STAPO: Stabilizing Reinforcement Learning for LLMs by Silencing Rare Spurious Tokens

Shiqi Liu[1,2,†], **Zeyu He**[1,2,†], **Guojian Zhan**[1,2,†], **Letian Tao**[1,2], **Zhilong Zheng**[1,2]
Jiang Wu[1], **Yinuo Wang**[1,2], **Yang Guan**[1], **Kehua Sheng**[2], **Bo Zhang**[2]
Keqiang Li[1], **Jingliang Duan**[1,2,✉], **Shengbo Eben Li**[1,✉]

[1] School of Vehicle and Mobility & College of AI, Tsinghua University
[2] Didi Voyager Labs, DiDi Autonomous Driving

[†] Equal contribution. [✉] Corresponding authors.

Abstract

Reinforcement Learning (RL) has significantly improved large language model reasoning, but existing RL fine-tuning methods rely heavily on heuristic techniques such as entropy regularization and reweighting to maintain stability. In practice, they often suffer from late-stage performance collapse, leading to degraded reasoning quality and unstable training. We identify a key factor behind this instability: a small fraction of tokens, termed spurious tokens (around 0.01%), which contribute little to the reasoning outcome but receive disproportionately amplified gradient updates due to inheriting the full sequence-level reward. We present a unified framework for evaluating token-level optimization impacts across spurious risk, gradient norms, and entropy changes. Building on the analysis of token characteristics that severely disrupt optimization, we propose the Silencing Spurious Tokens (S2T) mechanism to efficiently suppress their gradient perturbations. Incorporating this mechanism into a group-based objective, we propose Spurious-Token-Aware Policy Optimization (STAPO), which promotes stable and effective large-scale model refinement. Across six mathematical benchmarks and three model scales, STAPO consistently demonstrates superior entropy stability and achieves average accuracy improvements over baselines including GRPO, 20-Entropy, and JustRL under two widely-adopted evaluation settings.

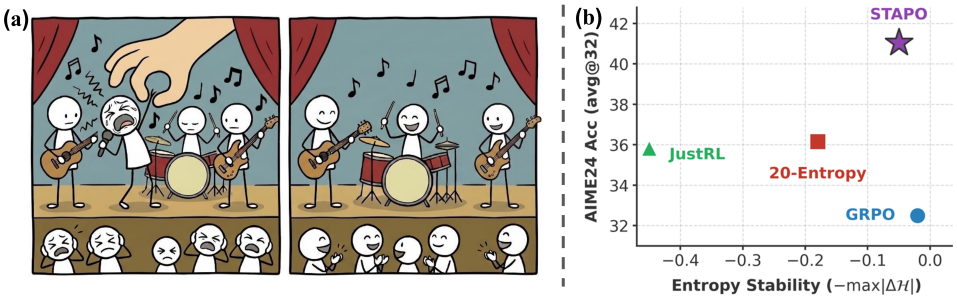


Figure 1: **Core Idea.** (a) In RLVR training, we argue that spurious tokens, which are rare and uninformative tokens within otherwise correct responses, can harm training stability, analogous to a dissonant vocalist disrupting the harmony of a performance. (b) By masking this negligible fraction (near 0.01%) of spurious tokens during the RL process of Qwen3-8B-Base, STAPO approaches the Pareto frontier of AIME24 Acc and entropy stability, compared to GRPO, 20-Entropy, and JustRL.

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

1 Introduction

Recent large language models (LLMs), including OpenAI-o1 [1], DeepSeek-R1 [2], and Qwen3 [3], have demonstrated remarkable capabilities in complex reasoning domains such as mathematics and programming. Central to this success is Reinforcement Learning (RL) [4], which optimizes directly for outcome-level correctness and has been empirically linked to the emergence of advanced, long-horizon reasoning behaviors [5].

Existing approaches have largely focused on enhancing exploration to improve the learning capacity of LLMs under RL training. For example, DAPO [6] increases the clipping threshold, while SAPO [7] adopts a soft-clipping strategy. Other works instead emphasize the exploration of forking tokens, as introduced in 20-Entropy [8]. However, in practice, these approaches often drive entropy toward explosive growth, ultimately leading to catastrophic degradation, where models collapse from coherent reasoning into shallow, repetitive, or even nonsensical outputs [9].

To mitigate such entropy-induced instability, prior efforts have introduced various stabilization techniques, including entropy regularization [10, 11], sample augmentation [12, 13], and advantage reweighting [14]. These methods regulate entropy at a global level but remain coarse-grained and fail to capture the heterogeneous roles of individual tokens during optimization. As a result, they may either over-suppress useful exploration or induce oscillatory entropy dynamics, leading to suboptimal performance.

In this work, we aim to strike a balance between effective exploration and strong reasoning capabilities. We first introduce the concept of *spurious tokens*, defined as a sparse subset of tokens within otherwise accurate responses that, rather than contributing to the underlying reasoning process, introduce logical misdirection or harmful noise, as illustrated in Figure 1. From a token-level perspective, we systematically analyze the detrimental impact of these spurious tokens on optimization dynamics, demonstrating that they introduce misleading and destabilizing update signals. Building on this insight, we propose an efficient method for identifying spurious tokens and formulate Spurious-Token-Aware Policy Optimization (STAPO). By masking a negligible fraction (approximately 0.01%) of such tokens during training, STAPO significantly stabilizes policy entropy and consistently yields performance improvements. Overall, our main contributions are summarized as follows:

- We identify spurious tokens as a key source of training instability: a minor fraction of tokens (around 0.01%) that contribute little to the actual reasoning yet receive disproportionately large gradient updates by inheriting the full sequence-level reward. To analyze this issue, we present a unified framework that systematically evaluates token-level optimization dynamics in terms of spurious risk, gradient norms, and entropy changes.
- By analyzing token characteristics that severely disrupt optimization, we propose the Silencing Spurious Tokens (S2T) mechanism to efficiently identify such disruptive tokens and suppress their gradient perturbations. We further integrate S2T into a group-based objective and develop STAPO for stable and effective large-scale model refinement.
- Across six mathematical reasoning benchmarks (AIME24, AIME25, AMC23, MATH500, Minerva, and OlympiadBench) and three model scales (Qwen3 1.7B, 8B, and 14B base models [3]), STAPO substantially stabilizes policy entropy and consistently improves reasoning accuracy under diverse evaluation settings.

2 Preliminaries

2.1 Problem Formulation

We consider the problem of fine-tuning large language models (LLMs) via reinforcement learning (RL) for reasoning tasks. Let $\mathcal{D}$ denote a distribution over input prompts. Given a prompt $\boldsymbol{x} \sim \mathcal{D}$, an LLM parameterized by θ acts as a stochastic policy π_θ that autoregressively generates an output sequence $\boldsymbol{y} = (y_1, \ldots, y_T)$. Specifically, at each step t, a token $y_t \in \mathcal{V}$ is sampled according to $\pi_\theta(y_t \mid \boldsymbol{x}, \boldsymbol{y}_{<t})$.

Supervision is provided via a sparse, sequence-level verifiable reward $R(\boldsymbol{x}, \boldsymbol{y}) \in \{-1, 1\}$, which evaluates the correctness of the generated sequence $\boldsymbol{y}$ using an external verifier (e.g., a code compiler

or a mathematical rule checker). The optimization objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\, \boldsymbol{y}\sim\pi_\theta(\cdot|\boldsymbol{x})}\left[R(\boldsymbol{x},\boldsymbol{y})\right].$$

2.2 Group Relative Policy Optimization (GRPO)

We briefly review Group Relative Policy Optimization (GRPO) [15], which estimates advantages without relying on an explicit value function. For each prompt $\boldsymbol{x}$, GRPO samples a group of G output sequences $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_G\}$ from a reference behavior policy $\pi_{\theta_{\text{old}}}$. The optimization objective is defined as the average clipped surrogate loss over the sampled group:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\{\boldsymbol{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|\boldsymbol{y}_i|}\sum_{t=1}^{|\boldsymbol{y}_i|}\min\Big(\rho_{i,t}(\theta)\hat{A}_i,\right.$$
$$\left.\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon\big)\hat{A}_i\Big)\right] - \beta\mathbb{D}_{\text{KL}}(\pi_\theta \mid \pi_{\text{ref}}),$$

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})}, \quad \hat{A}_i = \frac{R(\boldsymbol{x},\boldsymbol{y}_i) - \text{mean}(\{R(\boldsymbol{x},\boldsymbol{y}_j)\}_{j=1}^{G})}{\text{std}(\{R(\boldsymbol{x},\boldsymbol{y}_j)\}_{j=1}^{G})}, \tag{1}$$

where $\rho_{i,t}(\theta)$ is the importance sampling ratio and $\hat{A}_i$ is the sequence-level advantage signal, derived by standardizing the reward across the G samples within the group. The π_{ref} serves as the reference policy, and β is the scaling coefficient.

2.3 Clip-Higher and Token Normalization

Building upon the GRPO framework, several large-scale RL algorithms have recently emerged. Notably, DAPO [6] removes the KL penalty and introduces a set of training enhancements, with token-level normalization and an asymmetric clip-higher mechanism proving particularly effective for stabilizing optimization. Subsequent work, such as JustRL [16], has adopted these two components and demonstrated strong empirical performance. The corresponding objective is written as:

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\{\boldsymbol{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}\left[\frac{1}{\sum_{i=1}^{G}|\boldsymbol{y}_i|}\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{y}_i|}\min\Big(\rho_{i,t}(\theta)\hat{A}_i,\right.$$
$$\left.\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_i\Big)\right], \tag{2}$$

where ϵ_{low} and ϵ_{high} denote the asymmetric clipping parameters integral to the clip-higher mechanism. Given its simplicity and robust empirical success, we adopt this augmented configuration as our baseline objective.

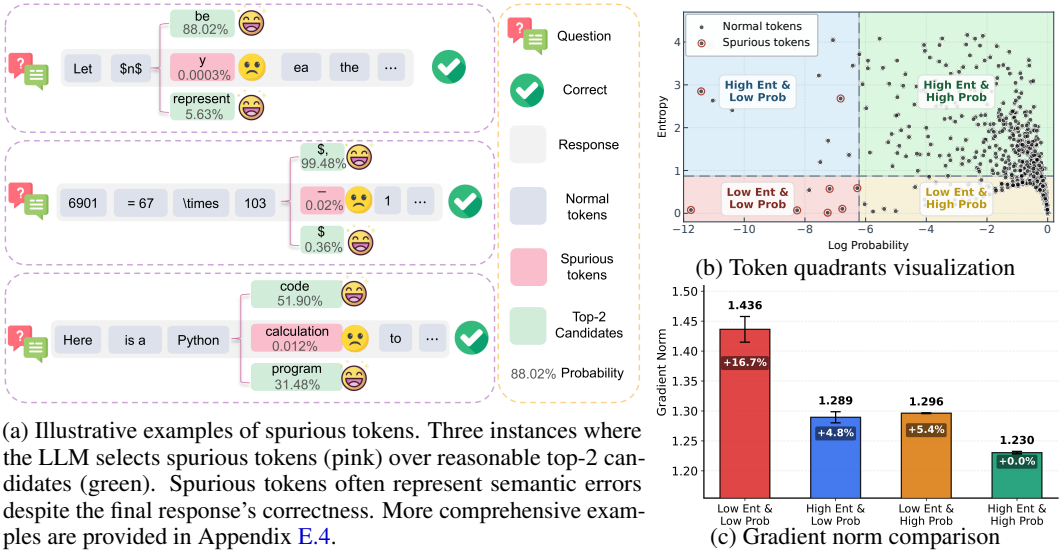
3 Methodology

3.1 The Hidden Threat: Spurious Tokens

As discussed in Section 2.1, rewards in Reinforcement Learning with Verifiable Rewards (RLVR) are typically derived solely from the final outcome. Consequently, all tokens $y_{i,t}$ within a given trajectory share an identical sequence-level advantage, $\hat{A}_i$. This coarse-grained credit assignment can inadvertently reinforce extraneous tokens. We formalize this phenomenon as follows:

Definition 3.1 (Spurious Tokens)**.** Spurious tokens are intermediate tokens $y_{i,t}$ that contribute negligibly to the correct reasoning outcome, yet receive disproportionately large positive updates.

To empirically examine this effect, we train a Qwen3-1.7B base model under the JustRL setting on DAPO-MATH-17K [6], and record all generated tokens along with their associated statistics during training. Figure 2a presents concrete instances of this phenomenon, showing that spurious tokens can induce misleading update signals that steer the policy toward detrimental directions and destabilize training.



(a) Illustrative examples of spurious tokens. Three instances where the LLM selects spurious tokens (pink) over reasonable top-2 candidates (green). Spurious tokens often represent semantic errors despite the final response's correctness. More comprehensive examples are provided in Appendix E.4.

(b) Token quadrants visualization

(c) Gradient norm comparison

Figure 2: **Comprehensive Analysis of Spurious Tokens.**

Motivated by this observation, we propose Spurious-Token-Aware Policy Optimization (STAPO), which introduces a binary mask, $\mathbb{I}_{i,t}^{\text{Spurious}}$, to discard gradient contributions from spurious tokens:

$$\mathbb{I}_{i,t}^{\text{Spurious}} = \begin{cases} 0, & \text{if } y_{i,t} \in \mathcal{S}_i, \\ 1, & \text{otherwise}, \end{cases}$$

where $\mathcal{S}_i$ denotes the set of identified spurious tokens within the i-th trajectory. Incorporating this, the STAPO objective is formulated as:

$$\mathcal{J}_{\text{STAPO}}(\theta) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D},\{\boldsymbol{y}_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|\boldsymbol{x})}\left[\frac{1}{\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{y}_i|}\mathbb{I}_{i,t}^{\text{Spurious}}}\sum_{i=1}^{G}\sum_{t=1}^{|\boldsymbol{y}_i|}\mathbb{I}_{i,t}^{\text{Spurious}}\cdot\min\Big(\rho_{i,t}(\theta)\hat{A}_i,\right.$$
$$\left.\text{clip}\big(\rho_{i,t}(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\big)\hat{A}_i\Big)\right]. \tag{3}$$

The terms $\rho_{i,t}$ and $\hat{A}_i$ follow standard definitions provided in (1). Comparing the STAPO objective in Eq. (3) with the standard DAPO objective in Eq. (2), two primary distinctions emerge. First, STAPO leverages this binary mask to selectively zero out the loss calculations for spurious tokens. Second, the normalization term in Eq. (3) is dynamically adjusted to average the loss exclusively over the remaining valid tokens.

3.2 Token-Level Optimization Analysis

Fundamentally, the generation dynamics of LLMs are shaped by pre-training, which assigns lower prior probabilities to ill-formed or semantically anomalous token sequences. As a result, spurious tokens, often appearing as incoherent or illogical reasoning steps, tend to exhibit *low sampling probabilities*. Motivated by this, we extend JustRL with full masking of low-probability tokens within correct answers (JustRL-FullMask) and compare its entropy dynamics with the default one. As shown in Figure 3, JustRL-FullMask leads to severe entropy collapse, resulting in insufficient exploration and degraded performance, while the default JustRL update maintains persistently high policy entropy in later stages, hindering convergence. This trade-off exposes a paradox in entropy control, explained by the following update dynamics:

Lemma 3.2 (Entropy Update Mechanism [17])**.** *Consider the language policy $\pi_\theta(\cdot)$ updated via a natural policy gradient step with learning rate η. Let y' denote a generic token variable distributed according to the current policy $\pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})$ at step t. The change in entropy $\Delta\mathcal{H}(y_{i,t})$ associated*

with a specific, already-sampled token $y_{i,t}$ between two consecutive policy iterates satisfies:

$$\Delta\mathcal{H}(y_{i,t}) \approx -\,\eta \cdot \pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) \cdot \Big[\log \pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) - \mathbb{E}_{y' \sim \pi_\theta}\big[\log \pi_\theta(y' \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})\big]\Big] \cdot \Big[\hat{A}_i(y_{i,t}) - \mathbb{E}_{y' \sim \pi_\theta}\big[\hat{A}_i(y')\big]\Big].$$

As formalized in Lemma 3.2, low-probability tokens within correct responses induce positive entropy updates, thereby sustaining policy exploration. This trade-off implies that the key challenge is to selectively remove truly detrimental spurious tokens among low-probability tokens while preserving stable exploration.

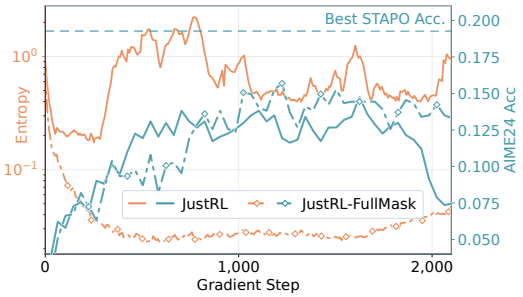


Figure 3: **Training curves of entropy and AIME24 accuracy.** JustRL-FullMask denotes a variant of JustRL that masks all low-probability tokens within correct responses.

To operationalize this distinction, we partition the representation space into four quadrants based on token statistics (Figure 2b). Spurious tokens predominantly occupy the low-probability regime, demonstrating a clear concentration within low-entropy states. We attribute this to the underlying generation dynamics: low-probability tokens in high-entropy states generally signify legitimate exploration and are therefore structurally reasonable. Conversely, low-entropy (high-confidence) states inherently possess highly probable, valid candidates; the realization of a low-probability token in such contexts is merely an artifact of random sampling, rendering it anomalous and significantly elevating the risk of spurious generation.

Furthermore, we analyze the gradient norm at the token level during training. Specifically, at decoding step t of sequence $\boldsymbol{y}_i$, the LLM produces a logit vector $\boldsymbol{a}_{i,t} \in \mathbb{R}^{|\mathcal{V}|}$ over the vocabulary $\mathcal{V}$, inducing the policy distribution $\pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})$ via the softmax function. We analyze the per-token gradient associated with $y_{i,t}$ as it propagates to intermediate layers.

Theorem 3.3 (Policy Gradient Norm Bounds)**.** *Consider the optimization objective at step t for sample i with target token $y_{i,t}$. The squared ℓ_2-norm of the gradient $\nabla_{\boldsymbol{a}_{i,t}}\mathcal{J}$ w.r.t. the logits $\boldsymbol{a}_{i,t} \in \mathbb{R}^{|\mathcal{V}|}$ is bounded by the entropy $\mathcal{H}(\pi_\theta)$ and the target probability $\pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})$ as follows:*

$$|w_{i,t}|^2\Big(1 - 2\pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) + e^{-\mathcal{H}(\pi_\theta)}\Big) \le \|\nabla_{\boldsymbol{a}_{i,t}}\mathcal{J}\|^2 \le |w_{i,t}|^2\Big(2 - 2\pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) - C_V\mathcal{H}(\pi_\theta)^2\Big),$$

where $C_V = \frac{|\mathcal{V}|-1}{|\mathcal{V}|(\ln|\mathcal{V}|)^2}$ and the scaling weight $w_{i,t}$ is defined as:

$$w_{i,t} = \begin{cases} 0, & \text{if}(\hat{A}_i > 0 \wedge \rho_{i,t} > 1 + \epsilon_{\text{high}}) \vee (\hat{A}_i < 0 \wedge \rho_{i,t} < 1 - \epsilon_{\text{low}}), \\ \frac{\pi_\theta(y_{i,t}|\boldsymbol{x},\boldsymbol{y}_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|\boldsymbol{x},\boldsymbol{y}_{i,<t})}\hat{A}_i, & \text{otherwise.} \end{cases}$$

Proof. See Appendix B for the detailed derivation. □

Theorem 3.3 demonstrates that the tokens characterized by both low probability and low entropy induce a larger gradient norm, a phenomenon empirically corroborated in Figure 2c. Such tokens are likely spurious, thereby introducing erroneous updates that can destabilize training.

To systematically analyze how different token types influence optimization, we focus on tokens within correct answers, which inherently possess a positive advantage ($\hat{A} > 0$), and categorize them along two binary axes: token probability (high/low) and entropy (high/low). As summarized in Table 1, tokens characterized by *low probability and low entropy* consistently exhibit detrimental optimization effects across all evaluation criteria. First, such tokens are highly likely to be spurious, leading to incorrect update directions. Second, they are associated with disproportionately large gradient norms, thereby amplifying their adverse impact during optimization. Finally, they contribute to entropy explosion in standard training, further exacerbating training instability.

Table 1: **Taxonomy of Token-Level Optimization Mechanisms.** Left: Token properties. Right: Influence on optimization. Green indicates beneficial updates, yellow indicates intermediate updates, and red indicates detrimental updates. τ_p and τ_h represent the thresholds for token probability and entropy, respectively.

Token Properties			**Influence on Optimization**		
Advantage	Token Prob.	Entropy	Spurious Risk	Gradient Norm	Entropy Change [17]
$\hat{A} > 0$	$\pi_t \geq \tau_p$	$\mathcal{H} \geq \tau_h$	Low	Low	$\downarrow$
	$\pi_t \geq \tau_p$	$\mathcal{H} < \tau_h$	Low	Medium	$\downarrow$
	$\pi_t < \tau_p$	$\mathcal{H} \geq \tau_h$	Medium	Medium	$\uparrow$
	$\pi_t < \tau_p$	$\mathcal{H} < \tau_h$	**High**	**High**	$\uparrow$

3.3 STAPO with the Silencing Spurious Tokens (S2T) Mechanism

Motivated by Table 1, we propose the **Silencing Spurious Tokens (S2T)** mechanism, which selectively discards the gradient contributions of tokens that fall into the destructive regime:

$$\mathbb{I}_{i,t}^{\text{S2T}} = \begin{cases} 0, & \text{if } \hat{A}_i > 0 \wedge \pi(y_{i,t}) < \tau_p \wedge \mathcal{H}_t < \tau_h^{(q)}, \\ 1, & \text{otherwise,} \end{cases}$$

where τ_p is a fixed probability threshold that defines an absolute notion of rarity, and $\tau_h^{(q)}$ denotes the q-th quantile of entropy, computed dynamically over the low-probability tokens within correct responses in each mini-batch.

The core idea is to first identify low-probability tokens within correct responses, which are characterized as exploratory tokens according to Lemma 3.2. Among these tokens, we further partition them based on entropy using the quantile threshold $\tau_h^{(q)}$. Tokens in the low-entropy region (i.e., the bottom q fraction), which correspond to the most destructive cases identified in Table 1, are suppressed. In contrast, the remaining high-entropy tokens (the top $1-q$ fraction) are preserved to maintain sufficient exploration, as illustrated in Figure 4. This adaptive Silencing mechanism achieves a principled balance between stabilizing optimization and preserving exploration capacity. To provide further insight, we include word cloud comparisons for both types of tokens in Section 4.4.

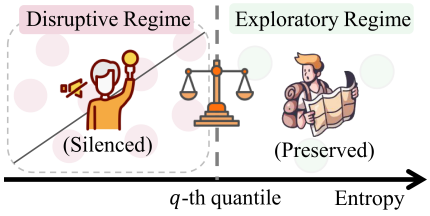


Figure 4: **Illustration of the Entropy Threshold.**

In practice, we approximate the ideal mask $\mathbb{I}^{\text{Spurious}}$ using the S2T mechanism, i.e., $\mathbb{I}_{i,t}^{\text{Spurious}} \approx \mathbb{I}_{i,t}^{\text{S2T}}$ in Equation (3), resulting in a computationally efficient instantiation of STAPO. The complete training procedure is outlined in Appendix C.

4 Experiments

4.1 Settings

Baselines. We compare our approach against several RL algorithms for LLMs, including GRPO [15], 20-Entropy [8], and JustRL [16]. For all baselines, we follow the parameter settings reported in their original papers. Unless otherwise specified, STAPO uses $\tau_p = 0.002$ and $q = 75\%$ across all experiments. For a fair comparison, we do not apply the dynamic sampling technique introduced in [6] to any method. Detailed training settings are provided in Appendix D.

Benchmarks. For mathematical reasoning, we evaluate three Qwen3 base models (1.7B, 8B, and 14B) on six benchmarks: AIME24 [18], AIME25 [19], AMC23 [18], MATH500 [20], Minerva [21], and OlympiadBench [22]. For code reasoning, we train Qwen3-1.7B-Base on Eurus-RL-Code [23] and evaluate on HumanEval+, MBPP+ [24], and LiveCodeBench v6 [25].

Table 2: **Results on Math Reasoning Benchmarks across Three Models.** Each cell reports performance under the training-aligned configuration (temperature ρ_T=1.0, top-p=1.0) and the JustRL [16] evaluation setting (temperature ρ_T=0.7, top-p=0.9, shown in gray). For each configuration, the best result is highlighted in **bold** (training-aligned setting) and in dark gray (JustRL setting), respectively.

Baseline	AIME24	AIME25	AMC23	MATH500	Minerva	Olympiad	Avg
RL from the Qwen3-1.7B Base Model							
GRPO	7.60 (10.42)	3.33 (3.33)	37.81 (39.84)	65.90 (66.40)	**33.73** (34.83)	31.16 (31.42)	29.92 (31.04)
20-Entropy	11.98 (17.40)	6.67 (11.67)	42.66 (47.58)	63.70 (64.90)	21.23 (24.45)	31.34 (32.31)	29.60 (33.05)
JustRL	9.17 (15.73)	6.67 (9.28)	43.20 (48.28)	66.25 (70.10)	30.61 (32.08)	30.71 (32.90)	31.10 (34.73)
STAPO	**18.44** (19.27)	**10.00** (13.33)	**54.22** (53.12)	**73.10** (71.80)	28.40 (26.10)	**36.94** (36.80)	**36.85** (36.74)
RL from the Qwen3-8B Base Model							
GRPO	31.25 (32.50)	24.69 (24.17)	75.23 (74.14)	88.90 (88.85)	55.88 (53.58)	61.50 (58.34)	56.24 (55.26)
20-Entropy	31.25 (36.15)	27.50 (27.39)	79.92 (80.00)	89.85 (89.85)	54.78 (54.23)	62.50 (60.39)	57.63 (58.00)
JustRL	25.21 (35.83)	21.98 (26.04)	73.52 (81.48)	84.90 (87.55)	47.33 (51.03)	51.26 (56.57)	50.70 (56.42)
STAPO	**38.33** (41.04)	**33.33** (29.37)	**83.12** (80.23)	**90.95** (91.40)	**56.80** (55.70)	**63.20** (63.32)	**60.96** (60.18)
RL from the Qwen3-14B Base Model							
GRPO	41.04 (42.40)	35.00 (26.67)	80.08 (82.81)	91.60 (91.00)	57.35 (56.89)	63.87 (63.39)	61.49 (60.53)
20-Entropy	42.08 (48.96)	34.06 (39.48)	84.45 (87.73)	91.65 (92.30)	51.10 (54.78)	61.30 (63.95)	60.77 (64.53)
JustRL	34.48 (50.83)	18.33 (38.33)	76.41 (89.06)	89.10 (93.30)	54.60 (59.93)	61.54 (68.81)	55.74 (66.71)
STAPO	**51.04** (52.81)	**41.67** (41.15)	**89.63** (90.78)	**93.60** (93.70)	**59.74** (59.38)	**70.92** (68.95)	**67.76** (67.80)

Evaluation Protocol. For mathematical benchmarks, we sample N independent responses per problem ($N = 4$ for MATH500, Minerva, and OlympiadBench; $N = 32$ for the others) under two decoding configurations: temperature $\rho_T = 1.0$ with top-p=1.0, and $\rho_T = 0.7$ with top-p=0.9. The maximum generation length is 20,480 tokens, and the latter configuration is used by default unless otherwise specified. We use CompassVerifier-3B [26] to correct errors in rule-based verification. For code benchmarks, following the default decoding configuration (temperature $\rho_T = 0.7$, top-p=0.9), we report avg@4 over four independently generated responses per problem.

4.2 Main Results

4.2.1 Training Behavior Analysis

Entropy Analysis. Entropy curves are a key indicator of learning progress in LLM training. As shown in Figure 5(b), JustRL and 20-Entropy suffer from entropy explosion, while GRPO exhibits entropy collapse. Conversely, STAPO maintains a stable, well-regulated entropy profile after warm-up. Token-level quantile analysis (Figure 5(d)) explains this stability: STAPO concentrates entropy exclusively at high quantiles (e.g., above the 80th percentile), keeping most tokens deterministic. This strategically preserves exploration for critical tokens while enhancing reasoning precision. In contrast, the uniformly high entropy in JustRL and 20-Entropy triggers repetitive generation, whereas GRPO's persistently low entropy stifles exploration and limits learning capacity.

Performance Comparison. Beyond maintaining a stable entropy profile and preserving critical exploratory tokens, STAPO demonstrates superior overall performance. During RL training, as shown in Figure 5(c), STAPO achieves the highest reward, maintaining a clear advantage over all baselines. This suggests that STAPO successfully explores and acquires reasoning capabilities that other methods fail to capture. Furthermore, Figure 5(a) illustrates that STAPO not only attains the highest accuracy on AIME24 but also sustains strong performance without noticeable degradation even after 3000 steps, evidencing remarkable training stability. Ultimately, the stark contrast with JustRL underscores the detrimental impact of spurious tokens in baseline methods, corroborating the validity of our prior analysis in Section 3.2.

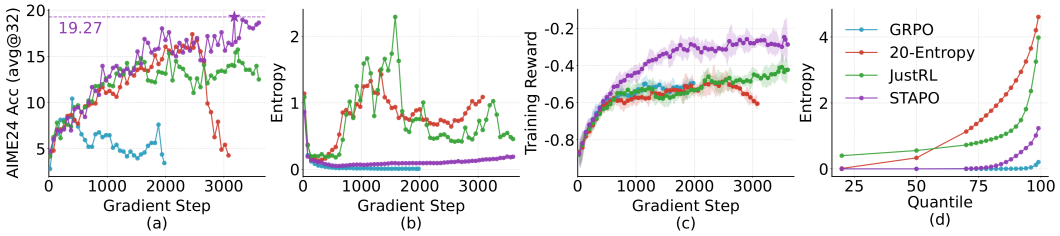


Figure 5: **Training Curves on the Qwen3-1.7B Base Model.** Notably, STAPO achieves superior performance while maintaining stable policy entropy. Training dynamics for larger models (8B and 14B) exhibit consistent trends and are provided in Appendix E.1.

4.2.2 Mathematical Reasoning Results

Table 2 summarizes performance across different model scales. In the training-aligned setting (ρ_T=1.0, top-p=1.0), STAPO demonstrates excellent scalability and enhanced intrinsic reasoning capability, surpassing the strongest baselines with average relative accuracy improvements of 18.49%, 5.78%, and 10.20% at the 1.7B, 8B, and 14B scales, respectively.

When evaluated under the JustRL configuration (ρ_T=0.7, top-p=0.9), STAPO continues to achieve state-of-the-art performance. The performance gap narrows slightly because high-entropy baselines (JustRL and 20-Entropy) benefit more from decoding heuristics that suppress tail generations. Crucially, STAPO consistently attains optimal results across varied evaluation settings, demonstrating strong robustness and overall performance. This confirms that its inherently stable distribution is less dependent on such decoding heuristics.

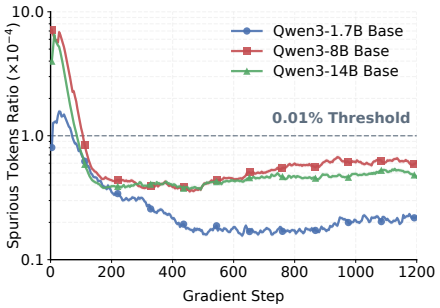


Figure 6: **Spurious Token Ratio Curves.** After warm-up, the ratio of spurious tokens stays below 0.01% across all models.

Notably, as shown in Figure 6, these significant gains are achieved by masking a negligible fraction of tokens (mostly $< 0.01\%$) across the 1.7B, 8B, and 14B models. This indicates that RL instability is caused by sparse spurious tokens, which disproportionately affect gradient updates and are precisely identified and effectively suppressed by STAPO. To better illustrate the impact of spurious tokens, we also provide additional examples, along with a simple case classification, in Appendix E.4.

4.2.3 Code Reasoning Results

As shown in Table 3, STAPO consistently achieves the best performance across all three coding benchmarks, outperforming the strongest baseline by 5.79% in relative average accuracy. Notably, these improvements are achieved without any task-specific hyperparameter tuning, demonstrating the generalizability of STAPO beyond mathematical reasoning. Furthermore, our qualitative analysis reveals that even positively rewarded coding responses can contain locally spurious tokens, such as flawed intermediate code, invalid syntax, and redundant formatting. We provide representative examples with detailed analyses in Appendix E.5, further illustrating that the spurious-token phenomenon is not limited to mathematical reasoning.

Table 3: **Code Reasoning Results with Qwen3-1.7B-Base.**

Method	HumanEval+	MBPP+	LCB v6	Avg.
GRPO	58.23	54.76	8.29	40.43
20-Entropy	59.15	55.95	12.43	42.51
JustRL	55.79	55.69	13.86	41.78
STAPO	**63.41**	**56.35**	**15.14**	**44.97**

4.3 Ablation and Sensitivity Analysis

Masking Strategy Ablation. We first ablate different masking strategies applied to positive-advantage, low-probability tokens on the Qwen3-1.7B-Base model. As shown in Figure 7, low-

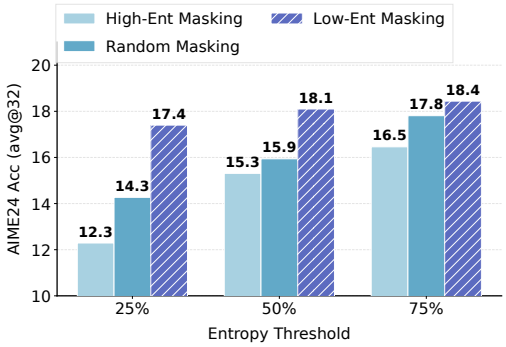


Figure 7: **Masking Strategy Ablation.** Various entropy masking strategies are applied to positive-advantage, low-probability tokens on the 1.7B base model with ρ_{T}=1.0, top-p=1.0.

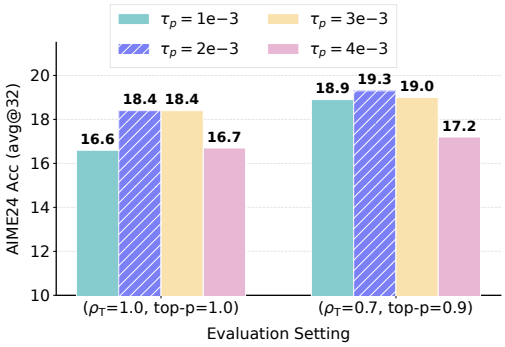


Figure 8: **Sensitivity of AIME24 Accuracy to τ_p.** The AIME24 Acc (avg@32) is evaluated across different probability thresholds in various settings on the 1.7B base model.

entropy masking (STAPO) consistently achieves the best performance, whereas high-entropy masking significantly degrades results, suggesting that masking high-entropy tokens can suppress useful exploratory signals. Random masking yields intermediate performance, gradually approaching that of low-entropy masking as the masking ratio increases. This indicates that while low-entropy masking remains the optimal strategy, random masking can function as a heuristic approximation when the masking budget is sufficiently large.

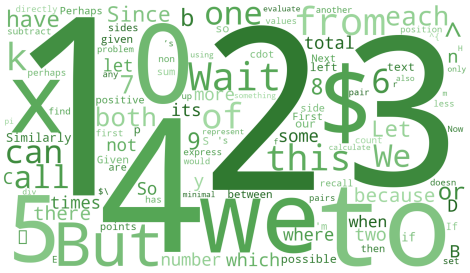
Hyperparameter Sensitivity. We further analyze the model's sensitivity to two key hyperparameters: the entropy quantile q and the probability threshold τ_p. As shown in Figure 7, decreasing q leads to a rapid degradation in performance for both high-entropy and random masking strategies. In contrast, STAPO exhibits only a marginal decline, demonstrating superior robustness to variations in the entropy quantile. Figure 8 illustrates the effect of varying τ_p, which establishes the threshold for identifying low-probability tokens. An insufficiently low τ_p may fail to isolate spurious tokens, leading to increased training entropy and subsequent performance degradation. Conversely, an excessively high τ_p may erroneously penalize legitimate tokens, rendering the policy overly conservative and impairing its reasoning capabilities. Despite these effects, overall performance remains relatively stable, indicating that the model is largely insensitive to τ_p.

4.4 Word Cloud Visualization

During the training of Qwen3-1.7B, we visualized word clouds for low-probability correct answers, categorizing tokens into low-entropy "spurious tokens" and high-entropy "exploratory tokens" as illustrated in Figures 9a and 9b. The most frequent spurious tokens primarily include specific digits (e.g., "1", "2", "3") and mathematical symbols (e.g., "$", "x"). While these tokens may appear in correct responses, they are often associated with formatting or calculation errors. When combined with large gradient updates, they can induce disproportionately unstable optimization dynamics. In contrast, the retained exploratory tokens correspond to the structural vocabulary of mathematical reasoning, including words like "Let", "This", "So", and "Wait". These tokens represent key procedural and logical components that sustain coherent reasoning chains. Retaining these tokens ensures that the model's core logical reasoning capabilities remain intact.

5 Related Work

Entropy Instability in RL. Entropy is a central issue in RL, as it directly affects exploration and training stability [27, 28]. In reasoning-oriented language models trained with RL [29], a persistent challenge is the rapid collapse of policy entropy during the early stages of optimization, which often leads to premature convergence and degraded performance. Prior work mitigates this issue through interventions such as selectively regularizing high-entropy tokens [8], increasing the proportion of entropy-enhancing samples [17], and modifying clipping strategies in policy optimization [6, 30, 7]. However, these methods often introduce the opposite failure mode, where entropy grows excessively or becomes unstable, degrading reasoning coherence and leading to repetitive or unstructured outputs.



(a) Spurious tokens.

(b) Exploratory tokens.

Figure 9: **Word clouds.** The font size of each token is proportional to its occurrence frequency. To facilitate a clearer analysis, common articles and prepositions are filtered out.

Although some studies analyze entropy dynamics during RL training [10, 31] and others explicitly enforce entropy stability as an optimization objective [11], existing approaches largely treat entropy as a surface-level training signal rather than addressing the underlying sources of instability.

Gradient Domination by Low-Probability Tokens. While various policy gradients aim to improve RL stability [32, 33], instability often stems from the disproportionate gradient influence of low-probability tokens. Rare tokens can generate excessively large updates [14], allowing unstable predictions to dominate fine-tuning optimization. Recent studies [14, 34] address this issue through probability-aware modulation. Token-Regulated GRPO (TR-GRPO) [35] employs probability shaping to suppress sharp update directions arising from low-confidence predictions. To avoid discarding informative exploratory signals, Low-Probability Regularization (Lp-Reg) [36] filters out noise while preserving meaningful low-probability tokens. However, relying mainly on scalar probability thresholds, these approaches lack a joint, fine-grained treatment of confidence and probability, and thus fail to distinguish useful exploration from aleatoric noise.

6 Conclusion

In this work, we formally introduce the concept of spurious tokens in RL training, demonstrating that a small fraction of these uninformative tokens can disproportionately skew optimization dynamics. To address this issue, we propose STAPO as a general paradigm designed to mitigate its detrimental effects. Within this paradigm, we implement the S2T mechanism as a specific instantiation to detect and curtail the gradient influence of spurious tokens. Empirical results show that STAPO substantially stabilizes policy entropy and consistently improves reasoning performance.

In future work, we plan to broaden this scope by analyzing tokens from incorrect responses and extending our empirical evaluation to diverse task domains. We hope that these efforts will provide further insights into improving the stability of RL for LLM reasoning.

Funding and Competing Interests

This study is supported by the National Natural Science Foundation of China (NSFC) under Grant No. 92582205 and the Shanghai Automotive Industry Science and Technology Development Foundation. The authors declare no competing interests.

References

[1] OpenAI. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

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

[4] Shengbo Eben Li. *Reinforcement Learning for Sequential Decision and Optimal Control*. Springer, Singapore, 2023.

[5] Yuji Cao, Huan Zhao, Yuheng Cheng, Ting Shu, Yue Chen, Guolong Liu, Gaoqi Liang, Junhua Zhao, Jinyue Yan, and Yun Li. Survey on large language model-enhanced reinforcement learning: Concept, taxonomy, and methods. *IEEE Transactions on Neural Networks and Learning Systems*, 36(6):9737–9757, 2025.

[6] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems*, volume 38, pages 125532–125554, 2025.

[7] Chang Gao, Chujie Zheng, Xiong-Hui Chen, Kai Dang, Shixuan Liu, Bowen Yu, An Yang, Shuai Bai, Jingren Zhou, and Junyang Lin. Soft adaptive policy optimization. *arXiv preprint arXiv:2511.20347*, 2025.

[8] Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for LLM reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pages 127985–128019, 2025.

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

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

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

[12] Marco Simoni, Aleksandar Fontana, Giulio Rossolini, Andrea Saracino, and Paolo Mori. GTPO: Stabilizing group relative policy optimization via gradient and entropy control. *Transactions of the Association for Computational Linguistics*, 14:2258–2280, 2026.

[13] Longtian Qiu, Shan Ning, Jiaxuan Sun, and Xuming He. NoisyGRPO: Incentivizing multimodal CoT reasoning via noise injection and bayesian estimation. In *Advances in Neural Information Processing Systems*, volume 38, pages 137670–137698, 2025.

[14] Zhihe Yang, Xufang Luo, Zilong Wang, Dongqi Han, Zhiyuan He, Dongsheng Li, and Yunjian Xu. Do not let low-probability tokens over-dominate in rl for llms. *arXiv preprint arXiv:2505.12929*, 2025.

[15] 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.

[16] 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.

[17] 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 *International Conference on Learning Representations*, 2026.

[18] Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q. Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. NuminaMath: The largest public dataset in AI4Maths with 860k pairs of competition math problems and solutions, July 2024.

[19] OpenCompass. Aime2025 dataset. https://huggingface.co/datasets/opencompass/AIME2025, 2025. Accessed: 2025-02-08.

[20] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021.

[21] 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. In *Advances in Neural Information Processing Systems*, volume 35, pages 3843–3857, 2022.

[22] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 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. Association for Computational Linguistics, 2024.

[23] Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, et al. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*, 2025.

[24] Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. Is your code generated by chatgpt really correct? rigorous evaluation of large language models for code generation. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 21558–21572. Curran Associates, Inc., 2023.

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

[26] Shudong Liu, Hongwei Liu, Junnan Liu, Linchen Xiao, Songyang Gao, Chengqi Lyu, Yuzhe Gu, Wenwei Zhang, Derek F. Wong, Songyang Zhang, and Kai Chen. CompassVerifier: A unified and robust verifier for LLMs evaluation and outcome reward. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33466–33494. Association for Computational Linguistics, 2025.

[27] Jingliang Duan, Yang Guan, Shengbo Eben Li, Yangang Ren, Qi Sun, and Bo Cheng. Distributional soft actor-critic: Off-policy reinforcement learning for addressing value estimation errors. *IEEE transactions on neural networks and learning systems*, 33(11):6584–6598, 2022.

[28] Yinuo Wang, Likun Wang, Yuxuan Jiang, Wenjun Zou, Tong Liu, Xujie Song, Wenxuan Wang, Liming Xiao, Jiang Wu, Jingliang Duan, et al. Diffusion actor-critic with entropy regulator. *Advances in Neural Information Processing Systems*, 37:54183–54204, 2024.

[29] David Hermann, Chao Wei, Luigi Mazzarella, Jed Judd, Jaren Devey, Klaus Stähle, Gereon Hinz, and Alois Knoll. Automated testing on proving grounds: Integration of intelligent test coordination and robot-guided vehicles. *Automotive Innovation*, 9(3):612–628, 2026.

[30] Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, et al. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*, 2025.

[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. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306 of *Proceedings of Machine Learning Research*, pages 131451–131468. PMLR, 2026.

[32] Guojian Zhan, Xiangteng Zhang, Feihong Zhang, Letian Tao, and Shengbo Eben Li. Bicriteria policy optimization for high-accuracy reinforcement learning. *IEEE Transactions on Neural Networks and Learning Systems*, 37(1):312–326, 2026.

[33] Guojian Zhan, Yuxuan Jiang, Jingliang Duan, Shengbo Eben Li, Bo Cheng, and Keqiang Li. Continuous-time policy optimization. In *2023 American Control Conference*, pages 3382–3388. IEEE, 2023.

[34] Xuan Cai, Xuesong Bai, Zhiyong Cui, Danmu Xie, Daocheng Fu, Haiyang Yu, and Yilong Ren. Text2Scenario: Text-driven scenario generation for autonomous driving test. *Automotive Innovation*, 9(1):102–127, 2026.

[35] Tue Le, Nghi DQ Bui, Linh Ngo Van, and Trung Le. Token-regulated group relative policy optimization for stable reinforcement learning in large language models. *arXiv preprint arXiv:2511.00066v1*, 2025.

[36] Guanhua Huang, Tingqiang Xu, Mingze Wang, Qi Yi, Xue Gong, Siheng Li, Ruibin Xiong, Kejiao Li, Yuhao Jiang, and Bo Zhou. Low-probability tokens sustain exploration in reinforcement learning with verifiable reward. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 24158–24188. Association for Computational Linguistics, 2026.

[37] Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*, chapter 5. Cambridge University Press, 2004.

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

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

A Gradient Norm Decomposition

We first establish an exact decomposition of the gradient norm with respect to the logits, which serves as the foundation for our bounds.

Lemma A.1 (Gradient Norm Decomposition [14]). *Let $\boldsymbol{a}_{i,t} \in \mathbb{R}^{|\mathcal{V}|}$ denote the logits and $\pi_\theta(v^n \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) = \frac{e^{a^n}}{\sum_m e^{a^m}}$ be the induced softmax distribution. Let $y_{i,t}$ be the target token with index k (i.e., $v^k = y_{i,t}$). The squared ℓ_2-norm of the gradient of the group-style objective $\mathcal{J}$ with respect to $\boldsymbol{a}_{i,t}$ satisfies:*

$$\|\nabla_{\boldsymbol{a}_{i,t}} \mathcal{J}(y_{i,t})\|^2 = |w_{i,t}|^2 \left(1 - 2\pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}) + \sum_{n=1}^{|\mathcal{V}|} \pi_\theta(v^n)^2\right),$$

where the weight $w_{i,t}$ is defined as:

$$w_{i,t} = \begin{cases} 0, & \text{if}(\hat{A}_i > 0 \wedge \rho_{i,t} > 1+\epsilon) \vee (\hat{A}_i < 0 \wedge \rho_{i,t} < 1-\epsilon), \\ \frac{\pi_\theta(y_{i,t}|\boldsymbol{x},\boldsymbol{y}_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|\boldsymbol{x},\boldsymbol{y}_{i,<t})}\hat{A}_i, & \text{otherwise.} \end{cases}$$

Proof. Recall that the gradient of the log-likelihood $\ln \pi_\theta(y_{i,t})$ with respect to the logit a^n is given by $\delta_{kn} - \pi_\theta(v^n)$, where δ_{kn} is the Kronecker delta. The gradient of the weighted objective $\mathcal{J}$ is therefore:

$$\frac{\partial \mathcal{J}(y_{i,t})}{\partial a^n} = w_{i,t} \cdot (\delta_{kn} - \pi_\theta(v^n)) = \begin{cases} w_{i,t}(1 - \pi_\theta(v^k)), & \text{if } n = k, \\ -w_{i,t}\pi_\theta(v^n), & \text{if } n \neq k. \end{cases}$$

Computing the squared ℓ_2-norm by summing over all $n \in \{1, \ldots, |\mathcal{V}|\}$:

$$\begin{aligned} \|\nabla_{\boldsymbol{a}_{i,t}} \mathcal{J}(y_{i,t})\|^2 &= \big(w_{i,t}(1 - \pi_\theta(v^k))\big)^2 + \sum_{n \neq k} \big(-w_{i,t}\pi_\theta(v^n)\big)^2 \\ &= |w_{i,t}|^2 \left(1 - 2\pi_\theta(v^k) + \pi_\theta(v^k)^2 + \sum_{n \neq k} \pi_\theta(v^n)^2\right), \\ &= |w_{i,t}|^2 \left(1 - 2\pi_\theta(v^k) + \sum_{n=1}^{|\mathcal{V}|} \pi_\theta(v^n)^2\right). \end{aligned}$$

Identifying v^k as $y_{i,t}$ completes the proof. □

B Proof of Theorem 3.3

Proof. We now complete the proof of Theorem 3.3. By Lemma A.1, the squared gradient norm can be written as

$$\|\nabla_{\boldsymbol{a}_{i,t}} \mathcal{J}(y_{i,t})\|^2 = |w_{i,t}|^2 \left(1 - 2\pi_\theta(y_{i,t}) + \sum_{n=1}^{|\mathcal{V}|} \pi_\theta(v^n)^2\right). \tag{4}$$

To simplify the notation throughout this proof, we denote the components of the distribution $\pi_\theta \in \Delta_{|\mathcal{V}|-1}$ as $\pi_n \triangleq \pi_\theta(v^n)$ for $n \in \{1, \ldots, |\mathcal{V}|\}$. The term $\sum_{n=1}^{|\mathcal{V}|} \pi_n^2$ represents the *collision probability* of the distribution, which measures its concentration. We universally denote the Shannon entropy as $\mathcal{H}(\pi_\theta) = -\sum_{n=1}^{|\mathcal{V}|} \pi_n \ln \pi_n$.

Lower bound. To establish a rigorous lower bound, we relate the collision probability to the Shannon entropy. Recall that the Rényi entropy of order 2 is defined as $\mathcal{H}_2(\pi_\theta) = -\ln\left(\sum_{n=1}^{|\mathcal{V}|} \pi_n^2\right)$. Expressing the argument of the logarithm as an expectation, we have $\sum_{n=1}^{|\mathcal{V}|} \pi_n^2 = \mathbb{E}_{v\sim\pi_\theta}[\pi_\theta(v)]$. Because the function $f(x) = -\ln(x)$ is strictly convex, applying Jensen's inequality yields:

$$\mathcal{H}_2(\pi_\theta) = -\ln\big(\mathbb{E}_{v\sim\pi_\theta}[\pi_\theta(v)]\big) \leq \mathbb{E}_{v\sim\pi_\theta}[-\ln \pi_\theta(v)] = \mathcal{H}(\pi_\theta).$$

Consequently, the collision probability satisfies $\sum_{n=1}^{|\mathcal{V}|} \pi_n^2 = e^{-\mathcal{H}_2(\pi_\theta)} \geq e^{-\mathcal{H}(\pi_\theta)}$. Substituting this inequality into Eq. (4) yields the entropy-based lower bound on the gradient norm:

$$\|\nabla_{\boldsymbol{a}_{i,t}} \mathcal{J}(y_{i,t})\|^2 \geq |w_{i,t}|^2 \left(1 - 2\pi_\theta(y_{i,t}) + e^{-\mathcal{H}(\pi_\theta)}\right).$$

Upper bound. We begin by formulating the objective via equivalent transformations. We introduce the Gini impurity $L = 1 - \sum_{n=1}^{|\mathcal{V}|} \pi_n^2$. The target inequality we aim to prove is:

$$\sum_{n=1}^{|\mathcal{V}|} \pi_n^2 \leq 1 - C_V \mathcal{H}(\pi_\theta)^2,$$

where $C_V = \frac{|\mathcal{V}|-1}{|\mathcal{V}|(\ln|\mathcal{V}|)^2}$. Rearranging the terms and applying the definition of L yields the equivalent condition $C_V \mathcal{H}(\pi_\theta)^2 \leq L$. At the vertices of the probability simplex (where a single component $\pi_n = 1$ and all others vanish), $\mathcal{H}(\pi_\theta) = 0$ and $L = 0$, rendering the inequality trivially satisfied ($0 \leq 0$). For any non-vertex distribution, $L > 0$, allowing us to reformulate the problem as bounding the functional $F(\pi_\theta) = \frac{\mathcal{H}(\pi_\theta)^2}{L} \leq \frac{1}{C_V}$. Thus, it suffices to prove that the global maximum of $F(\pi_\theta)$ on the simplex $\Delta_{|\mathcal{V}|-1}$ satisfies $F(\pi_\theta) \leq \frac{|\mathcal{V}|(\ln|\mathcal{V}|)^2}{|\mathcal{V}|-1}$.

To determine the interior constraints, let $\pi_\theta \in \Delta_{|\mathcal{V}|-1}$ denote a global maximizer of $F(\pi_\theta)$. Excluding the trivial vertices ensures $\mathcal{H}(\pi_\theta) > 0$ and $F(\pi_\theta) > 0$. By the Karush-Kuhn-Tucker (KKT) conditions [37], this extremum must reside in the relative interior of a sub-simplex characterized by k non-zero components ($1 < k \leq |\mathcal{V}|$). We apply the method of Lagrange multipliers to these non-zero components subject to the probability sum equality constraint. Because the logarithmic transformation is strictly monotonically increasing, it preserves the locations of extrema, allowing us to equivalently maximize $\Lambda = 2\ln\mathcal{H}(\pi_\theta) - \ln L - \lambda(\sum \pi_n - 1)$. For any $\pi_i > 0$, the first-order stationarity condition is:

$$\frac{\partial}{\partial \pi_i}[2\ln\mathcal{H}(\pi_\theta) - \ln L] = \lambda \implies \frac{2}{\mathcal{H}(\pi_\theta)}(-1 - \ln\pi_i) - \frac{1}{L}(-2\pi_i) = \lambda.$$

Rearranging this yields $\frac{\pi_i}{L} - \frac{\ln\pi_i}{\mathcal{H}(\pi_\theta)} = \frac{\lambda}{2} + \frac{1}{\mathcal{H}(\pi_\theta)}$. Defining the auxiliary function $g(t) = \frac{t}{L} - \frac{\ln t}{\mathcal{H}(\pi_\theta)}$ for $t \in (0,1)$, its second derivative is $g''(t) = \frac{1}{\mathcal{H}(\pi_\theta)t^2}$. Since $\mathcal{H}(\pi_\theta) > 0$, $g''(t) > 0$ universally on this domain, confirming that $g(t)$ is strictly convex. Consequently, the equation $g(\pi_i) = C$ can possess at most two distinct roots, implying the non-zero components of the extremum can assume at most two distinct values.

We now prove that no two-value stationary points can exist. Assume, toward a contradiction, that the non-zero components take exactly two distinct values, x and y (with $x, y \in (0,1)$ and $x \neq y$), occurring with multiplicities k' and m', respectively. The constraints dictate $k'x + m'y = 1$ and $k'x^2 + m'y^2 = 1 - L$. Because both x and y satisfy the stationarity condition, $g(x) = g(y)$, which implies:

$$\frac{x}{L} - \frac{\ln x}{\mathcal{H}(\pi_\theta)} = \frac{y}{L} - \frac{\ln y}{\mathcal{H}(\pi_\theta)} \implies \mathcal{H}(\pi_\theta) = L\frac{\ln y - \ln x}{y - x}.$$

Introducing the logarithmic difference quotient $\Delta = \frac{\ln y - \ln x}{y-x}$, we obtain $\mathcal{H}(\pi_\theta) = L\Delta$, which implies $\ln y = \ln x + \Delta(y - x)$. Substituting this into the entropy expression yields $\mathcal{H}(\pi_\theta) = -k'x\ln x - m'y(\ln x + \Delta(y-x))$. Grouping terms and applying $k'x + m'y = 1$ gives $\mathcal{H}(\pi_\theta) = -\ln x - m'y(y-x)\Delta$. Equating this with $L\Delta$ results in $-\ln x = [L + m'y(y-x)]\Delta$. Substituting $L = 1 - k'x^2 - m'y^2$ and simplifying the bracketed term reduces it to $1 - x(k'x + m'y) = 1 - x$. Consequently, $-\ln x = (1-x)\Delta$, yielding $\Delta = \frac{-\ln x}{1-x}$. By symmetry, we analogously obtain $\Delta = \frac{-\ln y}{1-y}$. Therefore, any two-value stationary point must satisfy:

$$\frac{-\ln x}{1-x} = \frac{-\ln y}{1-y}.$$

Let $u(t) = \frac{-\ln t}{1-t}$ for $t \in (0,1)$. Its derivative is $u'(t) = \frac{1 - 1/t - \ln t}{(1-t)^2}$. Let $v(t) = 1 - \frac{1}{t} - \ln t$ denote the numerator. Since $v'(t) = \frac{1-t}{t^2} > 0$ for all $t \in (0,1)$, $v(t)$ is strictly monotonically increasing.

Algorithm 1 Spurious-Token-Aware Policy Optimization (STAPO)

Require: Dataset $\mathcal{D}$, Initial Policy π_θ, Group size G, Thresholds τ_p, τ_h, Batch size B

Initialize policy parameters θ
for each training iteration **do**
  Synchronize: $\theta_{\text{old}} \leftarrow \theta$
  Sample prompts $\boldsymbol{x}^B \sim \mathcal{D}$ and generate responses $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_G\}^B \sim \pi_{\theta_{\text{old}}}(\cdot \mid \boldsymbol{x}^B)$
  Compute advantages $\hat{A}_i^B$ using Eq. (1)
  for each mini-batch $\subset \text{BatchData}$ of size B **do**
    for each response $\boldsymbol{y}_i$ and token t **do**
      Obtain $p_{i,t} = \pi_\theta(y_{i,t} \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t})$ and $h_{i,t} = \mathcal{H}(\pi_\theta(\cdot \mid \boldsymbol{x}, \boldsymbol{y}_{i,<t}))$
      $\mathbb{I}_{i,t}^{\text{S2T}} \leftarrow 1$ ▷ Default: Keep
      if $\hat{A}_i > 0 \wedge p_{i,t} < \tau_p \wedge h_{i,t} < \tau_h^{(q)}$ **then** $\mathbb{I}_{i,t}^{\text{S2T}} \leftarrow 0$ ▷ Spurious Tokens: Mask
    end for
    Update θ through Eq. (3) with $\mathbb{I}_{i,t}^{\text{S2T}}$
  end for
end for
return Final policy π_θ

Because $v(1) = 0$, it follows that $v(t) < 0$ on $(0, 1)$. This establishes that $u'(t) < 0$ globally on this interval, implying $u(t)$ is strictly monotonically decreasing. Thus, $u(x) = u(y)$ holds if and only if $x = y$, directly contradicting $x \neq y$.

The preceding contradiction establishes that the stationary point must be a uniform distribution over a support of size k ($1 < k \leq |\mathcal{V}|$), where each of the k non-zero components equals $\frac{1}{k}$. At this point, the functional evaluates to $h(k) = \frac{k(\ln k)^2}{k-1}$. Treating k as a continuous variable on $(1, |\mathcal{V}|]$ and differentiating yields:

$$h'(k) = \frac{\ln k[2(k-1) - \ln k]}{(k-1)^2}.$$

For $k > 1$, the standard inequality $\ln k < k - 1$ guarantees that $2(k-1) - \ln k > k - 1 > 0$. Because $\ln k > 0$, $h'(k) > 0$. This demonstrates that the objective function strictly monotonically increases with the support size k. Thus, the global maximum is uniquely attained at the full support $k = |\mathcal{V}|$:

$$\max_{\pi_\theta \in \Delta_{|\mathcal{V}|-1}} F(\pi_\theta) = h(|\mathcal{V}|) = \frac{|\mathcal{V}|(\ln |\mathcal{V}|)^2}{|\mathcal{V}| - 1}.$$

Given this global maximum, the inequality $\frac{\mathcal{H}(\pi_\theta)^2}{1 - \sum_{n=1}^{|\mathcal{V}|} \pi_n^2} \leq \frac{|\mathcal{V}|(\ln |\mathcal{V}|)^2}{|\mathcal{V}|-1}$ holds universally for any distribution $\pi_\theta \in \Delta_{|\mathcal{V}|-1}$. Multiplying both sides by the non-negative denominator gives $\mathcal{H}(\pi_\theta)^2 \leq \frac{|\mathcal{V}|(\ln |\mathcal{V}|)^2}{|\mathcal{V}|-1} \left(1 - \sum_{n=1}^{|\mathcal{V}|} \pi_n^2\right)$. Multiplying by the constant factor $\frac{|\mathcal{V}|-1}{|\mathcal{V}|(\ln |\mathcal{V}|)^2}$ and substituting the definition of C_V results in $C_V \mathcal{H}(\pi_\theta)^2 \leq 1 - \sum_{n=1}^{|\mathcal{V}|} \pi_n^2$. Isolating the summation term yields the desired upper bound:

$$\sum_{n=1}^{|\mathcal{V}|} \pi_n^2 \leq 1 - C_V \mathcal{H}(\pi_\theta)^2.$$

Combining these lower and upper bounds completes the proof of Theorem 3.3. □

C Algorithm

The complete STAPO procedure is summarized in Algorithm 1.

D Training Details

We implement our proposed STAPO algorithm based on the open-source alignment framework `veRL` [38]. We utilize the DAPO-Math-17K [6] as the training dataset, where each prompt is formatted

Table 4: Full Training Hyperparameters

Hyperparameter	Value
Advantage Estimator	GRPO
Reward Function	DAPO
Probability Threshold τ_p	0.002
Entropy Percentile q	75
Use KL Loss	No
Use Entropy Regularization	No
Train Batch Size	256
Max Prompt Length	1k
Max Response Length	15k
PPO Mini Batch Size	64
Dynamic Micro Batch Size	Enabled
Clip Ratio Range	[0.8, 1.28]
Grad Clip	1.0
Learning Rate	1e-6
Warm-up Step	10
Training Temperature	1.0
Training Top-p	1.0
Validation Temperature	1.0 or 0.7
Validation Top-p	1.0 or 0.9
Rollout N	8

with the instruction: "Please reason step by step, and put your final answer within \boxed{}". All models are trained using the AdamW optimizer with a constant learning rate of 1×10^{-6} and a warm-up phase of 10 steps. To ensure training stability, we apply a global gradient clipping norm of 1.0.

For efficient data generation, we utilize vLLM [39] as the inference backend. The training process employs a global batch size of 256, with each prompt generating a group of $G = 8$ rollouts. Following the DAPO formulation, we do not employ a separate value network or an additional KL divergence penalty term in the loss function; instead, we rely on group-relative advantages and the clipping mechanism to constrain policy updates. We conduct all experiments on 64 NVIDIA H20 GPUs, with each training session taking an average of 5 to 7 days. The complete set of hyperparameters used across all model scales is detailed in Table 4.

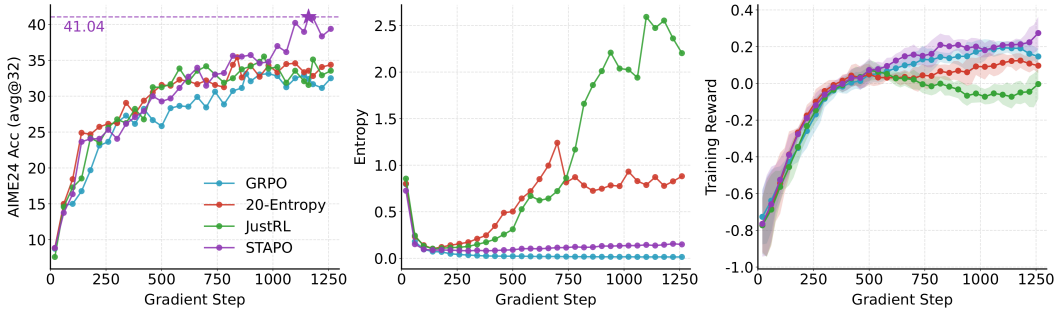
E Supplementary Experimental Results

E.1 Supplementary Training Dynamics

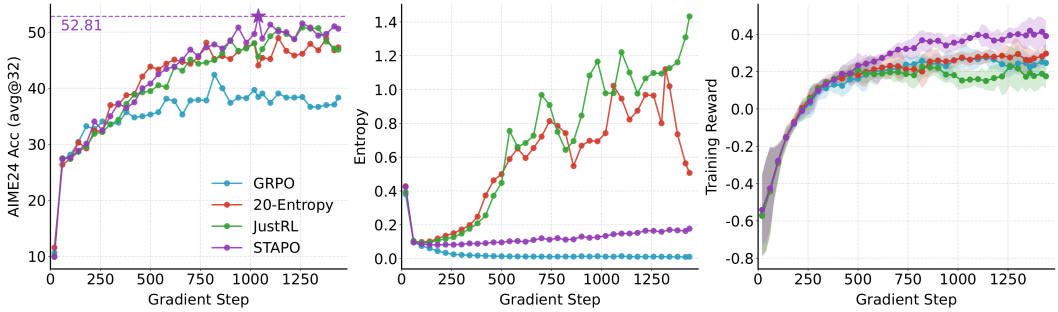
We further present the training dynamics for the larger model scales (Qwen3-8B and Qwen3-14B) in this section. As shown in Figure 10, the trends observed on the 1.7B model scale consistently to larger models. Across both 8B and 14B base models, STAPO successfully prevents premature entropy collapse, maintaining stable exploration throughout the training process and achieving state-of-the-art performance. To clearly visualize the extended training process, we downsampled the plotted curves, which were originally evaluated every 20 gradient steps, by retaining only every third data point. Crucially, the true peak performance of each accuracy curve remains strictly preserved.

E.2 Analysis of Computational and Memory Overhead

STAPO is a lightweight token-level optimization strategy that introduces no additional model parameters, forward passes, or backward passes. Its computational overhead is limited to entropy computation, entropy-based token selection, mask construction, and masked-objective application. To quantify the associated runtime and memory overhead, we profile STAPO under the same training configuration as JustRL, measuring the additional runtime and GPU memory consumption per micro-batch averaged over the first 100 training steps. The results are summarized in Table 5.



(a) AIME24 Acc, entropy, and training reward for Qwen3-8B base.



(b) AIME24 Acc, entropy, and training reward for Qwen3-14B base.

Figure 10: **Training Results on Larger Models.** The training dynamics for the Qwen3-8B and Qwen3-14B models. The results demonstrate that STAPO consistently maintains stable policy entropy and achieves higher accuracy across different parameter scales.

Table 5: **Computational and Memory Overhead of STAPO.** Additional overhead per micro-batch, averaged over the first 100 training steps. Entropy: entropy computation; Selection: entropy-based token selection; Masking: mask construction and masked-objective application.

Operation	**Time (ms)**	**Memory (MB)**
Entropy	11.939	0.474
Selection	0.258	0.947
Masking	0.452	2.487
Total	**12.650**	**3.908**

As shown in Table 5, the total additional computational overhead is only 12.650 ms per micro-batch, with an additional memory footprint of 3.908 MB. Entropy computation accounts for most of the additional runtime, while token selection and mask construction introduce only minor costs.

For reference, the baseline training configuration requires approximately 130 seconds per training update and occupies approximately 60 GB of GPU memory. The measured overhead is therefore negligible compared with the overall training cost, with both the additional runtime and memory footprint being approximately four orders of magnitude smaller than the corresponding baseline quantities. These results demonstrate that STAPO can be integrated into existing RL training pipelines with minimal computational and memory overhead.

E.3 Validation of Spurious Tokens

To assess whether the tokens masked by STAPO are genuinely spurious rather than legitimate low-probability tokens, we conduct a token-level validation study on a randomly sampled subset of 500 masked tokens. Following the taxonomy in Appendix E.4, we consider three categories of spurious tokens: *Uncommon Syntax* (Category I), *Hallucination or Mathematical Error* (Category II), and

Table 6: Validation of 500 randomly sampled tokens masked by STAPO using GPT-5.5.

Category	**Count**	**Ratio**
Uncommon Syntax (I)	269	53.8%
Hallucination / Math Error (II)	128	25.6%
Formatting Error (III)	77	15.4%
Spurious (Total)	**474**	**94.8%**
Legitimate (Non-spurious)	26	5.2%
Total	**500**	**100%**

Formatting Error (Category III). We employ the GPT-5.5 API with a fixed evaluation rubric to determine whether each masked token is spurious and, if so, assign it to one of the three categories. As an additional sanity check, we manually inspect 20 randomly selected cases, all of which agree with the GPT-5.5 annotations.

Table 6 summarizes the validation results. Among the 500 sampled masked tokens, GPT-5.5 identifies 474 (94.8%) as spurious, including 269 cases of uncommon syntax, 128 cases of hallucination or mathematical errors, and 77 cases of formatting errors. The remaining 26 tokens (5.2%) are classified as legitimate low-probability tokens. These results suggest that STAPO achieves a high estimated precision in identifying spurious tokens, despite requiring neither an auxiliary evaluator nor manual token annotations during training.

To better understand the remaining non-spurious cases, we examine a representative example in which a token is masked by STAPO but judged legitimate by GPT-5.5. The model-generated reasoning proceeds as follows:

> *For a cuboid with side lengths a, ar, and ar^2, the preceding derivation gives*
>
> $$a^3r^3 = 216 \implies (ar)^3 = 216 \implies ar = 6.$$
>
> *The model then generates*
>
> $$\begin{aligned} S &= 2(ab + bc + ca) \\ &= 2\big(a \times ar + \boxed{a} \times ar^2 + ar \times ar^2\big). \end{aligned}$$

Here, the boxed token $\boxed{a}$ is masked by STAPO. The top-5 token probabilities under the current policy are:

$$ar : 99.88\%, \quad a : 0.12\%, \quad ab : 0.00\%, \quad \backslash(: 0.00\%, \quad b : 0.00\%.$$

Despite its low probability of 0.12%, GPT-5.5 classifies the masked token a as legitimate, providing the following explanation:

> *"The target token a begins the term $a \times ar^2$, which is semantically correct as ca, and is a natural continuation after $a \times ar+$."*

This judgment is mathematically valid. The token a forms the product $a \times ar^2$, corresponding to the ca term in the standard surface-area formula $2(ab + bc + ca)$. Although the generated terms follow the order $ab + ca + bc$ instead of the conventional order $ab + bc + ca$, the two expressions are equivalent due to the commutativity of addition.

This case illustrates that a small fraction of STAPO-masked tokens can be semantically and mathematically valid despite having low probability. Such cases may arise from locally unexpected but equivalent reasoning continuations, such as reordering terms in an algebraic expression. Nevertheless, the substantial probability discrepancy indicates that these tokens deviate from the policy's preferred local continuation, explaining why they are selected by STAPO.

E.4 Spurious Tokens in Mathematical Reasoning

We provide additional examples of spurious tokens identified by STAPO in mathematical reasoning trajectories. These examples are organized into three categories: (I) uncommon syntax, (II) hallucinations and mathematical errors, and (III) formatting errors. For each case, we present the relevant generation context, the masked token, its probability and advantage, and the top-five alternative tokens. The masked tokens are highlighted in red.

Category I: Uncommon Syntax 5 cases

Case I.1: Unusual Word Choice

Context. The length of A is 1981, and the length of B is 1014. Since the length of U is 2012, for the minimum length of $A \cap B$, B should start immediately from the right of A. The length of $A \cap B$ is the sum of the lengths of A and B minus the length of U , so its minimum value is $1981 + 1014 - 2012 = 983$. Here is a Python calculation to verify the above reasoning...

Masked token: calculation **P:** 0.0115% **Adv.:** 1.62

Top-5: code(51.90%) • program(31.48%) • solution(11.58%) • script(3.32%) • function(0.58%)

Case I.2: Inconsistent Terminology

Context. We are given n islands with ferry lines between each pair of distinct islands, which is essentially a complete graph K_n. Each edge of the graph is colored by one of k colors, corresponding to the companies operating the ferry lines.

For any closure of one company, all its lines must be broken (the corresponding edges ignored) such that the remaining graph can no longer support a Hamiltonian Cycle, meaning it's no longer possible to visit all the islands exactly once without returning to the starting point.

According to the condition, every color, therefore, must serve to connect the graph in such a way that the removal of edges of any single color makes the graph Hamiltonian non-existent.

In a complete graph K_n, we can color the edges in such a way that removing any single color results in a graph that lacks a Hamiltonian Cycle. For this, we can use a result from graph theory, informally known as "edge coloring minimum spanning trees": for n nodes, we can divide the set of nodes into $2\lfloor \frac{n}{2} \rfloor$ color classes (as far as coloring spans, and where $\lfloor x \rfloor$ means "the greatest integer less than x").

So when we have $n = 2m$ (even) nodes, the maximal number of colors k we can use is $m - 1$, since we can divide our n nodes into m colors and deleting any one will still result in a complete graph K_{2m-1} which remains Hamiltonian (since K_{2m-1} is always Hamiltonian for $m \geq 1$).

For $n = 2m + 1$ (odd) nodes, we can divide our n nodes into m classes, so the discard of any one of these m classes will disintegrate the graph such that a Hamiltonian Cycle will become impossible because it contains a vertex of degree 1 or a subdivision thereof.

Masked token: broken **P:** 0.0471% **Adv.:** 2.47

Top-5: removed(85.53%) • closed(3.32%) • deleted(2.58%) • disconnected(1.57%) • taken(0.95%)

Masked token: discard **P:** 0.0001% **Adv.:** 2.47

Top-5: maximal(85.89%) • maximum(9.05%) • number(1.57%) • largest(0.45%) • answer(0.45%)

Case I.3: Redundant Answer Wording

Context. Therefore, the final boxed answer is:`\[ \boxed {41} \]<|endoftext|>`

Masked token: boxed **P:** 0.1486% **Adv.:** 0.54

Top-5: answer(98.88%) • result(0.52%) • value(0.19%) • boxed(0.15%) • solution(0.15%)

Case I.4: Awkward Preposition

Context. To solve the given problem, we need to find the smallest number of candies N that satisfies the given conditions. Specifically:

1. When N candies are divided between 21 people (Albert and his 20 friends), the remainder is 5.

Masked token: between **P:** 0.0667% **Adv.:** 0.72

Top-5: among(64.55%) • by(23.75%) • evenly(8.74%) • amongst(1.95%) • equally(0.92%)

Case I.5: Corrupted Word Fragments

Context. To trisegment, we must first count the number of points in the polygon.

Step 1:sume the number of points in the polygon

Let n yea the number of sides (or vertices) in a polygon P as given in the problem statement. We are given that $P_1, P_2, \ldots P_n/$ are the n vertices of the polygon.

Masked token: tr **P:** 0.0004% **Adv.:** 0.72

Top-5: olve(63.83%) • determine(30.15%) • find(5.24%) • triang(0.20%) • count(0.10%)

Masked token: y **P:** 0.0003% **Adv.:** 0.72

Top-5: be(88.02%) • represent(5.63%) • denote(3.41%) • =(0.76%) • and(0.59%)

Category II: Hallucinations and Math Errors 6 cases

Case II.1: Incorrect Formula Index

Context.

$$A = \frac{1}{2}\left|x_1(y_2 - y_3) + x_2(y_3 - y_2) + x_3(y_1 - y_2)\right|$$

Here, $ x$s and ys are the coordinates of the points.

Masked token: 2 **P:** 0.0142% **Adv.:** 0.54

Top-5: 1(99.95%) • 2(0.01%) • 3(0.01%) • 0(0.00%) • }_(0.00%)

Masked token: $s **P:** 0.0117% **Adv.:** 0.54

Top-5: _(94.78%) • 1(1.05%) • =(0.82%) • _{(0.82%) • $(0.50%)

Case II.2: Incorrect Numeric Value

Context. $(a - 10, b - 10) = (193, 11) \rightarrow a = 203,\ b = 213$

Masked token: 3 **P:** 0.0017% **Adv.:** 0.94

Top-5: \ (99.98%) • \n(0.01%) • 1(0.01%) • \\n(0.00%) • 3(0.00%)

Case II.3: Incorrect Expression

Context. $6901 = 67 \times 103 - 1$

Masked token: - **P:** 0.0202% **Adv.:** 1.21

Top-5: $,(99.48%) • $(0.36%) • +(0.13%) • -(0.02%) • =(0.00%)

Case II.4: Incorrect Decimal Value

Context. $-16a^3 = 350,$
$a^3 = -21.56625.$

Masked token: 5 **P:** 0.2433% **Adv.:** 1.62

Top-5: 8(98.16%) • 2(1.40%) • 5(0.24%) • 9(0.07%) • 7(0.04%)

Case II.5: Incorrect Derived Value

Context. Compute a for Each Pair:

$$
\begin{aligned}
(-5, 30) &\to a = -25 \\
(-4, 12) &\to a = -8 \\
(-3, 6) &\to a = -3 \\
(-2, 3) &\to a = -1 \\
(0, 0) &\to a = 0 \\
(-7, -42) &\to a = 49 \\
(-8, -24) &\to a = 32 \\
(-9, -18) &\to a = 9 \\
(-10, -15) &\to a = 5 \\
(-12, -12) &\to a = 24
\end{aligned}
$$

Masked token: 9 **P:** 0.0123% **Adv.:** 0.35

Top-5: 2(99.98%) • 9(0.01%) • 1(0.01%) • 3(0.00%) • 8(0.00%)

Case II.6: Incorrect Fraction Denominator

Context. Thus,

$$\sin x = \frac{435}{435} = \frac{435}{533}$$

Masked token: 4 **P:** 0.0867% **Adv.:** 1.62

Top-5: 5(95.28%) • 3(3.69%) • \(0.64%) • 6(0.11%) • 4(0.09%)

Category III: Formatting Errors — 6 cases

Case III.1: Unexpected Step Marker

Context. To solve this problem, let's start by using the given information about the function $f(x) = ax^2 + bx + c$:

1. Since $f(1) = 0$, we know that:

$$a + b + c = 0$$

This means $c = -a - b$.

Now, substitute $c = -a - b$ into the quadratic function:

$$f(x) = ax^2 + bx - (a + b)$$

Masked token: Now **P:** 0.0552% **Adv.:** 0.35

Top-5: 2(99.74%) • Next(0.12%) • Now(0.06%) • So(0.03%) • Sub(0.02%)

Case III.2: Malformed LaTeX Output

Context. \boxed{5\text{ agony}}

Masked token: \ **P:** 0.0015% **Adv.:** 0.35

Top-5: }\n(99.57%) • }(0.25%) • }\n\n(0.12%) • }\(0.03%) • }.\n(0.01%)

Case III.3: Broken Math Delimiters

Context. \therefore 10^y)$ \(\sqrt{25}=5

Masked token:)$ **P:** 0.0063% **Adv.:** 0.72

Top-5: =(84.69%) • =\(6.95%) • =(4.22%) • ␣\n(1.55%) • \n(0.35%)

Case III.4: Missing Mathematical Spacing

Context.

$$\begin{aligned}\Sigma_3 = 11^3 + 10^3 + 9^3 + 8^3 + 7^3 \\ + 6^3 + 5^3 + 4^3 + 3^3 \\ + 2^3 + 1^3 + 0^3\end{aligned}$$

Masked token: 3 **P:** 0.0710% **Adv.:** 1.62

Top-5: ␣(99.92%) • 3(0.07%) • \n(0.00%) • \(0.00%) • +(0.00%)

Case III.5: Extraneous Punctuation

Context. Repeating these steps three more times for each roll back to P_4:

- If A was rolled, $P_5 = (56, 368)$
- If B was rolled, $P_5 = (56, 184)$
- If C was rolled, $P_5 = (768, 184)$:

Masked token: : **P:** 0.0035% **Adv.:** 2.47

Top-5: \n\n(88.06%) • ((4.38%) • \n(4.38%) • ␣\n\n(1.26%) • ,(0.36%)

Case III.6: Corrupted Output Suffix

Context. Hence, the maximum value of `$\frac{n_i}{k}$` for `1 \leq i \leq 70` is:
`\[`
`\boxed{553}`
`\]`
`struck`user<|endoftext|>

Masked token: struck **P:** 0.0000% **Adv.:** 1.21

Top-5: \(98.58%) • \)(0.13%) • \(0.10%) • $(0.08%) • $$(0.07%)

These cases illustrate that even trajectories with correct final answers may contain locally undesirable tokens, ranging from unusual word choices to incorrect intermediate calculations and formatting artifacts. Such tokens may still be reinforced by response-level rewards, highlighting the need for finer-grained token-level optimization.

E.5 Spurious Tokens in Code Reasoning

We further examine spurious tokens identified by STAPO in code reasoning trajectories. Similar to mathematical reasoning, positively rewarded coding responses may contain locally unreliable or unnecessary tokens. We present three representative cases involving flawed code, invalid function calls, and formatting artifacts. For each case, we show the relevant code fragment and explain why the masked token is considered spurious and how the response may still receive a positive reward. The masked tokens are highlighted in red.

Case 1: Flawed Code

```
visited = [0 for _ in range(N + 1)]
i = 0
while any(discovered):
    ...
```

Spurious Reason

The masked token occurs within a flawed graph-search procedure, where the surrounding loop logic is unreliable and inconsistent with the intended component-search structure.

Why a Positive Reward Is Possible

The generated solution may still pass the evaluated test cases if the problematic execution path is not triggered.

Case 2: Invalid Python Syntax

```
if a < 3200:
    println("red")
```

Spurious Reason

The masked token `println` is not a valid Python built-in function; the intended function is `print`.

Why a Positive Reward Is Possible

The response also contains a valid Python solution block. The erroneous token appears in an intermediate or redundant fragment that is not used for evaluation.

Case 3: Formatting Artifact

```
return True
\n\n\n
maxNumCaptures = 0
```

Spurious Reason

The masked sequence consists of excessive newline characters, introducing unnecessary formatting between code fragments without contributing to the solution logic.

Why a Positive Reward Is Possible

The formatting artifact does not affect the final extracted executable code and therefore does not change the evaluation outcome.

These examples illustrate that positive response-level rewards do not necessarily imply the correctness or relevance of every generated token in code reasoning. In particular, coding responses may contain unreliable intermediate logic, invalid syntax in unevaluated fragments, or redundant formatting, while still producing an executable solution that passes the evaluation. By identifying and suppressing such locally spurious tokens, STAPO provides a finer-grained optimization signal beyond response-level rewards.

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: Yes, the main claims in the abstract and introduction accurately reflect the paper's contributions and scope. Please refer to the Abstract and Section 1.

   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: We discuss potential scope extensions and future work in Section 6, and address hyperparameter sensitivity in Section 4.3.

   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: Assumptions for Theorem 3.3 are stated in Section 3.2, with complete proofs provided in Appendix B.

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: Implementation details are described in Section 4.1 and Appendix D.

   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: The core code is provided in the supplemental material. Detailed information regarding the datasets and processing methods is available in Appendix D.

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: Complete training hyperparameters and configurations are provided in Table 4 within Appendix D.

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: Statistical variations and confidence intervals are reported in Figure 2c, Figure 5, and Figure 10.

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: Details regarding the type and number of GPUs (64 NVIDIA H20) are provided in Appendix D.

   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: Our research adheres to the established ethical guidelines for LLM training and alignment.

   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: [Yes]

    Justification: We address the scope of our reasoning-focused method and potential broader applications in Section 6.

    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: This work proposes an optimization algorithm and does not introduce new high-risk datasets or models.

    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: All used benchmarks (e.g., AIME24, MATH500) and base models (Qwen3) are properly cited in Section 4.1.

    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: [N/A]

    Justification: No new datasets or models are introduced as primary assets in this work.

    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: Our methodology does not involve human subjects or crowdsourcing.

    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: This research does not involve human subjects.

    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: The core method development in this research does not involve LLMs as any important, original, or non-standard components.

    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.