Understanding and Preventing Entropy Collapse in RLVR with On-Policy Entropy Flow Optimization

Huimin Xu[1], **Shuai Zhao**[1], **Xiaobao Wu**[2], **Anh Tuan Luu**[1,3]*
[1]Nanyang Technological University, Singapore
[2]Shanghai Jiao Tong University, China, [3]VinUniversity, Vietnam
{huimin.xu, shuai.zhao, xiaobao.wu, anhtuan.luu}@ntu.edu.sg
xiaobaowu@sjtu.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) has become an effective paradigm for improving the reasoning ability of large language models. However, widely used RLVR algorithms, such as GRPO, often suffer from entropy collapse, leading to premature determinism and unstable optimization. Existing remedies, including entropy regularization and ratio-based clipping heuristics, either control entropy in a coarse-grained manner or rely on approximate on-policy training. In this paper, we revisit entropy collapse from a token-level entropy flow perspective. Our analysis reveals that entropy-decreasing tokens consistently outweigh entropy-increasing ones, resulting in a severely imbalanced entropy flow. This perspective provides a unified explanation of entropy collapse in existing RLVR algorithms and highlights the importance of balancing entropy dynamics. Motivated by this analysis, we propose On-Policy Entropy Flow Optimization (OPEFO), an adaptive entropy flow balancing mechanism that rescales entropy-increasing and entropy-decreasing updates according to their contributions to entropy change, while remaining strict on-policy. Experiments on six mathematical reasoning benchmarks demonstrate that OPEFO improves training stability and final performance [1].

1 Introduction

Reinforcement Learning with Verifiable Rewards (RLVR) has emerged as an effective paradigm to advance reasoning capabilities in Large Language Models (LLMs, Lambert et al., 2024; Jaech et al., 2024; Guo et al., 2025; Yang et al., 2025a; Team et al., 2025). RLVR optimizes LLMs outputs via RL objectives guided by automated verifiable reward signals. However, recent RLVR algorithms, such as Proximal Policy Optimization (PPO, Schulman et al., 2017) and Group Relative Policy Optimization (GRPO, Shao et al., 2024), often suffer from ***Entropy Collapse***: policy entropy drops in the early stage of training and continues to decline towards near zero. This indicates the policy becomes prematurely deterministic, limiting policy exploration and thus hindering LLMs' reasoning capabilities (Cui et al., 2025; Hao et al., 2025b).

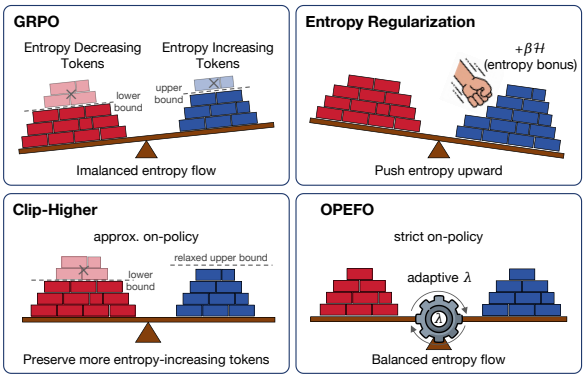


Figure 1: Entropy control mechanisms from the entropy flow perspective. GRPO suffers from imbalanced entropy flow. Entropy regularization increases entropy by adding an explicit entropy bonus; Clip-higher preserves more entropy-increasing tokens via relaxing the upper clipping bound. Differently, our OPEFO adaptively balances entropy flow via an adaptive scaling mechanism.

Recent approaches to mitigating entropy collapse fall into two categories, each with notable limitations, as shown in Figure 1. The first adopts entropy regularization, which introduces an explicit entropy bonus to encourage higher policy entropy (Mnih et al., 2016; Haarnoja et al., 2018). But this approach indiscriminately increases entropy and may cause excessive entropy growth in later training stages (Shen, 2025; Cheng et al., 2025). Besides, the entropy term is optimized independently of the policy-gradient objective, so it may dominate the advantage signal, leading to unstable updates and degraded performance (Zhang et al., 2025; Liu et al., 2025). The second category

[1]Our code, data, and models are available at https://github.com/Anna7355/Entropy-Flow.

Findings of the Association for Computational Linguistics: ACL 2026, pages 17759–17771

employs ratio- or clipping-based heuristics (Yu et al., 2025; Yang et al., 2025b). For instance, Clip-higher (Yu et al., 2025) relaxes the upper clipping bound of the importance sampling ratio to preserve more entropy-increasing tokens. But they are only approximately on-policy due to their reliance on an outdated reference policy, making them theoretically less grounded than strict on-policy optimization (Baird et al., 1995; Sutton et al., 1999; Hao et al., 2025a; Zheng et al., 2025).

In this paper, we address these challenges from a novel perspective. We revisit entropy collapse through the lens of **entropy flow**: how entropy-increasing and entropy-decreasing token-level updates jointly shape overall entropy dynamics during training. Our analysis reveals that entropy-decreasing tokens dominate the early training phase, resulting in a severely imbalanced entropy flow with a strongly negative net entropy change. This imbalance provides a unified explanation for entropy collapse in existing RLVR algorithms.

To mitigate this imbalance issue, we propose **On-Policy Entropy Flow Optimization (OPEFO)**. OPEFO introduces an adaptive entropy flow balancing mechanism that rescales the entropy-increasing and entropy-decreasing updates based on their contributions to entropy change. As such, OPEFO adaptively balances the entropy flow and thus stabilizes policy entropy, as illustrated in Figure 1. Moreover, it remains strict on-policy by avoiding reliance on reference policies as previous methods. Empirical results show that OPEFO achieves best performance across two base models and six challenging mathematical reasoning benchmarks. More importantly, it maintains the most stable entropy dynamics among strong baselines. Overall, we summarize our contributions as follows:

- From an entropy flow perspective, we conduct a token-level analysis of entropy dynamics and show that entropy collapse can be interpreted as an imbalance between entropy-increasing and entropy-decreasing updates.
- We propose OPEFO, a strict on-policy entropy flow balancing mechanism that rescales entropy-increasing and entropy-decreasing token-level updates to stabilize policy entropy.
- Extensive experiments on mathematical reasoning benchmarks show that OPEFO consistently improves training stability and final performance compared to existing RLVR methods.

2 Related Work

Reinforcement Learning with Verifiable Rewards (RLVR, Lambert et al., 2024) has recently emerged as an effective fine-tuning paradigm for large language models, achieving notable success in reasoning-intensive domains such as mathematics and programming (Shao et al., 2024; Guo et al., 2025; Yang et al., 2025a). Its core idea is to replace human feedback with automatically verifiable, objective criteria as reinforcement learning rewards, thereby avoiding the cost and complexity of training human preference models. OpenAI o1 (OpenAI, 2024) is among the first large-scale deployments of this paradigm, demonstrating substantial performance gains on mathematical competition problems and code generation benchmarks. Following this line of work, several subsequent models—including DeepSeek-R1 (Guo et al., 2025), Kimi-1.5 (Team et al., 2025), and Qwen-2.5 (Yang et al., 2024)—have reported matched or improved results on similar reasoning benchmarks. Overall, RLVR has shown clear advantages over prior approaches based solely on supervised fine-tuning for reasoning tasks.

These methods typically adopt GRPO-style training, yet empirical studies consistently report the emergence of entropy collapse during optimization, where the policy's entropy rapidly diminishes as training progresses. Early approaches draw on classical entropy regularization, introducing entropy bonuses or KL penalties to stabilize optimization (Mnih et al., 2016; Haarnoja et al., 2018). However, recent studies show that such techniques are highly sensitive to coefficient tuning in LLM and may even mislead optimization at critical states, yielding only coarse-grained effects on entropy control (Cui et al., 2025; Shen, 2025).

More recent efforts attempt to mitigate entropy collapse by modifying key components of policy optimization, including asymmetric or adaptive ratio clipping (Yu et al., 2025; Yang et al., 2025b), selective optimization over high-entropy tokens (Wang et al., 2025b), or balancing positive and negative samples (Zhu et al., 2025). Other methods introduce entropy-aware advantages or auxiliary objectives to encourage exploration (Cheng et al., 2025; Tan et al., 2025; Wang et al., 2025b,a; Deng et al., 2025). In this work, we adopt a token-level perspective to analyze entropy change and entropy flow, and study how these dynamics evolve under strict on-policy optimization.

3 Preliminaries

In this section, we introduce the preliminaries of RLVR and policy entropy of LLMs.

3.1 RLVR Algorithms

We consider reinforcement learning from verifiable rewards, where a policy model π_θ autoregressively generates a token sequence y given a prompt x. The objective is to maximize the expected reward received from a verifier:

$$\mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\big[r(x,y)\big] \tag{1}$$

where $\mathcal{D}$ denotes the training data. Following the policy gradient theorem (Williams, 1992), the gradient of this objective can be estimated as:

$$\nabla_\theta \mathcal{J}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)}\left[\sum_{t=1}^{|y|} \nabla_\theta \log \pi_\theta(y_t|x, y_{<t}) \cdot A_t\right] \tag{2}$$

where A_t denotes the estimated advantage of token y_t. To avoid training an additional value network, recent algorithms such as GRPO (Shao et al., 2024) estimate token-level advantages via group-wise normalization:

$$A_t = \frac{r(y) - \mathrm{mean}\big(r(y^{1:K})\big)}{\mathrm{std}(r(y^{1:K}))} \tag{3}$$

where $r(y^{1:K})$ denotes the rewards of K rollouts sampled for the same prompt. All tokens within the same response share the same normalized advantage value.

3.2 Policy Entropy of LLMs

For an autoregressive language model π_θ, uncertainty at each decoding step is characterized by the entropy of the next-token distribution. Given a state $s_t = (x, y_{<t})$, where a denotes a candidate next token sampled from the vocabulary, the token-level entropy $\mathcal{H}_t$ is defined as:

$$\mathcal{H}_t = -\,\mathbb{E}_{a\sim\pi_\theta(\cdot|s_t)}\left[\log \pi_\theta(a|s_t)\right] \tag{4}$$

A smaller $\mathcal{H}_t$ indicates higher confidence, while a larger value reflects greater uncertainty or exploration.

To capture uncertainty over an entire response and dataset $\mathcal{D}$, the policy entropy is defined as the average token entropy:

$$\mathcal{H}(\pi_\theta, \mathcal{D}) = \mathbb{E}_{x\sim\mathcal{D}, y\sim\pi_\theta(\cdot|x)}\left[\frac{1}{|y|}\sum_{t=1}^{|y|}\mathcal{H}_t\right] \tag{5}$$

While $\mathcal{H}_t$ captures local uncertainty at individual decoding steps, $\mathcal{H}(\pi_\theta, \mathcal{D})$ provides a global summary of the policy's exploration level.

3.3 Token-Level Entropy Change

While token entropy measures the model's uncertainty at each decoding step, entropy collapse is driven by how entropy changes during policy updates. Following prior works (Hao et al., 2025b; Cui et al., 2025), we therefore analyze training dynamics through token-level entropy change, decomposing the overall entropy evolution into local changes at individual decoding steps. Directly computing the exact entropy change for large autoregressive models is intractable, due to complex dependencies across tokens. As a result, existing works commonly adopt a simplified tabular-softmax assumption, where each token's logit is treated as conditionally independent.

Assumption 1 (Parameter-independent softmax). Assume the policy π_θ is a tabular softmax policy, where each state-action pair (s, a) is associated with an individual logit parameter $z_{s,a}(\theta) = \theta_{s,a}$.

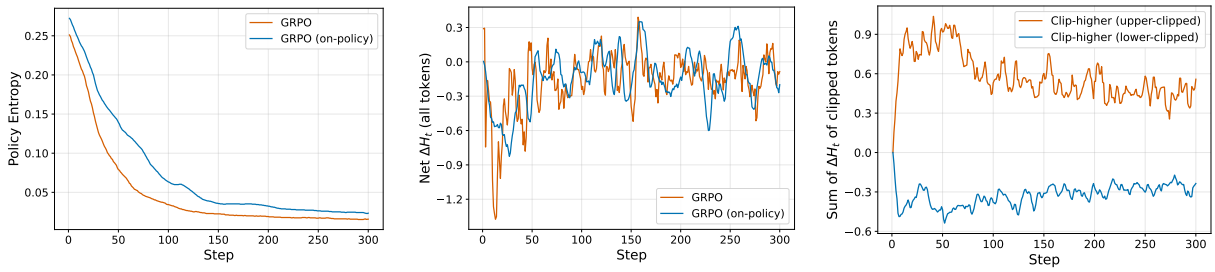
Theorem 1 (First-order entropy change). Under Assumption 1, the change of conditional entropy between two update steps is defined as $\Delta\mathcal{H}_t \triangleq \mathcal{H}(\pi_\theta^{k+1} \mid s_t) - \mathcal{H}(\pi_\theta^{k} \mid s_t)$. Then the first-order estimation of $\Delta\mathcal{H}_t$ is:

$$\begin{aligned}\Delta\mathcal{H}_t = -\,\eta\,\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\big[\, A_t(1-\pi_\theta^k(a\mid s_t))^2 \\ (\log \pi_\theta^k(a\mid s_t) + \mathcal{H}(\pi_\theta^k \mid s_t))\big]\end{aligned} \tag{6}$$

where η is the learning rate, and k indexes the policy update step.

This expression is derived from the first-order entropy change analysis introduced in Hao et al. (2025b), utilizing a Taylor expansion of the conditional entropy around the current policy logits. Compared to prior formulations based on importance ratios, we present the expression under the strict on-policy setting, where expectations are taken with respect to the current policy π_θ^k without reference policies or importance ratios.

We adopt $\Delta\mathcal{H}_t$ as a diagnostic quantity rather than a learning objective. Prior work shows that this first-order approximation closely tracks the exact entropy change in practice, justifying its use for analyzing training dynamics (Hao et al., 2025b). In our setting, it enables a token-level decomposition of entropy evolution, which we leverage to identify entropy imbalance and motivate our on-policy entropy flow balancing mechanism.



(a) (b) (c)

Figure 2: Empirical analysis of entropy dynamics in GRPO and Clip-higher. **(a)** Policy entropy over training steps for GRPO and its variant. **(b)** Net $\Delta\mathcal{H}_t$ per update step for GRPO and its variant. **(c)** Sum of $\Delta\mathcal{H}_t$ of upper- and lower-clipped tokens under Clip-higher.

4 Empirical Observations

In this section, we investigate how entropy evolves during RLVR training from an entropy flow perspective. We examine GRPO and its strict on-policy variant. GRPO performs 8 updates per batch, whereas the strict on-policy setting performs only one update per batch. In addition, we analyze Clip-higher as a representative clipping-based heuristic to understand, through token-level entropy change, why such methods can alleviate entropy collapse. All experiments are conducted using Qwen-2.5-Math-7B (Yang et al., 2024) as the base model, trained on DAPO-17k (Yu et al., 2025).

Entropy collapse under GRPO. Figure 2 (a) plots the evolution of policy entropy for GRPO and its strict on-policy variant. Both curves exhibit highly similar dynamics: starting at an entropy of approximately 0.25, dropping sharply within the first 50 training steps, and rapidly converging to nearly zero. This observation shows that entropy collapse is a common phenomenon in GRPO training, and motivates the need for a mechanism that can stabilize entropy dynamics.

Token-level entropy change in GRPO. We compute the aggregate token-level entropy change $\sum_t \Delta\mathcal{H}_t$ at each training step, based on the Eq. 6. Figure 2 (b) shows that this quantity is strongly negative during the first 50 steps, indicating that dynamics consistently bias entropy downward. This behavior directly explains the sharp entropy drop observed in Figure 2(a). As training proceeds, the net entropy change approaches zero, coinciding with the stabilization of policy entropy. Together, these observations indicate that the evolution of policy entropy is driven by the cumulative token-level entropy changes, highlighting the importance of stabilizing this quantity throughout training.

Clipping behavior of Clip-higher. Since Clip-higher has demonstrated strong empirical effectiveness in alleviating entropy collapse, we analyze it as a representative clipping-based heuristic to understand its behavior from a token-level entropy flow perspective. For each training step, we compute the total $\Delta\mathcal{H}_t$ contributed by tokens clipped at the upper and lower bounds, respectively. Figure 2 (c) reveals a clear pattern: tokens clipped at the upper bound tend to exhibit positive entropy change ($\Delta\mathcal{H}_t > 0$), whereas those clipped at the lower bound typically exhibit negative entropy change ($\Delta\mathcal{H}_t < 0$). This pattern provides a mechanistic explanation for the effectiveness of Clip-higher: by relaxing the upper bound, it selectively preserves entropy-increasing updates, thereby partially counteracting entropy collapse. However, as this effect relies on importance-ratio clipping, it is fundamentally incompatible with strict on-policy training.

Summary. These observations suggest that entropy collapse in RLVR can be interpreted from an entropy flow perspective, where entropy-decreasing updates consistently outweigh entropy-increasing ones, leading to a strongly negative net entropy change. Since clipping-based heuristics such as Clip-higher rely on importance-ratio clipping, they are incompatible with strict on-policy training, motivating the need for a direct mechanism to stabilize entropy dynamics under strict on-policy optimization.

5 On-Policy Entropy Flow Optimization

To address the imbalanced entropy flow observed in Section 4, we propose On-Policy Entropy Flow Optimization (OPEFO), a mechanism designed for stabilizing entropy dynamics under strict on-policy RLVR training.

5.1 Entropy Flow Decomposition

To formalize the entropy flow, we decompose entropy evolution into token-level contributions within each policy update. For each generated token y_t, we estimate the corresponding entropy change $\Delta\mathcal{H}_t$ (Eq. 6) induced by a policy-gradient update. Using the sign of $\Delta\mathcal{H}_t$, we partition the token-level updates within a batch into two sets:

- **Entropy-increasing set** ($\mathcal{S}^+$): Tokens with $\Delta\mathcal{H}_t > 0$, corresponding to updates that broaden the model's predictive distribution.
- **Entropy-decreasing set** ($\mathcal{S}^-$): Tokens with $\Delta\mathcal{H}_t < 0$, corresponding to updates that sharpen the distribution.

As observed in Section 4, GRPO-style training exhibits a pronounced imbalance in entropy flow, where entropy-decreasing updates consistently outweigh entropy-increasing ones across training steps, $\sum_{t\in\mathcal{S}^-}|\Delta\mathcal{H}_t| > \sum_{t\in\mathcal{S}^+}\Delta\mathcal{H}_t$. As this imbalance persists across updates, the policy entropy progressively collapses.

5.2 Balanced On-Policy Objective

To stabilize entropy dynamics under strict on-policy RLVR training, we introduce a reweighted on-policy gradient objective that rescales updates from the two token sets via a balancing coefficient λ:

$$\nabla_\theta J_{\text{OPEFO}}(\theta) = \mathbb{E}_{x\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|x)} \Bigg[(1+\lambda)\sum_{t\in\mathcal{S}^+} \nabla_\theta \log \pi_\theta(y_t|x, y_{<t}) \cdot A_t + (1-\lambda)\sum_{t\in\mathcal{S}^-} \nabla_\theta \log \pi_\theta(y_t|x, y_{<t}) \cdot A_t \Bigg] \tag{7}$$

where $\lambda \in (-1, 1)$ controls the balance between entropy-increasing and entropy-decreasing updates.

This objective offers two practical advantages. First, it is fully compatible with strict on-policy training, as it does not rely on importance sampling ratios or reference policies. Second, by only rescaling gradient components, it preserves the original optimization direction without introducing auxiliary entropy bonuses.

5.3 Adaptive Entropy Flow Scaling

The remaining question is how to determine an appropriate value of λ, which we compute adaptively from batch-level entropy statistics. Specifically, we adopt zero entropy flow as a local stabilizing criterion at the batch level. This criterion is motivated by the observation that entropy collapse arises when entropy-decreasing updates consistently outweigh entropy-increasing ones. By enforcing a balance between these two opposing forces, we establish a condition for stabilizing entropy dynamics without external intervention. Formally, we seek a value of λ such that:

$$\sum_{t\in\mathcal{S}^+}(1+\lambda)\Delta\mathcal{H}_t + \sum_{t\in\mathcal{S}^-}(1-\lambda)\Delta\mathcal{H}_t \approx 0 \tag{8}$$

Solving this constraint yields a unique solution:

$$\lambda^* = \frac{\sum_{t\in\mathcal{S}^-}|\Delta\mathcal{H}_t| - \sum_{t\in\mathcal{S}^+}\Delta\mathcal{H}_t}{\sum_{t\in\mathcal{S}^-}|\Delta\mathcal{H}_t| + \sum_{t\in\mathcal{S}^+}\Delta\mathcal{H}_t} \tag{9}$$

This closed-form expression provides an adaptive entropy flow balancing coefficient computed directly from the current batch. We do not claim λ^* to be globally optimal across tasks or training stages; rather, it acts as a self-adjusting mechanism that compensates for transient entropy imbalances.

5.4 Practical Implementation

OPEFO is straightforward to implement and requires only minimal modification to standard on-policy RLVR pipelines. Practically, implementing OPEFO requires only a few lines of modification to standard GRPO code: compute $\Delta\mathcal{H}_t$, group tokens by sign, compute λ^* using Eq. 9, and reweight gradients accordingly. This procedure is fully compatible with strict on-policy training and introduces negligible computational overhead. Figure 3 shows the implementation code of OPEFO.

```
def compute_opefo_loss(log_prob,
    advantages, delta_H, eps=1e-12, **
    args):
    pg_terms = - log_prob * advantages
    S_pos = delta_H > 0
    S_neg = delta_H < 0

    pos_mag = delta_H[S_pos].sum()
    neg_mag = delta_H[S_neg].abs().sum()

    denom = (neg_mag + pos_mag).
    clamp_min(eps)
    lambda_s = (neg_mag - pos_mag) /
    denom

    pg_terms[S_pos] *= (1.0 + lambda_s)
    pg_terms[S_neg] *= (1.0 - lambda_s)
    return pg_terms.mean()
```

Figure 3: Implementation code of OPEFO loss.

6 Experiments

6.1 Experimental setup

Training. Following prior work (Hao et al., 2025b; Cui et al., 2025), we use Qwen2.5-Math-7B (Yang et al., 2024) as the primary base model, and additionally include Qwen3-4B-Base (Yang et al., 2025a) to evaluate the generality of our method. The training codebase is adapted from Verl (Sheng et al., 2025). The training data is DAPO-17K (Yu et al., 2025), which contains high-quality reasoning trajectories annotated with verifiable rewards.

For all methods, each rollout step samples a batch of 32 prompts with 8 responses per prompt, yielding 256 responses in total. Strict on-policy methods perform one update per rollout using the full batch, while approximate on-policy baselines follow the standard GRPO setting with 8 sequential updates over mini-batches from the same rollout data. We optimize using AdamW (Loshchilov and Hutter, 2017), with learning rates of 2.83e-6 for strict on-policy methods [2] and 1e-6 for approximate on-policy baselines. All models are trained with a linear warm-up over the first 10 rollout steps. The maximum response length is set to 3000 tokens for Qwen2.5-Math-7B-base and 8000 tokens for Qwen3-4B-base. All experiments are conducted on 32 A100 GPUs.

Baselines. We compare OPEFO with several representative baselines. GRPO is used as the primary reference method, with both the upper and lower clipping bounds set to 0.2. Entropy regularization (Entropy-Reg) augments the objective with an explicit entropy regularization term, where the coefficient is fixed to 0.01. Clip-higher relaxes the upper clipping bound to 0.28 while keeping the lower bound unchanged at 0.2, following the standard configuration in prior work. In addition, Clip-Cov and KL-Cov are included, using the settings reported in Cui et al. (2025).

Evaluation. We evaluate models on six widely used mathematical reasoning benchmarks: AIME24, AIME25, AMC23 (Li et al., 2024), MATH500 (Hendrycks et al., 2021), Minerva Math (Lewkowycz et al., 2022), and OlympiadBench (He et al., 2024). Following prior work (Hao et al., 2025b), validation is performed with a top-p value of 0.7 and temperature 1.0 across all models and test sets. We report Avg@32 for AIME24 (30 problems), AIME25 (30 problems), and AMC23 (40 problems) due to their smaller size, and report Avg@1 for all other benchmarks. All evaluations are zero-shot with no additional prompts. All methods save a checkpoint every 10 steps, and the checkpoint achieving the highest average accuracy (Avg.) across benchmarks is selected for evaluation.

[2] Following common practice, the learning rate is scaled proportionally to the square root of the effective batch size increase (i.e., $\sqrt{8}$) to maintain a comparable optimization noise scale (Goyal et al., 2017; Smith and Le, 2017).

6.2 Main results

Table 1 reports the results of two base models on six mathematical reasoning benchmarks. We highlight three key observations.

First, among competitive baselines, OPEFO achieves the best overall performance. On Qwen2.5-Math-7B and Qwen3-Base-4B, OPEFO attains average accuracies of 52.4% and 51.9%, outperforming the second-best methods by 1.7% and 1.8%, respectively. These gains are observed across models and datasets, indicating that OPEFO improves reasoning ability in a robust manner.

Second, comparing GRPO with its strict on-policy variant reveals a clear and consistent empirical advantage of the strict on-policy setting over the approximate on-policy one. This provides empirical support for the claim discussed earlier that avoiding stale reference policies and distribution shifts can improve policy optimization in RLVR. However, strict on-policy GRPO alone, despite achieving higher accuracy, does not prevent entropy collapse, as shown in subsequent analyses.

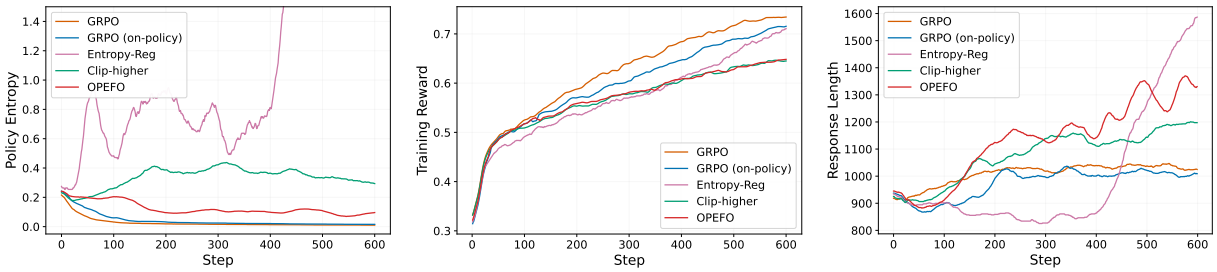
Third, under the same strict on-policy setting, OPEFO further improves performance over strict on-policy GRPO by explicitly balancing entropy-increasing and entropy-decreasing updates, yielding average gains of 2.3% and 1.8% on Qwen2.5-Math-7B and Qwen3-Base-4B, respectively. This highlights the complementary role of entropy flow balancing: while strict on-policy training provides a cleaner optimization signal, OPEFO addresses the remaining instability and brings additional performance improvements under this stabilized entropy dynamics.

6.3 Training Dynamics Analysis

In this section, we conduct an empirical analysis of the training dynamics of Qwen2.5-Math-7B under several representative training methods, examining

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
GRPO	26.7	13.0	69.8	80.3	37.1	46.1	45.5
GRPO (Strict on-policy)	32.6	15.4	81.8	83.5	38.9	48.1	50.1
Entropy-Reg	31.7	13.3	74.4	81.9	40.5	45.1	47.8
Clip-higher	30.5	17.4	78.5	83.5	39.7	49.4	49.8
Clip–Cov	32.2	18.5	77.9	85.1	40.3	50.2	50.7
KL–Cov	32.6	17.9	78.3	84.6	40.9	48.7	50.5
OPEFO	**34.5**	**19.2**	**82.2**	**85.3**	**41.6**	**51.8**	**52.4**
Qwen3-Base-4B	9.7	8.8	51.1	75.4	33.1	40.5	36.4
GRPO	19.4	17.3	66.1	80.6	36.3	49.4	44.8
GRPO (Strict on-policy)	26.2	22.3	71.5	86.3	38.9	55.7	50.1
Entropy-Reg	22.7	20.8	70.9	83.4	37.1	53.3	48.0
Clip-higher	23.5	21.7	70.3	84.8	39.2	55.3	49.1
Clip–Cov	24.2	23.1	71.9	85.7	39.7	56.9	48.9
KL–Cov	24.7	22.4	72.3	86.1	40.3	55.8	49.2
OPEFO	**27.7**	**24.8**	**73.1**	**87.5**	**41.2**	**57.4**	**51.9**

Table 1: Main results on mathematical reasoning benchmarks. All results are presented as percentages. The best are in **bold**.



(a) (b) (c)

Figure 4: Training dynamics under different methods: **(a)** policy entropy, **(b)** training reward, and **(c)** response length.

how these methods affect entropy, training reward, and response length over training.

Overall Entropy. As shown on the Figure 4 (a), both standard GRPO and its strict on-policy variant exhibit a rapid collapse of policy entropy during training. While strict on-policy GRPO achieves higher accuracy than the standard setting (Table 1), it does not inherently prevent entropy collapse. In contrast, entropy regularization leads to uncontrolled entropy growth in later training stages (after approximately 400 steps), whereas Clip-higher partially mitigates collapse but still suffers from noticeable oscillations. By comparison, OPEFO maintains a smooth and stable entropy trajectory, avoiding both premature determinism and uncontrolled entropy inflation.

Training Reward. Figure 4 (b) shows steady upward trends across all methods. Among them, GRPO and its strict on-policy variant achieve the highest training reward while also exhibiting the lowest policy entropy. When interpreted together with the entropy curves, this pattern is consistent with premature exploitation of a limited set of high-reward answer patterns, rather than sustained exploration throughout training. A more desirable objective is to improve training performance while maintaining sufficient entropy to preserve diversity in the learned policy. OPEFO follows this direction by stabilizing entropy while still achieving competitive reward growth.

Response Length. Figure 4 (c) shows the evolution of response length during training. For GRPO and its on-policy variant, the response length quickly saturates and stops increasing after around 200 steps, while Clip-higher stabilizes at a later stage, around 300 steps. Entropy regular-

Method	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg.
GRPO (Strict on-policy)	26.7	13.0	69.8	80.3	37.1	46.1	45.5
Static Scaling	30.0	16.5	81.0	83.8	38.0	49.5	49.8
One-side ($\mathcal{S}^+$ only)	32.8	18.2	83.1	84.2	38.7	47.5	50.8
One-side ($\mathcal{S}^-$ only)	32.3	16.3	83.6	83.9	38.3	47.2	50.3
OPEFO	**34.5**	**19.2**	**82.2**	**85.3**	**41.6**	**51.8**	**52.4**

Table 2: Ablation of balancing coefficient λ^* on Qwen2.5-Math-7B. GRPO (Strict on-policy) corresponds to $\lambda^* = 0$. "Static Scaling" uses a fixed $\lambda^* = 0.001$. "One-side ($\mathcal{S}^+$ only)" and "One-side ($\mathcal{S}^-$ only)" apply λ^* to entropy-increasing and entropy-decreasing updates, respectively. OPEFO applies λ^* to both sides.

ization exhibits a sharp increase in response length after around 400 steps, coinciding with the surge in policy entropy observed in Figure 4 (a), indicating that the model begins to generate longer but increasingly unconstrained responses. In contrast, OPEFO continues to produce longer and controlled responses throughout training. We do not claim longer responses are inherently better; rather, under the verifiable-reward setting, response length serves as a behavioral indicator of whether the model continues to explore more intermediate reasoning paths.

6.4 Exploration Analysis

One of the primary motivations of controlling entropy is to improve exploration during RL training. To evaluate whether OPEFO indeed enhances exploration, we measure Pass@k performance on Qwen2.5-Math-7B, which reflects the model's ability to discover diverse reasoning paths.

We report Pass@32 on four relatively small benchmarks (Table 3) and additionally evaluate the scaling behavior of Pass@k on AIME24 (Table 4). As shown in Table 3, OPEFO consistently outperforms all baselines across datasets, indicating improved coverage of correct solutions. More importantly, Table 4 shows that OPEFO achieves higher Pass@k across all k-values and exhibits stronger gains as k increases, suggesting that it explores a broader set of valid reasoning paths rather than repeatedly generating similar responses.

6.5 Analysis of the Balancing Coefficient λ^*

We analyze the role of the balancing coefficient λ^* on Qwen2.5-Math-7B, examining both different balancing strategies and the behavior of the analytically derived λ^* during training.

Ablation on λ^*. OPEFO adaptively reweights updates by amplifying entropy-increasing updates in $\mathcal{S}^+$ and attenuating entropy-decreasing updates in $\mathcal{S}^-$ using a dynamically computed λ^* (Eq. 9). We first compare OPEFO against a static scaling baseline with a fixed $\lambda = 0.001$, chosen as the average value of λ^* observed over the course of training. As shown in Table 2, static scaling yields moderate improvements over strict on-policy GRPO ($\lambda^* = 0$), but consistently underperforms OPEFO, highlighting the importance of dynamically adjusting λ^*.

We further consider two one-sided variants that apply λ^* only to entropy-increasing updates ($\mathcal{S}^+$ only) or only to entropy-decreasing updates ($\mathcal{S}^-$ only). Both variants outperform strict on-policy GRPO, but remain consistently inferior to full OPEFO, suggesting that jointly balancing leads to better overall performance.

Dynamics of λ^*. Beyond ablations on balancing strategies, we further examine how the balancing coefficient λ^* evolves during training. As shown in

Method	AIME24	AIME25	AMC23	MATH500
GRPO	59.8	38.4	93.7	92.5
GRPO (Strict)	61.3	41.2	94.8	93.3
Entropy-Reg	50.1	33.6	90.8	92.0
Clip-higher	57.7	37.6	94.6	92.7
Clip-Cov	59.0	41.7	95.3	93.5
KL-Cov	60.1	40.9	94.9	93.7
OPEFO	**62.4**	**43.3**	**95.6**	**94.1**

Table 3: Pass@32 performance comparison across different benchmarks.

Method	Pass@8	Pass@16	Pass@32	Pass@64
GRPO	45.1	54.3	59.8	65.6
GRPO (Strict)	50.9	56.2	61.3	66.5
Entropy-Reg	46.9	43.9	50.1	54.9
Clip-higher	50.2	51.7	57.7	63.3
Clip-Cov	51.3	53.2	59.0	64.7
KL-Cov	50.5	54.1	60.1	65.5
OPEFO	**52.5**	**56.5**	**62.4**	**68.4**

Table 4: Scaling behavior of Pass@k on the AIME24 benchmark.

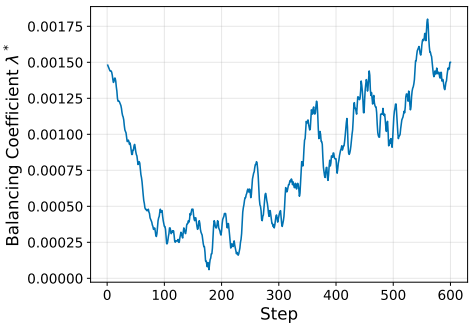


Figure 5: Evolution of the balancing coefficient λ^* over training steps.

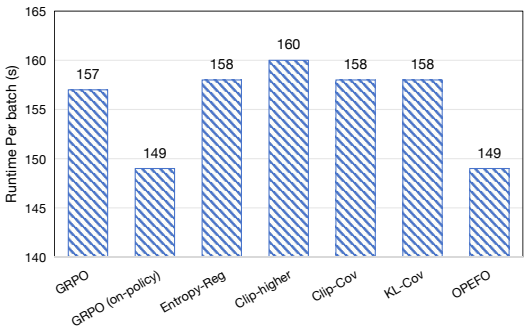


Figure 6: Per-batch runtime comparison under identical rollout settings.

Figure 5, λ^* exhibits a non-stationary pattern over training. Overall, λ^* remains positive throughout training, indicating a persistent need to encourage entropy-increasing updates to counteract entropy collapse. We observe that λ^* first decreases in the early stage and then gradually increases later in training, reflecting different entropy balancing demands across training phases. Taken together, λ^* displays bounded oscillations rather than converging to a fixed value, reflecting an adaptive balancing mechanism that adjusts to non-stationary entropy flow during training.

6.6 Training Efficiency

Strict on-policy methods are often considered computationally expensive due to the need for fresh rollouts. However, in our implementation, this is not the case. As described in Section 6.1, we adopt a single update per rollout with a larger effective batch size, while approximate on-policy methods perform 8 sequential updates per rollout batch. This difference in update strategy directly translates into runtime. As shown in Figure 6, strict on-policy methods have a per-batch runtime of 149s, while approximate on-policy methods take around 158s. Although the difference is modest, it shows that strict on-policy training does not introduce additional computational overhead in practice.

7 Conclusion

We proposed On-Policy Entropy Flow Optimization (OPEFO), a strict on-policy mechanism for stabilizing entropy dynamics in RLVR training. By analyzing entropy collapse from a token-level entropy flow perspective, we showed that it arises from a persistent imbalance between entropy-increasing and entropy-decreasing updates. OPEFO directly addresses this issue by balancing the two opposing entropy flows without relying on reference policies or heuristic entropy regularization. Experiments on six mathematical reasoning benchmarks across two base models demonstrate that OPEFO consistently improves training stability and final performance over strong RLVR baselines. These results suggest entropy flow balancing as a simple and effective principle for stabilizing reasoning-oriented reinforcement learning.

Limitations

Our method offers a principled and interpretable entropy flow perspective for stabilizing RLVR training under a strict on-policy setting, and demonstrates strong empirical performance across reasoning benchmarks. We nevertheless note several limitations.

First, our analysis relies on a first-order approximation of token-level entropy change under a simplified softmax assumption, which captures dominant entropy trends but abstracts away higher-order interactions in large transformer models.

Second, the proposed entropy flow balancing mechanism operates as a local, batch-level stabilizer, where the zero entropy flow criterion serves as a sufficient condition rather than a globally optimal objective, and entropy flow behaviors may vary across tasks.

Finally, our evaluation focuses on strict on-policy RLVR for mathematical reasoning. While OPEFO is conceptually applicable to other domains and reward structures, a systematic study under dense rewards or more complex credit assignment settings is left for future work.

We view these limitations not as fundamental constraints of the proposed method, but as opportunities for extending and refining entropy-aware optimization methods in future RLVR systems.

Acknowledgments

This research is supported by the Air Traffic Management Research Institute, NTU under the grant CAAS_MOA_REQ0392260_NTU.

We only use AI-assisted tools for language polishing and improving clarity.

References

Leemon Baird and 1 others. 1995. Residual algorithms: Reinforcement learning with function approximation. In *Proceedings of the twelfth international conference on machine learning*, pages 30–37.

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*.

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

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*.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. 2017. Accurate, large minibatch sgd: Training imagenet in 1 hour.

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. 2025a. On-policy rl with optimal reward baseline. *arXiv preprint arXiv:2505.23585*.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025b. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv preprint arXiv:2510.10150*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen 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. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

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*.

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*.

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.

Jiacai Liu. 2025. How does rl policy entropy converge during iteration. *Zhihu Zhuanlan*.

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

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

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. 2024. Learning to reason with llms.

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*.

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

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.

Samuel L Smith and Quoc V Le. 2017. A bayesian perspective on generalization and stochastic gradient descent. *arXiv preprint arXiv:1710.06451*.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Hongze Tan, Jianfei Pan, Jinghao Lin, Tao Chen, Zhihang Zheng, Zhihao Tang, and Haihua Yang. 2025. Gtpo and grpo-s: Token and sequence-level reward shaping with policy entropy. *arXiv preprint arXiv:2508.04349*.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, and 1 others. 2025. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*.

Jiakang Wang, Runze Liu, Fuzheng Zhang, Xiu Li, and Guorui Zhou. 2025a. 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. 2025b. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3):229–256.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

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

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*.

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

Haizhong Zheng, Jiawei Zhao, and Beidi Chen. 2025. Prosperity before collapse: How far can off-policy rl reach with stale data on llms? *arXiv preprint arXiv:2510.01161*.

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*.

A Theorem Proof Details

The following derivation is adapted from prior work (Hao et al., 2025b) and is included for reference and completeness.

Theorem 1 (First-order entropy change). Under Assumption 1, the change of conditional entropy between two update steps is defined as $\Delta\mathcal{H}_t \triangleq \mathcal{H}(\pi_\theta^{k+1} \mid s_t) - \mathcal{H}(\pi_\theta^k \mid s_t)$. Then the first-order estimation of $\Delta\mathcal{H}_t$ is:

$$\Delta\mathcal{H}_t = -\eta\, \mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\big[\, A_t(1-\pi_\theta^k(a\mid s_t))^2(\log\pi_\theta^k(a\mid s_t) + \mathcal{H}(\pi_\theta^k \mid s_t))\big] \tag{10}$$

where η is the learning rate, and k indexes the policy update step. Note that compared to the formulation in Hao et al. (2025b), Eq. 10 is specialized to the strict on-policy setting: the weight term w_t (defined there as $w_t = \mathbb{I}_{\text{clip}} r_t A_t$) reduces to the plain advantage A_t since no importance ratio or clipping is used in our updates.

Proof. The proof is similar to that of (Liu, 2025). Taking the first-order Taylor expansion, we have

$$\Delta H_t \triangleq \mathcal{H}(\pi_\theta^{k+1} \mid s_t) - \mathcal{H}(\pi_\theta^k \mid s_t) \approx \langle \nabla_\theta \mathcal{H}(\pi_\theta^k \mid s_t), z^{k+1} - z^k \rangle$$

Since we have the log trick $\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta \log\pi_\theta(a\mid s)] = 0$, the gradient term can be derived as

$$\begin{aligned}
\nabla_\theta\mathcal{H}(\pi_\theta \mid s) &= \nabla_\theta\mathcal{H}(\pi_\theta(\cdot \mid s)) \\
&= \nabla_\theta\left(-\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log\pi_\theta(a\mid s)]\right) \\
&= -\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\nabla_\theta\log\pi_\theta(a\mid s) + \log\pi_\theta(a\mid s)\nabla_\theta\log\pi_\theta(a\mid s)] \\
&= -\mathbb{E}_{a\sim\pi_\theta(\cdot|s)}[\log\pi_\theta(a\mid s)\nabla_\theta\log\pi_\theta(a\mid s)].
\end{aligned}$$

Then we have

$$\begin{aligned}
\Delta H_t &= \langle \nabla_\theta\mathcal{H}(\theta^k \mid s_t), (z^{k+1} - z^k)\rangle \\
&= -\left\langle \mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}[\log\pi_\theta(a\mid s_t)\nabla_\theta\log\pi_\theta(a\mid s_t)], \theta^{k+1} - \theta^k \right\rangle \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\left[\log\pi_\theta(a\mid s_t)\langle\nabla_\theta\log\pi_\theta(a\mid s_t), \theta^{k+1} - \theta^k\rangle\right] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\left[\log\pi_\theta(a\mid s_t)\sum_{a'\in\mathcal{A}}\frac{\partial\log\pi_\theta(a\mid s_t)}{\partial\theta_{s_t,a'}}(\theta_{s_t,a'}^{k+1} - \theta_{s_t,a'}^k)\right] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\left[\log\pi_\theta(a\mid s_t)\sum_{a'\in\mathcal{A}}(\mathbf{1}\{a=a'\} - \pi(a'\mid s_t))(\theta_{s_t,a'}^{k+1} - \theta_{s_t,a'}^k)\right] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s_t)}\Big[\Big(\log\pi_\theta(a\mid s_t) - \mathbb{E}_{\tilde{a}\sim\pi_\theta^k(\cdot|s_t)}\log\pi_\theta(\tilde{a}\mid s_t)\Big) \\
&\quad \Big(\theta_{s_t,a}^{k+1} - \theta_{s_t,a}^k - \mathbb{E}_{a'\sim\pi_\theta^k(\cdot|s_t)}(\theta_{s_t,a'}^{k+1} - \theta_{s_t,a'}^k)\Big)\Big] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\left[[\log\pi_\theta^k(a\mid s) + \mathcal{H}(\cdot\mid s)]\left[(1-\pi_\theta^k(a\mid s))(z_{s_t,a}^{k+1} - z_{s_t,a}^k)\right]\right] \\
&= -\mathbb{E}_{a\sim\pi_\theta^k(\cdot|s)}\left[[\log\pi_\theta^k(a\mid s) + \mathcal{H}(\cdot\mid s)]\left[w(s\mid a)(1-\pi_\theta^k(a\mid s))^2\right]\right],
\end{aligned}$$

where $w(s \mid a)$ is the weight in the policy gradient. □

B Detailed Information For Test Dataset

Test Datasets	#Questions	Level
AIME24	30	Olympiad
AIME25	30	Olympiad
AMC23	40	Intermediate
MATH500	500	Advanced
Minerva	272	Graduate
OlympiadBench	675	Olympiad

Table 5: Dataset statistics.