo, Yi Chang, and Yuan Wu. 2025. Lora-mgpo: Mitigating double descent in low-rank adaptation via momentum-guided perturbation optimization. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 648–659.

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, and 1 others. 2024. A survey on evaluation of large language models. *ACM transactions on intelligent systems and technology*, 15(3):1–45.

Jiawei Chen, Hande Dong, Xiang Wang, Fuli Feng, Meng Wang, and Xiangnan He. 2023. Bias and debias in recommender system: A survey and future directions. *ACM Transactions on Information Systems*, 41(3):1–39.

Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. 2025. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025a. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, and 1 others. 2025b. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*.

Yu Cui, Feng Liu, Jiawei Chen, Canghong Jin, Xingyu Lou, Changwang Zhang, Jun Wang, Yuegang Sun, and Can Wang. 2025c. Hatllm: Hierarchical attention masking for enhanced collaborative modeling in llm-based recommendation. *arXiv preprint arXiv:2510.10955*.

Jia Deng, Jie Chen, Zhipeng Chen, Wayne Xin Zhao, and Ji-Rong Wen. 2025. Decomposing the entropy-performance exchange: The missing keys to unlocking effective reinforcement learning. *arXiv preprint arXiv:2508.02260*.

Chongming Gao, Kexin Huang, Jiawei Chen, Yuan Zhang, Biao Li, Peng Jiang, Shiqi Wang, Zhong Zhang, and Xiangnan He. 2023a. Alleviating matthew effect of offline reinforcement learning in interactive recommendation. In *Proceedings of the 46th international ACM SIGIR conference on research and development in information retrieval*, pages 238–248.

Chongming Gao, Shiqi Wang, Shijun Li, Jiawei Chen, Xiangnan He, Wenqiang Lei, Biao Li, Yuan Zhang, and Peng Jiang. 2023b. Cirs: Bursting filter bubbles by counterfactual interactive recommender system. *ACM Transactions on Information Systems*, 42(1):1–27.

Google. 2025. A new era of intelligence with gemini 3. https://blog.google/products/gemini/gemini-3/.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024a. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Aaron Grattafiori and 1 others. 2024b. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr.

Yaru Hao, Li Dong, Xun Wu, Shaohan Huang, Zewen Chi, and Furu Wei. 2025. On-policy rl with optimal reward baseline. *arXiv preprint arXiv:2505.23585*.

Andre He, Daniel Fried, and Sean Welleck. 2025a. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, and 1 others. 2025b. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Jian Hu, Jason Klein Liu, Haotian Xu, and Wei Shen. 2025a. Reinforce++: An efficient rlhf algorithm with robustness to both prompt and reward models. *arXiv preprint arXiv:2501.03262*, 1(3):5.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. 2025b. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2.5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2024. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*.

Albert Q. Jiang and 1 others. 2023a. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Yihang Jiang, Xiaoyang Li, Guangxu Zhu, Hang Li, Jing Deng, Kaifeng Han, Chao Shen, Qingjiang Shi, and Rui Zhang. 2023b. 6g non-terrestrial networks enabled low-altitude economy: Opportunities and challenges. *arXiv preprint arXiv:2311.09047*.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, and 1 others. 2024. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, and 1 others. 2022. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, and 1 others. 2024. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9.

Mengdi Li, Jiaye Lin, Xufeng Zhao, Wenhao Lu, Peilin Zhao, Stefan Wermter, and Di Wang. 2025a. Curriculum-rlaif: Curriculum alignment with reinforcement learning from ai feedback. *arXiv preprint arXiv:2505.20075*.

Qingbin Li, Rongkun Xue, Jie Wang, Ming Zhou, Zhi Li, Xiaofeng Ji, Yongqi Wang, Miao Liu, Zheming Yang, Minghui Qiu, and 1 others. 2025b. Cure: Critical-token-guided re-concatenation for entropy-collapse prevention. *arXiv preprint arXiv:2508.11016*.

Siyi Lin, Chongming Gao, Jiawei Chen, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. 2025. How do recommendation models amplify popularity bias? an analysis from the spectral perspective. In *Proceedings of the Eighteenth ACM International Conference on Web Search and Data Mining*, pages 659–668.

Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, Chen Dong, and 1 others. 2025a. Deepseek-v3. 2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. 2025b. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. *arXiv preprint arXiv:2505.24864*.

Ziyu Liu, Yuhang Zang, Shengyuan Ding, Yuhang Cao, Xiaoyi Dong, Haodong Duan, Dahua Lin, and Jiaqi Wang. 2025c. Spark: Synergistic policy and reward co-evolving framework. *arXiv preprint arXiv:2509.22624*.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, and 1 others. 2025. Deepscaler: Surpassing o1-preview with a 1.5 b model by scaling rl. *Notion Blog*.

Shichao Ma, Zhiyuan Ma, Ming Yang, Xiaofan Li, Xing Wu, Jintao Du, Yu Cheng, Weiqiang Wang, Qiliang Liu, Zhengyang Zhou, and 1 others. 2026. Tspo: Breaking the double homogenization dilemma in multi-turn search policy optimization. *arXiv preprint arXiv:2601.22776*.

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

OpenAI. 2025. Introducing gpt-5.2. https://openai.com/index/introducing-gpt-5-2.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

Yuda Song, Julia Kempe, and Remi Munos. 2025. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*.

Hongze Tan and Jianfei Pan. 2025. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Haozhe Wang, Qixin Xu, Che Liu, Junhong Wu, Fangzhen Lin, and Wenhu Chen. 2025a. Emergent hierarchical reasoning in llms through reinforcement learning. *arXiv preprint arXiv:2509.03646*.

Hong Wang, Zhezheng Hao, Jian Luo, Chenxing Wei, Yao Shu, Lei Liu, Qiang Lin, Hande Dong, and Jiawei Chen. 2025b. Scheduling your llm reinforcement learning with reasoning trees. *arXiv preprint arXiv:2510.24832*.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. 2025c. Stabilizing knowledge, promoting reasoning: Dual-token constraints for rlvr. *arXiv preprint arXiv:2507.15778*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2025d. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Ziqing Wang, Yibo Wen, William Pattie, Xiao Luo, Weimin Wu, Jerry Yao-Chieh Hu, Abhishek Pandey, Han Liu, and Kaize Ding. 2025e. Polo: Preference-guided multi-turn reinforcement learning for lead optimization. *arXiv preprint arXiv:2509.21737*.

Ziqing Wang, Kexin Zhang, Zihan Zhao, Yibo Wen, Abhishek Pandey, Han Liu, and Kaize Ding. 2025f. A survey of large language models for text-guided molecular discovery: from molecule generation to optimization. *arXiv preprint arXiv:2505.16094*.

Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. 2025a. The invisible leash: Why rlvr may not escape its origin. *arXiv preprint arXiv:2507.14843*.

Fei Wu, Zhenrong Zhang, Qikai Chang, Jianshu Zhang, Quan Liu, and Jun Du. 2026. Step potential advantage estimation: Harnessing intermediate confidence and correctness for efficient mathematical reasoning. *arXiv preprint arXiv:2601.03823*.

Junkang Wu, Kexin Huang, Jiancan Wu, An Zhang, Xiang Wang, and Xiangnan He. 2025b. Quantile advantage estimation for entropy-safe reasoning. *arXiv preprint arXiv:2509.22611*.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025c. Table-r1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*.

Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2025. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024a. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Shihui Yang, Chengfeng Dou, Peidong Guo, Kai Lu, Qiang Ju, Fei Deng, and Rihui Xin. 2025a. Dcpo: Dynamic clipping policy optimization. *arXiv preprint arXiv:2509.02333*.

Weiqin Yang, Jiawei Chen, Xin Xin, Sheng Zhou, Binbin Hu, Yan Feng, Chun Chen, and Can Wang. 2024b. Psl: Rethinking and improving softmax loss from pairwise perspective for recommendation. *Advances in Neural Information Processing Systems*, 37:120974–121006.

Weiqin Yang, Jiawei Chen, Shengjia Zhang, Peng Wu, Yuegang Sun, Yan Feng, Chun Chen, and Can Wang. 2025b. Breaking the top-k barrier: Advancing top-k ranking metrics optimization in recommender systems. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, pages 3542–3552.

Weiqin Yang, Bohao Wang, Zhenxiang Xu, Jiawei Chen, Shengjia Zhang, Jingbang Chen, Canghong Jin, and Can Wang. 2026. Bear: Towards beam-search-aware optimization for recommendation with large language models. *arXiv preprint arXiv:2601.22925*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Zed Industries. 2025. zeta. https://huggingface.co/datasets/zed-industries/zeta.

Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. 2025. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*.

Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, and 1 others. 2025a. A survey of reinforcement learning for large reasoning models. *arXiv preprint arXiv:2509.08827*.

Kangning Zhang, Wenxiang Jiao, Kounianhua Du, Yuan Lu, Weiwen Liu, Weinan Zhang, and Yong Yu. 2025b. Looptool: Closing the data-training loop for robust llm tool calls. *arXiv preprint arXiv:2511.09148*.

Ruipeng Zhang, Ya-Chien Chang, and Sicun Gao. 2025c. When maximum entropy misleads policy optimization. *arXiv preprint arXiv:2506.05615*.

Shengjia Zhang, Weiqin Yang, Jiawei Chen, Peng Wu, Yuegang Sun, Gang Wang, Qihao Shi, and Can Wang. 2026a. Talos: Optimizing top-k accuracy in recommender systems. *arXiv preprint arXiv:2601.19276*.

Wenyuan Zhang, Xinghua Zhang, Haiyang Yu, Shuaiyi Nie, Bingli Wu, Juwei Yue, Tingwen Liu, and Yongbin Li. 2026b. Expseek: Self-triggered experience seeking for web agents. *Preprint*, arXiv:2601.08605.

Xingjian Zhang, Siwei Wen, Wenjun Wu, and Lei Huang. 2025d. Edge-grpo: Entropy-driven grpo with guided error correction for advantage diversity. *arXiv preprint arXiv:2507.21848*.

Xinglang Zhang, Yunyao Zhang, ZeLiang Chen, Junqing Yu, Wei Yang, and Zikai Song. 2026c. Logical phase transitions: Understanding collapse in llm logical reasoning. *Preprint*, arXiv:2601.02902.

Yunyao Zhang, Xinglang Zhang, Junxi Sheng, Wenbing Li, Junqing Yu, Yi-Ping Phoebe Chen, Wei Yang, and Zikai Song. 2026d. Semantic-aware logical reasoning via a semiotic framework. *Preprint*, arXiv:2509.24765.

Ziqi Zhao, Zhaochun Ren, Jiahong Zou, Liu Yang, Zhiwei Xu, Xuri Ge, Zhumin Chen, Xinyu Ma, Daiting Shi, Shuaiqiang Wang, and 1 others. 2026. Reinforced efficient reasoning via semantically diverse exploration. *arXiv preprint arXiv:2601.05053*.

Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. 2026. Look inward to explore outward: Learning temperature policy from llm internal states via hierarchical rl. *arXiv preprint arXiv:2602.13035*.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*.

Brian D Ziebart, Andrew Maas, J Andrew Bagnell, and Anind K Dey. 2008. Maximum entropy inverse reinforcement learning. In *AAAI*, volume 8, pages 1433–1438.

Contents

A Usage of LLMs

Throughout the preparation of this manuscript, Large Language Models (LLMs) were utilized as a writing and editing tool. Specifically, we employed LLMs to improve the clarity and readability of the text, refine sentence structures, and correct grammatical errors. All final content, including the core scientific claims, experimental design, and conclusions, was conceived and written by us, and we take full responsibility for the final version of this paper.

B Related Work

B.1 Reinforcement Learning with Verifiable Rewards in LLMs

Reinforcement Learning with Verifiable Rewards (RLVR) (Chang et al., 2024), in which rewards are derived from a rule-based verifier, has recently emerged as an effective paradigm for enhancing the reasoning performance of large language models (Zhang et al., 2025a; Jaech et al., 2024; Lambert et al., 2024; Gao et al., 2023a; Wang et al., 2025e; Liu et al., 2025c) beyond SFT (Chang et al., 2025, 2026). In most RLVR settings, the reward is determined by checking whether the model output matches a reference answer, typically yielding a binary signal that reflects success or failure. The development of core algorithms, notably PPO and GRPO (Shao et al., 2024), has substantially advanced RLVR. Several follow-up works further refine the optimization procedure, including reward-scheduling (Wang et al., 2025b), improved advantage estimation (Wu et al., 2026; Xie et al., 2025), and extensions to multi-turn settings (Ma et al., 2026). Another line of work applies RLVR at scale or to diverse tasks, such as DeepSeek-R1 (Guo et al., 2025), tool-augmented reasoning (Zhang et al., 2025b), and sample-efficient exploration (Zhang et al., 2026b; Zhao et al., 2026). Complementary approaches integrate curriculum learning (Li et al., 2025a), adaptive temperature control (Zhou et al., 2026), and benchmark-driven evaluation (An et al., 2025; Wu et al., 2025c) into the RLVR pipeline. Recent studies also explore the interplay between logical structure and RL dynamics (Zhang et al., 2026d,c; Chen et al., 2023; Lin et al., 2025; Cui et al., 2025c; Wang et al., 2025f; Gao et al., 2023b,a; Bai et al., 2026).

B.2 Entropy-Oriented RL Methods for LLM Reasoning

Entropy regularization (Mnih et al., 2016; Haarnoja et al., 2018; Chen et al., 2023; Lin et al., 2025; Gao et al., 2023b), an early line of work in traditional RL, may mislead actions at critical states (Zhang et al., 2025c) and has been shown to be highly sensitive to the coefficient in LLM training (Cheng et al., 2025; Cui et al., 2025b,c; Yang et al., 2024b, 2025b; Zhang et al., 2026a; Yang et al., 2026). (Liu et al., 2025b) argues that the KL penalty preserves entropy and acts as a regularizer, ensuring that the online policy remains close to a stable reference, which stabilizes learning and reduces overfitting to misleading reward signals. Nevertheless, the KL divergence term between the current policy π_θ and the reference policy π_{ref} in the original form (Shao et al., 2024) is excluded in our work, since its practical impact is often negligible or counterproductive for reasoning tasks, as demonstrated in recent works (Yu et al., 2025; Chu et al., 2025; Hu et al., 2025b). One typical approach to address entropy collapse is by raising the sampling temperature during inference. However, recent findings in (Luo et al., 2025) suggest that while this method postpones the onset of entropy collapse, it does not prevent it, as entropy continues to decrease progressively throughout the training process. Recent studies have sought to mitigate entropy collapse by adjusting key elements of policy optimization, such as PPO-style ratio clipping (Yu et al., 2025; Yang et al., 2025a), balancing positive and negative samples (Zhu et al., 2025), and applying KL regularization (Liu et al., 2025b). However, these methods are broad and lack fine-grained control at the token level, with their mechanisms often not fully explained in a unified or principled way. Several methods attempt to encourage exploration via an entropy-induced advantage (Cheng et al., 2025; Tan and Pan, 2025; Wang et al., 2025d,c; Deng et al., 2025), with the intuition that emphasizing uncertain states will promote exploration and raise overall policy entropy. In practice, however, we found this design often fails to reliably mitigate entropy collapse because it disproportionately strengthens learning on high-entropy tokens and thereby magnifies entropy change, leading to unreliable entropy control. Although prior work (Cui et al., 2025b) considers entropy change, the resulting estimation is distorted due to its unreasonable state-equivalence assumption. Notably, its entropy-

control scheme (i) enforces a hard binary split by entropy change without considering their intra-group differentiation, and (ii) may hinder the learning process, since high-entropy-change tokens that are informative for exploration are over-penalized.

C A Token-level Gradient Reweighting Perspective for Shaping Policy Entropy

To summarize existing methods more clearly, we reformulate them through the lens of token-level gradients. Existing entropy intervention methods can be unified into a gradient reweighting framework and subsequently examined their respective impacts on policy entropy.

The policy gradient of off-policy optimization can be expressed as follows:

$$\nabla_\theta J(\theta) = \mathbb{E}_{q\sim\mathcal{D},\, \{o_i\}\sim\pi_{\text{old}}(\cdot|q)} \left[\frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G} \sum_{t=1}^{|o_i|} w_{i,t}(q) \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t}) \right]. \tag{10}$$

For GRPO in Eq. (2), $w_{i,t}(q) = \mathbb{I}_{\text{clip}}\, r_{i,t} A_{i,t}$, where

$$\mathbb{I}_{\text{clip}} = \begin{cases} 0, & A_{i,t} > 0 \text{ and } r_{i,t} > 1+\varepsilon, \\ 0, & A_{i,t} < 0 \text{ and } r_{i,t} < 1-\varepsilon, \\ 1, & \text{otherwise}, \end{cases} \tag{11}$$

where $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\text{old}}(o_{i,t}|q,o_{i,<t})}$ denotes the importance sampling ratio. Advantage $A_{i,t}$ is calculated by reward $R_{i,t}$. For brevity and uniformity, let

$$w_{i,t}(q) = \mathbb{I}_{\text{clip}} r_{i,t} A_{i,t} + \beta \mathcal{R}(\pi_\theta), \tag{12}$$

where $\mathcal{R}(\pi_\theta)$ is the regularization. Table 7 briefly summarizes existing methods based on their interventions on token-level weight $w_{i,t}(q)$. It is evident that existing methods can be categorized into different token-level gradient reweighting schemes, depending on factors such as advantage $A_{i,t}$, generation probability $\pi_\theta(o_{i,t} \mid q, o_{i,<t})$, conditional entropy $\mathcal{H}_{i,t}$, etc. **These methods can be broadly summarized as increasing or suppressing the training weights of tokens that satisfy certain properties.** Our analysis explains why these methods are effective or not. Our proposed STEER adopts reweighting based on token-level entropy change, which is more fundamental for entropy control. Besides, recent studies (Wang et al., 2025d,a) highlight the importance of high-entropy tokens for reasoning and propose various mechanisms to strengthen their training. This does not conflict with our approach STEER as STEER explicitly controls token entropy changes to avert training collapse while preserving learning on critical tokens. This further confirms why (Zhang et al., 2025d) is effective in mitigating entropy collapse.

D Supplementary Empirical Analysis for RLVR Entropy

This section presents supplementary experiments that further support the analyses and claims in the main text.

D.1 Empirical Properties of Policy Entropy

In this subsection, we present several basic empirical properties of policy entropy dynamics in RLVR, providing a global view of how entropy behaves during training.

Figure 10 summarizes several empirical properties of policy entropy in RLVR. First, continuity across batch (panel 1) shows that, under a standard GRPO run, policy entropy decays smoothly step by step and entropy collapse appears as a gradual trend rather than sudden jumps. Second, dependence on model (panel 2) compares different backbones (e.g., Qwen2.5-Math-1.5B / 7B vs. Qwen2.5-7B) and reveals that while all models eventually experience entropy collapse, the initial level and decay speed vary with model size and pretraining. Third, dependence on dataset (panel 3) indicates that training on different math datasets (Math-3to5, DAPO-MATH, DeepScaleR) leads to distinct entropy curves, suggesting that entropy dynamics is also shaped by data distribution and difficulty. Finally, in-domain consistency (panel 4) shows that, within a fixed training run, entropy measured on several test sets in the same domain (Minerva, AMC23, AIME24, AIME25) follows highly similar monotonically decreasing trajectories, implying that a single global entropy trend governs a wide range of tasks. Together, these observations provide an overall picture of how policy entropy behaves in RLVR before any additional intervention is applied.

D.2 Entropy Change Estimation Comparison

To quantify this gap between entropy change estimator and ground-truth entropy change, we compute the Mean Squared Error (*MSE*), Pearson Correlation Coefficient (*PCC*), and Spearman's Rank Correlation Coefficient (*SRCC*) between each estimation and the ground-truth token-level entropy

Method	Intervention
DAPO / DCPO (Yu et al., 2025) (Yang et al., 2025a)	$\mathbb{I}_{\text{clip}} = \begin{cases} 0, & A_{i,t} > 0 \text{ and } r_{i,t} > 1 + \varepsilon_{\text{high}}, \\ 0, & A_{i,t} < 0 \text{ and } r_{i,t} < 1 - \varepsilon_{\text{low}}, \\ 1, & \text{otherwise} \end{cases}$
KL penalty (Shao et al., 2024)	$\mathcal{R}(\pi_\theta) = \frac{\pi_{\text{ref}}(o_{i,t}\|q,o_{i,<t})}{\pi_\theta(o_{i,t}\|q,o_{i,<t})}$
Entropy Regularization (He et al., 2025b)	$\mathcal{R}(\pi_\theta) = -\log \pi_\theta(o_{i,t} \mid q, o_{i,<t})$
Unlikeliness (He et al., 2025a)	$\hat{R}_{i,t} = R_{i,t}\left(1 - \beta_{\text{rank}} \frac{G - \text{rank}(o_i)}{G}\right), \ \beta_{\text{rank}} > 0$
W-REINFORCE (Zhu et al., 2025)	$\hat{A}_{i,t} = \begin{cases} \lambda, & A_{i,t} > 0 \\ 1, & A_{i,t} < 0 \end{cases}, \quad \lambda < 1$
Entropy Advantage (Cheng et al., 2025)	$\hat{A}_{i,t} = A_{i,t} + \min\left(\alpha \cdot \mathcal{H}_{i,t}^{\text{detach}}, \frac{\lvert A_{i,t}\rvert}{\kappa}\right), \quad \alpha > 0, \kappa > 1$
GTPO (Tan and Pan, 2025)	$\hat{R}_{i,t} = R_{i,t} + \alpha \frac{\mathcal{H}_{i,t}}{\frac{1}{d_t}\sum_{k=1}^{d_t} \mathcal{H}_{k,t}}, \text{ for } R_{i,t} > 0$
EDGE-GRPO (Zhang et al., 2025d)	$\hat{A}_i = \frac{A_i}{\hat{\mathcal{H}}_i}, \ \hat{\mathcal{H}}_i$: normalized entropy across the group
PPL-based (Deng et al., 2025)	$\hat{A}_{i,t} = A_{i,t}(1 - \alpha \text{log-PPL}(o_i)), \ \alpha > 0$
Position-based (Deng et al., 2025)	$\hat{A}_{i,t} = A_{i,t} + \gamma \text{sign}(A_{i,t})\sigma(r_{it})$ r_{it}: token's relative position
Forking Tokens (Wang et al., 2025d)	$\mathbb{I}_{\text{clip}} = \mathbb{I}_{\text{clip}} \wedge \mathbb{I}(\mathcal{H}_{i,t} > \tau_{\mathcal{B}}), \ \tau_{\mathcal{B}}$: threshold in batch $\mathcal{B}$

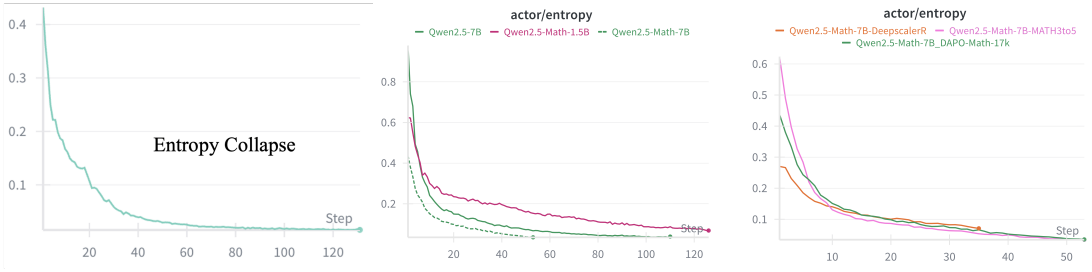
Table 7: A Token-level Gradient Reweighting Perspective for Shaping Policy Entropy.

change, as shown in Figure 1. Across all three metrics, $\Omega_{i,t}$ from Theorem 1 delivers orders-of-magnitude lower MSE and substantially higher *PCC* and *SRCC* than *Cov*. Furthermore, the SRCC between $\Omega_{i,t}$ and the ground-truth token entropy change exceeds 60% across all models, demonstrating a strong rank correlation. These results strongly validate the effectiveness of our estimator derived in Theorem 1 and the soundness of Assumption 1. A more comprehensive comparison is provided below.

We recorded the token entropy changes for the first 10 training steps across different models and datasets. Figure 12 and 13 show the results on dataset DAPO-Math-17k. The curve denotes the estimated vs. ground-truth entropy change (left axis) and histograms show token counts per bin (right axis) It can be observed that our method exhibits a clear positive correlation with ground-truth

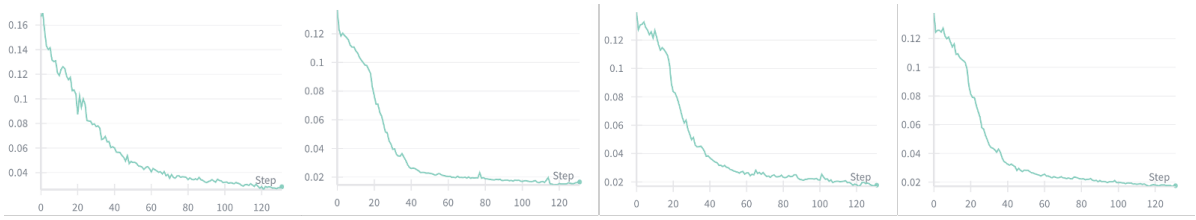
Model	*Method*	*MSE* ↓	*PCC* ↑	*SRCC* ↑
Math-1.5B	*Cov*	5.37	-6e-5	+0.04
	Ours	5e-4	+0.42	+0.65
7B	*Cov*	0.53	+0.05	+0.08
	Ours	8e-4	+0.39	+0.72
Math-7B	*Cov*	0.29	+0.03	+0.06
	Ours	4e-4	+0.42	+0.61

Table 8: Comparison of MSE, PCC and SRCC between covariance-based estimator (Cov) and ours.



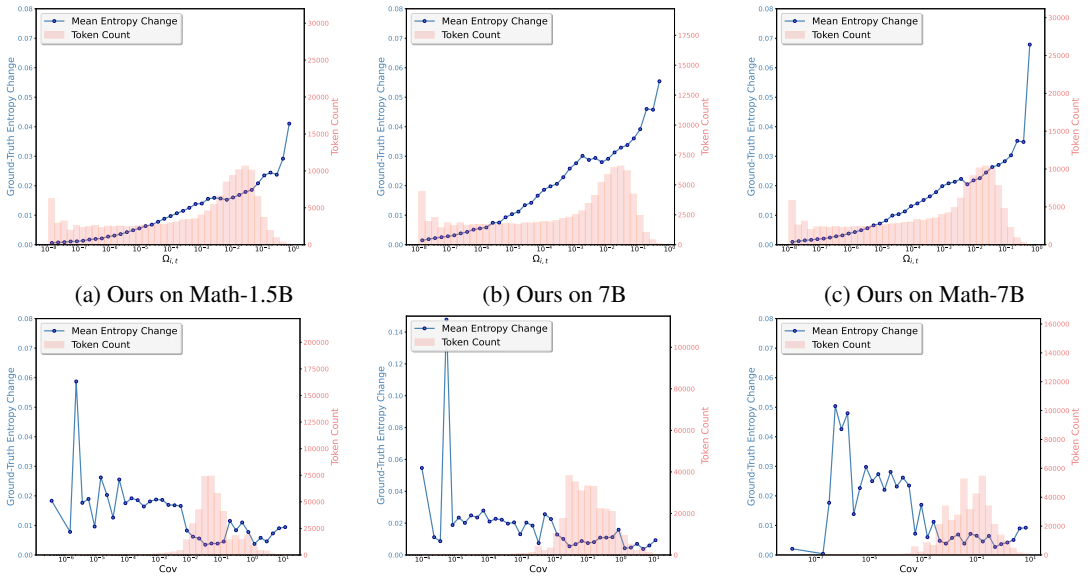
(a) Continuity across batches (b) Dependence on model (c) Dependence on dataset

Figure 10: Empirical properties of policy entropy.



(a) Minerva (b) AMC23 (c) AIME24 (d) AIME25

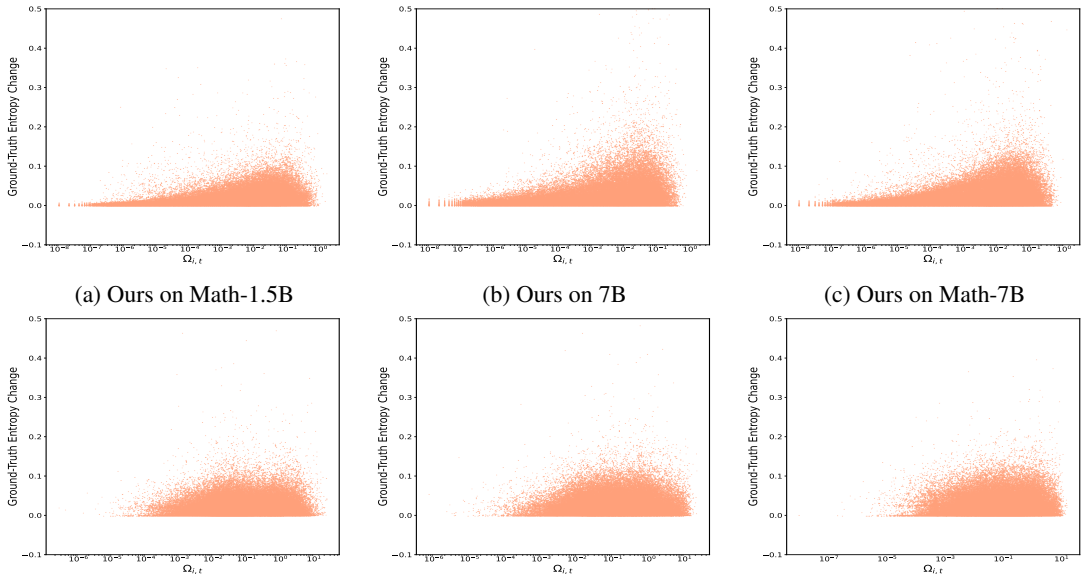
Figure 11: In-domain consistency.



(a) Ours on Math-1.5B (b) Ours on 7B (c) Ours on Math-7B

(d) Cov on Math-7B (e) Cov on Math-7B (f) Cov on Math-7B

Figure 12: Entropy Change on DAPO-Math-17k.



(a) Ours on Math-1.5B (b) Ours on 7B (c) Ours on Math-7B

(d) Cov on Math-1.5B (e) Cov on 7B (f) Cov on Math-7B

Figure 13: Entropy Change scatters on DAPO-Math-17k.

entropy change, which strongly supports our theoretical framework. By contrast, the estimation scheme in (Cui et al., 2025b) exhibits no clear correlation.

D.3 Theoretical analyses extended to other RL algorithms

While our main theoretical analysis is presented under the GRPO update, the derivation **is not specific to GRPO** and can be extended to other policy-gradient RL algorithms. In particular, our derivation is based on the policy-gradient formulation; and the GRPO policy-gradient expression can be adapted to other methods by making minor modifications to the following equation:

$$\nabla_\theta J(\theta) = \mathbb{E}_{q\sim\mathcal{D},\,\{o_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{L}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \mathbb{I}_{\text{clip}} r_{i,t} A_{i,t} \nabla_\theta \log \pi_\theta(o_{i,t} \mid q, o_{i,<t})\right]. \tag{13}$$

where $\mathbb{I}_{\text{clip}}$, $r_{i,t}$, $A_{i,t}$ denote clipping indicator, the importance sampling ratio and advantages, as defined before. Then, different RL algorithms can be recovered by:

- GRPO: $A_{i,t} = A_{\text{group}}$.
- PPO: PPO shares the same clipped-ratio structure as GRPO; the difference is the advantage estimator $A_{i,t} = A_{\text{GAE}}$.
- RLOO (Ahmadian et al., 2024): It is REINFORCE-style with a leave-one-out baseline, i.e., $r_{i,t} = 1$ and

$$A_{i,t} = R_i - \frac{1}{G-1}\sum_{j\neq i} R_j \tag{14}$$

- OPO (Hao et al., 2025): It is on-policy RL with an optimal reward baseline, so typically $r_{i,t} = 1$ and $A_{i,t}$ is a baseline-corrected advantage.

Substituting the above generalized policy-gradient form into the proof of Theorem 1 yields a corresponding generalized estimator of the entropy change:

$$\Omega(s) = -\frac{\eta}{L}\mathbb{E}_{\pi_\theta(\cdot|s)}\Big[\mathbb{I}_{\text{clip}} r_{i,t} A_{i,t} \pi_\theta(1-\pi_\theta)\left(\log \pi_\theta + \mathcal{H}(s)\right)\Big]. \tag{15}$$

This shows that a unified entropy-change estimator can be derived for multiple RL algorithms within the same analytical framework, with the specific choice of advantage function given by the definitions above. Consequently, the theoretical insights developed for GRPO are also applicable to other methods.

D.4 Influencing Entropy Dynamics by Strengthening or Weakening the Quadrants

We next ask whether these theoretical findings of quadrant-level tendencies can be used to actively

steer entropy in practice. Guided by the above quadrant-level tendencies, we design a simple intervention on 10% of tokens in each quadrant to increase entropy: for the entropy-increasing quadrants (II and IV), we double-weight their updates, whereas for the entropy-decreasing quadrants (I and III), we mask their updates, and then track the resulting policy entropy. As shown in Figure 14, all four interventions consistently increase policy entropy compared to the standard GRPO baseline, supporting our analysis. For experiments in Figure 14, we randomly select samples with a generation probability greater than 0.8 and an advantage greater than 0, as well as those with a generation probability less than 0.2 and an advantage less than 0, and randomly mask 10% of such tokens. Similarly, for samples with a generation probability greater than 0.8 and an advantage less than 0, or a generation probability less than 0.2 and an advantage greater than 0, we set the token weight for 10% of such tokens to twice the original token weight.

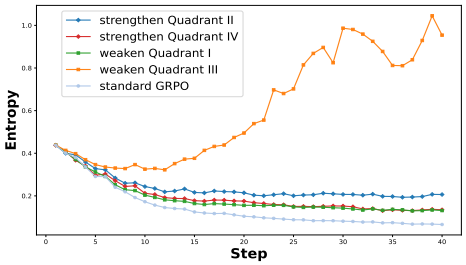


Figure 14: Four schemes to uplift entropy based on advantage-probability effect.

To further validate the patterns of entropy change with advantage and probability in Figure 2, we strengthen (up-weighting) or weaken (masking) each of the four quadrants at different intensities to induce entropy increases or decreases, respectively. Unlike the setup in Figure 14 where 10% tokens are intervened, we present a more comprehensive validation here.

Figure 15 shows interventions applied to each quadrant with the goal of increasing entropy, using standard GRPO ($\varepsilon_{\text{high}=0.2}, \varepsilon_{\text{low}=0.2}$) as the baseline; while Figure 16 presents interventions with the goal of decreasing entropy across the four quadrants, using GRPO w/ clip-high ($\varepsilon_{\text{high}=0.28}, \varepsilon_{\text{low}=0.2}$) as the baseline; In each case, the proportion of tokens masked or up-weighted ranges from 5% to 20%. Across all cases, it can be observed that the token-level intervention effects on entropy align with our quantitative analysis framework, and the impact becomes more pronounced as the intervention ratio increases (from 5% to 20%). For example, in Figure 15b, compared to standard GRPO, up-weighting Quadrant II yields a marked increase in policy entropy over standard GRPO (we exclude the 20% up-weight case because it produces excessively high entropy). This indicates that the clip-high mechanism in DAPO (Yu et al., 2025) and unlikeliness (He et al., 2025a)can be viewed as a special instance of this intervention. In summary, the overall entropy dynamics arise from the joint contributions of the four quadrants; perturbing any one of them can induce a predictable change in the total entropy from our analysis framework.

D.5 Entropy Effect of Clipping Operation

In this subsection, we adjust the clipping thresholds to steer entropy change in GRPO, and evaluate RLVR performance under different entropy levels.

The entropy dynamics induced by clip operation is shown in Figure 17. *Clip-high* ($\varepsilon_{\text{high}}$): entropy decreases in the early phase for all settings; larger ϵ_{high} leads to a clear late-stage entropy rebound and sustained growth, whereas small ϵ_{high} yields continued decay and low final entropy. *Clip-low* (ε_{low}): the behavior is more bifurcated—with $\epsilon_{\text{low}} = 0.1$, entropy increases monotonically over training, while $\epsilon_{\text{low}} \geq 0.2$ drives entropy rapidly toward (near) zero, exhibiting a much stronger tendency toward entropy collapse.

Parameters	0.1	0.2	0.5	0.8
$\varepsilon_{\text{high}}$	44.0	44.2	43.7	42.3
ε_{low}	43.7	44.2	42.5	42.0

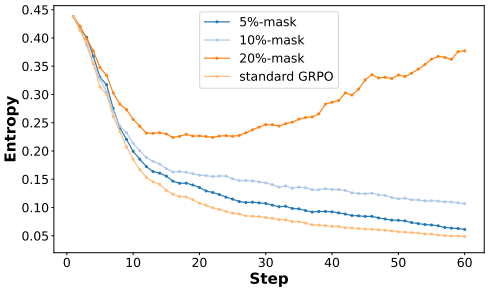
Table 9: Average math reasoning performance with different clip operations.

We evaluate these runs individually; the average math reasoning performance under different clipping operations is reported in Table 9. We observe that performance degrades under both entropy collapse and entropy explosion, whereas maintaining entropy within a stable range yields consistently better results. This highlights the importance of stabilizing entropy dynamics, which is aligned with our proposed STEER.

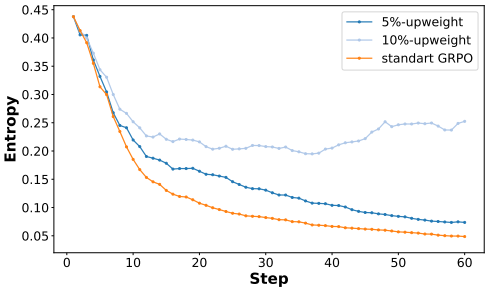
Why large entropy growth can hurt training? This phenomenon can be explained as follows: excessively high entropy makes the policy overly

Table 10: Dataset statistics.

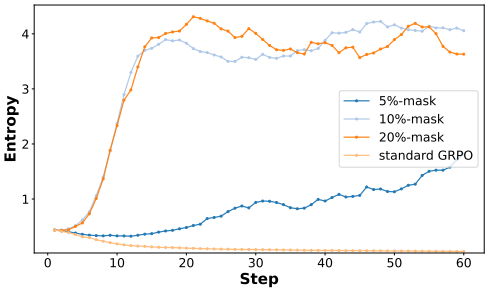
Test Datasets	#Questions	Level
AIME24 (Li et al., 2024)	30	Olympiad
AIME25 (Li et al., 2024)	30	Olympiad
AMC23 (Li et al., 2024)	40	Intermediate
MATH500 (Hendrycks et al., 2021)	500	Advanced
Minerva (Lewkowycz et al., 2022)	272	Graduate
OlympiadBench (He et al., 2024)	675	Olympiad



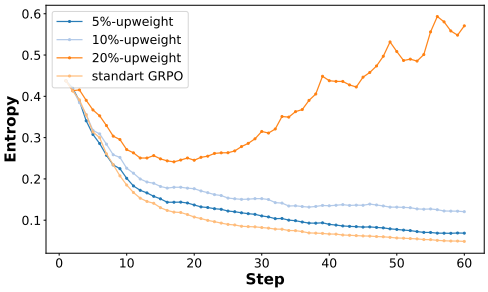
(a) Masking Quadrant I.



(b) Up-weighting Quadrant II.

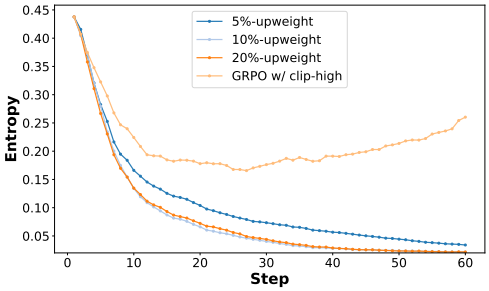


(c) Masking Quadrant III.

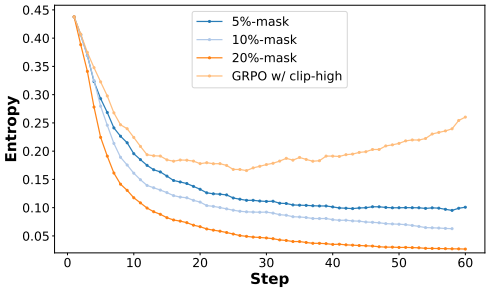


(d) Up-weighting Quadrant IV.

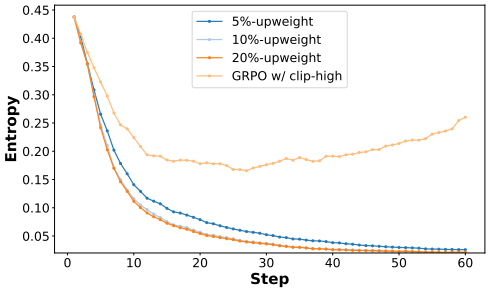
Figure 15: Increasing entropy in four cases.



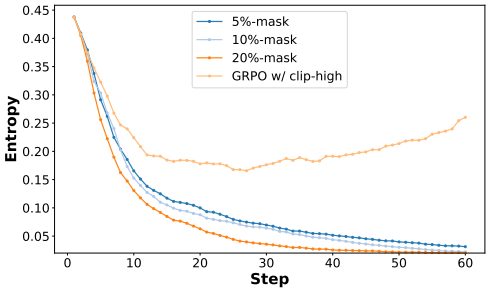
(a) Up-weighting Quadrant I.



(b) Masking Quadrant II.



(c) Up-weighting Quadrant III.



(d) Masking Quadrant IV.

Figure 16: Decreasing entropy in four cases.

stochastic, reducing the likelihood of sampling informative reasoning trajectories that provide useful training signals. In GRPO-style training, this may lead to a large fraction of rollouts receiving near-zero rewards, hindering optimization. Similar concerns on "entropy explosion" have also been raised in prior work (Wu et al., 2025b).

E Training Settings

E.1 Detailed Information for dataset

Math Reasoning We use DAPO-Math-17k as the training dataset for enhancing model's math reasoning, which is a competition-style math reasoning dataset introduced in the DAPO work. It contains roughly 17k problem–solution pairs of olympiad-level mathematics (covering algebra, number theory, combinatorics, geometry, etc.), derived from standard public math-reasoning benchmarks and reformatted into supervised trajectories suitable for RLVR training.

Table 10 reports detailed statistics of the test datasets used in our experiments, including the number of questions and difficulty levels. These benchmarks are all widely used to evaluate mathematical reasoning ability.

Code Tasks For code generation task, we adopt ArcherCodeR[2] for RLVR training, which contains 6000+ code generation tasks. And we adopt the widely used LiveCodeBench v5 (Jain et al., 2024) for evaluation. As for code edit task, we build a large-scale corpus of practical code-editing examples—over 50000 instances—collected from internal users and representative of their day-to-day software development workflows. We then evaluate the model on both our internal held-out test split (3314 cases) and the Zeta benchmark. Internal refers to our in-house, real-world code-editing evaluation set, and Zeta is a public code-edit benchmark (Zed Industries, 2025).

Each example contains two fields: <prompt> and <edit>. The <prompt> field bundles all required inputs, including the surrounding code context, an ordered sequence of prior edits, the target edit span annotated with the cursor position, and any user-provided hints. The <edit> field provides the reference (ground-truth) edit.

[2]Available at https://huggingface.co/datasets/Fate-Zero/ArcherCodeR-Dataset.

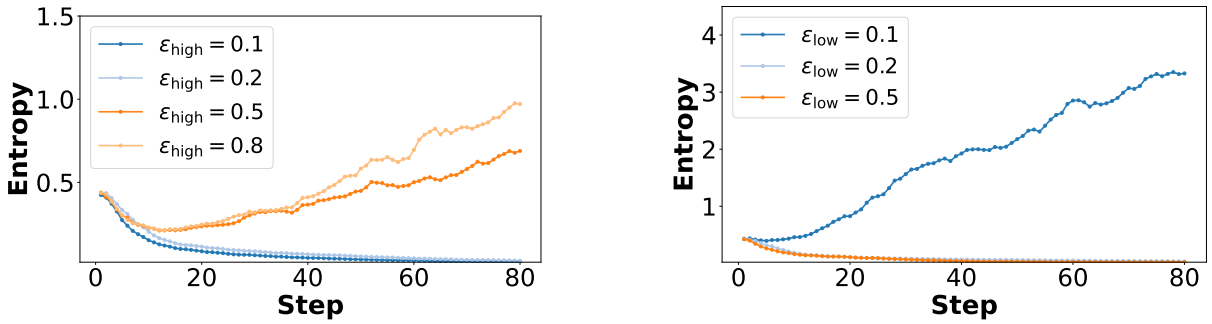
E.2 Training Details for our method and baselines

All algorithms are implemented based on the official GRPO codebase within the VeRL framework.

Training settings Generation batch size is set to 512, and update batch size is set to 32. The number of rollouts is set to 8. Both the KL-divergence and entropy loss terms are removed in our experiments. Training is performed with top-p value of 1.0 and temperature is 1.0. We use a learning rate of 1e-6 without warm-up across all experiments. At each rollout step, we generate 8 answers for each of 512 sampled questions with a mini-batch size of 32 for updating policy model. Models are trained for at most 200 rollout steps. Unless otherwise specified, we follow GRPO's default design choices with token-level loss normalization without dynamic sampling and KL regularization. For Qwen2.5 series models, the maximum input length is 1024 and the maximum output length is 3072. For Qwen2.5-Coder series models in code editing task, the maximum input length is 4096 and the maximum output length is 1024.

Evaluation settings Validation is performed with a top-p value of 0.7 and temperature is 1.0 across all models and test sets. We use Math-Verify and Qwen-Verify for both validation during training and final evaluation. All evaluations are *zero-shot* with no additional prompts. All methods save a checkpoint every 10 steps, and the checkpoint achieving the highest AIME24 accuracy is selected for test. All experiments were conducted on a cluster equipped with NVIDIA H20 GPUs.

Specific settings for baselines Table 3 reports the main results, and this subsection details the training setups for all compared baselines. For GRPO (Shao et al., 2024), we follow the official VeRL training recipe and keep all hyperparameters unchanged. For SimpleRL-Zoo (Zeng et al., 2025), we adopt the original training recipe from the official repository, replacing the original math corpus with DAPO-Math-17k as the training data. For Eurus-PRIME (Cui et al., 2025a), we directly use the publicly released checkpoint that is trained on Qwen2.5-Math-7B with process reward. For OPO (Hao et al., 2025), we follow the reference implementation in VeRL without new hyperparameters introduced. GRPO w/ clip-high (Yu et al., 2025) sets the upper clipping threshold to $\varepsilon_{\text{high}} = 0.28$ while keeping all other settings iden-



(a) Effect of $\varepsilon_{\text{high}}$

(b) Effect of ε_{low}

Figure 17: Empirical validation of entropy-change mechanisms via ratio clipping.

tical to GRPO. GRPO w/ Entro. Loss (Schulman et al., 2017) augments the GRPO objective with an entropy loss term; we tune its weight over $\{0.01, 0.001, 0.0001\}$ and report the best result. GRPO w/ Fork Tokens follows the strategy in (Wang et al., 2025d): using only policy gradients of the top 20%, 30%, or 40% highest-entropy tokens and report the best-performing configuration. W-REINFORCE (Zhu et al., 2025) is implemented by assigning a reduced weight to positive samples, tuning $\lambda \in \{0.1, 0.2\}$ and reporting the best result. Entro. Adv. (Cheng et al., 2025) is reproduced following the original paper, fixing $\kappa = 2$ for all experiments and setting $\alpha = 0.4$. Clip-Cov and KL-Cov (Cui et al., 2025b) apply clipping or KL-penalty constraints only to a small subset of generated tokens. Following the original implementation, we set the fraction of constrained tokens to 0.0002.

F Supplementary Performance Evaluation

This section presents additional performance results that assess the robustness and generality of STEER under varied settings.

F.1 Ablation Study

Besides the exponential mapping in Eq. (9), we consider the following linear mapping and binary mapping for ablation:

$$\text{linear:} \quad \lambda_{i,t} = \lambda_{\max} - \frac{\lambda_{\max} - \lambda_{\min}}{\Omega_{\max} - \Omega_{\min}} (\Omega_{i,t} - \Omega_{\min}),$$

$$\text{binary:} \quad \lambda_{i,t} = \begin{cases} \lambda_{\min}, & \Omega_{i,t} > Q_{\xi}(\Omega), \\ 1, & \text{otherwise.} \end{cases}$$

We set $\lambda_{\min} = 0.7$ throughout and $\lambda_{\max} = 1.2$ for the linear mapping. For the binary mapping, the quantile threshold Q_{ξ} is set to $Q_{0.8}$—i.e., the top 20% of tokens by $\Omega_{i,t}$ are assigned weight $\lambda_{\min}$ and the remaining 80% of tokens keep weight 1. The three mapping schematics are illustrated in Figure 18, and their performance on Qwen2.5-Math-7B is reported in Table 11, each is the average of two runs. It can be seen that the binary mapping degrades performance, whereas the linear mapping does not materially harm performance. This highlights the necessity of continuous token-level reweighting, as truncation cannot precisely control entropy change.

We also assess the sensitivity of the experimental results to hyperparameters $\lambda_{\min}$ in Eq. (9). An excessively small $\lambda_{\min}$ may hinder the model's learning and lead to unstable training, while an excessively large $\lambda_{\min}$ reduces the model's ability to control entropy. As shown in Figure 19, our method performs consistently well when $\lambda_{\min} \in [0.6, 0.8]$.

Figure 20 presents the test accuracy curves for different hyperparameters, demonstrating both stability and superiority of STEER.

F.2 Empirical Results Extended to Other Base Models and Other RL Algorithms

We add experiments on additional backbones, including Qwen2.5-Math-1.5B, Llama-3.2-3B-Instruct (Grattafiori et al., 2024b) for math reasoning and Mistral-7B-v0.3 (Jiang et al., 2023a) for code editing. For each task, we use the same STEER hyperparameter as in the main experiments, rather than re-tuned per backbone. The results are reported in Appendix Figure 21, and we find that STEER yields consistent performance improvements.

As observed, STEER improves the average score from 19.5 $\rightarrow$ 21.6 and clearly mitigates entropy collapse (0.22 $\rightarrow$ 0.85 at Step 200). On Mistral-

Table 11: Ablation study on different weight mapping modes.

Mapping	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
exponential	36.2	16.1	72.1	82.2	41.7	43.0	48.6
linear	36.0	15.7	73.6	81.8	39.4	41.8	48.0
binary	32.5	14.7	71.3	80.9	38.2	41.5	46.5

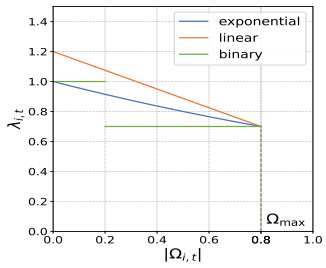


Figure 18: Weight Mapping.

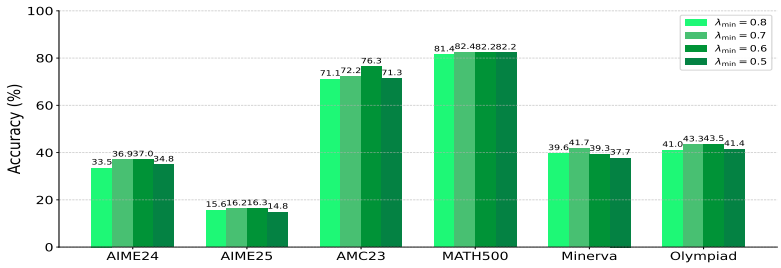


Figure 19: Hyperparameter Sensitivity on λ_{min}.

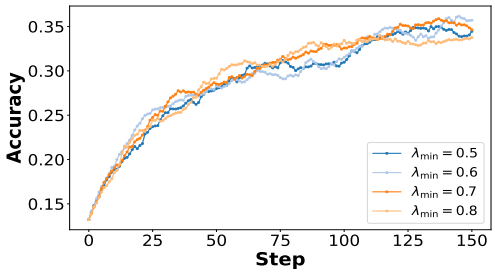


Figure 20: Test accuracy dynamics under different λ_{min}.

7B-v0.3 for code-edit task, STEER improves exact-match on both Internal (12.93 → 14.38) and Zeta (7.64 → 8.35), again keeping entropy much higher at the end (0.04 → 0.38 at Step 200). These improvements suggest our proposed method and its hyperparameters transfer beyond the Qwen2.5 family.

We evaluate STEER under RLOO and OPO with the default setting in verl implementations. The results are shown in Figure 22. The rest of the core setup is the same as in the main results.

As shown, STEER improves the average score from 45.8 → 46.8 on RLOO and 46.4 → 47.5 on OPO, indicating the effectiveness of STEER is not GRPO-specific.

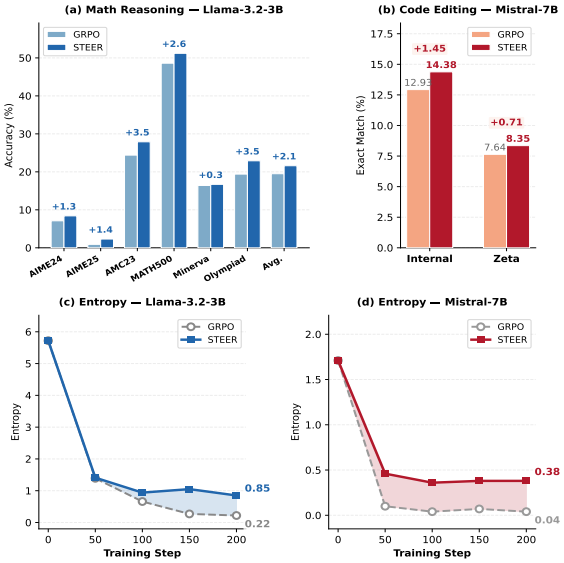


Figure 21: Performance comparison and entropy on math reasoning and coding tasks on other base models.

Table 12: Benchmark results of different methods on Qwen2.5-Math-1.5B. We report avg@32 for AIME24, AIME25, and AMC23 and avg@1 for others. All results are presented as percentages.

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
Qwen2.5-Math-1.5B							
Base	4.1	2.1	24.7	29.0	9.2	20.5	14.9
GRPO	16.2	7.6	56.0	74.4	26.1	34.6	35.8
OPO	14.8	9.0	58.2	72.2	26.1	35.9	36.0
Entro. Adv.	15.0	9.1	55.7	70.2	26.8	34.9	35.3
Clip-Cov	14.7	8.4	56.0	72.8	26.4	34.9	35.5
STEER	**17.4**	**9.7**	**61.6**	**75.5**	**28.2**	**36.6**	**38.2**

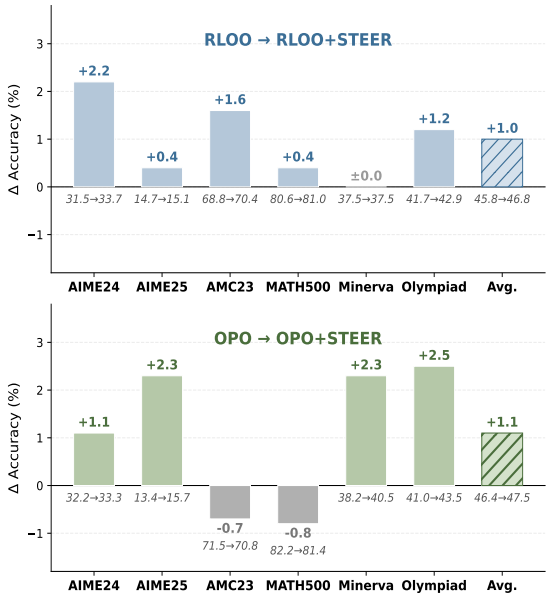


Figure 22: Results on math reasoning tasks based on other RL algorithms.

G Theorem Proof Details

During the RLVR training, token logits are shaped by entangled internal parameters, making entropy change difficult to quantify. To capture the essence of distribution shifts during training, we adopt the following weak assumption.

Assumption 1 (Parameter-independent softmax)**.** *For any context (state) $s = (q, o_{<t})$, each token (action) a in the vocabulary $\mathcal{V}$ is associated with an independent logit parameter $z_{s,a}(\theta)$. At update step k in training, the next-token distribution of π_θ^k then follows*

$$\pi_\theta^k(\cdot \mid s) = \mathrm{softmax}(z^k(s)),$$

where $z^k(s)$ is the vector of logit parameters for all actions under state s.

Assumption 1 states that a gradient step on the sampled token does not substantially affect the logits of the other tokens in the vocabulary. Given this assumption, we derive the following theorem on token entropy change.

Theorem 1 (First–order entropy change estimation)**.** *Let the policy model π_θ satisfy Assumption 1. For any context (state) $s = (q, o_{<t})$, define the token–level entropy change between two update steps as*

$$\Delta\mathcal{H}(s) \triangleq \mathcal{H}(\pi_\theta^{k+1} \mid s) - \mathcal{H}(\pi_\theta^k \mid s).$$

Under a single GRPO update in Eq. (2), $\Delta\mathcal{H}(s)$ admits the decomposition

$$\Delta\mathcal{H}(s) = \Omega(s) + \Phi(s), \tag{16}$$

where the first–order estimation term is

$$\Omega(s) = -\frac{\eta}{L}\,\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\Big[\frac{\mathbb{I}_{clip}(s,a)\,A(s,a)}{\pi_{old}(a \mid s)}\,\pi_\theta^k(a \mid s)\,\big(1-\pi_\theta^k(a \mid s)\big)\,\big(\log\pi_\theta^k(a \mid s)+\mathcal{H}(\pi_\theta^k \mid s)\big)\Big], \tag{17}$$

and the higher–order remainder term $\Phi(s)$ satisfies

$$|\Phi(s)| \;\le\; C\,\eta^2\left[\frac{A_{\max}\,r_{\max}}{L}\right]^2, \tag{18}$$

where L is the total decoded length in the GRPO update and $A_{\max}, r_{\max}$ bound the token–level advantage and importance ratio and and $C > 0$ is a constant depending on the policy parameterization.

Proof. We prove the theorem in five steps as follows.

Step 1: First-order Taylor expansion. Taking the first-order Taylor expansion of $\Delta\mathcal{H}(s)$ around $z^k(s)$, we have

$$\begin{aligned}\Delta\mathcal{H}(s) &= \mathcal{H}(\pi_\theta^{k+1} \mid s) - \mathcal{H}(\pi_\theta^k \mid s)\\ &= \underbrace{\left\langle \frac{\partial\mathcal{H}(\pi_\theta^k \mid s)}{\partial z},\, z^{k+1}(s) - z^k(s)\right\rangle}_{\text{first-order term }\Omega(s)} + \underbrace{\mathcal{O}\big(\|z^{k+1}(s) - z^k(s)\|_2^2\big)}_{\text{higher-order remainder }\Phi(s)},\end{aligned} \tag{19}$$

where $z(s) = (z_{s,a})_{a\in\mathcal{V}}$ is the logit vector for all actions under state s.

Step 2: Gradient of entropy w.r.t. logits. For a fixed state s, the conditional entropy of the policy $\pi_\theta^k(\cdot \mid s)$ is

$$\mathcal{H}(\pi_\theta^k \mid s) = -\sum_{a\in\mathcal{V}} \pi_\theta^k(a \mid s)\,\log\pi_\theta^k(a \mid s).$$

Under the parameter–independent softmax parameterization, each token $a \in \mathcal{V}$ has a logit $z_{s,a}^k$ and

$$\pi_\theta^k(a \mid s) = \frac{\exp(z_{s,a}^k)}{\sum_{b\in\mathcal{V}}\exp(z_{s,b}^k)}.$$

We differentiate $\mathcal{H}(\pi_\theta^k \mid s)$ with respect to a single logit $z_{s,b}$. Using the softmax derivative, we have

$$\frac{\partial \pi_\theta^k(a \mid s)}{\partial z_{s,b}} = \pi_\theta^k(a \mid s)\,\big(\mathbf{1}\{a=b\} - \pi_\theta^k(b \mid s)\big).$$

Then, we obtain

$$\begin{aligned}
\frac{\partial \mathcal{H}(\pi_\theta^k \mid s)}{\partial z_{s,b}} &= -\sum_{a\in\mathcal{V}} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \frac{\partial \pi_\theta^k(a \mid s)}{\partial z_{s,b}} \\
&= -\sum_{a\in\mathcal{V}} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \pi_\theta^k(a \mid s) \big(\mathbf{1}\{a=b\} - \pi_\theta^k(b \mid s)\big) \\
&= -\Big(\log \pi_\theta^k(b \mid s) + 1\Big) \pi_\theta^k(b \mid s) \big(1 - \pi_\theta^k(b \mid s)\big) + \pi_\theta^k(b \mid s) \sum_{a\neq b} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \pi_\theta^k(a \mid s) \\
&= -\Big(\log \pi_\theta^k(b \mid s) + 1\Big) \pi_\theta^k(b \mid s) \big(1 - \pi_\theta^k(b \mid s)\big) \\
&\quad + \pi_\theta^k(b \mid s) \left[\sum_{a\in\mathcal{V}} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \pi_\theta^k(a \mid s) - \Big(\log \pi_\theta^k(b \mid s) + 1\Big) \pi_\theta^k(b \mid s)\right] \\
&= -\Big(\log \pi_\theta^k(b \mid s) + 1\Big) \pi_\theta^k(b \mid s) + \pi_\theta^k(b \mid s) \sum_{a\in\mathcal{V}} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \pi_\theta^k(a \mid s),
\end{aligned}$$

where $\mathcal{V}$ denotes the vocabulary. Note that

$$\sum_{a\in\mathcal{V}} \pi_\theta^k(a \mid s) = 1, \qquad \sum_{a\in\mathcal{V}} \log \pi_\theta^k(a \mid s)\, \pi_\theta^k(a \mid s) = -\mathcal{H}(\pi_\theta^k \mid s).$$

Therefore,

$$\sum_{a\in\mathcal{V}} \Big(\log \pi_\theta^k(a \mid s) + 1\Big) \pi_\theta^k(a \mid s) = -\mathcal{H}(\pi_\theta^k \mid s) + 1.$$

Substituting back, we get

$$\begin{aligned}
\frac{\partial \mathcal{H}(\pi_\theta^k \mid s)}{\partial z_{s,b}} &= -\Big(\log \pi_\theta^k(b \mid s) + 1\Big) \pi_\theta^k(b \mid s) + \pi_\theta^k(b \mid s) \big(-\mathcal{H}(\pi_\theta^k \mid s) + 1\big) \\
&= -\,\pi_\theta^k(b \mid s)\, \Big(\log \pi_\theta^k(b \mid s) + \mathcal{H}(\pi_\theta^k \mid s)\Big).
\end{aligned}$$

Thus, for any $a \in \mathcal{V}$,

$$\frac{\partial \mathcal{H}(\pi_\theta^k \mid s)}{\partial z_{s,a}} = -\,\pi_\theta^k(a \mid s)\, \big(\log \pi_\theta^k(a \mid s) + \mathcal{H}(\pi_\theta^k \mid s)\big). \tag{20}$$

Step 3: One-step GRPO update in logit space. The policy gradient of GRPO in Eq. (2) can be written as

$$\nabla_\theta J(\theta) = \mathbb{E}_{\substack{q\sim\mathcal{D}, \\ \{o_i\}_{i=1}^G \sim \pi_{\text{old}}(\cdot|q)}} \left[\frac{1}{\sum_{i=1}^G |o_i|} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \mathbb{I}_{\text{clip}}(i,t)\, \frac{\pi_\theta^k\big(o_{i,t} \mid q, o_{i,<t}\big)}{\pi_{\text{old}}\big(o_{i,t} \mid q, o_{i,<t}\big)} A_{i,t}\, \nabla_\theta \log \pi_\theta\big(o_{i,t} \mid q, o_{i,<t}\big)\right], \tag{21}$$

where $\mathbb{I}_{\text{clip}}(i,t)$ is the clipping indicator and $A_{i,t}$ is the token-level advantage.

For notational convenience, fix a particular token position (i,t) and write

$$s \triangleq (q, o_{i,<t}), \qquad a \triangleq o_{i,t}.$$

We also write $\mathbb{I}_{\text{clip}}(s,a)$ and $A(s,a)$ to denote the same quantities as functions of the state–action pair (s,a). Under Assumption 1, each logit $z_{s,a}$ is treated as an independent parameter. Under the softmax parameterization, the derivative of the log-probability with respect to its own logit is

$$\frac{\partial}{\partial z_{s,a}} \log \pi_\theta^k(a \mid s) = 1 - \pi_\theta^k(a \mid s),$$

while the derivatives with respect to logits of non-sampled actions $a' \neq a$ at the same state s are neglected.

An update of $z_{s,a}$ in the direction of the GRPO gradient with learning rate η therefore contributes

$$\begin{aligned} z_{s,a}^{k+1} - z_{s,a}^k &= \eta \frac{1}{\sum_{i'=1}^G |o_{i'}|} \mathbb{I}_{\text{clip}}(i,t) \frac{\pi_\theta^k(a \mid s)}{\pi_{\text{old}}(a \mid s)} A_{i,t} \frac{\partial}{\partial z_{s,a}} \log \pi_\theta^k(a \mid s) \\ &= \eta \frac{1}{\sum_{i'=1}^G |o_{i'}|} \mathbb{I}_{\text{clip}}(i,t) \frac{\pi_\theta^k(a \mid s)}{\pi_{\text{old}}(a \mid s)} A_{i,t} \big(1 - \pi_\theta^k(a \mid s)\big). \end{aligned} \tag{22}$$

By switching to the (s,a) notation explicitly, we obtain the simplified update

$$z_{s,a}^{k+1} - z_{s,a}^k = \eta \frac{1}{\sum_{i'=1}^G |o_{i'}|} \mathbb{I}_{\text{clip}}(s,a) \frac{\pi_\theta^k(a \mid s)}{\pi_{\text{old}}(a \mid s)} A(s,a) \big(1 - \pi_\theta^k(a \mid s)\big). \tag{23}$$

For non-sampled actions a' at the same state s, the corresponding logits $z_{s,a'}$ remain unchanged in this update.

Step 4: First–order estimation $\Omega(s)$. Plugging (20) and (23) into the inner product, we have

$$\begin{aligned} \Omega(s) &= \left\langle \frac{\partial \mathcal{H}(\pi_\theta^k \mid s)}{\partial z}, z^{k+1}(s) - z^k(s) \right\rangle \\ &= \sum_{a \in \mathcal{V}} \frac{\partial \mathcal{H}(\pi_\theta^k \mid s)}{\partial z_{s,a}} \left(z_{s,a}^{k+1} - z_{s,a}^k\right) \\ &= \sum_{a \in \mathcal{V}} \Big[-\pi_\theta^k(a \mid s) \left(\log \pi_\theta^k(a \mid s) + \mathcal{H}(\pi_\theta^k \mid s)\right) \Big] \Big[\frac{\eta}{L} \frac{\mathbb{I}_{\text{clip}}(s,a)\, A(s,a)}{\pi_{\text{old}}(a \mid s)} \pi_\theta^k(a \mid s) \left(1 - \pi_\theta^k(a \mid s)\right) \Big] \\ &= -\frac{\eta}{L} \sum_{a \in \mathcal{V}} \frac{\mathbb{I}_{\text{clip}}(s,a)\, A(s,a)}{\pi_{\text{old}}(a \mid s)} \left[\pi_\theta^k(a \mid s)\right]^2 \big(1 - \pi_\theta^k(a \mid s)\big) \big(\log \pi_\theta^k(a \mid s) + \mathcal{H}(\pi_\theta^k \mid s)\big) \\ &= -\frac{\eta}{L} \mathbb{E}_{a \sim \pi_\theta^k(\cdot \mid s)} \Big[\frac{\mathbb{I}_{\text{clip}}(s,a)\, A(s,a)}{\pi_{\text{old}}(a \mid s)} \pi_\theta^k(a \mid s) \left(1 - \pi_\theta^k(a \mid s)\right) \left(\log \pi_\theta^k(a \mid s) + \mathcal{H}(\pi_\theta^k \mid s)\right) \Big], \end{aligned}$$

where L denotes $\sum_{i'=1}^G |o_{i'}|$ for short. In the last line, we rewrote the sum as an expectation under $a \sim \pi_\theta^k(\cdot \mid s)$ by absorbing one factor $\pi_\theta^k(a \mid s)$ into the measure. This is exactly Eq. (17).

Step 5: Remainder term $\Phi(s)$. By multivariate Taylor's theorem, the higher–order remainder can be written as a quadratic form in the logit increment, so there exists a constant $C > 0$ such that for any state s,

$$|\Phi(s)| \;\leq\; C \left\| z^{k+1}(s) - z^k(s) \right\|_2^2. \tag{24}$$

For a given state s, only the sampled action a has a nonzero logit update, hence

$$\left\| z^{k+1}(s) - z^k(s) \right\|_2^2 = \left(z_{s,a}^{k+1} - z_{s,a}^k\right)^2.$$

Denote $L \triangleq \sum_{i'=1}^G |o_{i'}|, r(s,a) \triangleq \frac{\pi_\theta^k(a|s)}{\pi_{\text{old}}(a|s)}$, and assume the importance ratios and the token–level advantage are uniformly bounded:

$$|r(s,a)| \leq r_{\max}, \qquad |A(s,a)| \leq A_{\max} \quad \text{for all } (s,a).$$

Using the GRPO update in Eq. (23), we obtain

$$\begin{aligned}
\left\| z^{k+1}(s) - z^k(s) \right\|_2^2 &= \left(z_{s,a}^{k+1} - z_{s,a}^k \right)^2 \\
&= \eta^2 \left[\frac{\mathbb{I}_{\text{clip}}(s,a)\, A(s,a)}{L}\, r(s,a) \left(1 - \pi_\theta^k(a \mid s) \right) \right]^2 \\
&\leq \eta^2 \left[\frac{\mathbb{I}_{\text{clip}}(s,a)\, A(s,a)}{L}\, r(s,a) \right]^2 \\
&\leq \eta^2 \left[\frac{A(s,a)\, r(s,a)}{L} \right]^2 \\
&\leq \eta^2 \left[\frac{A_{\max}\, r_{\max}}{L} \right]^2,
\end{aligned}$$

Combining this bound with (24) and absorbing all fixed constants into C yields

$$|\Phi(s)| \;\leq\; C\, \eta^2 \left[\frac{A_{\max}\, r_{\max}}{L} \right]^2. \tag{25}$$

Thus the remainder term is of order $\mathcal{O}(\eta^2)$ and is further suppressed by the per-update normalization factor L^{-2} in GRPO.

Combining the above steps completes the proof. □