Entropy Polarity in Reinforcement Fine-Tuning: Direction, Asymmetry, and Control

Jiazheng Zhang[1*†] **Ziche Fu**[1,*] **Junrui Shen**[1,*] **Yunbin Zhao**[2]
Yunke Zhang[2] **Zhiheng Xi**[1] **Long Ma**[1] **Chenxin An**[3] **Zhihao Zhang**[1] **Shichun Liu**[1]
Dingwei Zhu[1] **Shihan Dou**[1] **Shaofan Liu**[1] **Han Li**[4] **Wiggin Zhou**[5] **Aiden Adams**[5]
Tao Gui[1,†] **Fei Huang**[2,†] **Qi Zhang**[1,†] **Xuanjing Huang**[1]

[1] **Fudan University** [2] **Honor Device Co., Ltd**
[3] **The University of Hong Kong** [4] **Shanghai Jiao Tong University** [5] **Tencent**

Abstract

Policy entropy has emerged as a fundamental measure for understanding and controlling exploration in reinforcement learning with verifiable rewards (RLVR) for LLMs. However, existing entropy-aware methods mainly regulate entropy through global objectives, while the underlying mechanism of entropy dynamics remains unclear. In this work, we thoroughly analyze how policy optimization affects entropy evolution at the token level. Our analysis yields a first-order approximation of the entropy change $\Delta\mathcal{H}$, giving rise to ***entropy polarity***, a signed token-level quantity that predicts how much a policy update expands or contracts entropy. This analysis further reveals a structural asymmetry: reinforcing frequent high-probability tokens triggers contraction tendencies, whereas expansive tendencies typically require lower-probability samples or stronger distributional correction. Empirically, we verify that entropy polarity reliably predicts entropy changes, and that positive and negative polarity branches play complementary roles in preserving exploration while strengthening exploitation. Motivated by these findings, we propose **P**olarity-**A**ware **P**olicy **O**ptimization (**PAPO**), a framework that adaptively balances entropy-expanding and entropy-contracting updates through polarity-aware advantage reweighting. Specifically, PAPO distinguishes the two branches according to the sign of entropy polarity and adopts the observed trajectory as an online feedback signal to reallocate their relative optimization pressure. Experiments on mathematical reasoning and agentic benchmarks show that PAPO consistently outperforms baselines, while delivering superior training efficiency and substantial reward improvements.

Code & Sources: https://github.com/JiazhengZhang/PAPO

1 Introduction

Reinforcement fine-tuning (RFT) has substantially improved the capabilities of large language models (LLMs) [1, 2, 3, 4], especially on complex tasks such as mathematics [5, 6], coding [7, 8, 9], and agentic decision-making [10, 11, 12]. A pivotal post-training paradigm driving these gains is Reinforcement Learning with Verifiable Rewards (RLVR) [2, 13, 14], where model behavior is optimized against task-specific objectives through automated correctness verification [15, 16]. Despite the rapid progress of RLVR, the exploration and exploitation trade-off remains a central challenge for

*Equal Contribution
†Correspondence to: jzzhang24@m.fudan.edu.cn, {tgui, qz}@fudan.edu.cn, huangbo6@honor.com

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

RFT: excessive exploitation can prematurely narrow the policy and limit further discovery, whereas insufficient exploitation can slow or destabilize learning [17, 18, 19, 20, 21].

As a practical proxy for policy behavior, entropy has become a critical lens for understanding RLVR [22, 23, 24]. Recent work highlights the role of high-entropy tokens in exploratory reasoning [25, 26]. Prior research has proposed a variety of entropy-aware interventions to mitigate entropy collapse, including entropy regularization [27], clamped bonuses [28], and dynamic clipping [29, 30] for entropy control. Despite these advances, an in-depth understanding of entropy dynamics remains unresolved: beyond identifying high-entropy regions or designing heuristic interventions, it is still unclear how policy-gradient optimization reshape entropy along policy trajectories, how these effects evolve over training, and to what extent they drive performance gains.

To this end, we develop an analytical framework for entropy mechanics in RLVR to study how policy optimization affects entropy dynamics. Instead of viewing entropy as a global statistic alone, our framework characterizes the microscopic mechanism linking policy gradient to logit perturbation, next-token probability redistribution, and subsequent policy entropy variation. At context state s_t, we characterize the policy behavior via the token-level entropy change:

$$\Delta\mathcal{H}_t \;=\; \mathcal{H}\big(\pi_\theta^{k+1}(\cdot \mid s_t)\big) - \mathcal{H}\big(\pi_\theta^{k}(\cdot \mid s_t)\big)\,, \tag{1}$$

whose sign indicates whether it expands or contracts the policy's exploratory capacity. We show that $\Delta\mathcal{H}_t$ admits a first-order decomposition into a sampled-token contribution and a state-wise distributional correction. This gives rise to a computable signed quantity, ***entropy polarity***, that predicts how much reinforcing a given token will increase or decrease local entropy. The polarity view further reveals an asymmetric property: reinforcing high-probability continuations tends to sharpen the distribution, whereas entropy expansion requires low-probability samples or strong distributional correction, explaining why entropy collapse is easier to trigger than to reverse.

Building on the analysis above, we empirically investigate the predictive validity of entropy polarity in practical training. Section 3.3 demonstrates that entropy polarity reliably predicts measured entropy changes between consecutive steps, supporting it as a behaviorally meaningful token-level signal. Section 4.1 shows that positive and negative-polarity induce distinct training dynamics: the former preserve exploratory capacity but yield weake reward, while the latter strengthen reward-aligned exploitation at the cost of faster entropy contraction. Since neither branch is sufficient for stable RL in Figure 2, an effective strategy should therefore keep both branches while controlling their relative influence over training instead of favoring any direction or treating them uniformly.

Motivated by these findings, we propose PAPO (**P**olarity-**A**ware **P**olicy **O**ptimization), a framework that adaptively balances entropy-expanding and entropy-contracting updates through polarity-aware advantage reweighting. Specifically, PAPO distinguishes the two branches according to the sign of entropy polarity and adopts the observed trajectory as an online feedback signal to adjust their relative optimization pressure. In this way, the proposed method preserves exploratory capacity when entropy collapse is imminent, while restoring reward-aligned exploitation when training needs to consolidate useful behaviors. Extensive experiments validate the effectiveness of PAPO spanning mathematics reasoning and agentic tool-calling. Our method outperforms baselines overall on in-distribution benchmarks and achieves competitive performance on out-of-distribution benchmarks. Overall, our main contributions are:

- We develop an analytical framework for entropy mechanics in RLVR. This leads to entropy polarity, a signed quantity that predicts how much a policy update expands or contracts entropy.
- We demonstrate that entropy polarity reliably predicts entropy change in empirical study, making it a behaviorally meaningful signal. Positive and negative polarity play complementary roles: the former preserve exploration, whereas the latter promote exploitation.
- We propose **PAPO**, a polarity-aware optimization method that adaptively coordinates entropy-expanding and entropy-contracting updates using online entropy feedback, achieving consistent overall improvements on mathematical reasoning and agentic benchmarks.

2 Preliminary

Group Relative Policy Optimization. Given a prompt x, LLM π_θ generates response $\pi_\theta(y \mid x) = \prod_{t=1}^{T} \pi_\theta(y_t \mid s_t)$, $s_t := (x, y_{<t})$ denotes the decoding context. In GRPO, the rollout policy

samples a group of G responses $\{y^{(i)}\}_{i=1}^{G}$, and each response $y^{(i)} = (y_1^{(i)}, \ldots, y_{T_i}^{(i)})$ receives a response-level reward $r^{(i)} = r(x, y^{(i)})$. It optimizes the clipped surrogate objective

$$\mathcal{J}_{\text{GRPO}}(\theta) = \frac{1}{G}\sum_{i=1}^{G}\frac{1}{T_i}\sum_{t=1}^{T_i}\min\left(\rho_{i,t}(\theta)A^{(i)}, \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\,A^{(i)}\right), \tag{2}$$

where $\rho_{i,t}(\theta) = \frac{\pi_\theta(y_t^{(i)}|s_t^{(i)})}{\pi_{\theta_{\text{old}}}(y_t^{(i)}|s_t^{(i)})}$ is the importance sampling ratio and $A^{(i)} = \frac{r^{(i)} - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$ denotes the normalized advantages within each group.

Policy Entropy. In RFT, entropy has become a practical indicator of policy uncertainty. For a decoding state s_t, let $p_v = \pi_\theta(v \mid s_t)$ denote the probability assigned to token $v \in \mathcal{V}$. The local policy entropy and its change between consecutive steps are defined as

$$\mathcal{H}_t := -\sum_{v\in\mathcal{V}} p_v \log p_v, \qquad \Delta\mathcal{H}_t := \mathcal{H}(\pi_\theta^{k+1}(\cdot \mid s_t)) - \mathcal{H}(\pi_\theta^{k}(\cdot \mid s_t)). \tag{3}$$

Here π_θ^k and π_θ^{k+1} denote the policies before and after one optimization step. A positive $\Delta\mathcal{H}_t$ indicates that the update locally spreads probability mass over alternative continuations, whereas a negative value indicates local concentration around fewer continuations.

3 Entropy Mechanics

We study entropy mechanics through two complementary quantities. The *intrinsic entropy tendency* $\mathcal{T}(s_t, y_t)$ captures how reinforcing a sampled token would reshape entropy based on the token and distribution alone, while the *entropy polarity* $\mathcal{P}(s_t, y_t, A)$ incorporates the signed training signal and captures the realized first-order entropy effect, including both direction and magnitude.

3.1 A Token-Level View of Entropy Change

We examine how individual policy-gradient updates affect next-token entropy under a local softmax policy. Given a decoding state s_t, consider the next-token policy $\pi_\theta(\cdot \mid s_t)$, parameterized by softmax logits $z_{s_t,v}$ over $v \in \mathcal{V}$. Suppose token $y_t \in \mathcal{V}$ is sampled from state s_t, with an associated advantage $A \in \mathbb{R}$. For compactness, we write

$$p_v := \pi_\theta(v \mid s_t), \qquad p_t := \pi_\theta(y_t \mid s_t), \qquad \mathcal{H}_t := \mathcal{H}(\pi_\theta(\cdot \mid s_t)).$$

Proposition 1 (Logit Update Under Policy Gradient)**.** *Under the local softmax policy, one gradient-ascent step on $A \log \pi_\theta(y_t \mid s_t)$ with step size $\eta > 0$ gives*

$$z'_{s_t,v} = z_{s_t,v} + \eta A\big(\mathbb{I}[v = y_t] - p_v\big), \quad \forall v \in \mathcal{V}.$$

Following this sampled-update perspective, we characterize the resulting first-order entropy response. Equivalent first-order formulations have also been established in prior work [31].

Theorem 1 (First-order Entropy Change via Sampled Updates)**.** *Under the setup of Proposition 1, the entropy change at state s_t satisfies*

$$\Delta\mathcal{H}_t = -\eta A\, t_1(s_t, y_t) + \eta A\, t_2(s_t) + O(\eta^2), \tag{4}$$

where

$$t_1(s_t, y_t) := p_t(\mathcal{H}_t + \log p_t), \qquad t_2(s_t) := \sum_{v\in\mathcal{V}} p_v^2(\mathcal{H}_t + \log p_v). \tag{5}$$

Proof sketch. We derive the result by applying a first-order Taylor expansion to policy entropy, and incorporating the gradient-induced logit update derived in Proposition 1. Under the softmax parametrization,

$$\frac{\partial\mathcal{H}}{\partial z_{s_t,v}} = -p_v(\mathcal{H}_t + \log p_v), \qquad \delta z_{s_t,v} = \eta A\big(\mathbb{I}[v = y_t] - p_v\big). \tag{6}$$

Substituting $\delta z_{s_t,v}$ into the first-order expansion of $\mathcal{H}$ and collecting the sampled-token term and the distribution term yields Eq. (4). Full derivations are deferred to Appendix A.

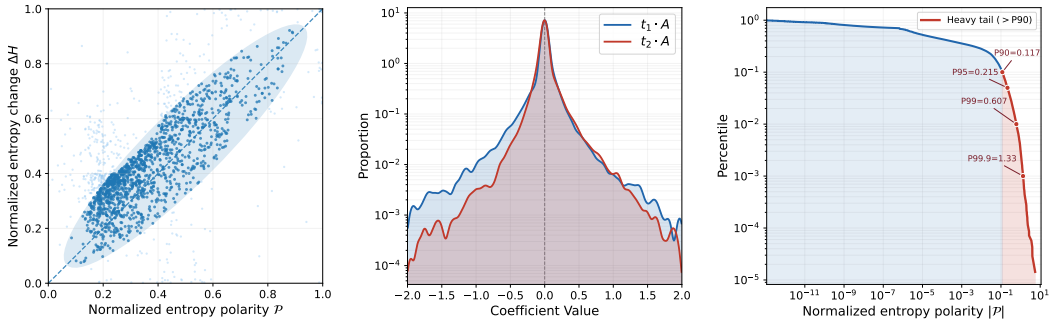


Figure 1: (a) Correlation between polarity score $\mathcal{P}$ and measured entropy change $\Delta\mathcal{H}$. (b) Histograms for the two decomposed components derived from $\mathcal{P}$. (c) Percentile curve of the polarity score.

Theorem 1 shows that the first-order entropy change induced by a sampled token update factorizes into the signed training signal A and an A-independent token-state quantity $-t_1(s_t, y_t)+t_2(s_t)$. We therefore take this combination as the mechanism-level object of our analysis.

Definition 1 (Intrinsic Entropy Tendency)**.** For a sampled token-state pair (s_t, y_t), we define

$$\mathcal{T}(s_t, y_t) := -t_1(s_t, y_t) + t_2(s_t). \tag{7}$$

Here, $\mathcal{T}$ measures the local entropy tendency induced by positively reinforcing y_t at state s_t: $\mathcal{T} > 0$ indicates entropy expansion, while $\mathcal{T} < 0$ indicates entropy contraction. Because it is independent of the signed advantage, $\mathcal{T}$ characterizes the intrinsic tendency from the distribution view.

3.2 Intrinsic Entropy Tendency and Asymmetry

Recall that the intrinsic entropy tendency is $\mathcal{T}(s_t, y_t) = -t_1(s_t, y_t) + t_2(s_t)$, which combines a sampled-token effect and a state-wise distributional correction. The distributional term $t_2(s_t)$ is shared by all sampled tokens at the same state and acts as a correction toward entropy expansion; it is always non-negative, as it can be expressed as the covariance between token probabilities and their log-probabilities. The sampled-token term enters with a negative sign: since $t_1(s_t, y_t) = p_t(\mathcal{H}_t + \log p_t)$, its sign is governed by the threshold $p_t = \exp(-\mathcal{H}_t)$. Thus, low-probability tokens make $-t_1$ expansion-aligned, whereas high-probability tokens make $-t_1$ contractive.

This sign structure explains the asymmetry of $\mathcal{T}$. For low-probability samples, $-t_1$ and t_2 jointly favor expansion. For high-probability samples, $-t_1$ competes against the positive correction t_2, and contraction is triggered when the token-specific concentration effect dominates. Because policy sampling encounters high-probability tokens more often, reinforcing already dominant tokens tends to further concentrate the local distribution, making contractive tendencies easier to trigger. Conversely, expansion requires lower-probability samples or a sufficiently large correction term.

This asymmetry is structural: it is already encoded in $\mathcal{T}$ before the signed training signal is applied; detailed sign analysis are deferred to Appendix A.3. However, policy updates are signed and scaled by the advantage, so structural tendency alone does not yet determine the realized first-order entropy effect of an update. The aggregate first-order effect depends on the joint distribution of sampled tokens and advantages.

3.3 Entropy Polarity and Empirical Insights

From tendency to polarity. Because actual RL updates are signed by the advantage, we define the entropy polarity of a sampled update as

$$\mathcal{P}(s_t, y_t, A) := A\,\mathcal{T}(s_t, y_t), \qquad \Delta\mathcal{H}_t = \eta\,\mathcal{P}(s_t, y_t, A) + O(\eta^2). \tag{8}$$

Thus, $\mathcal{P}$ is the realized first-order entropy effect: positive entropy polarity corresponds to an entropy-expanding update, whereas negative entropy polarity corresponds to an entropy-contracting update. State entropy measures current uncertainty and is shared by all candidate tokens at a given state. Polarity additionally depends on the sampled token and signed advantage, so different samples at the same entropy level can have opposite $\mathcal{P}$ and induce opposite first-order entropy changes.

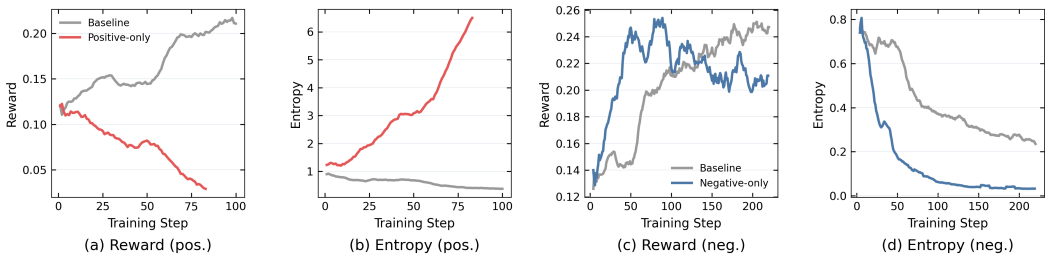


Figure 2: **Ablation study of entropy polarity branches.** We retain exclusively positive or negative polarity tokens to compare training dynamics. Positive-polarity updates boost entropy but impair reward learning, whereas negative-polarity updates obtain significant reward improvements alongside unstable optimization and declining entropy.

Entropy polarity tracks measured entropy change. Theorem 1 predicts that the signed first-order entropy change is governed by the entropy polarity $\mathcal{P} = A\mathcal{T}$. As shown in Figure 1(a), normalized $\mathcal{P}$ aligns closely with the measured local entropy movement during RL training, confirming that entropy polarity is a practical token-level predictor of entropy direction under sampled updates. This agreement suggests that $\mathcal{P}$ is not merely an analytical artifact, but a reliable signal for analyzing how sampled updates shape local entropy.

Polarity effects are structured and non-uniform. Figure 1(b) plots the two signed components $t_1 \cdot A$ and $t_2 \cdot A$ that enter the first-order entropy change. Their substantial overlap indicates that, under practical RLVR training, both the sampled-token contribution and the state-wise distributional correction materially shape the realized entropy effect. Figure 1(c) further shows a long-tailed magnitude distribution of $\mathcal{P}$, where a small fraction of high-magnitude updates accounts for a large share of the cumulative entropy effect.

Together, these results support a two-level view of entropy mechanics: $\mathcal{T}$ captures the intrinsic entropy tendency of a sampled token-state pair, while $\mathcal{P}$ incorporates the signed advantage and captures the realized first-order entropy effect from both direction and magnitude.

4 Method

4.1 Motivation: Expansion and Contraction Branches

Positive and negative entropy polarity induce opposite training dynamics. To explore the effect of polarity sign in RL, we conduct a single-polarity ablation of RL (Details are presented in Appendix B.5). Figure 2 shows that retaining only positive polarity updates preserves or expands entropy but weakens reward, whereas retaining only negative polarity updates improves reward early but rapidly contracts entropy and reduces later exploration. This indicates that entropy polarity separates entropy-expanding and entropy-contracting branches, and that neither branch alone supports stable reinforcement fine-tuning.

Characterizing the functional roles of entropy-polarity branches. To understand this distinction, we inspect tokens from each polarity branch. Quantitatively, Figure 3 shows that positive-polarity tokens are associated with both higher context entropy and greater context diversity, indicating that they tend to occur in more open next-token distributions. We further examine their lexical profiles in Figure 4. Positive-polarity tokens cluster around connective and structural tokens that keep continuations open, whereas negative-polarity tokens concentrate on formulaic reasoning fragments and answer-oriented transitions in more constrained contexts. More discussion about the position concentration of entropy

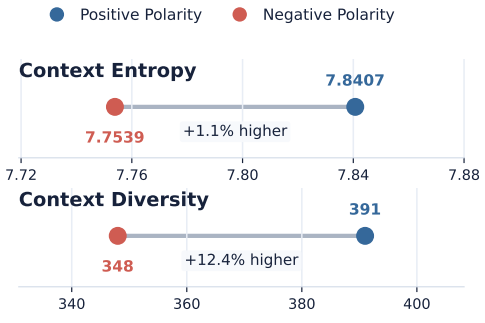
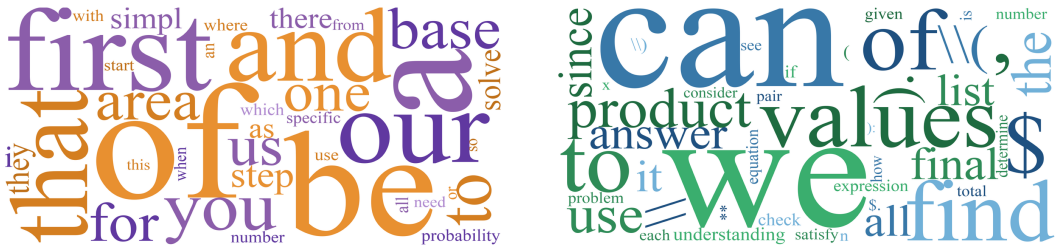


Figure 3: Contextual statistics of positive and negative polarity.



(a) Positive Polarity (b) Negative Polarity

Figure 4: Word clouds of positive and negative polarity branches with the high absolute $\|\mathcal{P}\|$ polarity values, respectively. The positive branch ($\mathcal{P} > 0$) is enriched in connective and structural expressions, e.g., "and", "that", "first", and "step", consistent with more open local continuations; the negative branch ($\mathcal{P} < 0$) instead concentrates on formulaic reasoning and answer-oriented expressions, e.g., "find", "values", "product", and "final", consistent with more constrained solution trajectories.

polarity is supplemented in Appendix C.1. Together, these observations suggest that positive-polarity updates help preserve action diversity, while negative-polarity updates sharpen reward-aligned exploitation. Additional continuation statistics are reported in Appendix C.2.

Motivation for branch-level control. Taken together, these insights indicate complementary roles: **the positive polarity branch preserves exploratory capacity, whereas the negative polarity branch strengthens exploitation**. An effective RFT strategy should therefore keep both branches while controlling their relative influence over training, rather than favoring one entropy direction or treating them uniformly, as in GRPO. Since the desired exploration–exploitation balance changes over time, this control should be adaptive rather than fixed.

4.2 Polarity-Aware Policy Optimization

To this end, we propose **P**olarity-**A**ware **P**olicy **O**ptimization (PAPO), which performs branch-level control by reweighting token advantages according to entropy polarity. Rather than targeting a fixed entropy level, PAPO uses the empirical entropy trajectory as an online phase signal to redistribute optimization pressure between entropy-expanding and entropy-contracting updates, protecting exploration when entropy declines rapidly and strengthening exploitation once the trajectory stabilizes.

General Formulation. We formalize this design by modulating the token advantage. For training step k, the polarity-weighted advantage $\tilde{A}_t^{(i)}$ is defined as:

$$\tilde{A}_t^{(i)} = \begin{cases} A^{(i)} \cdot \omega_{\text{pos}}(k), & \mathcal{P}(s_t^{(i)}, y_t^{(i)}, A^{(i)}) \geq 0, \\ A^{(i)} \cdot \omega_{\text{neg}}(k), & \mathcal{P}(s_t^{(i)}, y_t^{(i)}, A^{(i)}) < 0, \end{cases} \tag{9}$$

Here $A^{(i)}$ is the group-normalized advantage, while the branch assignment is token-specific through the sign of $\mathcal{P}(s_t^{(i)}, y_t^{(i)}, A^{(i)})$. This formulation preserves the original reward objective and recovers GRPO when $\omega_{\text{pos}} = \omega_{\text{neg}} = 1$.

Adaptive Entropy Modulation. Since the desired exploration–exploitation balance depends on the training dynamics, fixed polarity weights are generally suboptimal. Let $s_k \triangleq \text{EMA}_\beta(\Delta H)$ denote the smoothed entropy slope, PAPO uses the empirical entropy trajectory through s_k as an online phase signal. After a short warmup with neutral weights, we record the warmup slope and entropy as the reference rate $s_{\text{ref}} < 0$ and reference level h_{ref}, respectively.

By rescaling the current slope against s_{ref}, we instantiate a normalized progress metric p_k that measures how much the current entropy decay has recovered from the warmup contraction rate:

$$p_k = \text{clip}\left(\frac{s_k - s_{\text{ref}}}{-s_{\text{ref}} + \varepsilon}, 0, 1\right). \tag{10}$$

Table 1: **Performance (%, Mean@32) on in-domain and out-of-domain benchmarks.** Cells with deeper background color correspond to better performance within each model group.

Methods	In-Domain Performance						Out-of-Domain		
	Olympiad	AMC23	AIME24	AIME25	MATH500	Minerva	CRUX	IFEval	MMLU-Pro
Qwen2.5-7B-Base									
- DAPO	34.68	46.09	10.83	6.04	66.19	25.14	41.49	30.30	44.55
- Ent-Reg	34.65	45.86	9.79	7.71	67.05	26.88	41.63	30.49	44.56
- 80/20	34.12	47.42	9.58	5.62	65.97	24.82	41.43	30.54	44.45
- STEER	34.47	47.11	9.17	6.04	66.23	26.14	41.41	30.40	44.69
- Clip-Cov	34.87	48.36	10.83	6.77	66.89	26.11	41.46	30.56	44.58
- KL-Cov	34.47	48.20	10.62	7.08	66.64	25.72	41.39	31.01	44.60
- PAPO	35.57	49.22	11.46	7.19	67.46	27.22	44.79	30.65	44.76
Qwen2.5-14B-Base									
- DAPO	37.73	55.47	11.77	9.90	70.46	30.62	51.26	34.43	53.92
- Ent-Reg	37.92	55.08	12.19	10.52	71.19	31.00	51.20	34.19	53.86
- 80/20	37.47	54.53	11.98	10.83	70.37	30.41	50.36	34.23	53.80
- STEER	36.82	55.39	11.04	9.58	70.06	30.46	51.13	34.22	53.89
- Clip-Cov	37.56	55.86	11.88	10.21	70.31	29.99	49.18	34.25	53.72
- KL-Cov	37.21	55.62	11.67	8.65	70.28	30.51	49.23	34.24	53.63
- PAPO	39.23	58.36	13.44	12.19	71.31	31.70	51.82	34.27	53.79

Smaller p_k indicates ongoing entropy collapse, while larger p_k indicates recovery. We then map p_k to polarity weights through a quadratic rule with reciprocal coupling:

$$\omega_{\text{neg}}(k) = \omega_{\min} + (\omega_{\max} - \omega_{\min})p_k^2, \qquad \omega_{\text{pos}}(k) = \frac{1}{\omega_{\text{neg}}(k)}. \tag{11}$$

When $p_k \approx 0$, the controller protects exploration by suppressing entropy-contracting updates and strengthening entropy-expanding ones; as p_k increases, this bias relaxes toward neutral weighting and can mildly favor exploitation. The quadratic mapping keeps the response conservative, the reciprocal coupling focuses control on relative branch pressure, and a gate at $\gamma_{\text{gate}} \cdot h_{\text{ref}}$ disables reweighting in the extreme low-entropy regime. The pseudocode of PAPO is detailed in Algorithm 1.

5 Experiments

5.1 Experimental Setup

Datasets and Models. For the main mathematical reasoning experiments, we train on DAPO-Math-17K [32], a widely adopted high-quality RLVR dataset. We use Qwen2.5-7B-Base and Qwen2.5-14B-Base [33] as base models, ensuring consistency with baseline comparisons. Additional agentic reasoning experiments use a separate training setup, described in Section 5.3.

Evaluation. To comprehensively assess the model capabilities, we employ a diverse suite of benchmarks encompassing mathematical reasoning, code reasoning, and general multi-task proficiency. Specifically: (1) Mathematical reasoning: OlympiadBench [34], AMC [35], MATH500 [36], and AIME [37, 38], and Minerva [39]; (2) Code reasoning: CRUX [40]; (3) Instruction-following and general abilities: IFEval [41] and MMLU-Pro [42].

Implementation Details. We implement our algorithms based on the VeRL framework and conduct experiments on 16 $\times$ H20 GPUs. During RL training, the learning rate is set to 1×10^{-6}, with a global batch size of 128, and gradient accumulation over 64 steps. For each group, 8 candidate responses are sampled. PAPO computes polarity statistics using a top-k approximation with $k = 2048$ and reweights existing advantages, adding no extra forward pass, auxiliary loss, or trainable parameters. More implementation illustrations are in Appendix B.4.

5.2 Main Results

Superior performance on mathematical reasoning. Table 1 reports the results of mathematical reasoning on six benchmarks. PAPO achieves the best performance among the compared methods on most benchmarks and delivers consistent overall improvements across both 7B and 14B model

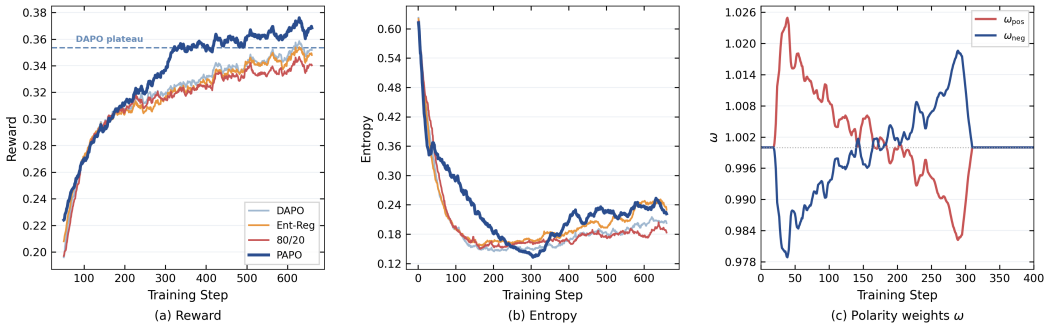


Figure 5: **Training dynamics of PAPO.** Reward (left), entropy (middle) and polarity weight (right) of Qwen2.5-14B-Base trained on DAPO-Math-17k. PAPO delivers significant reward improvement while maintaining a healthier entropy dynamic, and its adaptive entropy modulation relaxes back to near-neutral weights after the transient collapse-recovery phase.

scales: For Qwen2.5-14B, PAPO outperforms DAPO by 1.67 percentage points on AIME24 and 2.29 points on AIME25; it obtains gains of 2.08 points on Minerva and 3.13 points on AMC23 on Qwen2.5-7B. Entropy regularization works well on MATH500, but its effects on competition benchmarks vary across tasks and model sizes. At 14B, PAPO achieves clearer gains over entropy regularization on AMC23 and AIME while performing comparably on MATH500. Thus PAPO's token-level polarity-aware reweighting provides finer-grained control, yielding more consistent improvements across benchmarks and scales.

PAPO reshapes policy training through phase-adaptive control. Figure 5 shows that PAPO regulates optimization by adaptively rebalancing exploration and exploitation across training phases. In the early stage, the controller prioritizes exploration protection by down-weighting the entropy-contracting branch ($\omega_{\text{neg}} < 1$), which helps avoid premature contraction. As entropy decay slows, the controller gradually restores and then mildly favors exploitation ($\omega_{\text{neg}} > 1$). After the late-stage gate is triggered, the reweighting is deactivated, and training proceeds with neutral branch weights. This phase-wise control produces a non-monotonic entropy trajectory: PAPO exhibits a transient mid-training contraction, then stabilizes at a higher reward and entropy level than DAPO.

PAPO achieves significant training efficiency. The reward curve in Figure 5 displays significant training efficiency: our method enables the policy model to reach the mature reward performance of DAPO withapproximately 60% of the training steps, and continues to increase throughout the training stage. Such rapid capability advancement does not stem from over-aggressive early exploitation. As validated by entropy trajectories, the policy preserves strong exploratory capacity, manifesting that the rapid performance growth of our model does not compromise reasoning diversity.

5.3 Analysis

Generalization on out-of-domain benchmarks. We further evaluate the cross-domain generalization of math-optimized checkpoints, including code reasoning (CRUX), instruction following (IFEval), and general knowledge (MMLU-Pro). Despite being trained exclusively on math, PAPO achieves comparable or even superior results over DAPO. For the 14B variant, our method performs comparably to DAPO across the three benchmarks. In the 7B setting, PAPO improves over DAPO by 3.30 percentage points on CRUX, with comparable performance on IFEval and MMLU-Pro. The transfer gain on CRUX, alongside broadly stable instruction-following and knowledge scores, suggests that polarity-aware training strengthens reasoning beyond the math domain while retaining broader capabilities.

Transferability in agentic tool-calling scenarios. We further evaluate PAPO in agentic tool-call generation scenarios, which place high demands on agent-level capabilities such as precise tool intent comprehension and accurate function invocation decision-making, differing from open-ended mathematical reasoning. This setting tests whether polarity-aware reweighting captures a transferable optimization signal. We train Qwen2.5-14B-Instruct on Tool-N1 [43] and evaluate on BFCL-

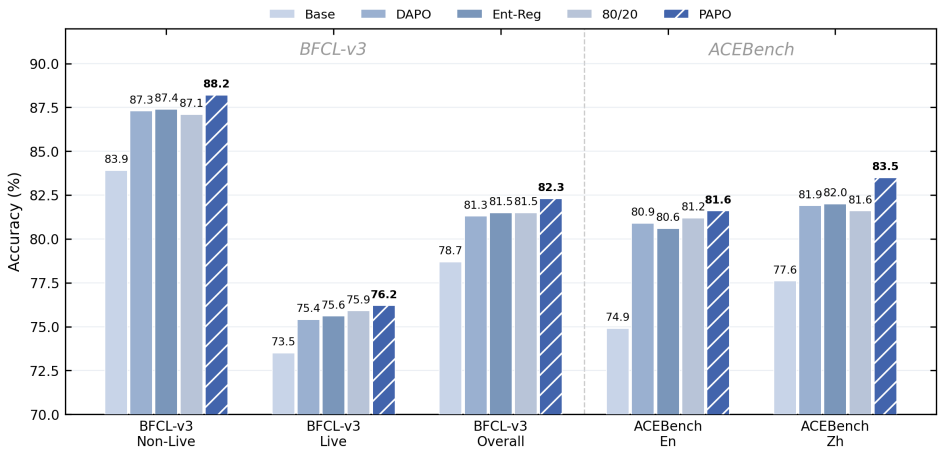


Figure 6: **Performance (%) on agentic tool-using benchmarks.** PAPO achieves the strong overall performance. It consistently surpasses baselines across different evaluation settings, covering non-live and live scenarios in BFCL, as well as English and Chinese tasks in ACEBench.

v3 [44] and ACEBench [45]. As shown in Figure 6, PAPO achieves the strongest overall performance among matched methods on both benchmarks, achieving an average +1.0% improvement on BFCL-v3 and SOTA performance on ACEBench among baselines. This suggests that effective-polarity reweighting transfers beyond mathematical reasoning to structured tool-call actions.

Effect of polarity-aware strength on policy dynamics. Figure 7 compares training dynamics with different polarity-control strengths; detailed settings are deferred to Appendix B.6. The exploration-oriented setting preserves entropy more aggressively by suppressing entropy-contracting updates, but consequently weakens the optimization pressure needed for reward improvement, leading to slower gains and a lower final reward. The exploitation-oriented setting exhibits the opposite behavior: reward improves rapidly early on because the contractive branch is less restrained, but this acceleration is achieved at the expense of entropy, inducing premature collapse of exploratory capacity while reaching the highest training reward. The moderate setting maintains a controlled entropy decline with partial recovery while eventually reaching a competitive reward level. In short, polarity control needs to balance the entropy cost of reward gains, keeping the contractive branch sufficiently robust to sustain learning and avoid premature entropy collapse.

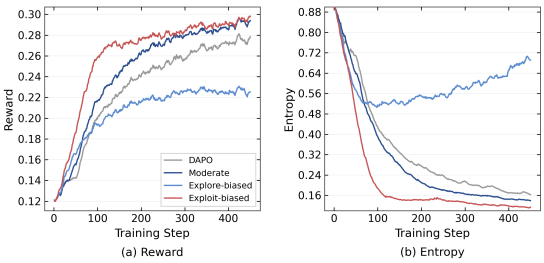


Figure 7: Effect of polarity-control strength ω on policy dynamics.

6 Related Work

Reinforcement Learning with Verifiable Rewards. RLVR has become a dominant paradigm for improving the reasoning capabilities of large language models [13, 5, 46], motivating a broad line of work on stable policy optimization. GRPO [5] pioneered critic-free optimization for reasoning LLMs, while later methods such as DAPO [32] and VAPO [47] further improved training stability through refined optimization and value modeling. Subsequent work has explored complementary directions, including scalable training recipes [6, 48], sequence-level stabilization for MoE models [49, 50], and enhanced credit assignment or self-improving training schemes [51, 52].

Entropy Dynamics and Control. Entropy has emerged as a useful lens for understanding exploratory capacity and reasoning behavior in RLVR [25, 22, 53, 20, 29, 54]. Prior work studies entropy dynamics from both analytical and empirical perspectives, showing that RLVR updates often concentrate on a small set of high-impact reasoning tokens [25, 53, 20]. To mitigate entropy collapse, existing methods mainly rely on global or heuristic interventions, including clipping-based trust-region modifications [32, 55, 30] and advantage-shaping or adaptive-control schemes [56, 26]. In contrast, our work focuses on how sampled token updates directionally expand or contract entropy, and uses this token-level mechanism to implement entropy control through advantage reweighting.

7 Conclusion, Limitation and Future Work

Conclusion. Building on first-order analyses of entropy dynamics, this work develops an entropy-polarity framework that distinguishes entropy-expanding and entropy-contracting updates in RLVR. Through polarity-specific interventions and behavioral analyses, we demonstrate the complementary training dynamics of the two branches, highlighting their distinct tendencies toward exploration and exploitation. Motivated by these findings, we propose PAPO, which adaptively rebalances the two branches using the observed entropy trajectory as an online feedback signal. Experiments on mathematical reasoning and agentic benchmarks demonstrate consistent overall performance improvements, supporting polarity-aware control as a practical approach to balancing exploration and exploitation in RLVR.

Limitation and Future Work. Our study is conducted in text-only RLVR settings with 7B/14B-scale LLMs and standard verifiable rewards, and does not yet examine long-horizon reasoning or interaction regimes involving hundreds of sequential steps. Extending the polarity perspective to larger-scale models, multimodal settings, and more diverse reward designs remains an important direction for future work.

Acknowledgments and Disclosure of Funding

The authors wish to thank the anonymous reviewers for their helpful comments. This work was partially funded by National Natural Science Foundation of China (No. 62576106, 62521004, 62476061, 62376061).

References

[1] Richard S. Sutton and Andrew G. Barto. Reinforcement learning: An introduction. *IEEE Trans. Neural Networks*, 9(5):1054–1054, 1998.

[2] D Guo, D Yang, H Zhang, J Song, P Wang, Q Zhu, R Xu, R Zhang, S Ma, X Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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

[4] Zhiheng Xi, Xin Guo, Jiaqi Liu, Jiazheng Zhang, Yutao Fan, Zhihao Zhang, Shichun Liu, Mingxu Chai, Xiaowei Shi, Yitao Zhai, Xunliang Cai, Tao Gui, Qi Zhang, and Xuanjing Huang. Can RL improve generalization of LLM agents? an empirical study. *CoRR*, abs/2603.12011, 2026.

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

[6] Chenxin An, Zhihui Xie, Xiaonan Li, Lei Li, Jun Zhang, Shansan Gong, Ming Zhong, Jingjing Xu, Xipeng Qiu, Mingxuan Wang, and Lingpeng Kong. Polaris: A post-training recipe for scaling reinforcement learning on advanced reasoning models, 2025.

[7] Anthropic. Claude code, 2025.

[8] Shihan Dou, Jiazheng Zhang, Jianxiang Zang, Yunbo Tao, Weikang Zhou, Haoxiang Jia, Shichun Liu, Yuming Yang, Shenxi Wu, Zhiheng Xi, et al. Multi-programming language sandbox for llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pages 40–50, 2025.

[9] Yujiong Shen, yajie yang, Zhiheng Xi, Binze Hu, Huayu Sha, Qiyuan Peng, Jiazheng Zhang, Junlin Shang, Jixuan Huang, Yutao Fan, Jingqi Tong, Ming Zhang, Shihan Dou, Zhenfei Yin, Xingjun Ma, LEI BAI, Tao Gui, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Sciagent-gym: Benchmarking multi-step scientific tool-use in LLM agents. In *Forty-third International Conference on Machine Learning*, 2026.

[10] Zhiheng Xi, Jixuan Huang, Chenyang Liao, Baodai Huang, Honglin Guo, Jiaqi Liu, Rui Zheng, Junjie Ye, Jiazheng Zhang, Wenxiang Chen, Wei He, Yiwen Ding, Guanyu Li, Zehui Chen, Zhengyin Du, Xuesong Yao, Yufei Xu, Jiecao Chen, Tao Gui, Zuxuan Wu, Qi Zhang, Xuanjing Huang, and Yu-Gang Jiang. Agentgym-rl: Training LLM agents for long-horizon decision making through multi-turn reinforcement learning. *CoRR*, abs/2509.08755, 2025.

[11] Lang Feng, Zhenghai Xue, Tingcong Liu, and Bo An. Group-in-group policy optimization for LLM agent training. *CoRR*, abs/2505.10978, 2025.

[12] Jiazheng Zhang, Long Ma, Yunxian Yang, Zhiheng Xi, Zhikai Lei, Yajie Yang, Chenyang Liao, Enyu Zhou, Yang Nan, Yuchen Tian, et al. Scaling verifiable environments for long-horizon work agents. *arXiv preprint arXiv:2610.04906*, 2026.

[13] Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiaze Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models. *CoRR*, abs/2509.08827, 2025.

[14] Boyang Liu, Senjie Jin, Peixin Wang, Zhangyue Yin, Yibo Wang, Yuhao Zhou, Zhihao Zhang, Xinbing Liang, Shizheng Zhu, Yuhui Wang, et al. Cafe: Self-improving search agents need co-evolving feedback. *arXiv preprint arXiv:2608.24794*, 2026.

[15] Wenxiang Chen, Wei He, Zhiheng Xi, Honglin Guo, Boyang Hong, Jiazheng Zhang, Nijun Li, Tao Gui, Yun Li, Qi Zhang, et al. Better process supervision with bi-directional rewarding signals. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 14471–14485, 2025.

[16] Jiazheng Zhang, Ziche Fu, Zhiheng Xi, Wenqing Jing, Mingxu Chai, Wei He, Guoqiang Zhang, Chenghao Fan, Chenxin An, Wenxiang Chen, Zhicheng Liu, Haojie Pan, Dingwei Zhu, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentV-RL: Scaling reward modeling with agentic verifier. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens, editors, *Findings of the Association for Computational Linguistics: ACL 2026*, pages 23078–23100, San Diego, California, United States, July 2026. Association for Computational Linguistics.

[17] Evan Zheran Liu, Aditi Raghunathan, Percy Liang, and Chelsea Finn. Decoupling exploration and exploitation for meta-reinforcement learning without sacrifices. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, Proceedings of Machine Learning Research, pages 6925–6935. PMLR, 2021.

[18] Xinyu Tang, Yuliang Zhan, Zhixun Li, Wayne Xin Zhao, Zhenduo Zhang, Zujie Wen, Zhiqiang Zhang, and Jun Zhou. Rethinking sample polarity in reinforcement learning with verifiable rewards. *CoRR*, abs/2512.21625, 2025.

[19] Jiazheng Zhang, Wenqing Jing, Zizhuo Zhang, Zhiheng Xi, Shihan Dou, Rongxiang Weng, Jiahuan Li, Jingang Wang, Mingxu Chai, Shibo Hong, et al. Two minds better than one: Collaborative reward modeling for llm alignment. *arXiv preprint arXiv:2505.10597*, 2025.

[20] Kexin Huang, Haoming Meng, Junkang Wu, Jinda Lu, Chiyu Ma, Ziqian Chen, Xue Wang, Bolin Ding, Jiancan Wu, Xiang Wang, Xiangnan He, Guoyin Wang, and Jingren Zhou. Beyond magnitude: Leveraging direction of RLVR updates for LLM reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026.

[21] Chengjun Pan, Shichun Liu, Jiahang Lin, Dingwei Zhu, Jiazheng Zhang, Shihan Dou, Songyang Gao, Zhenhua Han, Binghai Wang, Rui Zheng, et al. Evpo: Explained variance policy optimization for adaptive critic utilization in llm post-training. *arXiv preprint arXiv:2604.19485*, 2026.

[22] Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *CoRR*, abs/2505.22617, 2025.

[23] Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor, editors, *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 30377–30385. AAAI Press, 2026.

[24] Peter Chen, Xiaopeng Li, Ziniu Li, Wotao Yin, Xi Chen, and Tianyi Lin. Exploration vs exploitation: Rethinking RLVR through clipping, entropy, and spurious reward. *CoRR*, abs/2512.16912, 2025.

[25] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[26] Aleksei Petrenko, Ben Lipkin, Kevin Chen, Erik Wijmans, Marco F. Cusumano-Towner, Raja Giryes, and Philipp Krähenbühl. Entropy-preserving reinforcement learning. *CoRR*, abs/2603.11682, 2026.

[27] Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *CoRR*, abs/2505.22660, 2025.

[28] Han Shen. On entropy control in LLM-RL algorithms. *CoRR*, abs/2509.03493, 2025.

[29] Zhiheng Xi, Xin Guo, Yang Nan, Enyu Zhou, Junrui Shen, Wenxiang Chen, Jiaqi Liu, Jixuan Huang, Xun Deng, Zhihao Zhang, Honglin Guo, Zhikai Lei, Miao Zheng, Guoteng Wang, Peng Sun, Rui Zheng, Hang Yan, Tao Gui, Qi Zhang, and Xuanjing Huang. BAPO: Stabilizing off-policy reinforcement learning for LLMs via balanced policy optimization with adaptive clipping. In *The Fourteenth International Conference on Learning Representations*, 2026.

[30] Kun Chen, Peng Shi, Fanfan Liu, Haibo Qiu, Zhixiong Zeng, Siqi Yang, and Wenji Mao. Flexible entropy control in RLVR with gradient-preserving perspective. *CoRR*, abs/2602.09782, 2026.

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

[32] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

[33] Qwen, An Yang, Baosong Yang, et al. Qwen2.5 technical report, 2024.

[34] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 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, 2024.

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

[36] Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024.

[37] Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2024. 2024.

[38] Yifan Zhang and Team Math-AI. American invitational mathematics examination (aime) 2025. 2025.

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

[40] Alex Gu, Baptiste Rozière, Hugh Leather, Armando Solar-Lezama, Gabriel Synnaeve, and Sida I Wang. Cruxeval: a benchmark for code reasoning, understanding and execution. In *Proceedings of the 41st International Conference on Machine Learning*, pages 16568–16621, 2024.

[41] Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*, 2023.

[42] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, et al. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. *Advances in Neural Information Processing Systems*, 37:95266–95290, 2024.

[43] Shaokun Zhang, Yi Dong, Jieyu Zhang, Jan Kautz, Bryan Catanzaro, Andrew Tao, Qingyun Wu, Zhiding Yu, and Guilin Liu. Nemotron-research-tool-n1: Exploring tool-using language models with reinforced reasoning. *arXiv preprint arXiv:2505.00024*, 2025.

[44] Shishir G Patil, Huanzhi Mao, Fanjia Yan, Charlie Cheng-Jie Ji, Vishnu Suresh, Ion Stoica, and Joseph E Gonzalez. The berkeley function calling leaderboard (bfcl): From tool use to agentic evaluation of large language models. In *Forty-second International Conference on Machine Learning*, 2025.

[45] Chen Chen, Xinlong Hao, Weiwen Liu, Xu Huang, Xingshan Zeng, Shuai Yu, Dexun Li, Shuai Wang, Weinan Gan, Yuefeng Huang, et al. Acebench: Who wins the match point in tool usage? *arXiv preprint arXiv:2501.12851*, 2025.

[46] Zhihao Zhang, Qiaole Dong, Qi Zhang, Jun Zhao, Enyu Zhou, Zhiheng Xi, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Mingqi Wu, et al. Why reinforcement fine-tuning enables mllms preserve prior knowledge better: A data perspective. *arXiv preprint arXiv:2506.23508*, 2025.

[47] Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

[48] Bingxiang He, Zekai Qu, Zeyuan Liu, Yinghao Chen, Yuxin Zuo, Cheng Qian, Kaiyan Zhang, Weize Chen, Chaojun Xiao, Ganqu Cui, et al. Justrl: Scaling a 1.5 b llm with a simple rl recipe. *arXiv preprint arXiv:2512.16649*, 2025.

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

[50] Chang Gao, Chujie Zheng, Xionghui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *CoRR*, abs/2511.20347, 2025.

[51] Fengwei Teng, Jinyi Bai, Xinhao Yao, Demi Ruohan Wang, Jiahao Zhao, and Zhijiang Guo. Skip-connected policy optimization for implicit advantage. *arXiv preprint arXiv:2604.08690*, 2026.

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

[53] Haoming Meng, Kexin Huang, Shaohang Wei, Chiyu Ma, Shuo Yang, Xue Wang, Guoyin Wang, Bolin Ding, and Jingren Zhou. Sparse but critical: A token-level analysis of distributional shifts in RLVR fine-tuning of LLMs. In *The Fourteenth International Conference on Learning Representations*, 2026.

[54] Safa Messaoud, Billel Mokeddem, Zhenghai Xue, Linsey Pang, Bo An, Haipeng Chen, and Sanjay Chawla. S^2ac: Energy-based reinforcement learning with stein soft actor critic. *arXiv preprint arXiv:2405.00987*, 2024.

[55] Zhenpeng Su, Leiyu Pan, Minxuan Lv, Yuntao Li, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. CE-GPPO: coordinating entropy via gradient-preserving clipping policy optimization in reinforcement learning. *CoRR*, abs/2509.20712, 2025.

[56] Xinzhu Chen, Xuesheng Li, Zhongxiang Sun, and Weijie Yu. Beyond high-entropy exploration: Correctness-aware low-entropy segment-based advantage shaping for reasoning llms. *CoRR*, abs/2512.00908, 2025.

[57] Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. Rethinking entropy interventions in RLVR: An entropy change perspective, 2025.

A Theoretical Derivations

This appendix provides the full derivations for Proposition 1 and Theorem 1, and further analyzes the sign and magnitude structure of the intrinsic entropy tendency $\mathcal{T}$. Throughout this appendix, we fix a decoding state s_t, and write

$$p_v := \pi_\theta(v \mid s_t), \qquad p_t := \pi_\theta(y_t \mid s_t), \qquad \mathcal{H}_t := -\sum_{v \in \mathcal{V}} p_v \log p_v.$$

All quantities are evaluated at the pre-update policy unless otherwise specified.

A.1 Proof of the Sampled Logit Update

Restatement. For a fixed decoding state s_t, suppose token y_t is sampled and the local policy is updated by the single policy-gradient term $A \log \pi_\theta(y_t \mid s_t)$ with step size η. Then the first-order logit update satisfies

$$z'_{s_t,v} = z_{s_t,v} + \eta A \left(\mathbb{I}[v = y_t] - p_v\right), \qquad \forall v \in \mathcal{V}.$$

Proof. Under the local softmax parameterization,

$$p_v = \pi_\theta(v \mid s_t) = \frac{\exp(z_{s_t,v})}{\sum_{u\in\mathcal{V}} \exp(z_{s_t,u})}.$$

For the sampled token y_t, we have

$$\log \pi_\theta(y_t \mid s_t) = z_{s_t,y_t} - \log\left(\sum_{u\in\mathcal{V}} \exp(z_{s_t,u})\right).$$

Differentiating with respect to $z_{s_t,v}$ gives

$$\frac{\partial \log \pi_\theta(y_t \mid s_t)}{\partial z_{s_t,v}} = \mathbb{I}[v = y_t] - p_v.$$

Therefore,

$$\frac{\partial}{\partial z_{s_t,v}} \left(A \log \pi_\theta(y_t \mid s_t)\right) = A\left(\mathbb{I}[v = y_t] - p_v\right).$$

Applying one gradient-ascent step yields

$$z'_{s_t,v} = z_{s_t,v} + \eta A\left(\mathbb{I}[v = y_t] - p_v\right),$$

which proves Proposition 1. □

A.2 Proof of the Entropy-Polarity Theorem

Restatement. Under the same local softmax setting, the one-step entropy change induced by the sampled update in Proposition 1 satisfies

$$\Delta\mathcal{H}_t = -\eta A\, t_1(s_t, y_t) + \eta A\, t_2(s_t) + O(\eta^2),$$

where

$$t_1(s_t, y_t) := p_t(\mathcal{H}_t + \log p_t), \qquad t_2(s_t) := \sum_{v\in\mathcal{V}} p_v^2(\mathcal{H}_t + \log p_v).$$

Equivalently, with

$$\mathcal{T}(s_t, y_t) := -t_1(s_t, y_t) + t_2(s_t),$$

we have

$$\Delta\mathcal{H}_t = \eta A\, \mathcal{T}(s_t, y_t) + O(\eta^2).$$

Proof. We first compute the gradient of the local entropy with respect to the logits. Since

$$\mathcal{H}_t = -\sum_{u\in\mathcal{V}} p_u \log p_u,$$

and the softmax derivative satisfies

$$\frac{\partial p_u}{\partial z_{s_t,v}} = p_u(\mathbb{I}[u = v] - p_v),$$

we obtain

$$\begin{aligned}
\frac{\partial \mathcal{H}_t}{\partial z_{s_t,v}} &= -\sum_{u\in\mathcal{V}} \frac{\partial p_u}{\partial z_{s_t,v}} (\log p_u + 1) \\
&= -\sum_{u\in\mathcal{V}} p_u(\mathbb{I}[u = v] - p_v)(\log p_u + 1) \\
&= -p_v(\log p_v + 1) + p_v \sum_{u\in\mathcal{V}} p_u(\log p_u + 1). \qquad (12)
\end{aligned}$$

Using

$$\sum_{u\in\mathcal{V}} p_u(\log p_u + 1) = -\mathcal{H}_t + 1,$$

we have

$$\frac{\partial \mathcal{H}_t}{\partial z_{s_t,v}} = -p_v(\mathcal{H}_t + \log p_v).$$

Let

$$\delta z_{s_t,v} := z'_{s_t,v} - z_{s_t,v}.$$

By the first-order Taylor expansion of entropy around the pre-update logits,

$$\Delta\mathcal{H}_t = \sum_{v\in\mathcal{V}} \frac{\partial \mathcal{H}_t}{\partial z_{s_t,v}} \delta z_{s_t,v} + O(\|\delta z_t\|^2).$$

From Proposition 1,

$$\delta z_{s_t,v} = \eta A(\mathbb{I}[v = y_t] - p_v).$$

Substituting the entropy gradient and the logit update gives

$$\begin{aligned}\Delta\mathcal{H}_t &= -\eta A \sum_{v\in\mathcal{V}} p_v(\mathcal{H}_t + \log p_v)(\mathbb{I}[v = y_t] - p_v) + O(\eta^2) \\ &= -\eta A p_t(\mathcal{H}_t + \log p_t) + \eta A \sum_{v\in\mathcal{V}} p_v^2(\mathcal{H}_t + \log p_v) + O(\eta^2). \end{aligned} \tag{13}$$

By the definitions in Eq. (5), this becomes

$$\Delta\mathcal{H}_t = -\eta A\, t_1(s_t, y_t) + \eta A\, t_2(s_t) + O(\eta^2),$$

which proves Eq. (4).

Using Eq. (7),

$$\mathcal{T}(s_t, y_t) = -t_1(s_t, y_t) + t_2(s_t),$$

we further obtain

$$\Delta\mathcal{H}_t = \eta A\, \mathcal{T}(s_t, y_t) + O(\eta^2).$$

Together with the definition

$$\mathcal{P}(s_t, y_t, A) = A\mathcal{T}(s_t, y_t)$$

in Eq. (8), this also gives

$$\Delta\mathcal{H}_t = \eta\, \mathcal{P}(s_t, y_t, A) + O(\eta^2).$$

This completes the proof of Theorem 1. □

A.3 Sign Analysis and Directional Asymmetry of Entropy Tendency

Theorem 1 shows that the intrinsic entropy tendency is

$$\mathcal{T}(s_t, y_t) = -t_1(s_t, y_t) + t_2(s_t).$$

This section further analyzes how t_1 and t_2 determine the sign and magnitude of $\mathcal{T}$.

The sampled-token term t_1. The term

$$t_1(s_t, y_t) = p_t(\mathcal{H}_t + \log p_t)$$

is the only component in $\mathcal{T}$ that depends on the sampled token y_t. Since $p_t > 0$, its sign is fully determined by $\mathcal{H}_t + \log p_t$:

$$\operatorname{sgn}(t_1(s_t, y_t)) = \operatorname{sgn}(\mathcal{H}_t + \log p_t)\,.$$

Therefore,

$$t_1(s_t, y_t) > 0 \quad \Longleftrightarrow \quad p_t > \exp(-\mathcal{H}_t),$$

and

$$t_1(s_t, y_t) < 0 \quad \Longleftrightarrow \quad p_t < \exp(-\mathcal{H}_t).$$

Thus, t_1 separates sampled tokens by an entropy-induced probability threshold. Tokens above this threshold have a positive t_1, and hence contribute negatively to $\mathcal{T}$ through the term $-t_1$. Tokens below this threshold have a negative t_1, and hence contribute positively to $\mathcal{T}$.

The distribution term t_2. The term

$$t_2(s_t) = \sum_{v \in \mathcal{V}} p_v^2 (\mathcal{H}_t + \log p_v)$$

does not depend on the sampled token. Instead, it is shared by all sampled-token updates at the same decoding state and depends only on the shape of the local next-token distribution. A useful equivalent form shows that t_2 is always non-negative:

$$\begin{aligned} t_2(s_t) &= \sum_{v \in \mathcal{V}} p_v^2 \log p_v + \mathcal{H}_t \sum_{v \in \mathcal{V}} p_v^2 \\ &= \mathbb{E}_{v \sim p}[p_v \log p_v] - \mathbb{E}_{v \sim p}[p_v]\, \mathbb{E}_{v \sim p}[\log p_v] \\ &= \mathrm{Cov}_{v \sim p}(p_v, \log p_v) \geq 0. \end{aligned}$$

The last inequality follows because p_v and $\log p_v$ are monotone functions of the same probability value. Therefore, $t_2(s_t)$ acts as a non-negative state-wise distributional correction. Since it enters $\mathcal{T}(s_t, y_t) = -t_1(s_t, y_t) + t_2(s_t)$ with a positive sign, it shifts the intrinsic tendency toward entropy expansion for all sampled tokens at the same state.

Sign analysis of $\mathcal{T}$. Combining the two terms, the sign of $\mathcal{T}$ is determined by the competition between the sampled-token term t_1 and the distribution term t_2:

$$\mathcal{T}(s_t, y_t) < 0 \quad \Longleftrightarrow \quad t_1(s_t, y_t) > t_2(s_t),$$

and

$$\mathcal{T}(s_t, y_t) > 0 \quad \Longleftrightarrow \quad t_1(s_t, y_t) < t_2(s_t).$$

Therefore, when the sampled-token contraction effect t_1 dominates the distributional correction t_2, the token has a negative intrinsic tendency and tends to decrease local entropy under positive reinforcement. Conversely, when t_2 dominates t_1, or when $t_1 < 0$, the token has a positive intrinsic tendency and tends to increase local entropy under positive reinforcement. For negative-advantage updates, the realized direction is reversed by the sign of A, which motivates the effective entropy polarity

$$\mathcal{P}(s_t, y_t, A) = A\mathcal{T}(s_t, y_t).$$

Pointwise asymmetry and realized-scale comparability. The decomposition reveals a subtle distinction between the pointwise form of the intrinsic terms and the empirical scale of the realized components that appear in the entropy change. For a fixed sampled token, the sampled-token term has magnitude

$$|t_1(s_t, y_t)| = p_t |\mathcal{H}_t + \log p_t|,$$

which is pointwise first order in the sampled-token probability p_t, up to the logarithmic factor. The distribution term is

$$t_2(s_t) = \sum_{v \in \mathcal{V}} p_v^2 (\mathcal{H}_t + \log p_v),$$

and therefore aggregates squared-probability contributions over the whole next-token distribution.

This pointwise first-order versus second-order form should not be interpreted as implying that the realized distributional component is negligible in practice. Indeed, under the same policy that generates the sampled token, and for a fixed signed training signal A,

$$\mathbb{E}_{y_t \sim \pi(\cdot|s_t)} [A\, t_1(s_t, y_t)] = A \sum_{v \in \mathcal{V}} p_v\, p_v (\mathcal{H}_t + \log p_v) = A\, t_2(s_t).$$

Thus, when rollout statistics are collected over sampled token instances, the sampling process itself gives the sampled-token component an additional probability weighting, making the empirical scale of At_1 naturally comparable to that of At_2. This explains why the two realized components can appear on the same order in practice, despite the different pointwise dependence of t_1 and t_2 on token probability.

The resulting asymmetry is therefore directional rather than a universal magnitude dominance. High-probability tokens are sampled more frequently and fall on the contractive side of $-t_1$; when this token-specific concentration effect exceeds the positive correction t_2, the intrinsic tendency becomes contractive. In contrast, entropy expansion is associated with lower-probability samples or cases where the distributional correction is strong enough to offset the sampled-token concentration effect.

Algorithm 1 PAPO: Polarity-Aware Policy Optimization

Require: Policy π_θ; EMA coefficient β; bounds $\omega_{\min}, \omega_{\max}$; warmup N_{w}; gate ratio γ_{gate}

$s_0 \leftarrow 0$; $s_{\mathrm{ref}}, h_{\mathrm{ref}} \leftarrow$ `None`; `active` $\leftarrow$ `True`
for step $k = 1, 2, \ldots$ **do**
  Sample responses; compute advantages $A^{(i)}$
  for all response i, position t **do** ▷ Polarity
    $\mu \leftarrow \sum_v \pi(v \mid s_t) \log \pi(v \mid s_t)$; $t_1 \leftarrow \pi(y_t \mid s_t)(\log \pi(y_t \mid s_t) - \mu)$
    $t_2 \leftarrow \sum_v \pi(v \mid s_t)^2 (\log \pi(v \mid s_t) - \mu)$; $\mathcal{P}_t^{(i)} \leftarrow A^{(i)} \cdot (-t_1 + t_2)$
  end for
  $h_k \leftarrow$ mean entropy; $s_k \leftarrow \beta\, s_{k-1} + (1-\beta)(h_k - h_{k-1})$
  if $k = N_{\mathrm{w}}$ **then** $s_{\mathrm{ref}} \leftarrow s_k$; $h_{\mathrm{ref}} \leftarrow h_k$ ▷ Lock references
  end if
  if `active` **and** $\mathrm{EMA}(h_k) < \gamma_{\mathrm{gate}} h_{\mathrm{ref}}$ **then** `active` $\leftarrow$ `False` ▷ Deactivate
  end if
  if $k > N_{\mathrm{w}}$ **and** `active` **then** ▷ Adaptive weights
    $p_k \leftarrow \mathrm{clip}_{[0,1]}\left(\frac{s_k - s_{\mathrm{ref}}}{-s_{\mathrm{ref}}}\right)$
    $\omega_{\mathrm{neg}} \leftarrow \omega_{\min} + (\omega_{\max} - \omega_{\min})\, p_k^2$; $\omega_{\mathrm{pos}} \leftarrow 1/\omega_{\mathrm{neg}}$
  else $\omega_{\mathrm{pos}}, \omega_{\mathrm{neg}} \leftarrow 1$
  end if
  for all response i, position t **do** ▷ Reweight (Eq. (9))
    $\tilde{A}_t^{(i)} \leftarrow A^{(i)} \cdot \omega_{\mathrm{pos}}$ if $\mathcal{P}_t^{(i)} > 0$, $A^{(i)} \cdot \omega_{\mathrm{neg}}$ if $\mathcal{P}_t^{(i)} < 0$, $A^{(i)}$ otherwise
  end for
  Update θ via clipped policy gradient with $\tilde{A}_t^{(i)}$
end for

B Experimental Details

B.1 Pseudo Code of PAPO

To intuitively present the pipeline of PAPO, we summarize the pseudo codes of PAPO in Algorithm B.1.

B.2 Evaluation Details

We evaluate across benchmarks spanning mathematical reasoning, code reasoning, instruction following, and general multi-task abilities.

Mathematical reasoning. We evaluate on six benchmarks: OlympiadBench [34], AMC23 [35], AIME24, AIME25 [37], MATH500 [36], and Minerva. For all math benchmarks, answers are extracted from model outputs via rule-based parsing and compared against ground-truth solutions using exact match. In the main table (Table 1), we report Mean@32 accuracy: each problem is solved 32 times independently and the average accuracy across all attempts is reported.

Out-of-domain benchmarks. To assess generalization beyond the math training distribution, we evaluate on three out-of-domain benchmarks: CRUX [40] for code reasoning, IFEval [41] for instruction following, and MMLU-Pro [42] for general knowledge. CRUX is evaluated via the ZeroEval library; IFEval and MMLU-Pro are evaluated using lm-evaluation-harness. For all OOD benchmarks, we report Mean@32 scores.

Agentic benchmarks. We evaluate on BFCL-v3 [44] and ACEBench [45], reporting accuracy for all results. For BFCL, we follow the official benchmark setup and evaluate on the Live and Non-live subsets while excluding the Multi-turn split. Non-live consists of synthetically generated or curated test cases, whereas Live contains real user-contributed queries. Each subset includes four categories: Simple, Multiple, Parallel, and Parallel Multiple. For ACEBench, we exclude multi-turn cases and focus on two subsets under the Normal category: Atom and Single-turn.

B.3 Baselines

DAPO. DAPO [32] serves as the vanilla RLVR baseline. It uses the clip-higher mechanism ($\epsilon_{\text{high}} > \epsilon_{\text{low}}$) to widen the trust region for advantaged tokens, and employs dynamic sampling: for each prompt, more candidate responses are generated than the training batch requires, and prompts whose responses are all correct or all incorrect are filtered out so that only prompts with mixed outcomes contribute to the gradient. DAPO applies no explicit entropy control beyond these mechanisms.

Entropy Regularization. This baseline augments the GRPO objective with an entropy bonus term $\alpha\mathcal{H}(\pi_\theta)$ added to the loss, encouraging the policy to maintain higher entropy throughout training. The regularization is applied uniformly across all tokens regardless of their role in entropy dynamics.

80/20. Following [25], this method retains only the top 20% of tokens ranked by per-token entropy and computes the policy gradient exclusively on this high-entropy subset. The motivation is that reasoning-critical tokens concentrate at high-entropy positions; by discarding low-entropy tokens, the gradient signal focuses on the subset most relevant to exploratory reasoning.

STEER. STEER [57] estimates the entropy change induced by each token update from its advantage, probability, and conditional entropy, and uses this estimate to reweight the policy gradient. Tokens with larger estimated entropy changes in absolute value receive smaller weights, moderating abrupt shifts in policy entropy during training.

Clip-Cov. Clip-Cov [22] limits entropy contraction by suppressing updates to tokens with high covariance between log-probability and advantage. It randomly selects a small fraction of tokens within a specified covariance interval and masks their policy-gradient contributions, preserving updates on the remaining tokens.

KL-Cov. KL-Cov [22] uses the same covariance score as Clip-Cov to identify tokens associated with entropy contraction and applies an additional KL penalty relative to the rollout policy to those with the highest scores. This selectively constrains their policy updates to mitigate entropy collapse, while leaving the remaining tokens under the policy-gradient objective.

B.4 Training Configuration

All math experiments are built on the VeRL framework and trained on 16 $\times$ H20 GPUs. We train for 6 epochs with a learning rate of 1×10^{-6} and a sampling temperature of 1.0. The generation batch size is 256; the training batch size is 128 with gradient accumulation over 64 steps. For each prompt, 8 candidate responses are sampled with a maximum completion length of 8,192 tokens. clipping range $[\epsilon_{\text{low}}, \epsilon_{\text{high}}] = [0.2, 0.28]$ follows DAPO. No KL penalty is used except for the token-selective penalty in KL-Cov.

For PAPO, the polarity weight bounds are set to $[\omega_{\min}, \omega_{\max}] = [0.97, 1.02]$ for the 14B backbone and $[0.98, 1.03]$ for the 7B backbone. The entropy-slope EMA uses $\beta_{\text{warm}} = 0.95$ during the warmup phase and $\beta_{\text{run}} = 0.9$ thereafter. The warmup phase lasts $N_{\text{w}} = 20$ steps, during which training runs with neutral weights ($\omega_{\text{pos}} = \omega_{\text{neg}} = 1$) to estimate the reference collapse rate s_{ref}. The gate threshold is set to $\gamma_{\text{gate}} = 0.3$: once the smoothed entropy falls below $0.3 \cdot h_{\text{ref}}$, the adaptive reweighting deactivates permanently. All baselines (DAPO, Ent-Reg, 80/20) share the same training budget, batch sizes, and evaluation protocol to ensure a fair comparison. For STEER, we use a linear mapping with token weights bounded by $[0.8, 1.2]$.For Clip-Cov, we set the token-selection ratio to 2×10^{-4} (0.02%), with covariance lower and upper bounds of 1 and 5, respectively.For KL-Cov, we set the selected-token ratio to 2×10^{-4} (0.02%) and the KL coefficient to 1.

Agentic experiments are based on Tool-N1 dataset for 7 epochs with a learning rate of 1×10^{-6} and a sampling temperature of 0.7. The generation batch size is 256; the training batch size is 128 with gradient accumulation over 64 steps. Group size is 5 and with a maximum completion length of 8,192 tokens. For PAPO, the polarity weight bounds are set to $[\omega_{\min}, \omega_{\max}] = [0.99, 1.05]$. The entropy-slope EMA uses $\beta_{\text{warm}} = 0.95$ during the warmup phase and $\beta_{\text{run}} = 0.9$ thereafter. The warmup phase lasts $N_{\text{w}} = 25$ steps, during which training runs with neutral weights ($\omega_{\text{pos}} = \omega_{\text{neg}} = 1$) to estimate the reference collapse rate s_{ref}.

B.5 Details of Single-Polarity Ablation

To verify the functional meaning of effective polarity, we conduct the single-polarity ablation shown in Figure 2. For each sampled token update, we compute the effective polarity $\mathcal{P}(s_t, y_t, A) = A\mathcal{T}(s_t, y_t)$. The positive-only variant keeps token-level policy-gradient terms with $\mathcal{P} > 0$ and masks out terms with $\mathcal{P} < 0$. The negative-only variant keeps terms with $\mathcal{P} < 0$ and masks out terms with $\mathcal{P} > 0$. This diagnostic ablation is implemented on vanilla GRPO rather than the DAPO-based recipe used in the main experiments. This choice isolates the effect of polarity sign and avoids confounding from DAPO-specific sampling or clipping mechanisms and from PAPO's adaptive reweighting. Apart from this objective choice and the polarity mask, rewards, advantage normalization, and other hyperparameters follow the main math experimental setup.

B.6 Details of Polarity-Aware Strength Ablation

This ablation (Figure 7) examines how the polarity weight bounds $[\omega_{\min}, \omega_{\max}]$ affect the exploration–exploitation trade-off during training. All three PAPO configurations use Qwen2.5-7B-Base on the same math training setup described in Appendix B.4; the only difference is the polarity weight range:

- **Explore-biased:** $[\omega_{\min}, \omega_{\max}] = [0.96, 1.00]$. The contraction weight stays at or below one, while its reciprocal favors the expansion branch.
- **Exploit-biased:** $[\omega_{\min}, \omega_{\max}] = [0.99, 1.06]$. The range allows stronger contraction weighting as the recovery signal rises.
- **Moderate:** $[\omega_{\min}, \omega_{\max}] = [0.98, 1.03]$. This is the same setting used in the 7B main experiment.

The DAPO baseline shares the same training budget and hyperparameters but applies no polarity reweighting ($\omega_{\text{pos}} = \omega_{\text{neg}} = 1$ throughout).

B.7 Prompt Templates

Math reasoning. All math experiments use the Qwen chat template with the following prompt format:

```
<|im_start|>system
You are a helpful assistant.<|im_end|>
<|im_start|>user
{input}
Please reason step by step, and put your final answer within \boxed{}.<|im_end|>
<|im_start|>assistant
```

where `{input}` is replaced by the problem statement from DAPO-Math-17K.

Agentic Reasoning All agentic experiments use the Qwen chat template with the following prompt format:

```
<|im_start|>system
You are an expert in composing functions.  You are given a question and a set of
possible functions.  Based on the question, you will need to make one or more
function/tool calls to achieve the purpose.  If none of the function can be
used, point it out.  If the given question lacks the parameters required by the
function, also point it out.  You should only return the function call in tools
call sections.  Here is a list of functions in JSON format that you can invoke:
# Tool

You are provided with function signatures within <tools></tools> XML tags:

<tools>
{tools}
</tools>

In each action step, you MUST:
1.  Think about the reasoning process in the mind and enclosed your reasoning
within <think></think> XML tags.
2.  Then, provide a json object with function names and arguments
within <tool_call></tool_call> XML tags.  i.e., <tool_call>[{"name":
<function-name>, "arguments":  <args-json-object>}, {"name":
<function-name2>, "arguments":  <args-json-object2>}, ...]</tool_call>
3.  Make sure both the reasoning and the tool call steps are included together in
one single reply.
A complete reply example is:  <think> To address the query, I need to send the
email to Bob and then buy the banana through walmart.  </think> <tool_call>
[{"name":  "email", "arguments":  {"receiver":  "Bob", "content":  "I will
bug banana through walmart"}}, {"name":  "walmart", "arguments":  {"input":
"banana"}}]</tool_call>.  Please make sure the type of the arguments is
correct.<|im_end|>
<|im_start|>user
{input}
<|im_end|>
<|im_start|>assistant
```

where `{tools}` and `{input}` are replaced by the tool list and the problem statement from the ToolN1 dataset, respectively.

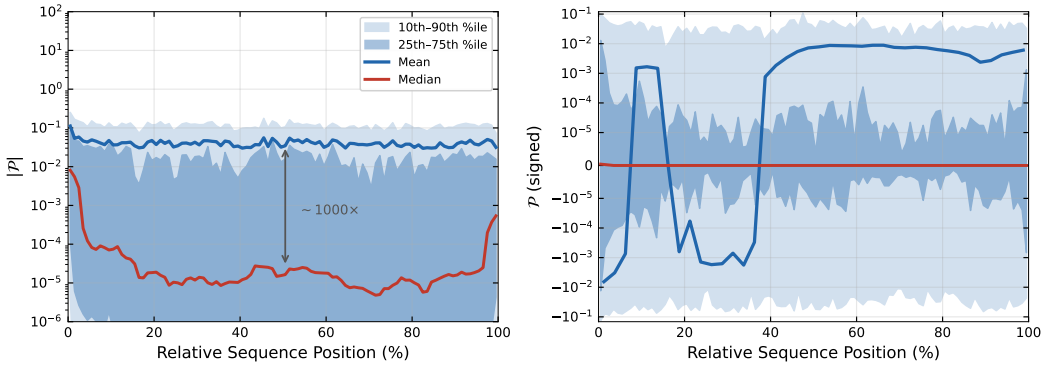
C Additional Results

C.1 Positional Distribution of Entropy Polarity

We explore how $\mathcal{P}$ varies with token position within each rollout in Figure 8.

Magnitude distribution. The mean $|\mathcal{P}|$ remains nearly constant across all positions (≈ 0.04), while the median sits approximately three orders of magnitude lower ($\sim 10^{-5}$), confirming the heavy-tail sparsity reported in Section 3.3. This gap implies that more than half of all tokens at every position contribute negligibly to entropy dynamics; the entropy-shaping effect is concentrated in a sparse tail of high-$|\mathcal{P}|$ tokens. The median exhibits a mild boundary effect—elevated at the start and end of the rollout ($\sim 10^{-2}$ to 10^{-4}) compared to $\sim 10^{-5}$ in the interior—suggesting that tokens near sequence boundaries carry moderately higher polarity on average. The flat mean indicates comparable average polarity magnitudes across positions. This justifies a position-agnostic reweighting scheme: methods that filter tokens by position would discard informative gradient signal at every location, whereas polarity-based selection captures the sparse, high-impact tokens regardless of where they appear.

Signed polarity reveals a positional trend. The signed mean transitions from net negative (contraction-dominated) in the first $\sim 15\%$ of the sequence to net positive (expansion-dominated) thereafter, while the median remains at zero throughout. This pattern admits a natural interpretation: early tokens in reasoning chains typically establish the solution direction (e.g., selecting a proof strategy or computation framework) and tend to be high-confidence, so reinforcement con-



(a) Magnitude (b) Signed polarity

Figure 8: Positional distribution of entropy polarity $\mathcal{P}$. Shaded bands show the 10th–90th and 25th–75th percentile ranges; solid lines show mean (blue) and median (red). (a) Magnitude $|\mathcal{P}|$: the mean remains flat while the median sits $\sim$1000$\times$ lower, reflecting heavy-tail sparsity at every position. (b) Signed $\mathcal{P}$: the mean transitions from net contraction in early positions to net expansion in later positions, while the median remains at zero.

Table 2: Continuation statistics following high-polarity events. Relative differences use the negative-polarity value as the denominator.

Behavioral aspect	**Metric**	**Positive**	**Negative**	**Difference (%)**
Long-tail coverage	Rare next-token continuations	290	227	+27.8
Usable diversity	Effective two-token branches	560.8	457.7	+22.5
Local repetition ↓	Two-token repetition probability	0.00047	0.00087	−46.0

centrates probability further and produces a contracting effect. Later tokens handle more uncertain, problem-specific reasoning steps where the model distributes probability more broadly, yielding a net expansion tendency. The near-zero median indicates a small central signed polarity at every position; the nonzero mean reflects contributions from the tails, consistent with the heavy-tail structure observed in panel (a).

C.2 Branch-Conditioned Continuation Statistics

To characterize the behavioral differences between the two polarity branches, we examine local continuations following high-polarity events. Rare next-token continuations measure access to infrequently visited alternatives. Effective two-token branches quantify the number of short continuation paths that remain actively used, while two-token repetition probability captures the degree of local concentration on previously repeated continuations.

Table 2 shows that positive-polarity events exhibit broader coverage, more active continuation paths, and lower repetition, whereas negative-polarity events correspond to a narrower and more concentrated continuation distribution. Together with the single-polarity ablation in Figure 2 and the contextual and lexical analyses in Figures 3 and 4, these results support the exploration-oriented tendency of the positive branch and the exploitation-oriented tendency of the negative branch.

C.3 Complete Training Dynamics

Qwen2.5-7B-Base (Math). Figure 9 shows the full training curves for all four methods on the 7B backbone. The overall trends mirror the 14B results in Figure 5: PAPO achieves the highest final reward with a stable entropy trajectory. Unlike at 14B, PAPO's entropy settles slightly below DAPO, suggesting that the smaller model naturally favors a more exploitative regime; crucially, entropy remains stable without collapse, indicating sufficient exploratory capacity for continued learning. The adaptive weights ω exhibit the same transient rebalancing pattern observed at 14B scale.

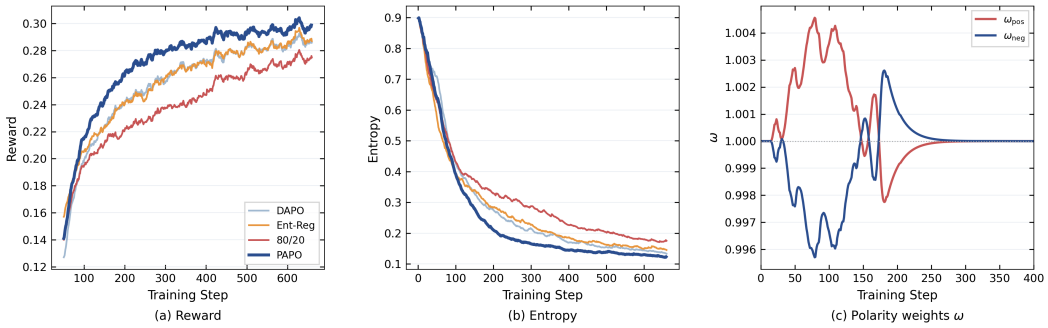


Figure 9: **Training dynamics on Qwen2.5-7B-Base (Math).** (a) Reward, (b) entropy, and (c) polarity weights ω for all methods trained on DAPO-Math-17k. Trends are consistent with the 14B results in Figure 5.

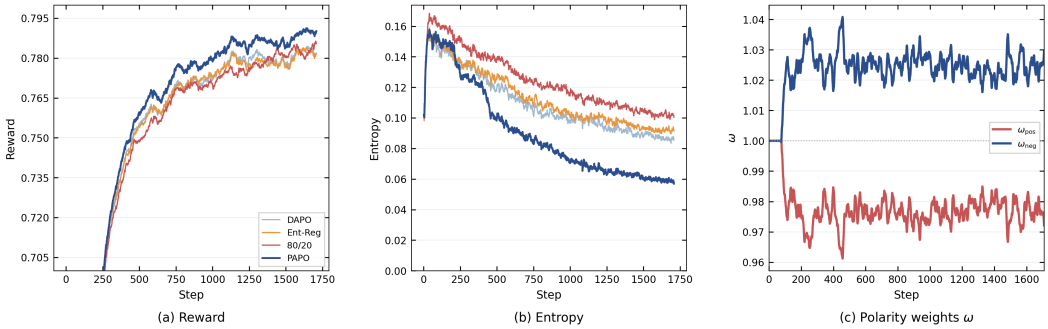


Figure 10: **Training dynamics on tool-call reasoning (Qwen2.5-14B-Instruct).** (a) Reward, (b) entropy, and (c) polarity weights ω. The adaptive weights consistently favor the contracting branch, reflecting the more constrained action space of structured tool-call generation.

Qwen2.5-14B-Instruct (Agentic). Figure 10 presents the training dynamics for the agentic setting. Unlike math reasoning, the adaptive weights consistently favor the contracting branch ($\omega_{\text{neg}} > \omega_{\text{pos}}$), reflecting the more constrained action space of structured tool-call generation where exploitation is naturally preferred.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We state our main claims in the abstract and introduction.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

Question: Does the paper discuss the limitations of the work performed by the authors?

Answer: [Yes]

Justification: Please see the limitation in Section 7.

Guidelines:

- The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
- The authors are encouraged to create a separate "Limitations" section in their paper.
- The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
- The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
- The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
- The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
- If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
- While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: Please see the theoretical proofs in Section 3 and Appendix A.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

Justification: The experimental details and results are included in Section 5, Appendix B and C.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We have released the code on github.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.

- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes],

   Justification: The experimental details and results are included in Section 5, Appendix B and C.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: This paper correctly defined or other appropriate information about the statistical significance of the experiments.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.
   - It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
   - For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
   - If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: : All the experiments are conducted on 16 X H20 GPUs.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: This work comply with the code of Ethics.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: There is no societal impact of this work.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

Answer: [N/A]

Justification: NA

Guidelines:

- The answer [N/A] means that the paper poses no such risks.
- Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

Answer: [Yes]

Justification: We ensure the appropriate use and citation of resources such as models and datasets in the paper.

Guidelines:

- The answer [N/A] means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: NA

Guidelines:

- The answer [N/A] means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing nor research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve above risks.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are used to assist in writing and formatting the implementation code. They did not participate in the design of core methodology and have no impact on the scientific rigor or originality of this work.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.