Low-probability Tokens Sustain Exploration in Reinforcement Learning with Verifiable Reward

Anonymous ACL submission

Abstract

Reinforcement Learning with Verifiable Rewards (RLVR) has propelled Large Language Models in complex reasoning, yet its scalability is often hindered by a training bottleneck where performance plateaus as policy entropy collapses, signaling a loss of exploration. While previous methods attempt to maintain high entropy, we argue that unselective entropy maximization risks amplifying irrelevant noise rather than fostering meaningful exploration. In this paper, we identify a deeper issue: the gradual elimination of valuable low-probability exploratory tokens, which we term ***reasoning sparks***, driven by RLVR over-penalization. To address this, we introduce Low-probability Regularization (Lp-Reg). Leveraging the statistical distinction where reasoning sparks exhibit higher probabilities than noise, Lp-Reg constructs a filtered, re-normalized proxy distribution. By penalizing deviations from this proxy via forward KL divergence, our method selectively shields these valuable tokens from elimination. Experiments demonstrate that Lp-Reg enables stable on-policy training for over $3,000$ steps (81,204 GPU-hours), sustaining exploration in regimes where baselines typically collapse. Validated across extensive evaluations totaling over 300,000 cumulative GPU-hours, Lp-Reg consistently achieves state-of-the-art performance across diverse model families, sizes, and domains, with relative accuracy improvements ranging from 3.06% to 7.98%.

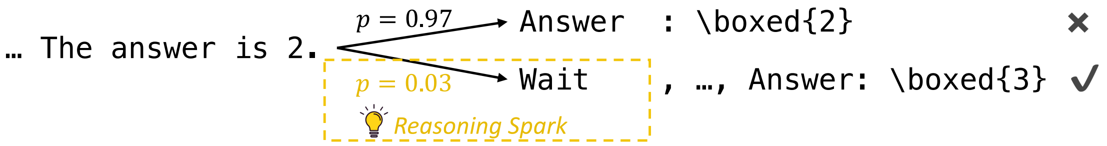
1 Introduction

The advent of large reasoning models, such as OpenAI o1 (OpenAI et al., 2024) and DeepSeek-R1 (DeepSeek-AI et al., 2025), has reshaped AI. A central technique underpinning these systems is reinforcement learning with verifiable reward (RLVR), which assigns reward to verifiable solutions through rule-based verification. These models generate extended chain-of-thought (CoT) reasoning (Wei et al., 2023) to solve challenging problems in domains like mathematical olympiads (He et al., 2024b). However, a notable bottleneck emerges during RL training that limits its scalability, frequently culminating in a performance plateau and subsequent collapse. This failure is consistently accompanied by a rapid decay in policy entropy, indicating a severe loss of exploration capacity (Yu et al., 2025; Cui et al., 2025; Wang et al., 2025a).

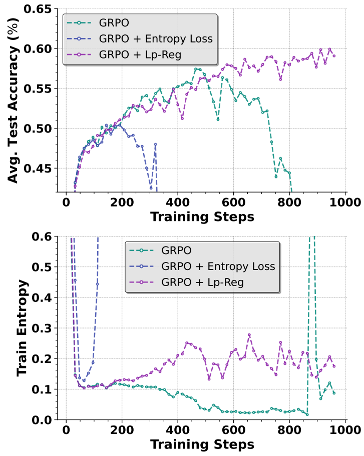
Previous approaches attempt to mitigate this by maintaining higher entropy through adaptive regularization or selective updates (He et al., 2025; Cui et al., 2025; Wang et al., 2025a). However, relying on overall entropy can be an indirect and imprecise tool. An indiscriminate focus on maximizing randomness risks amplifying noise and destabilizing training (Ömer Veysel Çağatan and Akgün, 2025).

Our analysis suggests the performance bottleneck may stem from the systematic elimination of **Reasoning Sparks**, which empirically defined as low-probability exploratory tokens (Figure 1a) like "wait", "however", or "perhaps", "low-probability" describes their statistical attribute, while "exploratory" describes their semantic function to start a new exploration fork. As shown in Figure 1c, standard GRPO suppresses the sampling of these sparks. Conversely, indiscriminately boosting randomness by entropy loss amplifies irrelevant tokens (e.g., "cost", "fine"), which are semantically out of context for mathematical reasoning. This noise amplification accelerates performance collapse compared to the baseline (Figure 1b).

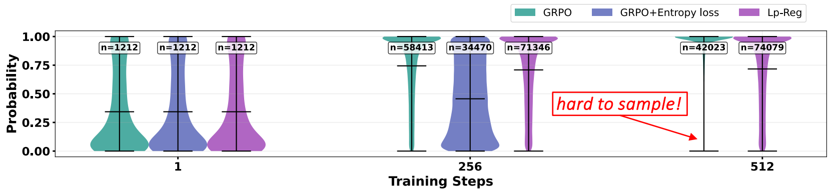
This presents a central challenge: **to protect *reasoning sparks* without amplifying irrelevant noise.** To address this, we introduce **Low-probability Regularization (Lp-Reg)**. Our method leverages a critical statistical distinction observed in Section 6.1: within the low-probability tail, meaningful exploratory tokens (e.g., "wait") consistently exhibit higher probabilities than irrelevant noise (e.g., "cost"). Operating directly on the predictive distribution (logits), Lp-Reg first filters



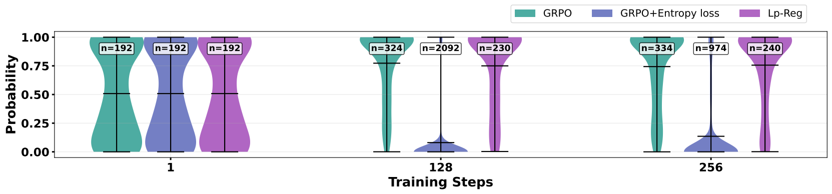
(a) `Reasoning Sparks`: low-probability exploratory tokens that initiate potential diverse reasoning paths.



(b) Training Dynamics



(c) Aggregated distribution of observed sampling probabilities for exploratory tokens (e.g., "wait", "however"). "n" represents sampling number.



(d) Aggregated distribution of observed sampling probabilities for meaningless tokens (e.g., "cost", "fine"). "n" represents sampling number.

Figure 1: Selectively preserving low-probability tokens is key to overcoming performance plateaus. **(a)** Example of *reasoning sparks*. **(b)** Standard GRPO collapses with decaying entropy, whereas indiscriminate entropy loss accelerates collapse. **(c)** GRPO systematically suppresses *reasoning sparks*, collapsing their distribution. Entropy loss fails to mitigate this, but Lp-Reg preserves them. **(d)** Entropy bonuses fail by amplifying meaningless noise. Plots (c) and (d) show aggregated statistics; single-token details are in Appendix H.1.

out presumed noise via a probability threshold, then renormalizes the probability mass. This process constructs a less-noisy proxy distribution in which valuable low-probability tokens are preserved, and their relative probabilities are amplified. Finally, by penalizing deviations from this proxy using forward KL divergence, Lp-Reg selectively preserves the remaining low-probability tokens against suppression.

Experiments show that Lp-Reg enables stable on-policy training for over 3,000 steps (totaling 81,204 GPU-hours), effectively sustaining exploration in regimes where baselines typically collapse. Across more than 300,000 cumulative GPU-hours of experiments, Lp-Reg consistently achieves state-of-the-art performance across diverse model families (including Qwen and Llama) and scales (ranging from 8B to 32B). Furthermore, its robustness is validated across multiple domains, demonstrating superior relative accuracy improvements from 3.06% to 7.98% in three domains: mathematics, science, and code. Our contributions are summarized as follows:

- We identify the suppression of low-probability tokens, rather than overall entropy decay, as a deeper issue for sustained RLVR performance.
- We introduce Lp-Reg, a method that creates a stable exploratory environment by filtering presumed noise to selectively protect low-probability exploratory tokens.
- We demonstrate that Lp-Reg achieves state-of-the-art performance across diverse benchmarks and enables stable, continuous on-policy scaling where baselines fail.

2 Related Work

Reinforcement learning for LLMs Reinforcement learning (RL) has established itself as the dominant framework for enhancing LLM reasoning (OpenAI et al., 2024; DeepSeek-AI et al., 2025). Notably, RL with verifiable rewards (RLVR) has driven significant breakthroughs by leveraging automatic verification (Shao et al., 2024a; Yang et al., 2025a; Team et al., 2025). Building on RLVR and GRPO, recent methods such as DAPO (Yu et al., 2025), VAPO (Yue et al., 2025), and other policy optimization variants (Zhao et al., 2025; Cui et al., 2025; Zheng et al., 2025) have been proposed to further improve the stability and scalability of reasoning models.

Entropy collapse in RL training A major bottleneck in reasoning RL is the rapid collapse of policy entropy, reflecting insufficient exploration. To mitigate this, researchers have proposed strategies including selectively regularizing high-entropy "forking" tokens (Wang et al., 2025a), amplifying advantages at exploratory positions (Cheng et al., 2025), or modifying clipping strategies (Yu et al., 2025; MiniMax et al., 2025; Su et al., 2025). However, these methods primarily monitor policy entropy, a metric correlational rather than causal to exploration. In contrast, our approach directly analyzes the next-token prediction distribution. This enables a more semantically grounded investigation into individual candidate probabilities and their specific roles in exploration dynamics.

Intrinsic confidence of LLMs LLMs demonstrate strong intrinsic confidence signals that can effectively guide complex reasoning (Saurav et al., 2022; Loka et al., 2024; Amir et al., 2025). Research indicates that tokens with relatively higher probabilities are often more contextually appropriate than lower-probability counterparts (Nguyen et al., 2025; Xu et al., 2025; Fu et al., 2025b). While related works utilize entropy minimization to sharpen confidence for consistent inference (Gao et al., 2025; Agarwal et al., 2025), we leverage this intrinsic confidence differently: using it to distinguish between valuable *reasoning sparks* and irrelevant noise within the low-probability range.

3 Preliminaries

3.1 Reinforcement Learning with Verifiable Rewards

Reinforcement learning (RL) enhances LLMs (Murphy, 2024) by maximizing the expected reward:

$$\mathcal{J}_{\mathrm{RL}}(\boldsymbol{\theta}) = \mathbb{E}_{(q,a)\sim D, o\sim\pi_\theta(\cdot|q)}\big[r(o,a)\big], \tag{1}$$

where $r(o,a)$ is determined by rule-based functions (e.g., Math-Verify[1]). Recent studies show that large-scale RLVR substantially improves complex problem-solving by encouraging chain-of-thought reasoning (DeepSeek-AI et al., 2025). Typically, this objective is optimized using policy gradient methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017) or Group Relative Policy Optimization (GRPO) (Shao et al., 2024b).

[1] https://github.com/huggingface/Math-Verify

3.2 Group-Relative Policy Optimization

GRPO is an actor-only policy gradient method that estimates advantages by leveraging multiple outputs $\{o_1, \ldots, o_G\}$ sampled from the same prompt. The advantage for the t-th token in o_i is defined as:

$$A_{i,t} = \frac{R(o_i) - \mathrm{mean}(\mathcal{G})}{\mathrm{std}(\mathcal{G})}, \tag{2}$$

where $\mathcal{G} = \{R(o_1), \ldots, R(o_G)\}$ denotes the group rewards. The policy is optimized using the PPO surrogate objective:

$$\begin{aligned}\mathcal{J}_{\mathrm{GRPO}}(\boldsymbol{\theta}) =&\mathbb{E}_{(q,a)\sim D,\, \{o_i\}_{i=1}^G\sim\pi_{\boldsymbol{\theta}_{\mathrm{old}}}(\cdot|q)}\frac{1}{\sum_{i=1}^G|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\\ &\Big[\min\big[r_{i,t}\,A_{i,t},\, \mathrm{clip}\big(r_{i,t}, 1-\epsilon, 1+\epsilon\big)\,A_{i,t}\big]\\ &-\beta\, D_{\mathrm{KL}}\left(\pi_{\boldsymbol{\theta}}\,\|\,\pi_{\mathrm{ref}}\right)\Big],\end{aligned} \tag{3}$$

where β scales the KL regularization against the reference policy π_{ref}. The importance sampling weight $r_{i,t} = \frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\mathrm{old}}}(o_{i,t}|q,o_{i,<t})}$ is computed relative to the behavior policy $\pi_{\theta_{\mathrm{old}}}$. Finally, ϵ defines the clipping ratio to prevent excessive policy deviation and ensure optimization stability.

4 Low-probability Regularization

We propose **Low-probability Regularization (Lp-Reg)** to prevent the elimination of potentially valuable exploratory tokens. Integrated into policy gradient algorithms, Lp-Reg leverages the model's own predictive distribution to construct a noise-filtered reference, thereby preserving low-probability tokens. The additional computational overhead remains below 0.5% (Appendix F).

4.1 Proxy Distribution

Lp-Reg builds a proxy distribution by filtering noise from the current policy $\pi_{\boldsymbol{\theta}}$ and renormalizing the remaining probability mass.

Filtering Noise Tokens We identify noise tokens as those with probability $\pi_{\boldsymbol{\theta}}(o|\cdot) < \tau$. We consider two threshold strategies: (1) **Fixed:** τ is a constant (e.g., 0.02); (2) **Min-p:** $\tau = \kappa \cdot \max_{o'} \pi_{\boldsymbol{\theta}}(o'|\cdot)$, where $\kappa \in (0,1)$ makes the threshold adaptive to distribution sharpness (Nguyen et al., 2025). We primarily employ the Min-p strategy, though fixed thresholds are also effective (Section 5.4).

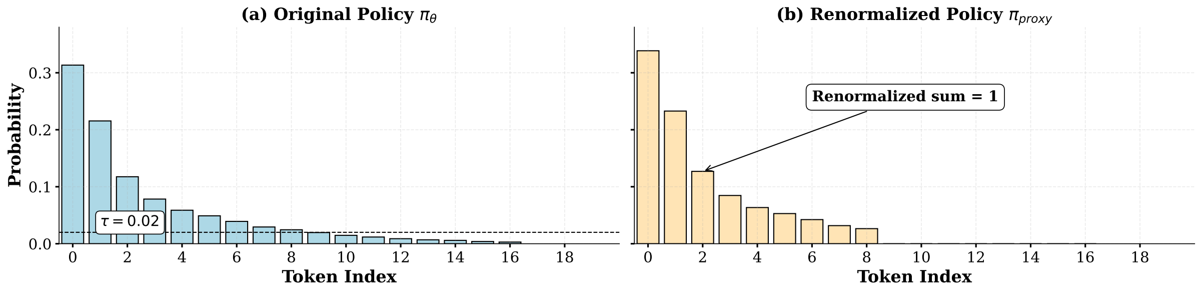


Figure 2: An example of probability renormalization. π_{proxy} assigns zero probability to tokens with $\pi_{\boldsymbol{\theta}} \leq \tau$ and renormalizes the probability mass to tokens with $\pi_{\boldsymbol{\theta}} > \tau$.

Probability Renormalization As shown in Figure 2, the proxy distribution π_{proxy} zeros out filtered tokens and renormalizes the remaining mass:

$$\pi_{\text{proxy}}(o|\cdot) = \begin{cases} \frac{\pi_{\boldsymbol{\theta}}(o|\cdot)}{\sum_{o' \text{ s.t. } \pi_{\boldsymbol{\theta}}(o'|\cdot)>\tau} \pi_{\boldsymbol{\theta}}(o'|\cdot)} & \text{if } \pi_{\boldsymbol{\theta}}(o|\cdot) > \tau \\ 0 & \text{otherwise} \end{cases}. \tag{4}$$

4.2 Low-probability Regularization Objective

We integrate the Lp-Reg penalty into the GRPO framework as a selective regularization term:

$$\begin{aligned} J_{\text{Lp-Reg}}(\boldsymbol{\theta}) =& \mathbb{E}_{train}\Bigg[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\Big[\text{clip}(r_{i,t},0,U)A_{i,t} \\ & - \beta \cdot \mathbb{I}\left[\pi_{\theta,t} < \delta_\rho^{\mathcal{B}} \wedge \pi_{\text{proxy},t} > 0 \wedge A_{i,t} < 0\right] \\ & \cdot \mathcal{D}_{\text{KL}}\left(\pi_{\text{proxy},t} \| \pi_{\theta,t}\right)\Big]\Bigg], \end{aligned} \tag{5}$$

where $\mathbb{E}_{train}$ denotes expectation over batch $\mathcal{B}$, queries, and samples $\{o_i\}_{i=1}^G \sim \pi_{\boldsymbol{\theta}_{\text{old}}}$. The first term is the GRPO objective, modified with one-sided clipping (no lower bound, large upper bound U) to preserve high-variance exploration.

The second term is the Lp-Reg penalty, activated by $\mathbb{I}[\cdot]$ only when three conditions are met: (1) **Low-probability**: sampling probability $\pi_{\boldsymbol{\theta}}$ falls below a dynamic batch-percentile threshold $\delta_\rho^{\mathcal{B}}$; (2) **Non-noise**: the token is valid in the proxy distribution ($\pi_{\text{proxy}} > 0$); and (3) **Negative Samples**: the token receives a negative advantage ($A_{i,t} < 0$). This ensures regularization selectively targets valuable tokens under threat of elimination, without interfering with positive updates. We provide data-driven guidelines for selecting hyperparameters κ and ρ in Appendix E.1, with sensitivity analysis in Appendix E.2.

We use the forward KL divergence, $\mathcal{D}_{\text{KL}}(\pi_{\text{proxy}} \| \pi_{\boldsymbol{\theta}})$ as the regularization function, supported by Appendix D. It imposes a significant penalty when $\pi_{\boldsymbol{\theta}}(o|\cdot)$ approaches zero for a token o with non-zero probability in π_{proxy}, providing a targeted penalty against token elimination without forcing the policy to strictly match the heuristic proxy distribution.

5 Experiments

5.1 Experimental Setup

Baselines We compare Lp-Reg against a comprehensive suite of baselines, including **GRPO** (Shao et al., 2024a) and several state-of-the-art methods designed to enhance exploration through entropy control. These include **GRPO + Entropy Loss**, **Clip-Higher** (Yu et al., 2025), **Selective High-Entropy Training (80/20)** (Wang et al., 2025a), **KL-Cov** (Cui et al., 2025), and **GSPO** (Zheng et al., 2025). Detailed are provided in Appendix B.1.

Implementation Details All experiments are conducted within the `veRL` (Sheng et al., 2024) framework. We train all baselines for an extended duration ranging from 600 to 1,000 steps, ensuring a long-horizon training that facilitates comprehensive comparisons and robust conclusions. Complete details are listed in Appendix B.2.

Evaluation We assess model performance across eight benchmarks spanning mathematical, coding, and scientific domains, as summarized in Table 1. Further details can be found in Appendix B.3.

5.2 Main Results

Stable Long-Horizon Training. As shown in Figure 3, Lp-Reg enables stable RL training for 3,000 steps, totaling 81,204 GPU-hours on Qwen2.5-32B-Base. Figure 4 and Table 2 further confirm that Lp-Reg achieves state-of-the-art performance across five mathematical reasoning benchmarks on both 14B and 32B scales. On

Domains	Training datasets	Evaluation Benchmarks
Math	Dapo-Math-17K (Yu et al., 2025)	AIME24 (MAA), AIME25 (MAA), MATH-500 (Hendrycks et al., 2021), OlympiadBench (He et al., 2024a), Minerva Math (Lewkowycz et al., 2022)
Code	AReaL-boba-2-RL-Code (Fu et al., 2025a)	LCB-v5, LCB-v6 (Jain et al., 2024)
Science	SCP-116k (Lu et al., 2025)	GPQA-diamond (Rein et al., 2024)

Table 1: Overview of training datasets and evaluation benchmarks across Math, Code, and Science domains.

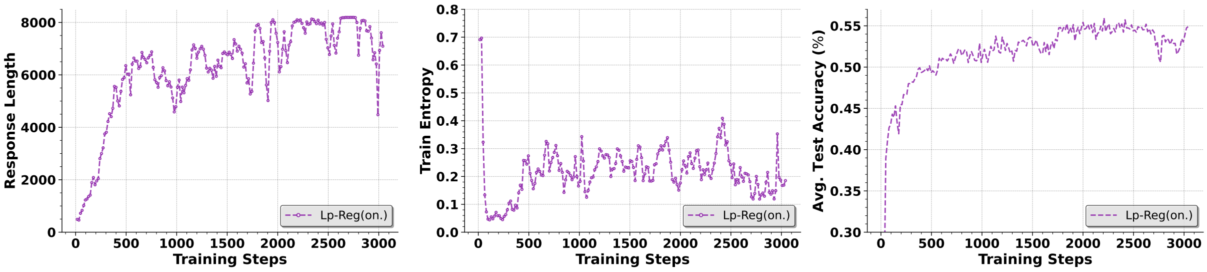


Figure 3: **Stable training over 3,000 training steps, totaling 81,204 GPU-hours**, for Lp-Reg (on-policy) on the Qwen2.5-32B-Base model.

Qwen3-14B, on-policy Lp-Reg sets a new benchmark with an average accuracy of 60.17%, surpassing the next best method by 2.66% (+4.63% relatively). This advantage grows with stronger base models (Qwen3-14B-Base > Qwen2.5-32B-Base), which we hypothesize provide richer *reasoning sparks* for Lp-Reg to leverage. The reported scores reflect the checkpoint with the highest average accuracy. A per-benchmark peak score analysis in Appendix C.2 reveals an even more substantial performance superiority of Lp-Reg, showcasing its full potential on individual tasks.

Superiority of On-Policy Training Our experiments consistently demonstrate the stability advantage of on-policy over off-policy methods. Off-policy methods like Clip-Higher rely on importance sampling clipping, leading to instability. While competitive on Qwen2.5-32B, Clip-Higher's performance degrades on Qwen3-14B. In contrast, on-policy training avoids distribution shifts caused by mismatched policies. Lp-Reg's policy-intrinsic regularization ensures robustness in both settings, distinguishing it from competitors heavily reliant on off-policy clipping.

Distinct Entropy Signature Beyond raw performance, Lp-Reg exhibits a unique entropy trajectory indicative of a healthy exploration-exploitation balance. Unlike methods that induce artificial entropy increases (e.g., Clip-Higher, Figure 4), Lp-Reg facilitates a multi-phase dynamic: entropy initially decreases (learning core patterns), then increases (fostering exploration), and finally stabilizes. This adaptive behavior stems from our confidence-aware regularization, which selectively protects *reasoning sparks* without amplifying high-entropy noise.

Method	AIME24	AIME25	Math-500	Minerva	Olympiad	Avg.
Qwen2.5-32B-Base (800 training steps)						
GRPO (off.)	30.63	22.29	88.00	41.18	54.37	47.29
GSPO (off.)	33.33	22.29	87.60	**48.53**	55.56	49.46
Clip-Higher (off.)	**38.33**	**29.79**	87.60	45.22	56.44	51.48
KL-Cov (off.)	35.62	27.50	87.40	44.49	55.11	50.02
80/20	38.12	28.75	87.00	45.22	58.37	51.49
Lp-Reg (off.)	37.71	24.58	**90.20**	40.81	59.70	50.60
GRPO (on.)	28.54	22.50	86.60	44.85	60.30	48.56
Entropy Loss (on.)	3.75	1.88	60.80	27.94	22.22	23.32
80/20 (on.)	32.50	28.54	89.40	45.59	57.63	50.73
Lp-Reg (on.)	38.12	27.08	90.00	46.32	**61.19**	**52.54**
Qwen3-14B-Base (1, 000 training steps)						
GRPO (off.)	34.38	27.08	89.20	49.26	55.70	51.13
GSPO (off.)	41.46	34.58	88.60	**50.74**	59.85	55.05
Clip-Higher (off.)	41.67	32.71	**95.00**	47.43	64.00	56.16
KL-Cov (off.)	49.17	34.79	93.00	47.43	62.07	57.29
80/20 (off.)	43.96	34.58	91.80	48.16	60.89	55.88
Lp-Reg (off.)	46.25	34.17	92.40	48.16	64.44	57.08
GRPO (on.)	46.04	34.38	93.00	48.53	65.19	57.43
Entropy Loss (on.)	37.29	25.21	88.20	46.32	54.96	50.40
80/20 (on.)	47.29	32.50	91.60	50.37	65.78	57.51
Lp-Reg (on.)	**50.83**	**37.92**	94.40	49.26	**68.44**	**60.17**

Table 2: Main results on five math benchmarks. We denote off-policy and on-policy settings as off. and on. Scores represent the checkpoint with the highest average accuracy. **Bold** and underlined indicate best and second-best results. See Appendix C.2 for peak and pass@k scores.

5.3 Extended Models and Domains

To further validate the generalizability of Lp-Reg, we extend our evaluation across different model families and domains.

Extension to Llama3 We first examine architectural robustness using Llama3-OctoThinker-

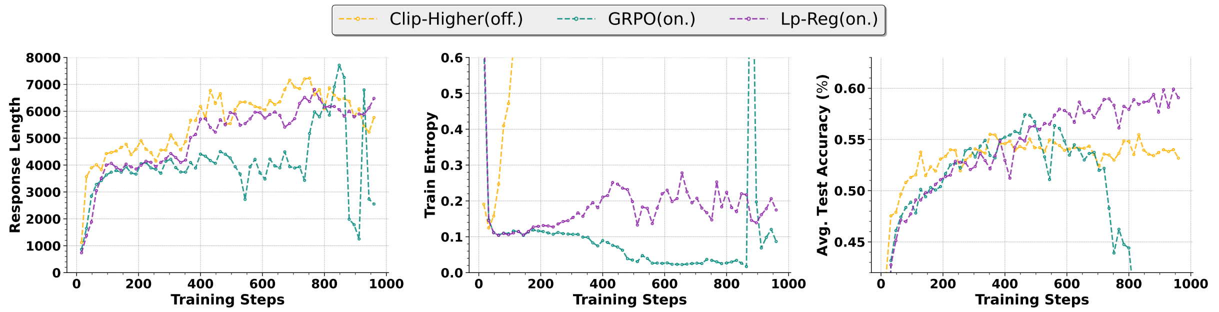


Figure 4: Training dynamics on the Qwen3-14B-Base model. To best illustrate the performance differences, we compare the top-performing methods. Lp-Reg demonstrates superior and stable performance. Full training dynamics are available in Figure 10.

8B (Wang et al., 2025b), despite the Llama3 series' known challenges for RLVR (Gandhi et al., 2025). As presented in Table 3, Lp-Reg consistently outperforms baselines. Specifically, Lp-Reg(on.) achieves an absolute gain of 2.88% (+7.98% relatively) over GRPO, while Lp-Reg(off.) surpasses the nearest off-policy competitor by 3.62%. These results align with our findings observed on Qwen models, confirming the robustness of Lp-Reg across model architectures.

Extension on Science and Code We further evaluate performance on code generation (LCB-v5/v6 (Jain et al., 2024)) and science (GPQA-diamond (Rein et al., 2024)) benchmarks. As shown in Table 4, Lp-Reg achieves top-tier performance in both domains. In code generation, it secures the highest average scores, outperforming the next-best method by a relative margin of 3.06%. Similarly, on the challenging GPQA-diamond benchmark, both on-policy and off-policy Lp-Reg variants surpass their respective baselines (GRPO and KL-Cov) with a relative improvement of at least 4.62%. This consistency across mathematics, science, and code demonstrates the broad applicability of our approach.

Method	AIME24	AIME25	Math-500	Minerva	Olympiad	Avg.
Llama3-OctoThinker-8B (600 training steps)						
GRPO (off.)	4.38	4.58	60.00	26.47	25.93	24.27
GSPO (off.)	4.58	2.50	58.80	29.41	25.33	24.13
Clip-Higher (off.)	11.88	3.75	61.80	23.16	26.96	25.51
KL-Cov (off.)	7.71	4.58	55.00	23.16	22.96	22.68
80/20 (off.)	10.00	7.50	59.00	18.75	27.56	24.56
Lp-Reg (off.)	9.58	8.33	68.80	27.21	31.70	29.13
GRPO (on.)	15.42	12.50	76.20	33.09	43.26	36.09
80/20 (on.)	11.67	4.17	73.60	27.21	37.48	30.82
Lp-Reg (on.)	**18.33**	**16.88**	**79.00**	**35.29**	**45.33**	**38.97**

Table 3: Main results on five mathematical reasoning benchmarks on **Llama3-OctoThinker-8B**. Formatting conventions follow Table 2.

Methods	Code LCB-v5	Code LCB-v6	Code Avg.	Science GPQA-diamond
Qwen3-8B-Base (600 training steps)				
GRPO (off.)	27.32	27.43	27.38	39.71
GSPO (off.)	28.29	26.57	27.43	47.16
Clip-Higher (off.)	27.10	27.57	27.34	48.61
KL-Cov (off.)	28.74	27.43	28.09	49.18
80/20 (off.)	26.57	27.64	27.11	45.90
Lp-Reg(off.)	**29.57**	27.57	28.57	51.77
GRPO (on.)	27.47	27.86	27.67	50.63
80/20 (on.)	28.29	27.36	27.83	48.42
Lp-Reg(on.)	28.89	**29.00**	**28.95**	**52.97**

Table 4: Results on science and code on Qwen3-8B-Base. Formatting conventions follow Table 2.

5.4 Ablation Study

We validate the core design choices of Lp-Reg in Equation 5. Below, we demonstrate the criticality of low-probability tokens and filtering noise. Due to space constraints, analyses for **selection of negative samples** and the **choice of KL divergence formulation** are detailed in Appendix D.

Importance of Low-Probability Token We verify the superiority of targeting low-probability tokens over the conventional approach of targeting high entropy. Figure 5 compares Lp-Reg (lowest 1% probability) against regularizing the highest 1% entropy tokens. The entropy-based approach fails to improve performance and leads to entropy collapse. This confirms that high entropy is a poor proxy for exploration, as it predominantly targets function words rather than the meaningful, low-probability exploratory tokens essential for reasoning breakthroughs.

Necessity of Noise Filtering Figure 6 demonstrates that removing the noise filter (Lp-Reg w/o τ) causes catastrophic performance collapse and entropy explosion. This confirms that filtering the extreme tail of the distribution is critical to avoid re-

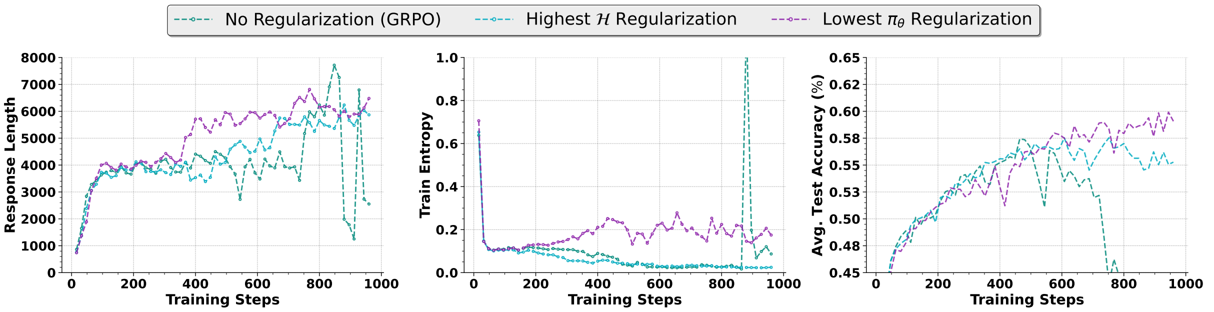


Figure 5: Ablation study comparing **low-probability token regularization versus high-entropy token regularization** for Lp-Reg (on-policy) on the Qwen3-14B-Base model.

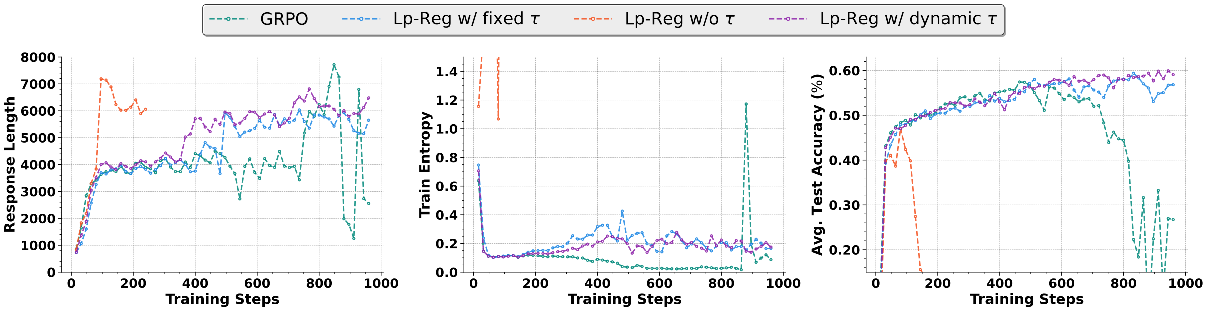


Figure 6: Ablation studies for **noise filtering** of Lp-Reg (on-policy) on the Qwen3-14B-Base model. The results confirm that targeting our noise filtering threshold τ is critical for stable performance. The adaptiveness of the min-p threshold is also shown to be beneficial over a fixed one.

inforcing irrelevant noise. Regarding the threshold strategy, while a fixed threshold outperforms standard GRPO, the dynamic min-p threshold yields superior results (Figure 6). Its adaptiveness allows for a more robust estimation of noise across varying contexts, better preserving *reasoning sparks*.

6 Analysis

This section investigates the mechanisms underlying Lp-Reg through three key questions: (1) **Can we empirically distinguish *reasoning sparks* from noise in the low-probability region?** (Sec. 6.1); (2) **Why does low-probability regularization outperform entropy-based methods?** We provide empirical evidence (Sec. 6.2) supported by theoretical proofs (App. G.2); and (3) **How does Lp-Reg sustain exploration?** We analyze sampling distribution (Sec. 6.3 and App. G.1) and trajectory-level behavior (App. G.3). Additional comparisons and a case study are provided in Appendices G.4 and I, respectively.

6.1 Distinguishing Reasoning Sparks from Noise

Effective exploration requires distinguishing valuable *reasoning sparks* from destructive noise. To validate this feasibility, we analyze the training dynamics of tokens within the low-probability range (< 0.1). Figure 8 compares the average probability trajectories of meaningful exploratory tokens (e.g., *"wait," "perhaps"*) against irrelevant ones (e.g., *"cost," "fine"*).

We observe a consistent probabilistic gap: exploratory tokens maintain persistently higher probabilities than irrelevant noise across training stages. This distinction, likely stemming from the model's intrinsic confidence (Nguyen et al., 2025; Xu et al., 2025; Fu et al., 2025b), provides the empirical justification for the threshold τ in Lp-Reg (Section 4.1). By filtering out the lowest-probability noise, τ enables targeted regularization that preserves viable reasoning sparks without destabilizing training.

6.2 The Probability-Entropy Gap

Figure 9 contrasts the top 1% lowest probability and highest entropy tokens. Low-probability tokens frequently feature meaningful exploratory markers (e.g., *"But," "Let," "Perhaps"*) that signal reasoning shifts, whereas high-entropy tokens are dominated by non-informative function words (e.g., *"times," "sqrt"*). This indicates that entropy-based regularization often fails by conflating noise

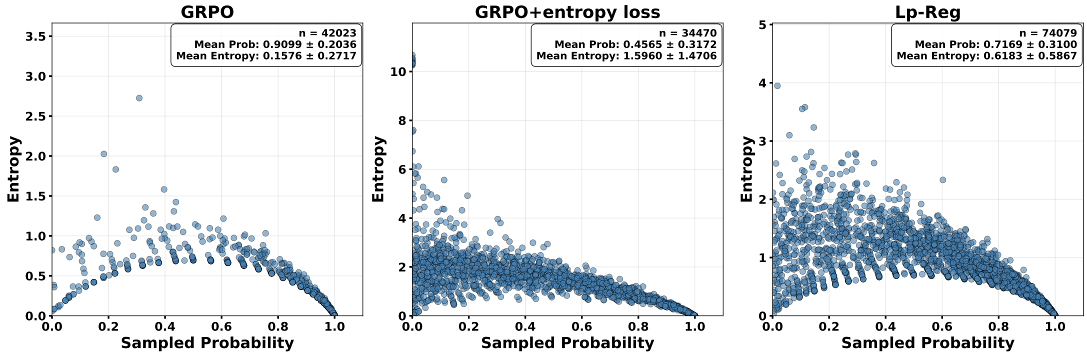


Figure 7: Probability–Entropy scatter plots of explorative tokens, displaying a random sample of 5% of all data points. See Figure 25 for individual token details.

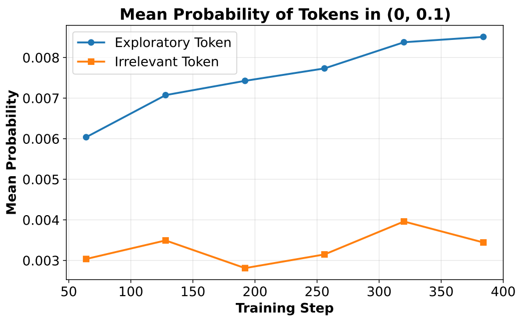


Figure 8: Probabilistic distinction between exploratory and irrelevant tokens across training steps in standard GRPO training.

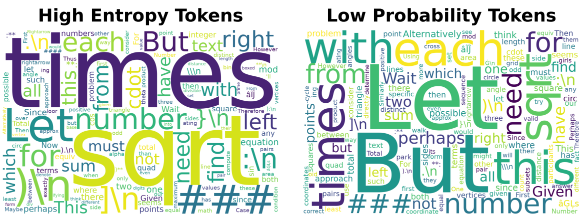


Figure 9: The word cloud statistics.

with exploration. As theoretically proven in Appendix G.2, entropy methods inherently miss exploration in low-entropy regions. However, since low-probability tokens also contain destabilizing artifacts (e.g., \n), Lp-Reg employs a threshold τ to filter them. Ablation studies (Section 5.4) confirm that this selective exclusion is critical for training stability and performance.

6.3 Probability-Entropy Distribution of Explorative Tokens

Figure 7 analyzes the probability–entropy distributions of key explorative tokens (e.g., *"but," "wait," "perhaps"*). In standard GRPO, these tokens cluster in low-entropy, high-probability regions, reducing them to deterministic patterns rather than uncertain exploration. Conversely, adding entropy loss disperses tokens uncontrollably into high-entropy zones, boosting diversity superficially without meaningful integration into the reasoning.

In contrast, Lp-Reg maintains a balanced dynamic where explorative tokens span a broad range of entropy values. This prevents probability collapse under negative feedback while ensuring tokens remain informative. Consequently, markers like *"wait"* remain viable options throughout training, allowing the model to initiate alternative reasoning paths rather than overfitting to fixed patterns. Appendix G.3 further reveals the trajectory-level analysis.

7 Conclusion

In this work, we investigated the exploration collapse in Reinforcement Learning with Verifiable Rewards, identifying a key mechanism driving this failure: the systematic elimination of valuable, low-probability *reasoning sparks*. To address this, we introduced Low-probability Regularization (Lp-Reg), a method designed to selectively preserve these crucial exploratory pathways. Lp-Reg leverages the insight that *reasoning sparks* often exhibit higher relative probabilities than meaningless noise in their immediate predictive context. By filtering out the noise tokens and regularizing the policy towards the remainder, our method effectively protects valuable sparks from being extinguished. This focus on exploration quality enables stable on-policy training for over 3,000 steps (81, 204 GPU-hours) where baselines collapse. Validated across over 300,000 cumulative hours, Lp-Reg achieves state-of-the-art performance across diverse models and domains, yielding relative improvements of 3.06% to 7.98%.

Limitations

Although Lp-Reg demonstrates excellent performance across various model scales, architectures, and domains, it still has several limitations:

(1) Our study investigates exploration dynamics exclusively within domains where ground-truth labels provide deterministic signals (e.g., math, code, science). We did not evaluate Lp-Reg in standard RLHF scenarios involving open-ended generation (e.g., creative writing, chat), where rewards are derived from preference models rather than symbolic verification. As recent studies demonstrate that verifiable environments are the primary drivers for eliciting complex reasoning capabilities and "System 2" behaviors, whereas open-ended tasks with subjective rewards often focus on stylistic alignment and show limited efficacy in scaling fundamental reasoning (OpenAI et al., 2024; DeepSeek-AI et al., 2025). Future work may explore how our findings translate to preference-based optimization.

(2) Our experiments mainly focused on English corpora, and while our proposed method is general, we did not explore its performance on multilingual corpora. We leave the detailed analysis of multilingual datasets for future work.

References

Shivam Agarwal, Zimin Zhang, Lifan Yuan, Jiawei Han, and Hao Peng. 2025. The unreasonable effectiveness of entropy minimization in llm reasoning. *Preprint*, arXiv:2505.15134.

Taubenfeld Amir, Sheffer Tom, Ofek Eran, Feder Amir, Goldstein Ariel, Gekhman Zorik, and Yona Gal. 2025. Confidence improves self-consistency in llms. *arXiv preprint arXiv:2502.06233*.

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

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

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Wei Fu, Jiaxuan Gao, Xujie Shen, Chen Zhu, Zhiyu Mei, Chuyi He, Shusheng Xu, Guo Wei, Jun Mei, Jiashu Wang, Tongkai Yang, Binhang Yuan, and Yi Wu. 2025a. Areal: A large-scale asynchronous reinforcement learning system for language reasoning. *Preprint*, arXiv:2505.24298.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025b. Deep think with confidence. *Preprint*, arXiv:2508.15260.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *Preprint*, arXiv:2503.01307.

Zitian Gao, Lynx Chen, Haoming Luo, Joey Zhou, and Bryan Dai. 2025. One-shot entropy minimization. *Preprint*, arXiv:2505.20282.

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. 2024a. 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, Bangkok, Thailand. Association for Computational Linguistics.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024b. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *Preprint*, arXiv:2402.14008.

Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, Siyuan Li, Liang Zeng, Tianwen Wei, Cheng Cheng, Bo An, Yang Liu, and Yahui Zhou. 2025. Skywork open reasoner 1 technical report. *Preprint*, arXiv:2505.22312.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Xiao Hu, Xingyu Lu, Liyuan Mao, YiFan Zhang, Tianke Zhang, Bin Wen, Fan Yang, Tingting Gao, and Guorui Zhou. 2025. Why distillation can outperform zero-rl: The role of flexible reasoning. *Preprint*, arXiv:2505.21067.

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

Aitor Lewkowycz, Anders Johan Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Venkatesh Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*.

Li Loka, Chen Zhenhao, Chen Guangyi, Zhang Yixuan, Su Yusheng, Xing Eric, and Zhang Kun. 2024. Confidence matters: Revisiting intrinsic self-correction capabilities of large language models. *arXiv preprint arXiv:2402.12563*.

Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi. 2025. Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated extraction in the higher education science domain. *Preprint*, arXiv:2501.15587.

MAA. n.d. American invitational mathematics examination (AIME). Mathematics Competition Series.

MiniMax, :, Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, Chengjun Xiao, Chengyu Du, Chi Zhang, Chu Qiao, Chunhao Zhang, Chunhui Du, Congchao Guo, and 109 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *Preprint*, arXiv:2506.13585.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Kevin Murphy. 2024. Reinforcement learning: an overview. *arXiv preprint arXiv:2412.05265*.

Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. 2025. Turning up the heat: Min-p sampling for creative and coherent llm outputs. *Preprint*, arXiv:2407.01082.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. 2025. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in llm reasoning. *Preprint*, arXiv:2506.02867.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Kadavath Saurav, Conerly Tom, Askell Amanda, Henighan Tom, Drain Dawn, Perez Ethan, Schiefer Nicholas, Hatfield-Dodds Zac, DasSarma Nova, Tran-Johnson Eli, Johnston Scott, El-Showk Sheer, Jones Andy, Elhage Nelson, Hume Tristan, Chen Anna, Bai Yuntao, Bowman Sam, Fort Stanislav, and 17 others. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024a. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*.

Zhenpeng Su, Leiyu Pan, Xue Bai, Dening Liu, Guanting Dong, Jiaming Huang, Wenping Hu, Fuzheng Zhang, Kun Gai, and Guorui Zhou. 2025. Klear-reasoner: Advancing reasoning capability via gradient-preserving clipping policy optimization. *Preprint*, arXiv:2508.07629.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 77 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, Yuqiong Liu, An Yang, Andrew Zhao, Yang Yue, Shiji Song, Bowen Yu, Gao Huang, and Junyang Lin. 2025a. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Preprint*, arXiv:2506.01939.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025b. Octothinker: Mid-training incentivizes reinforcement learning scaling. *Preprint*, arXiv:2506.20512.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Zenan Xu, Zexuan Qiu, Guanhua Huang, Kun Li, Siheng Li, Chenchen Zhang, Kejiao Li, Qi Yi, Yuhao Jiang, Bo Zhou, Fengzong Lian, and Zhanhui Kang. 2025. Adaptive termination for multi-round parallel reasoning: An universal semantic entropy-guided framework. *Preprint*, arXiv:2507.06829.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

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

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, Xiangpeng Wei, Xiangyu Yu, Gaohong Liu, Juncai Liu, Lingjun Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Chi Zhang, and 8 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *Preprint*, arXiv:2504.05118.

Yuzhong Zhao, Yue Liu, Junpeng Liu, Jingye Chen, Xun Wu, Yaru Hao, Tengchao Lv, Shaohan Huang, Lei Cui, Qixiang Ye, Fang Wan, and Furu Wei. 2025. Geometric-mean policy optimization. *Preprint*, arXiv:2507.20673.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *Preprint*, arXiv:2507.18071.

Ömer Veysel Çağatan and Barış Akgün. 2025. Failure modes of maximum entropy rlhf. *Preprint*, arXiv:2509.20265.

Appendix

A AI Assistants Usage Statement **11**

B Experimental Setup **12**
B.1 Baselines 12
B.2 Training Settings 12
B.3 Evaluation 12

C Further Experiments **13**
C.1 Details of Training Dynamics 13
C.2 Per-Benchmark Peak Performance Analysis 13

D Additional Ablation Study **13**

E Hyperparameter Analysis **15**
E.1 Guidelines for Hyperparameter Selection 15
E.2 Hyperparameter Sensitivity Analysis . 16

F Computational Overhead Analysis **16**

G Further Analysis on Low-Probability and High-Entropy Tokens **17**
G.1 Frequency Dynamics of Exploratory Tokens 17
G.2 Theoretical Discussion on Low-Probability vs. High-Entropy Tokens 17
G.3 Trajectory-Level Token Analysis . . . 20
G.4 Comparison with Concurrent Work (Lopti) 21

H Details of Analysis **22**
H.1 Details of Sampling Probability Density 22
H.2 Details of Probability-Entropy Distribution 23
H.3 Training Dynamics of Regularized Token 23

I Case Study **23**

A AI Assistants Usage Statement

We disclose the use of a large language model (LLM) as a general-purpose writing assistant during the preparation of this manuscript. The LLM's role was strictly limited to improving the clarity, grammar, and readability of our author-written text, such as spell-checking and rephrasing sentences for better flow. Crucially, the LLM did not contribute to any of the core scientific aspects of this work, including research ideation, experimental design, data analysis, or the generation of novel insights. The authors have carefully reviewed all LLM-modified text and take full responsibility for the intellectual substance and final content of this paper.

B Experimental Setup

B.1 Baselines

We compare Lp-Reg against a suite of strong baselines, including a foundational algorithm and several state-of-the-art methods designed to enhance exploration through entropy control. Our primary baseline is **GRPO** (Shao et al., 2024a), a value-free policy optimization algorithm that employs group-relative advantage estimation, making it a common choice for RLVR. To represent classical entropy regularization methods, we implement **GRPO + Entropy Loss**, which directly incorporates the principles of Maximum Entropy RL by adding a policy entropy bonus to the GRPO objective function. We also compare against several advanced methods: **Clip-Higher** (Yu et al., 2025), a core component of DAPO that encourages higher entropy by using an asymmetric clipping range in the PPO objective; **Selective High-Entropy Training (80/20)** (Wang et al., 2025a), a method that restricts policy gradient updates to only the top 20% of tokens with the highest generation entropy; **KL-Cov** (Cui et al., 2025), which prevents entropy collapse by applying a selective KL-divergence penalty to tokens with the highest covariance between their log probabilities and advantages; and **GSPO** (Zheng et al., 2025), which modifies the clipping mechanism to operate at the sequence level to promote higher training entropy.

B.2 Training Settings

All experiments are conducted within the `veRL` (0.3.0) (Sheng et al., 2024) framework, encompassing **over 300,000 cumulative GPU-hours in total**. To ensure equitable evaluation despite varying convergence speeds, we train models until performance saturation or reach the maximum training steps. For our main comparisons (Section 5.2), this entails approximately 1,000 steps for Qwen3-14B-Base ($\sim$8,000 GPU-hours on 32 NVIDIA H20s) and 800 steps for Qwen2.5-32B-Base ($\sim$16,000 GPU-hours on 64 NVIDIA H20s). Extended generalization experiments (Section 5.3) consume between 1,000 and 4,000 GPU-hours per run. Conversely, for baselines exhibiting training collapse—defined as a performance drop exceeding 10 percentage points—we implement early stopping to conserve resources. Finally, to strictly assess whether Lp-Reg sustains long-term exploration, we conduct an extended stability test on Qwen2.5-32B for 3,000 steps, consuming 81,204 GPU-hours.

For the reinforcement learning from verifier rewards (RLVR) phase, models are trained with a maximum response length of $8,192$ tokens.We use a global batch size of 256. For off-policy methods, we use a mini-batch size of 32, resulting in 8 gradient updates per rollout. It should be noted that in our experimental results, "step" refers to the rollout step for both on-policy and off-policy methods. To ensure a fair comparison, a “step” in our experimental results consistently refers to a single rollout for all methods. A constant learning rate of 1×10^{-6} is applied without a warmup schedule. We set the group number as 8 for all GRPO-based methods. To ensure numerical stability, we set the policy gradient’s clipping by setting the upper bound of the importance sampling ratio to $U = 10$. For our proposed Lp-Reg, which uses the min-p threshold, we set the probability percentile threshold ρ to 0.5% for Qwen2.5-32B, Qwen3-8B-Base, Llama3-OctoThinker-8B and 1% for Qwen3-14B-Base, the KL regularization coefficient β to 1.0, and the min-p ratio κ to 0.02. The proxy distribution, π_{proxy}, is constructed from the data-generating policy ($\pi_{\theta_{\text{old}}}$ in the off-policy setting and the current policy $\pi_{\boldsymbol{\theta}}$ in the on-policy setting). For all baseline methods, we adopt the hyperparameters specified in their original public implementations to ensure a faithful reproduction. Specifically for the GRPO + Entropy Loss baseline, we set the entropy coefficient to 0.002 within the `verl` framework.

B.3 Evaluation

For evaluation, we assess model performance across eight reasoning benchmarks in Table 1, spanning various domains including math, code, and science. For small benchmarks, we use sampled decoding with a temperature of 0.6 to obtain a robust performance estimate, generating 16 independent responses per problem for AIME24 and AIME25, and 8 for GPQA-diamond, LCB-v5, and LCB-v6. For larger benchmarks like MATH-500, OlympiadBench, and Minerva, we utilize greedy decoding.

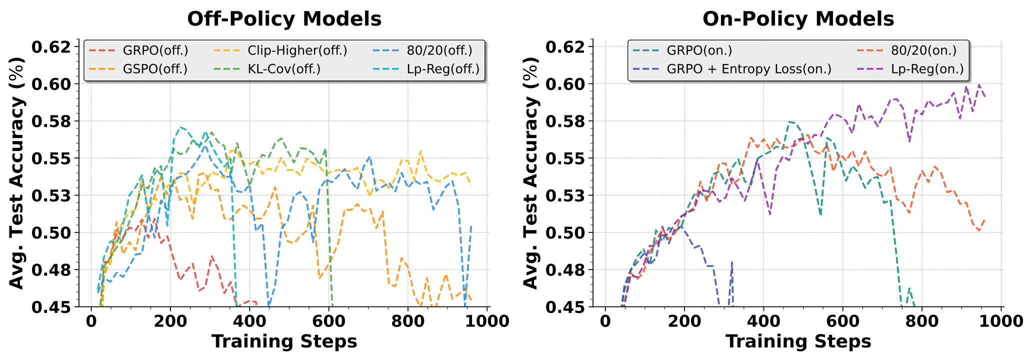


Figure 10: Training dynamics on the Qwen3-14B-Base model. On-policy training exhibits better training stability and testing performance compared to off-policy training.

C Further Experiments

C.1 Details of Training Dynamics

Figure 10 shows the superiority of on-policy training over off-policy methods. This is due to the inherent stability of on-policy updates, which avoid distribution shifts caused by mismatched data-sampling and training policies. Off-policy methods, such as Clip-Higher, often rely on importance sampling clipping, leading to instability. In contrast, Lp-Reg's self-contained, policy-intrinsic regularization ensures its effectiveness in both on-policy and off-policy settings, unlike competing methods that are heavily reliant on off-policy importance sampling.

C.2 Per-Benchmark Peak Performance Analysis

Peak Score In Section 5.2, we reported performance based on a single checkpoint selected for the best average test accuracy across five mathematical benchmarks. However, aggregating results can obscure the model's peak potential on individual tasks. To address this, we present the per-benchmark best scores in Table 5. As shown, our on-policy Lp-Reg achieves the highest peak scores on all benchmarks with the exception of Minerva. Even on Minerva, where Lp-Reg(on.) is not the best performer, the gap is marginal: on Qwen2.5-32B-Base, it trails the highest score by only 1.47 percentage points (a relative difference of -2.96%). Conversely, the gains on other benchmarks are substantial, particularly on the most challenging reasoning tasks such as AIME24, AIME25, and Olympiad Bench. Notably, on Qwen2.5-32B-Base, Lp-Reg(on.) outperforms the second-best method, 80/20(off.), by a relative margin of 10.78% on AIME24. Similarly, on Qwen3-14B-Base, it achieves a 10.77% relative improvement on AIME25. These significant improvements on the hardest benchmarks underscore the effectiveness of Lp-Reg in solving complex reasoning problems.

Pass@K Score We further evaluate the exploration capability of our method by comparing the best pass@k rates. As detailed in Table 6, Lp-Reg(on.) consistently achieves the highest pass@k scores on both AIME24 and AIME25 across both model scales, often by a wide margin. For the Qwen2.5-32B model, Lp-Reg(on.) demonstrates a minimum relative improvement of 5.97% in pass@k metrics on AIME24. Furthermore, on the Qwen3-14B model, it shows impressive gains on AIME25, achieving relative improvements ranging from 7.81% to 9.33%. These robust pass@k results provide strong evidence that Lp-Reg effectively sustains meaningful exploration throughout long-horizon RLVR training, resulting in more diverse and successful reasoning rollouts.

D Additional Ablation Study

In this section, we present the ablation studies regarding the **Effect of Negative Samples** and the **Forward vs. Reverse KL Divergence**, complementing the main results in Section 5.4.

Effect of Negative Samples. Figure 11 analyzes the impact of regularizing negative versus positive samples. While applying Lp-Reg to *all* samples improves over GRPO, restricting regularization to **negative samples only** achieves the fastest learning rate. This is because vulnerable exploratory tokens are most prevalent in failed (negative) trajectories. Protecting them is sufficient to sustain

Methods	AIME24	AIME25	Math-500	Minerva	Olympiad Bench
Qwen2.5-32B-Base					
GRPO (Shao et al., 2024a) (off.)	30.63	23.75	88.00	46.69	56.00
GSPO (Zheng et al., 2025) (off.)	36.88	26.46	89.00	**49.63**	56.30
Clip-Higher (Yu et al., 2025) (off.)	39.58	**32.71**	88.80	<u>48.90</u>	58.22
KL-Cov (Cui et al., 2025) (off.)	36.88	29.38	89.00	48.16	56.89
80/20 (Wang et al., 2025a) (off.)	<u>40.62</u>	30.21	<u>90.80</u>	48.16	58.81
Lp-Reg (off.)	37.71	26.88	90.20	43.38	60.15
GRPO (Shao et al., 2024a) (on.)	32.50	23.54	88.80	47.79	<u>60.30</u>
GRPO + Entropy Loss (on.)	3.75	2.50	60.80	32.72	22.22
80/20 (Wang et al., 2025a) (on.)	35.00	28.54	90.00	47.79	58.81
Lp-Reg (on.)	**45.00**$_{+10.78\%}$	**32.71**$_{+0.00\%}$	**93.00**$_{+2.42\%}$	48.16$_{-2.96\%}$	**64.15**$_{+6.38\%}$
Qwen3-14B-Base					
GRPO (Shao et al., 2024a) (off.)	35.83	27.71	91.00	48.16	59.56
GSPO (Zheng et al., 2025) (off.)	43.75	<u>36.67</u>	91.60	50.74	61.04
Clip-Higher (Yu et al., 2025) (off.)	44.79	33.75	**95.00**	49.63	65.19
KL-Cov (Cui et al., 2025) (off.)	<u>49.38</u>	35.83	94.20	**51.84**	64.44
80/20 (Wang et al., 2025a) (off.)	44.17	34.58	92.80	50.37	62.81
Lp-Reg (off.)	48.75	34.79	94.40	49.63	<u>65.78</u>
GRPO (Shao et al., 2024a) (on.)	46.04	35.42	93.80	50.37	65.63
GRPO + Entropy Loss (on.)	37.29	28.54	90.60	48.53	57.93
80/20 (Wang et al., 2025a) (on.)	47.29	35.00	94.00	50.37	<u>65.78</u>
Lp-Reg (on.)	**51.88**$_{+5.06\%}$	**40.62**$_{+10.77\%}$	**95.00**$_{+0.00\%}$	<u>51.47</u>$_{-0.71\%}$	**70.37**$_{+6.98\%}$

Table 5: Per-benchmark peak performance on five mathematical reasoning benchmarks. **Note that the scores reported represent the maximum value achieved for each specific benchmark individually; thus, scores within a single row may originate from different training checkpoints.** Best scores are **bolded** while second-best scores are <u>underlined</u>. The relative accuracy improvement of Lp-Reg over the next best method is indicated as a subscript.

Methods	AIME24			AIME25		
	Pass@2	Pass@4	Pass@8	Pass@2	Pass@4	Pass@8
Qwen2.5-32B-Base						
GRPO (Shao et al., 2024a) (off.)	40.06	49.87	58.10	29.11	36.25	44.75
GSPO (Zheng et al., 2025) (off.)	46.83	57.62	66.78	32.86	38.84	45.04
Clip-Higher (Yu et al., 2025) (off.)	48.11	57.80	68.32	<u>35.92</u>	<u>43.27</u>	<u>51.29</u>
KL-Cov (Cui et al., 2025) (off.)	46.89	55.94	64.61	35.44	41.60	49.39
80/20 (Wang et al., 2025a) (off.)	48.97	56.52	64.29	34.08	41.35	49.47
Lp-Reg (off.)	<u>49.69</u>	<u>59.75</u>	<u>69.21</u>	33.75	42.44	50.80
GRPO (Shao et al., 2024a) (on.)	42.08	51.74	61.95	29.19	35.83	43.20
GRPO + Entropy Loss (on.)	6.89	11.88	19.08	4.00	6.06	10.11
80/20 (Wang et al., 2025a) (on.)	45.06	55.33	63.40	35.28	41.64	48.54
Lp-Reg (on.)	**53.33**$_{+7.33\%}$	**63.50**$_{+6.28\%}$	**73.34**$_{+5.97\%}$	**38.28**$_{+6.57\%}$	**45.52**$_{+5.20\%}$	**53.12**$_{+3.57\%}$
Qwen3-14B-Base						
GRPO (Shao et al., 2024a) (off.)	45.31	54.81	64.09	34.14	41.00	48.29
GSPO (Zheng et al., 2025) (off.)	54.11	63.67	71.05	<u>44.39</u>	<u>51.97</u>	<u>59.67</u>
Clip-Higher (Yu et al., 2025) (off.)	56.00	<u>66.85</u>	<u>74.91</u>	40.19	48.31	57.35
KL-Cov (Cui et al., 2025) (off.)	<u>59.47</u>	66.84	74.52	42.22	49.98	58.65
80/20 (Wang et al., 2025a) (off.)	57.14	66.25	72.05	41.50	49.26	59.03
Lp-Reg (off.)	58.08	64.23	71.41	40.86	46.30	52.39
GRPO (Shao et al., 2024a) (on.)	55.19	63.93	70.48	42.86	49.90	57.85
GRPO + Entropy Loss (on.)	47.44	57.53	66.34	34.86	41.62	48.09
80/20 (Wang et al., 2025a) (on.)	56.97	63.66	71.66	42.28	49.76	57.39
Lp-Reg (on.)	**62.67**$_{+5.38\%}$	**71.04**$_{+6.27\%}$	**79.85**$_{+6.59\%}$	**48.53**$_{+9.33\%}$	**56.03**$_{+7.81\%}$	**64.95**$_{+8.85\%}$

Table 6: Per-benchmark peak pass@k results on the challenging AIME24 and AIME25 benchmarks. Similar to Table 5, **scores reported denote the peak pass@k rate for each metric separately, implying they may be derived from different checkpoints.** Best scores are **bolded** and second-best scores are <u>underlined</u>. The relative improvement of Lp-Reg is indicated as a subscript.

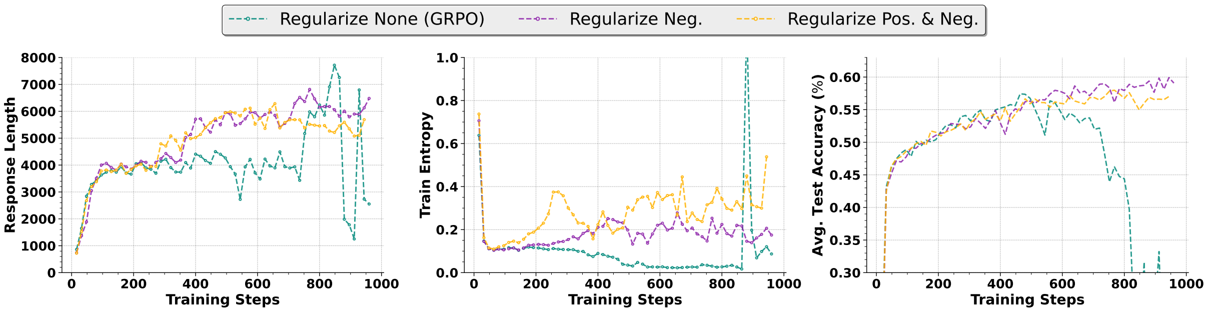


Figure 11: Ablation study comparing **positive sample regularization and negative sample regularization** for the Lp-Reg penalty (on-policy) on the Qwen3-14B-Base model. Negative sample regularization exhibits better performance.

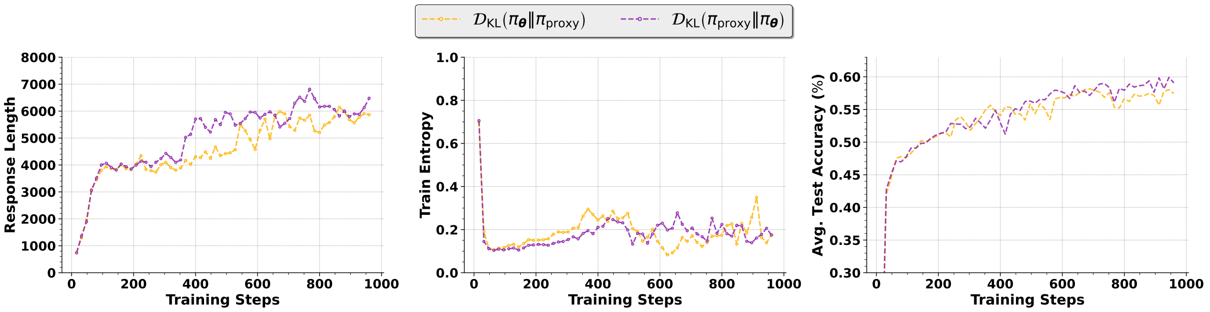


Figure 12: Ablation study comparing the **forward and reverse KL** formulations for the Lp-Reg penalty (on-policy) on the Qwen3-14B-Base model. The results demonstrate the superiority of the forward KL, which uses the heuristic proxy distribution as a soft guide, over the reverse KL, which forces a strict imitation.

exploration; extending penalties to positive samples introduces unnecessary noise into the gradient updates, slowing overall training progress without adding significant exploratory value.

Forward KL vs. Reverse KL. Figure 12 confirms that the Forward KL ($\mathcal{D}_{\mathrm{KL}}(\pi_{\mathrm{proxy}} \| \pi_{\boldsymbol{\theta}})$) significantly outperforms the Reverse KL. Since our proxy distribution π_{proxy} is a heuristic reference rather than a ground truth, Reverse KL hinders learning by forcing strict imitation of this heuristic. In contrast, Forward KL acts as a soft constraint: it prevents mode collapse by penalizing the elimination of plausible tokens, yet grants the policy sufficient freedom to explore and optimize beyond the proxy distribution.

E Hyperparameter Analysis

E.1 Guidelines for Hyperparameter Selection

In this section, we provide a data-driven guideline for selecting the initial values of the two core hyperparameters in Lp-Reg: the low-probability percentile ρ and the min-p ratio κ. Here, ρ determines the regularization threshold $\delta_\rho^{\mathcal{B}}$, while κ defines the noise filtering threshold $\tau = \kappa \cdot \max_{o' \in V} \pi_{\boldsymbol{\theta}}(o'|\cdot)$. Instead of heuristic guessing, we derive the rational ranges for these parameters by analyzing the training dynamics of the standard GRPO baseline.

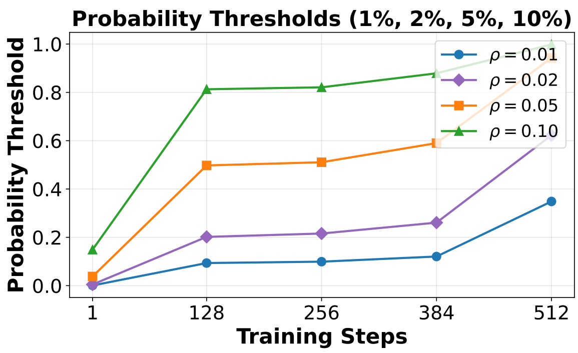


Figure 13: Evolution of probability thresholds for different percentiles (ρ) during standard GRPO training. The bottom 1% ($\rho = 0.01$) consistently captures the low-probability tail (< 0.1), whereas higher percentiles include high-confidence tokens.

Selection of ρ. Figure 13 visualizes the upper probability bound of tokens falling within the lowest ρ percentile during standard GRPO training. As illustrated, RLVR training causes the policy distribution to collapse, concentrating mass on high-probability tokens. From step 128 to 384, the probability of tokens in the bottom 1% consistently remains in the strictly low-probability regime (< 0.1). In contrast, tokens in the bottom 5% span a much wider range, reaching probabilities as high as 0.5, which are no longer low-probability candidates requiring protection. Consequently, setting $\rho \approx 1\%$ (0.01) is a logical and robust choice to target the true tail of the distribution without inadvertently regularizing high-probability tokens. The sensitivity analysis in Figure 15 confirms that performance is stable around this empirically derived value.

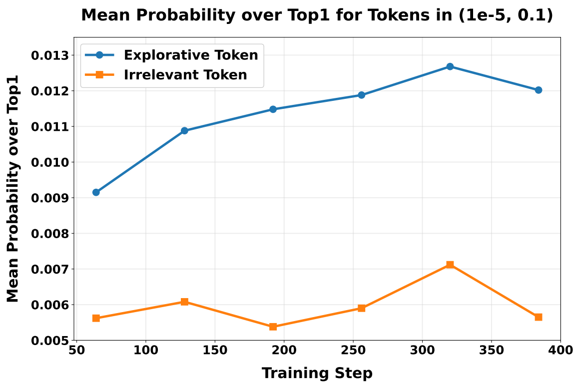


Figure 14: Comparison of relative probability ratios between exploratory tokens and meaningless noise tokens during training. A clear gap exists, supporting the selection of $\kappa \approx 0.01$.

Selection of κ. Figure 14 compares the average relative probability ratio $(\frac{\pi_{\boldsymbol{\theta}}(o|\cdot)}{\max_{o' \in V} \pi_{\boldsymbol{\theta}}(o'|\cdot)})$ between a set of meaningful exploratory tokens ($S_{explore}$ = {"but", "wait", "perhaps", "alternatively", "however"}) and a set of meaningless noise tokens (S_{noise} = {"cost", "fine", "balanced", "ere", "trans"}) that are irrelevant with the reasoning task. The statistics, derived from standard GRPO training, reveal a distinct and persistent separability gap: the relative probability of meaningful exploratory tokens consistently exceeds that of noise tokens throughout the training process. This empirical gap justifies setting the min-p ratio κ (which determines the noise threshold $\tau = \kappa \cdot \max_{o' \in V} \pi_{\boldsymbol{\theta}}(o'|\cdot)$) within this separation region. As shown in the figure, most noise tokens typically fall below a ratio of 0.01, while exploratory tokens remain above it. Therefore, values of κ around 0.01 (or slightly higher) serve as effective initial settings to filter noise while preserving reasoning sparks. The robustness of Lp-Reg with $\kappa \in \{0.01, 0.02, 0.03\}$, as verified in Section E.2, further validates this selection strategy.

E.2 Hyperparameter Sensitivity Analysis

In this section, we analyze the sensitivity of two core hyperparameters in Lp-Reg to demonstrate the robustness of our method: the low-probability percentile ρ and the min-p ratio κ. The results are presented in Figure 15.

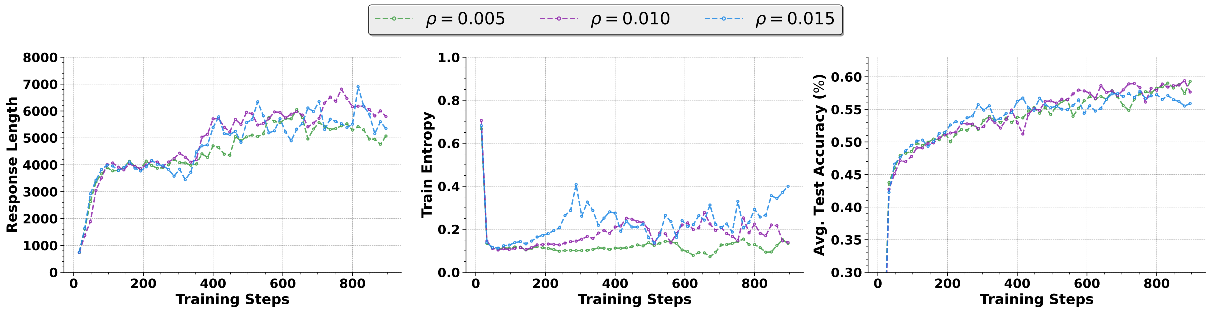
The parameter ρ, as defined in our objective function (Equation 5), determines the percentile threshold for identifying low-probability tokens that are candidates for regularization. A higher ρ means a wider range of tokens are protected. As shown in the top panel of Figure 15, we evaluated ρ with values of 0.005, 0.010, and 0.015. The training trajectories for average test accuracy are nearly identical, and the final performance across all three settings is highly comparable. This indicates that Lp-Reg is not overly sensitive to the precise scope of tokens being protected within this reasonable range.

The hyperparameter κ controls the adaptiveness of the min-p filtering threshold, which defines the boundary for what is treated as noise. A smaller κ results in a more conservative filtering strategy, removing fewer tokens. Our sensitivity analysis for κ, presented in the bottom panel of Figure 15, shows a similar trend of stability. Across the tested values of 0.01, 0.02, and 0.03, the training curves and final performance remain consistently high and tightly clustered. Taken together, these results demonstrate the robustness of Lp-Reg. The method's effectiveness is not contingent on extensive, fine-grained hyperparameter tuning, highlighting its practical applicability.

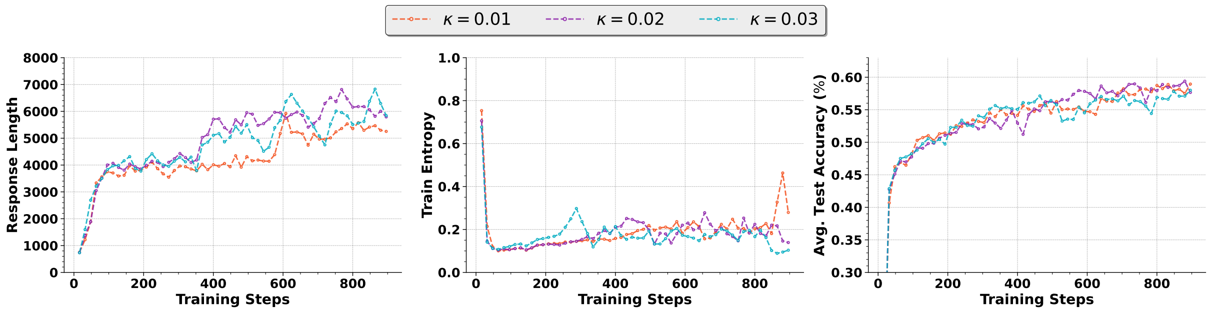
F Computational Overhead Analysis

To analyze the computational overhead of Lp-Reg, particularly with large vocabularies, we analyze the complexity of its two core components: proxy distribution construction in Equation 4 and loss computation in Equation 5. We provide the PyTorch-style implementation for proxy distribution renormalization in Listing 1 and the Lp-Reg loss calculation in Listing 2.

Proxy Distribution Renormalization: As shown in Listing 1, the renormalization process involves computing the maximum probability and re-evaluating log-probabilities. While these operations scale linearly with the vocabulary size $\mathcal{O}(|V|)$, they are structurally identical to the standard Softmax and Log-Softmax operations already required by the base model. These element-wise



(a) Sensitivity analysis on ρ which defined the low-probability percentile threshold $\delta_{\rho}^{\mathcal{B}}$.



(b) Sensitivity analysis on κ which defined the noise threshold $\tau = \kappa \cdot \max_{o' \in V} \pi_{\theta}(o'|\cdot)$

Figure 15: Training dynamics of Lp-Reg method with different hyperparameters.

vector operations are highly parallelizable on GPUs and are memory-bandwidth bound rather than compute-bound. Consequently, their cost is negligible compared to the $\mathcal{O}(d_{model}^2)$ complexity of the Transformer's matrix multiplications, regardless of the vocabulary size.

Loss Computation: The regularization term requires identifying the lowest-probability tokens, which involves a Top-K selection (Listing 2). The computational complexity is $\mathcal{O}(N \log K)$ (using a heap) or $\mathcal{O}(N)$ (using QuickSelect), where N is the total number of tokens in a micro-batch (typically $\approx 30,000$) and $K = \rho \cdot N$ ($\rho \approx 0.01$) is the number of selected tokens. Given that N is relatively small and the operation is performed only once per optimization step (not during every inference step), this sorting overhead is computationally trivial.

Empirical Verification: We empirically validate this analysis by comparing the training runtime of GRPO and Lp-Reg in Table 7. To ensure a strictly fair comparison, we loaded checkpoints at 256, 512, and 768 steps and executed exactly one training update for each method under identical conditions of the same rollout data. The results show that Lp-Reg introduces a marginal overhead of approximately $0.3\% \sim 0.5\%$. This confirms that Lp-Reg is computationally lightweight and does not affect the scalability of training.

G Further Analysis on Low-Probability and High-Entropy Tokens

G.1 Frequency Dynamics of Exploratory Tokens

Figure 16 corroborates this by comparing token frequencies. Lp-Reg consistently maintains a higher fraction of explorative tokens than GRPO, demonstrating that our method not only broadens their distribution but also sustains their practical use throughout training.

G.2 Theoretical Discussion on Low-Probability vs. High-Entropy Tokens

While previous works have primarily utilized policy entropy as a proxy for exploration (Wang et al., 2025a), our approach distinguishes between high-entropy tokens and low-probability tokens. Empirical results presented in Table 2 and Figure 5 demonstrate that regularizing low-probability tokens yields significantly better stability and performance than targeting high-entropy tokens.

In this section, we provide a theoretical foundation for these results. We formally demonstrate that the set of tokens targeted by high-entropy methods is a *subset* of those captured by low-probability

```
def forward_micro_batch(logits, kappa):
    # Standard Log-Softmax calculation
    log_prob = log_softmax(logits)

+   # 1. Calculate dynamic threshold
+   prob = exp(log_prob)
+   threshold = kappa * max(prob, axis=-1)

+   # 2. Filter noise
+   mask = prob < threshold
+   proxy_logits = logits.clone()
+   proxy_logits[mask] = -infinity

+   # 3. Re-normalization
+   proxy_log_prob = log_softmax(proxy_logits)

    return log_prob, proxy_log_prob
```

Listing 1: Pseudo-code of Proxy Distribution Construction

```
def compute_policy_loss_lp_reg(old_log_prob, log_prob, proxy_log_prob, advantage, **args):
    # Standard PPO/GRPO Loss
    ratio = exp(log_prob - old_log_prob)
    pg_loss = maximum(-ratio * advantage, -clip(ratio) * advantage)

+   # 1. Identify tokens receiving negative feedback
+   neg_idx = indices(advantage < 0)

+   # 2. Select bottom rho% lowest probability tokens
+   k = int(len(neg_idx) * args["rho"])
+   low_prob_idx = topk(log_prob[neg_idx], k=k, largest=False)

+   # 3. Apply Regularization
+   mask = log_prob[low_prob_idx] < proxy_log_prob[low_prob_idx]
+   reg_idx = low_prob_idx[mask]

+   # 4. Calculate KL penalty term
+   pg_loss[reg_idx] += args["ppo_kl_coef"] * kl_penalty(
+       log_prob[reg_idx], proxy_log_prob[reg_idx])

    return pg_loss.mean()
```

Listing 2: Pseudo-code of Lp-Reg Loss Calculation

Steps	Avg. Response Length	GRPO (s)	Lp-Reg (s)	Overhead
256	4058.53	698.49	700.60	+0.30%
512	5794.25	973.74	978.62	+0.50%
768	6640.69	1137.24	1141.49	+0.37%

Table 7: Runtime comparison between GRPO and Lp-Reg under different training steps. Lp-Reg introduces only marginal overhead compared with GRPO.

methods. Crucially, high-entropy strategies inherently overlook the region of low-probability tokens within low-entropy distributions, which is important for exploration, proven by empirical experi-

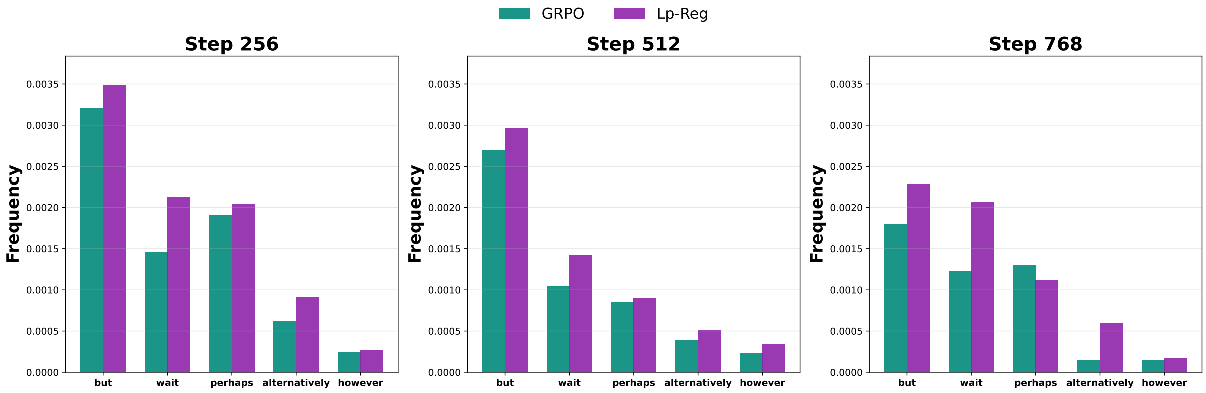


Figure 16: Frequency of exploratory tokens during training.

ments.

Proposition 1 *Given the policy $\pi_\theta(\cdot|s)$ over a vocabulary V, and the policy entropy defined as $\mathcal{H}(\pi_\theta) = -\sum_{o \in V} \pi_\theta(o|s)\log(\pi_\theta(o|s))$, the following holds: $\forall \epsilon \in (1/|V|, 1), \exists \delta > 0$, s.t. if $\mathcal{H}(\pi_\theta) > \delta$, then $\pi_\theta(o|s) < \epsilon, \forall o \in V$.*

Proof Let $p_{\max} = \max_{o \in V} \pi_\theta(o|s)$ be the max token probability in the policy, and let $o^* = \arg\max_{o \in V} \pi_\theta(o|s)$. Accordingly, $\pi_\theta(o^*|s) = p_{\max}$.

Firstly, we decompose the entropy term by separating the maximal probability token o^* from the rest of the vocabulary $V \setminus \{o^*\}$:

$$\mathcal{H}(\pi_\theta) = -p_{\max}\log p_{\max} - \sum_{o \neq o^*} \pi_\theta(o|s)\log \pi_\theta(o|s). \tag{6}$$

Let $K = |V| - 1$. The remaining probability mass is $1 - p_{\max}$. Since $f(x) = x\log x$ is a convex function, according to Jensen's Inequality, the entropy of the remaining tokens is maximized when the distribution is uniform, i.e., $\pi_\theta(o|s) = \frac{1-p_{\max}}{K}$ for all $o \neq o^*$. Substituting this into the equation, we obtain the upper bound function $g(p_{\max})$:

$$\begin{aligned}\mathcal{H}(\pi_\theta) &\leq -p_{\max}\log p_{\max} \\ &\quad - \sum_{o \neq o^*} \frac{1-p_{\max}}{K} \log\left(\frac{1-p_{\max}}{K}\right) \\ &= -p_{\max}\log p_{\max} - (1-p_{\max})\log\left(\frac{1-p_{\max}}{K}\right) \\ &\triangleq g(p_{\max}).\end{aligned}$$

Then, we analyze the monotonicity of the function $g(x) = -x\log x - (1-x)\log\frac{1-x}{K}$ for $x \in (1/|V|, 1)$. Taking the derivative with respect to x:

$$\begin{aligned}g'(x) &= -(\log x + 1) - \Bigg[(-1)\cdot\log\left(\frac{1-x}{K}\right) \\ &\quad + (1-x)\cdot\frac{K}{1-x}\cdot\left(-\frac{1}{K}\right)\Bigg] \\ &= -\log x - 1 + \log\left(\frac{1-x}{K}\right) + 1 \\ &= \log\left(\frac{1-x}{Kx}\right).\end{aligned}$$

Since $K = |V| - 1$, we have $\frac{1-x}{Kx} < 1$ for any $x > \frac{1}{|V|}$. Thus, $g'(x) < 0$ when $x \in (\frac{1}{|V|}, 1)$, which means $g(x)$ is strictly monotonically decreasing on the interval $(\frac{1}{|V|}, 1)$.

Finally, Let $\delta = g(\epsilon)$. Since $\epsilon \in (1/|V|, 1)$, δ is a well-defined positive value. Assume the condition $\mathcal{H}(\pi_\theta) > \delta$ holds. By the upper bound established above, we have:

$$g(p_{\max}) \geq \mathcal{H}(\pi_\theta) > \delta = g(\epsilon). \tag{7}$$

Thus, $g(p_{\max}) > g(\epsilon)$. Since we have proved that $g(x)$ is strictly monotonically decreasing for $x > 1/|V|$, the inequality of function values implies the reverse inequality of arguments:

$$p_{\max} < \epsilon. \tag{8}$$

By definition, $\pi_\theta(o|s) \leq p_{\max}$ for all $o \in V$. Therefore, $\pi_\theta(o|s) < \epsilon$ for all $o \in V$.

Proposition 1 theoretically establishes that high entropy strictly implies low probability for all tokens. In other words, the set of tokens targeted by high-entropy methods is almost a subset of those targeted by low-probability regularization.

However, the converse does not hold. Low-probability tokens can be sampled not only from high-entropy positions but also from low-entropy positions. The latter scenario constitutes a blind spot for entropy-based methods: when the model is in a low entropy position, entropy methods ignore the step. Yet, as shown in Figure 17, valuable explorative tokens (e.g., “but”, “wait”) frequently appear in this low-probability, low-entropy region. The theoretical upper bound visualized in Figure 18 further confirms that entropy maximization is restricted to the left-most region, whereas our Lp-Reg remains effective across the entire region. This explains why Lp-Reg outperforms high-entropy regularization, as validated by our experiments.

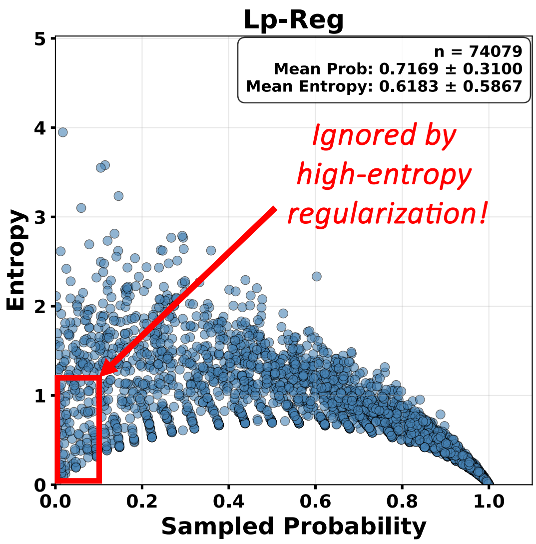


Figure 17: Probability–Entropy scatter plots of explorative tokens: “but”, “wait”, “perhaps”, “alternatively”, and “however”. It displays a random sample of 5% of all data points

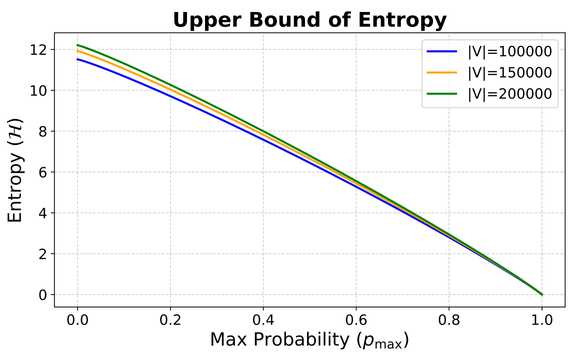


Figure 18: Theoretical bound of entropy $\mathcal{H}$ vs. max probability $p_{max} = \max_{o \in V} \pi_\theta(o|\cdot)$. The curve represents the maximum possible entropy for a given peak probability $p_{\max}$ with $|\mathcal{V}| = 100000, 150000, 200000$.

G.3 Trajectory-Level Token Analysis

In this section, we conduct a fine-grained trajectory-level analysis to characterize the sampling probability distributions of specific tokens. We decompose the token sampling distributions based on the advantage values of their corresponding trajectories: positive ($A_i > 0$), negative ($A_i < 0$), and neutral/invalid ($A_i = 0$). The comparative results between exploratory tokens (e.g., “but”, “wait”) and irrelevant tokens (e.g., “cost”) are visualized in Figure 19.

As shown in Figure 19, we observe that the probability distributions of exploratory tokens are remarkably similar across Positive and Negative trajectories, under both standard GRPO and Lp-Reg. This indicates that these tokens function as reasoning patterns: they represent the mechanism of the reasoning attempt, rather than the determinant of the final outcome. Just as scratchpad paper is utilized for both correct and incorrect solutions, a negative trajectory containing “wait” represents a failed reasoning attempt, which is fundamentally different from a failure due to a lack of reasoning. This is further corroborated by the contrast in sampling density between active learning groups ($A \neq 0$) and static groups ($A = 0$). The former exhibits a significantly higher density of low-probability tokens, while the latter shows much less. This is consistent with the intuition that active exploration yields diverse outcomes (both successes and failures), whereas a lack of exploration leads to concentrated, often stagnant results. Because these tokens appear abundantly in negative trajectories simply due to the high volume of failed attempts during exploration, standard GRPO tends to systematically suppress them. Lp-Reg successfully preserves these essential patterns, ensuring the model retains the capacity to reason even when individual attempts fail.

Importantly, a distinct divergence emerges when comparing standard GRPO and Lp-Reg. As illustrated in Figure 19 (Step 512), standard GRPO exhibits a significant reduction in low-probability token sampling in later training stages, signaling a diminishing of exploration attempts when uncertain (low probability). This collapse directly corresponds to the performance bottleneck observed in Figure 20. In contrast, Lp-Reg maintains robust low-probability token sampling throughout long-horizon training, coinciding with a continuously increasing accuracy score. This demonstrates the effectiveness of Lp-Reg in sustaining exploration.

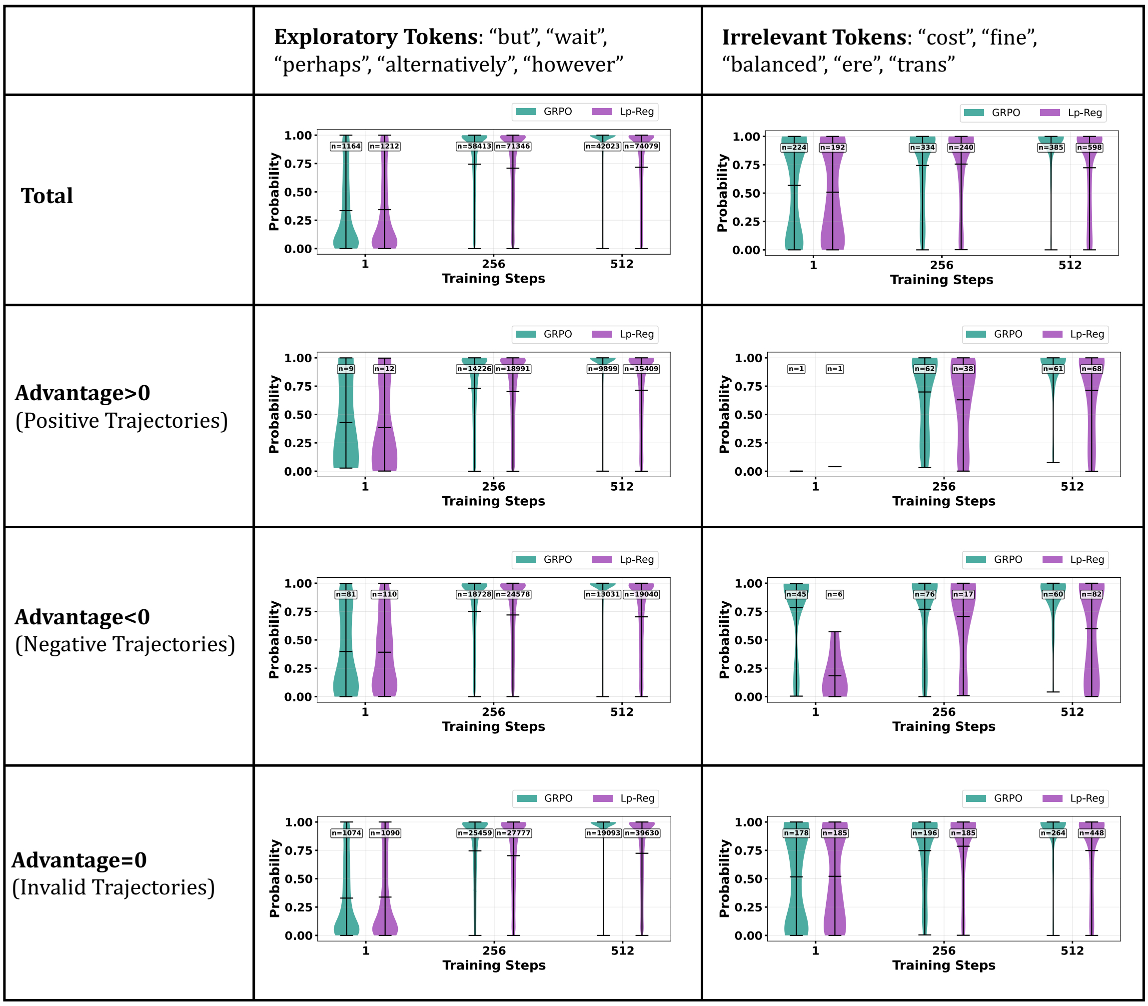


Figure 19: Trajectory-level probability analysis distinguishing exploratory tokens (left) from irrelevant tokens (right). The distributions are decomposed into positive ($A > 0$), negative ($A < 0$), and invalid ($A = 0$) trajectories, where n represents the sampling token number.

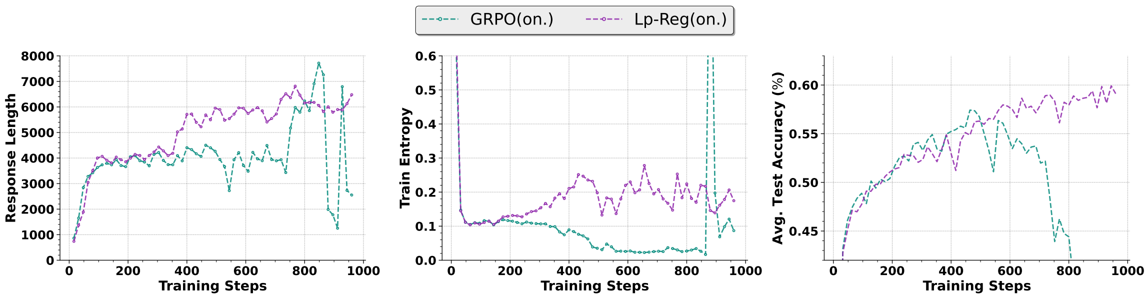


Figure 20: Comparison between standard GRPO and Lp-Reg on Qwen3-14B-Base.

G.4 Comparison with Concurrent Work (Lopti)

In this section, we discuss the difference between our Lp-Reg and Lopti (Yang et al., 2025b), a recent work that also investigates low-probability tokens. It is important to note that Lp-Reg and Lopti are not in conflict; rather, they identify and address two distinct orthogonal challenges in RLVR training. Lopti focuses on improving gradient dynamics for better data efficiency, while Lp-Reg focuses on ensuring long-horizon training stability. The distinction is from three perspectives: the core research problem, the methodological approach, and new, direct experimental comparisons.

(1) Different Core Research Problems: Lopti focuses on the training efficiency, whereas Lp-Reg

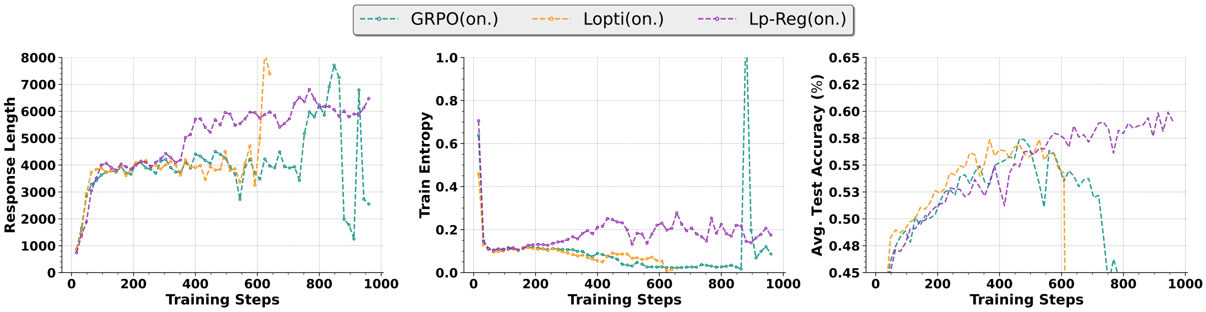


Figure 21: Comparison on standard GRPO, Lopti, and Lp-Reg.

focuses on the training stability. These represent two orthogonal axes of improvement for RLVR. In detail, Lopti identifies that low-probability tokens generate gradients with disproportionately large norms. Its core focus is on how this "over-domination" suppresses gradient signals from high-probability tokens, thereby reducing the data efficiency of the training process. In contrast, our Lp-Reg identifies the systematic elimination of low-probability tokens with exploratory semantics (e.g., "wait"), which we term "*reasoning sparks*". Our core focus is on how the over-penalization of these tokens leads to a loss of exploration capacity with the entropy collapse phenomenon, hindering the model from achieving higher performance in long-horizon stable training.

(2) Different Methodological Approaches: Lopti's method of separate gradient updates and Lp-Reg's regularization are distinct and non-conflicting algorithms. Specifically, to prevent large-norm gradients from low-probability tokens suppressing signals from high-probability tokens, Lopti separates the loss computation for these two groups and updates the model parameters twice per micro-batch. For another goal to protect low-probability tokens from over-penalization in RLVR, Lp-Reg introduces a regularization on them via a KL divergence between the current policy and a filtered proxy policy.

(3) Empirical evidence from long-horizon experiments: To empirically validate our claims, we have conducted a long-horizon training experiment comparing Lopti, Lp-Reg, and the GRPO baseline for 1,000 steps. As shown in Figure 21, Lopti shows a faster initial rise in test accuracy, confirming its effectiveness at accelerating learning, consistent with the findings in their paper. However, after approximately 600 steps, Lopti's performance plateaus, and its training entropy collapses in the same manner as the GRPO baseline. This shows that improving data efficiency does not inherently solve the long-term exploration problem. In contrast, our Lp-Reg demonstrates stable performance improvement throughout the 1,000 steps, correlated with its ability to maintain policy entropy. This sustained exploration allows it to achieve a significantly higher final accuracy.

In conclusion, Lp-Reg and the Lopti study address distinct, orthogonal challenges in RLVR. The choice between these methods may depend on the specific training objectives. While investigating a potential combination could be an interesting avenue for future research, our primary contribution here is to formally identify the exploration stability and provide an effective solution for it. We have added this detailed comparison to our revised manuscript to contextualize our work better and highlight its unique conceptual contribution.

H Details of Analysis

H.1 Details of Sampling Probability Density

This section provides a detailed, token-by-token breakdown of the aggregated distributions presented in Figure 1c and Figure 1d of the main paper, reinforcing the conclusions drawn from our analysis.

Figure 23 exhibits the individual distribution of observed sampling probabilities for meaningful exploratory tokens, also known as *reasoning sparks*: "but", "wait", "perhaps", "alternatively", and "however". These tokens are also frequently analyzed as representative cases in previous studies (DeepSeek-AI et al., 2025; Muennighoff et al., 2025; Hu et al., 2025; Qian et al., 2025; Wang et al., 2025a). A consistent trend is observable across all five tokens, validating our claims in the introduction. With standard GRPO training, the ability to sample these tokens at low probabilities is systematically eliminated, causing their distributions to collapse and shift towards higher probabilities. The indiscriminate entropy bonus (GRPO + Entropy Loss) is largely ineffective at restoring this crucial low-probability tail. In stark contrast, our proposed method, Lp-Reg, consistently maintains a healthy, wide distribution for each of these tokens, demonstrating its effectiveness in preserving the model's capacity for nuanced exploration.

Conversely, Figure 24 details the behavior of a class of meaningless noise tokens: "cost", "fine", "balanced", "ere", and "trans". These individual plots clearly illustrate the detrimental side effect of a simple entropy bonus. For nearly every token, the GRPO + Entropy Loss baseline significantly amplifies the sampling of this irrelevant noise, which, as shown in our main analysis, contributes to a faster performance collapse. Lp-Reg, by design, avoids this amplification and maintains a suppressed probability distribution for these tokens, comparable to or even more constrained than the standard GRPO baseline.

These detailed visualizations confirm that the phenomena of reasoning spark elimination and noise amplification are not artifacts of aggregation but are consistent patterns at the individual token level. This provides strong, granular evidence for the central challenge our paper addresses and highlights the necessity of a selective preservation mechanism like Lp-Reg.

H.2 Details of Probability-Entropy Distribution

To supplement the aggregated analysis presented in Figure 7 of the main text, this section provides a detailed breakdown of the probability-entropy distributions for individual *reasoning sparks*. Figure 25 shows a consistent pattern across all representative tokens, ranging from "but" (Figure 25a) to "however" (Figure 25e). For frequently occurring tokens such as "but", "wait", and "perhaps", we randomly subsample one out of every 20 instances for visualization. Under the baseline GRPO, these sparks are consistently confined to a low-entropy, high-probability region, indicating a collapse into deterministic usage. In contrast, the addition of an entropy loss pushes these tokens into highly scattered, often excessively high-entropy states, suggesting an uncontrolled and potentially noisy form of exploration. Our method, Lp-Reg, strikes a crucial balance, maintaining a structured and broad distribution across a healthy range of entropy values. This consistent behavior demonstrates that the trends identified in the aggregated data are not artifacts of averaging. The individual plots offer strong, disaggregated evidence for our central claim: Lp-Reg effectively preserves the exploratory potential of reasoning sparks by preventing both the deterministic collapse seen in the baseline and the chaotic scattering induced by the indiscriminate entropy bonus.

H.3 Training Dynamics of Regularized Token

To better understand how Lp-Reg operates during training, we analyze the dynamics of the probability threshold $\delta_\rho^{\mathcal{B}}$ and the proportion of low-probability tokens subjected to regularization. As shown in Figure 22, the threshold $\delta_\rho^{\mathcal{B}}$ gradually decreases with training steps. In parallel, the regularization ratio also declines steadily. This trend suggests that as training progresses, an increasing share of extremely low-probability tokens correspond to meaningless noise, while the semantically meaningful tokens are lifted into higher-probability regions and thus require less regularization.

I Case Study

To further illustrate the effect of the filter applied on low-probability tokens, Figure 26 to Figure 28 presents a case study of a model-generated response, where low-probability tokens are highlighted according to whether they were preserved or filtered. Tokens with probability greater than τ are those retained by the filter, while tokens with probability smaller than τ are suppressed. The preserved tokens include meaningful exploratory markers such as "Then", "Wait", which guide the reasoning trajectory, whereas the discarded set largely consists of noisy tokens such as "We", "also", "that". This qualitative evidence complements our quantitative analysis, demonstrating that Lp-Reg effectively leverages min-p distribution re-normalization to reliably distinguish between semantically meaningful exploratory reasoning sparks and destabilizing noise.

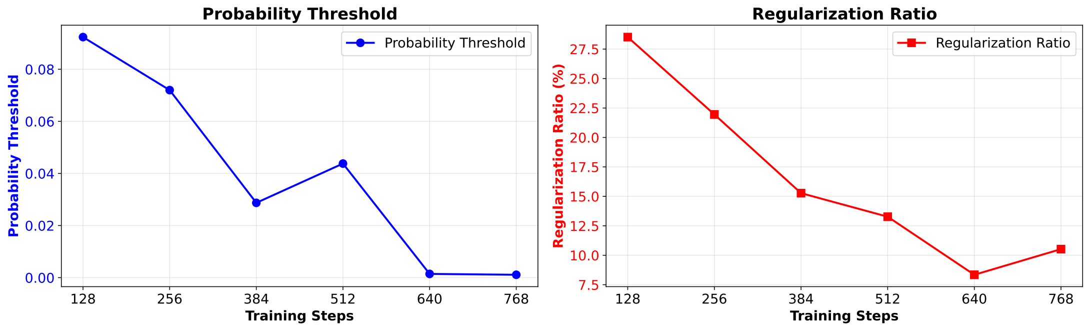
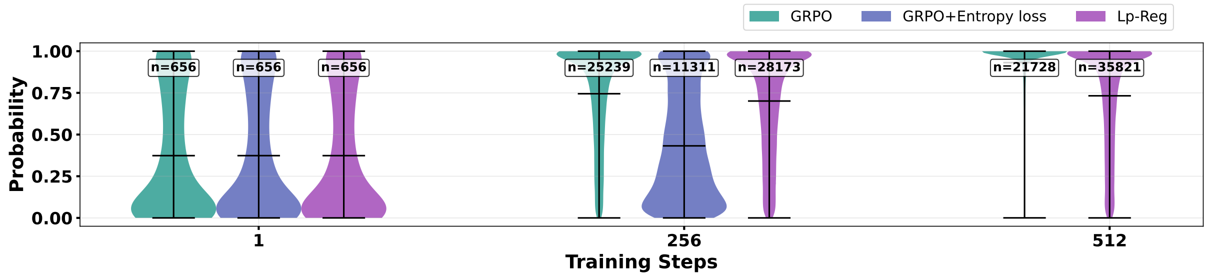
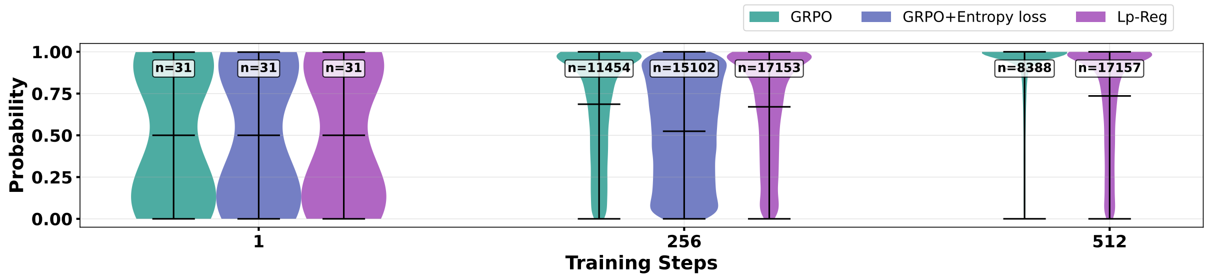


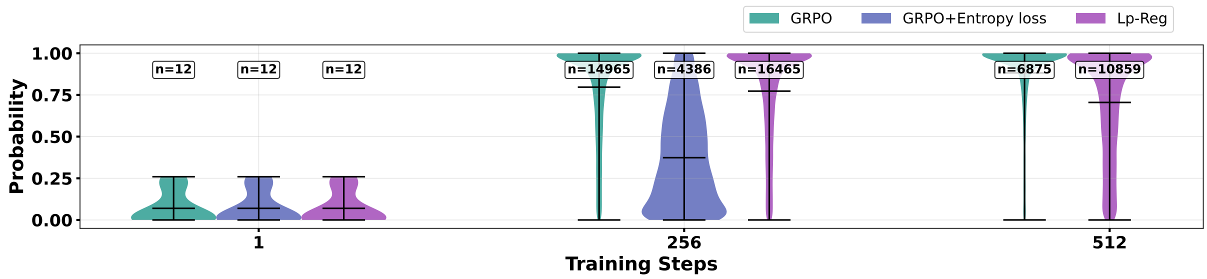
Figure 22: Training dynamics of the probability threshold and regularization ratio.



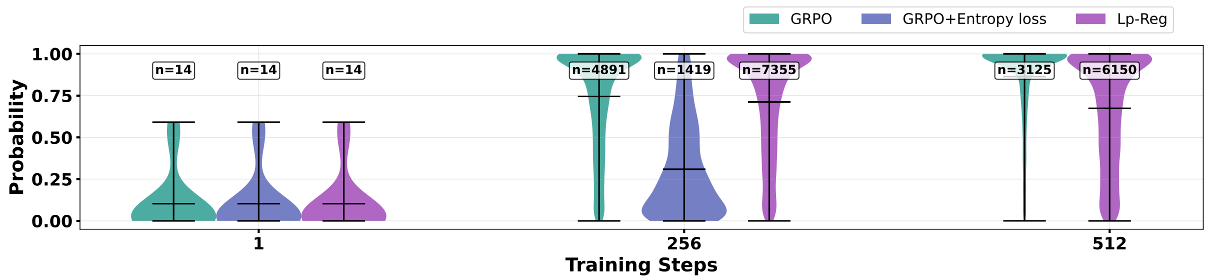
(a) Density of observed sampling probabilities for token “but".



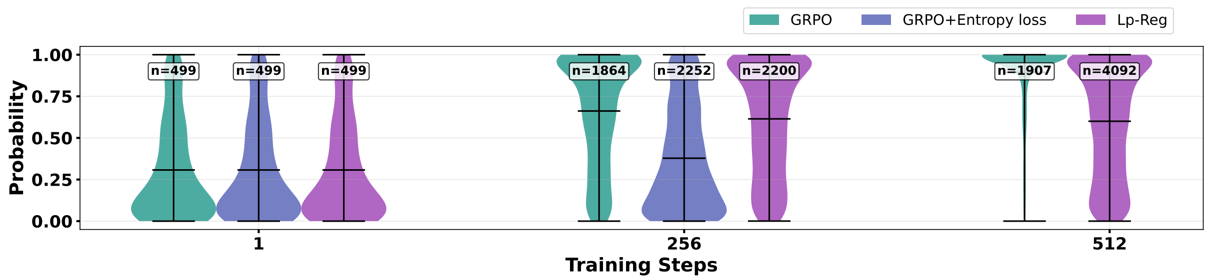
(b) Density of observed sampling probabilities for token “wait".



(c) Density of observed sampling probabilities for token “perhaps".

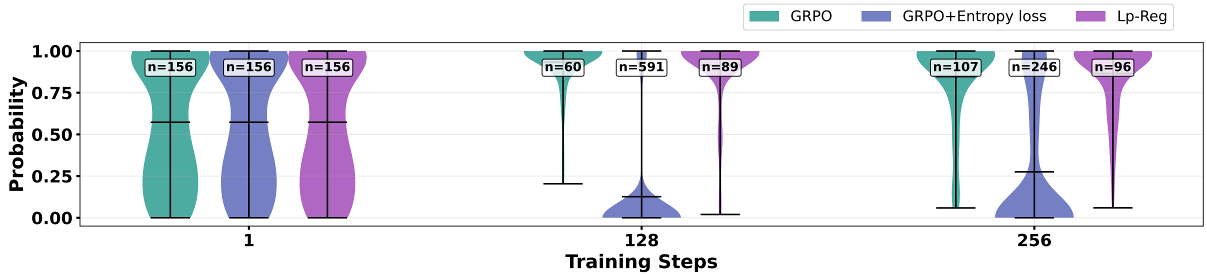


(d) Density of observed sampling probabilities for token “alternatively".

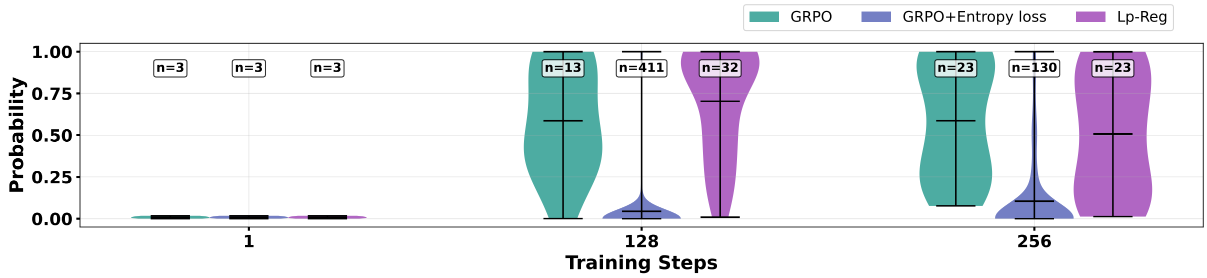


(e) Density of observed sampling probabilities for token “however".

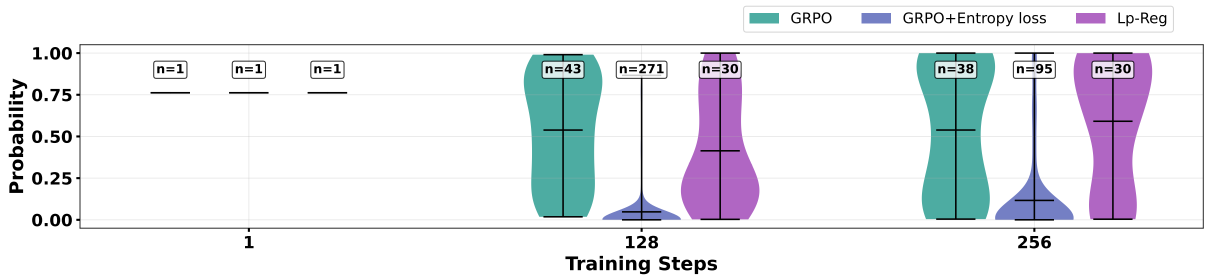
Figure 23: Individual Density of observed sampling probabilities for meaningful exploratory tokens: “but", “wait", “perhaps", “alternatively", and “however".



(a) Density of observed sampling probabilities for token “cost".

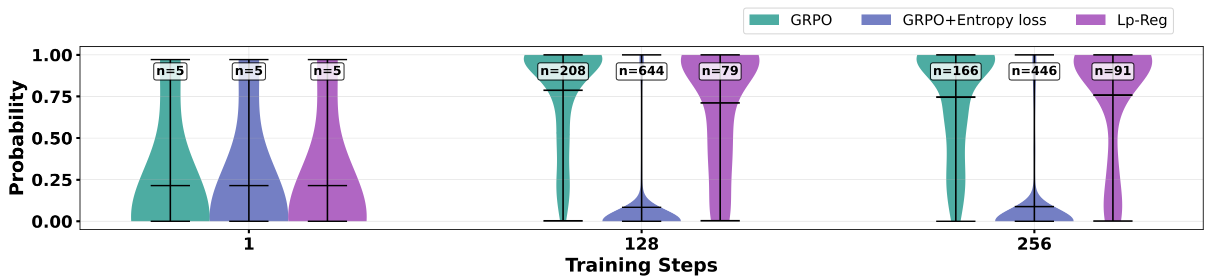


(b) Density of observed sampling probabilities for token “fine".



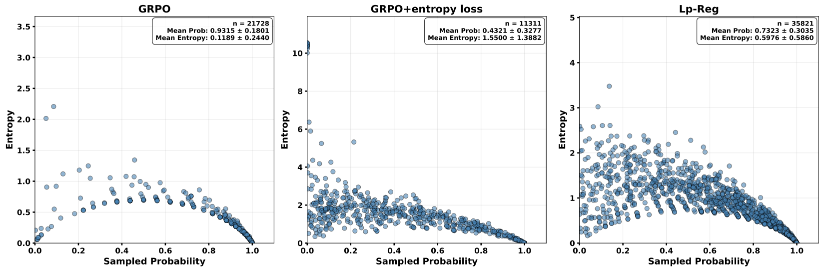
(c) Density of observed sampling probabilities for token “balanced".

(d) Density of observed sampling probabilities for token “ere".

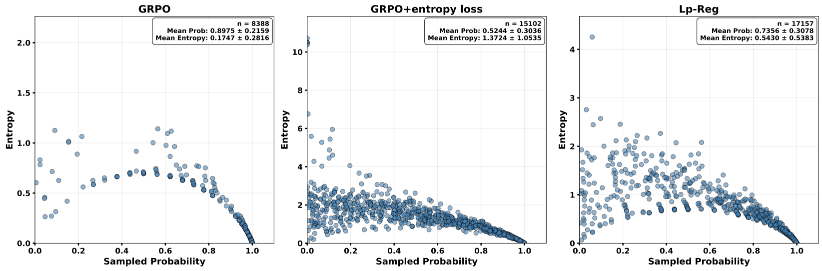


(e) Density of observed sampling probabilities for token “trans".

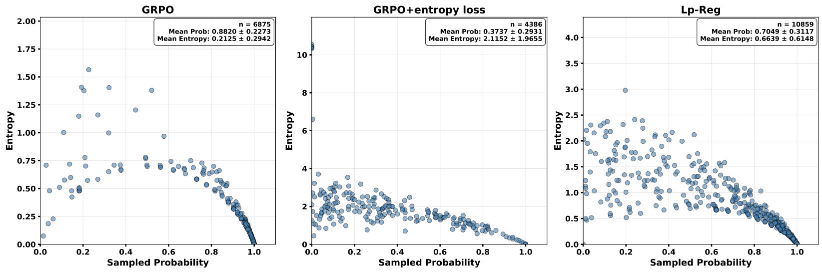
Figure 24: Individual Density of observed sampling probabilities for meaningless tokens: “cost", “fine", “balanced", “ere", and “trans".



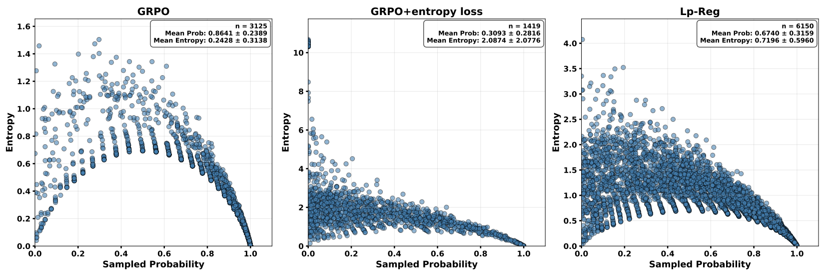
(a) Scattered probability–entropy plot of observed sampling instances for the token “but".



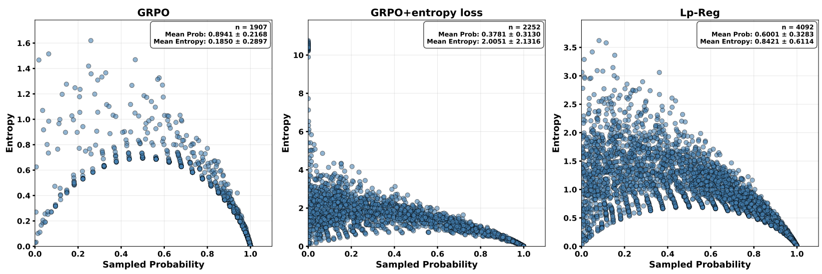
(b) Scattered probability–entropy plot of observed sampling instances for the token “wait".



(c) Scattered probability–entropy plot of observed sampling instances for the token “perhaps".



(d) Scattered probability–entropy plot of observed sampling instances for the token “alternatively".



(e) Scattered probability–entropy plot of observed sampling instances for the token “however".

Figure 25: Individual scattered probability–entropy plot of observed sampling instances for meaningful tokens: “but", “wait", “perhaps", “alternatively", and “however".

Single Response Case Study
Lowest Probability Tokens threshold: $\delta_\rho^\mathcal{B} = 0.026208$
(Part 1 of 3)

Reserved low-probability tokens: $\pi_\theta < \delta_\rho^\mathcal{B}$ and $\pi_\theta > \tau$
Filtered low-probability tokens: $\pi_\theta < \delta_\rho^\mathcal{B}$ and $\pi_\theta < \tau$
Normal tokens: $\pi_\theta > \delta_\rho^\mathcal{B}$

Assistant :\n Alright , I have this problem about a square that 's divided into smaller squares , and the process keeps repeating indefinitely . Some parts are shaded , and I need to find out what fraction of the whole figure is shaded . The final answer should be in the form of a fraction \ (\ frac { m }{ n }\ ), and I need to provide the value of \(m + n \ ). Let 's break this down step by step .\n\n ### Understanding the Problem \n\n First , let 's try to understand what 's happening :\n\n 1 . ** Initial Square **: We start with one big square .\n 2 . ** First Division **: This square is divided into nine smaller squares of equal area . So , it 's like a 3 x 3 grid .\n 3 . ** Center Division **: The center square of this 3 x 3 grid is then further divided into nine even smaller squares (3 x 3 again).\n 4 . ** Pattern Contin ues **: This process repeats indefinitely âĢĶ each new center square is divided into nine smaller squares .\n\n From the As ym pt ote code , it seems that at each step , certain parts (corn ers) of the squares are shaded . Specifically , the four corner squares of each smaller subdivision are shaded .\n\n Wait , looking at the code :\n\n ``` asy
fill ((- n ,n)-- (- 1 / 3 *n ,n)-- (- 1 / 3 *n , 1 / 3 *n)-- (- n , 1 / 3 *n)-- cycle);\n fill ((- n ,- n)-- (- 1 / 3 *n ,- n)-- (- 1 / 3 *n ,- 1 / 3 *n)-- (- n ,- 1 / 3 *n)-- cycle);\n fill ((n ,- n)-- (1 / 3 *n ,- n)-- (1 / 3 *n ,- 1 / 3 *n)-- (n ,- 1 / 3 *n)-- cycle);\n fill ((n ,n)-- (1 / 3 *n ,n)-- (1 / 3 *n , 1 / 3 *n)-- (n , 1 / 3 *n)-- cycle);\n `` `\n\n This code seems to shade the four corner squares of each subdivision at each level . So , at each step , four of the nine smaller squares are shaded .\n\n ### Visual Representation \n\n Let 's try to visualize this :\n\n 1 . ** Level 0 **: The original big square .\n 2 . ** Level 1 **: Div ided into 9 squares , and the four corners are shaded .\n 3 . ** Level 2 **: The center square of Level 1 is divided into 9 even smaller squares , and its four corners are shaded .\n 4 . ** Level 3 **: The center of that latest division is divided again , and so on , infinitely .\n\n Wait , actually , looking back at the As ym pt ote code , it seems that at every level , the four corner squares of each current smallest subdivision are shaded . But in the description , it mentions the center square is divided , not necessarily the corners . Maybe I mis interpreted .\n\n Wait , the code sh ad ings correspond to the corners of each subdivision , but the description says the center square is divided . Hmm , perhaps the shaded parts are always the four corners of the current center square .\n\n But perhaps it 's better to assign a value .\n\n ### Assign ing Areas \n\n Let 's assign an area to the original square to make calculations easier . Let 's say the area of the original square is 1 .\n\n - ** Level 0 **: Total area = 1 . Fraction shaded = 0 ? Wait , no , according to the As ym pt ote code , the parts shaded are the four corners of each subdivision , not clearly defined yet .\n\n Wait , perhaps it 's better to think in terms of geometric series .\n\n At each level :\n\n - The entire figure is divided into 9 equal smaller squares .\n - Out of these , 4 are shaded (the corners).\n\n Wait , perhaps it 's not exactly that . Looking at the code , for each square at a certain level , its four corners in the next subdivision are shaded . But the center square continues to be subdiv ided .\n\n Wait , perhaps it 's better to think recursively .\n\n ### Recursive Thinking \n\n Let 's denote :\n\n - \(S _n \ ) as the total shaded area after \(n \ ) levels of subdivision .\n\n But since it 's infinite , we need to consider the limit as \(n \ ) approaches infinity .\n\n At each step :\n\n - The current smallest squares are divided into 9 even smaller squares .\n - Out of these 9 , 4 are shaded .\n\n Wait , perhaps it 's better to think of the entire figure and how much is shaded at each stage .\n\n Let 's consider the first level :\n\n - Original square is divided into 9 .\n - Sh aded area : the four corners , each has area \(\ frac { 1 }{ 9 } \ ), so total shaded at level 1 : \(4 \ times \ frac { 1 }{ 9 } = \ frac { 4 }{ 9 } \ ).\n\n Wait , but then the center square is further divided , and in its corners , etc .\n\n Wait , perhaps it 's better to think of it as :\n\n At each subdivision , for each existing square , we shade four of its nine smaller parts , and then the center is subdiv ided further .\n\n This seems like it might lead to a geometric series where at each step , a fraction is added to the shaded area .\n\n But I 'm not entirely clear on the pattern . Maybe I should try to compute the shaded area at each step .\n\n ### Computing Sh aded Area \n\n Let 's try to compute the shaded area after a few levels to see if a pattern emerges .\n\n ** Level 0 **: Sh aded area = 0 .\n\n ** Level 1 **: The original square is divided into 9 , and 4 are shaded :

Figure 26: An Example generated by Qwen3-14B-Base model trained by Lp-Reg from math reasoning. (Part 1)

Single Response Case Study
Lowest Probability Tokens threshold: $\delta_\rho^B = 0.026208$
(Part 2 of 3)

Reserved low-probability tokens: $\pi_\theta < \delta_\rho^B$ and $\pi_\theta > \tau$
Filtered low-probability tokens: $\pi_\theta < \delta_\rho^B$ and $\pi_\theta < \tau$
Normal tokens: $\pi_\theta > \delta_\rho^B$

\(S _ 1 = \ frac { 4 }{ 9 } \ ).\n\n Wait , but then the process continues . The center square (of area \(\ frac { 1 }{ 9 } \ )) is further divided into 9 squares , each of area \(\ frac { 1 }{ 8 1 } \ ), and 4 of these are shaded : \(\ frac { 4 }{ 8 1 } \ ).\n\n So , total shaded after Level 2 : \(S _ 2 = \ frac { 4 }{ 9 } + \ frac { 4 }{ 8 1 } \ ).\n\n Similarly , the center of this latest subdivision (which was a square of area \(\ frac { 1 }{ 8 1 } \ )) has a center that gets divided further , and so on .\n\n Wait , perhaps it 's better to think of it as an infinite series .\n\n Each time we divide a square into 9 and shade 4 , and then the center is divided again .\n\n This seems like a geometric series where each term is a fraction of the previous shaded area .\n\n Wait , let 's see :\n\n At Level 1 : shaded area \(\ frac { 4 }{ 9 } \ ).\n\n At Level 2 : shaded area \(\ frac { 4 }{ 8 1 } \ ) (from the center square at Level 1).\n\n At Level 3 : shaded area \(\ frac { 4 }{ 7 2 9 } \ ) (from the center square at Level 2), and so on .\n\n Wait , perhaps the total shaded area is the sum of those :\n\n \(S = \ frac { 4 }{ 9 } + \ frac { 4 }{ 8 1 } + \ frac { 4 }{ 7 2 9 } + \ cd ots \ )\n\n This is a geometric series with first term \(a = \ frac { 4 }{ 9 } \ ) and common ratio \(r = \ frac { 1 }{ 9 } \ ).\n\n The sum of an infinite geometric series is \(S = \ frac { a }{ 1 - r } \ ).\n\n So , \(S = \ frac {\ frac { 4 }{ 9 }} { 1 - \ frac { 1 }{ 9 }} = \ frac {\ frac { 4 }{ 9 }} {\ frac { 8 }{ 9 }} = \ frac { 4 }{ 8 } = \ frac { 1 }{ 2 } \ ).\n\n Wait , but I 'm not entirely confident this captures the entire shaded area because the shaded parts are at each subdivision level , and the pattern might be different .\n\n Wait , perhaps I should consider that at each subdivision , the shaded parts are not just in the outer squares but are nested within the subdivisions .\n\n Alternatively , perhaps the correct approach is to consider that at each step , the shaded area added is \(\ frac { 4 }{ 9 } \ ) of the area being subdiv ided , and the un sh aded center continues to be subdiv ided .\n\n Wait , perhaps a better approach is to consider the self -s im ilarity of the problem .\n\n ### Self -S im ilarity Approach \n\n Let 's denote \(S \ ) as the total shaded area of the original square .\n\n Looking at the first subdivision :\n\n - We have 9 smaller squares .\n - 4 of them are shaded immediately : each has area \(\ frac { 1 }{ 9 } \ ), so total \(\ frac { 4 }{ 9 } \ ).\n - The center square (also of area \(\ frac { 1 }{ 9 } \ )) is a smaller version of the original figure , and its shaded area would be \(\ frac { S }{ 9 } \ ).\n\n Wait , no . Actually , the center square is undergoing the same pattern without the immediate shading of its corners initially (in the context of the original shading).\n\n Wait , perhaps it 's better to think that the entire figure maintains self -s im ilarity where :\n\n Total shaded area \(S = \ ) shaded in first level \(+ \ ) shaded in scaled -down version \(= \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ).\n\n Wait , let 's think carefully :\n\n At the first level :\n\n - Sh aded area : \(\ frac { 4 }{ 9 } \ ) (the four corner squares).\n - The center square (of area \(\ frac { 1 }{ 9 } \ )) also follows the same pattern , but in its divisions , its corners will be shaded in the next level , and so on .\n\n Wait , perhaps the recursive relation is :\n\n \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ).\n\n Wait , let 's see :\n\n The total shaded area is the shaded area from the first subdivision plus the shaded area from the pattern that would have been applied to the center square if it were the original , but scaled down by a factor of \(\ frac { 1 }{ 9 } \ ).\n\n Wait , but in the first subdivision , only the outer squares are immediately shaded , and the center continues to be subdiv ided indefinitely .\n\n Wait , perhaps it 's better to think of it as :\n\n At each step , we shade four out of nine parts of each square at the current level , and then the center square is treated as a new " original " square at the next level .\n\n Thus , the total shaded area \(S \ ) satisfies :\n\n \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ).\n\n Wait , let 's verify :\n\n The first term \(\ frac { 4 }{ 9 } \ ) accounts for the four shaded corner squares at the first level .\n\n Then , the center square (of area \(\ frac { 1 }{ 9 } \ )) has its own version of the problem , so its shaded area is \(\ frac { S }{ 9 } \ ).\n\n Thus , the equation \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ) seems correct .\n\n Now , solve for \(S \ ):\n\n \(S - \ frac { S }{ 9 } = \ frac { 4 }{ 9 } \ )\n\n \(\ frac { 8 S }{ 9 } = \ frac { 4 }{ 9 } \ )\n\n Multiply both sides by 9 :\n\n \(8 S = 4 \ )\n\n Div ide both sides by 8 :\n\n \(S = \ frac { 4 }{ 8 } = \ frac { 1 }{ 2 } \ ).\n\n So , the total shaded fraction is \(\ frac { 1 }{ 2 } \ ).\n\n But the problem states that the answer should be in the form \(\ frac { m }{ n } \ ) where \(\ gcd (m , n) = 1 \ ), and we need to

Figure 27: An Example generated by Qwen3-14B-Base model trained by Lp-Reg from math reasoning. (Part 2)

Single Response Case Study
Lowest Probability Tokens threshold: $\delta_\rho^B = 0.026208$
(Part 3 of 3)

Reserved low-probability tokens: $\pi_\theta < \delta_\rho^B$ and $\pi_\theta > \tau$
Filtered low-probability tokens: $\pi_\theta < \delta_\rho^B$ and $\pi_\theta < \tau$
Normal tokens: $\pi_\theta > \delta_\rho^B$

find \(m + n \ ).\n\n Here , \(\ frac { m }{ n } = \ frac { 1 }{ 2 } \ ), so \(m = 1 \ ), \(n = 2 \ ), and thus \(m + n = 3 \ ).\n\n Wait , but before final izing , let 's double -check the recursive relation .\n\n ### Ver ifying the Recursive Relation \n\n Let 's consider a simpler case to verify :\n\n Sup pose we only do the first level of subdivision . Then , the shaded area is \(\ frac { 4 }{ 9 } \ ).\n\n According to our recursive relation \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ), if we only do level 1 , then perhaps the \(\ frac { S }{ 9 } \ ) should be zero , but our initial assumption that \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ ) holds for infinite subdivisions .\n\n Alternatively , perhaps it 's better to think that at each level , the shaded area added is correctly captured by \(\ frac { 4 }{ 9 } \ ) of the current area being considered , and the process continues similarly .\n\n Given that , our calculation seems consistent .\n\n ### Final Answer \n\n The fractional part of the figure that is shaded is \(\ frac { 1 }{ 2 } \ ). Therefore , \(m = 1 \ ), \(n = 2 \ ), and \(m + n = 3 \ ).\n\n Wait , but looking back at the problem , it mentions that the answer is in the form \(\ frac { m }{ n } \ ) with \(\ gcd (m , n) = 1 \ ), and we need to find \(m + n \ ). Here , \(\ frac { 1 }{ 2 } \ ) satisfies this with \(m = 1 \ ), \(n = 2 \ ), so \(m + n = 3 \ ).\n\n But before final izing , I recall that sometimes such problems have the shaded area as \(\ frac { 5 }{ 1 2 } \ ), but based on our reasoning , it 's \(\ frac { 1 }{ 2 } \ ). Double -check ing the recursive relation seems correct .\n\n Alternatively , perhaps the shaded parts are not just the immediate corners but also within the subdivisions in a different pattern . However , based on the As ym pt ote code and the description focusing on the center 's subdivision , our approach seems valid .\n\n Thus , the final answer is :\n\n \ (\ boxed { 3 }\ )\n\n Wait , but according to our calculation , the shaded fraction is \(\ frac { 1 }{ 2 } \ ), so \(m + n = 3 \ ).\n\n But to ensure accuracy , let 's consider another approach .\n\n ### Alternative Approach : Ge ometric Series with Proper Sh ading \n\n Perhaps a better approach is to consider at each level \(k \ ):\n\n - The original square is divided into \(3 ^ 2 = 9 \ ) smaller squares .\n - At each subdivision , 4 of these squares are shaded .\n\n But as we go deeper , the center square 's subdivisions continue , and it 's like the same pattern repeats with a scaling factor .\n\n Let 's think differently : label the initial square as level 0 . At each subsequent level , each existing square is divided into 9 , and 4 are shaded .\n\n But the center square continues to be subdiv ided , effectively repeating the process .\n\n Thus , the total shaded area can be thought of as the shaded area directly plus the shaded area from all subsequent levels of the center squares ' subdivisions .\n\n Math em atically , if \(S \ ) is the total shaded area , then :\n\n \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ )\n\n Because at each level , we add \(\ frac { 4 }{ 9 } \ ) of the current area being subdiv ided , and the remaining one -n inth contributes to the next level 's scaling .\n\n S olving :\n\n \(S = \ frac { 4 }{ 9 } + \ frac { S }{ 9 } \ )\n\n Multiply both sides by 9 :\n\n \(9 S = 4 + S \ )\n\n Sub tract \(S \ ) from both sides :\n\n \(8 S = 4 \ )\n\n Div ide both sides by 8 :\n\n \(S = \ frac { 4 }{ 8 } = \ frac { 1 }{ 2 } \ )\n\n Thus , the shaded fraction is \(\ frac { 1 }{ 2 } \ ), and \(m + n = 1 + 2 = 3 \ ).\n\n Therefore , the final answer is :\n\n \ (\ boxed { 3 }\ )\n\n ** Answer :** \ (\ boxed { 3 }\ ) <|endoftext|>

Figure 28: An Example generated by Qwen3-14B-Base model trained by Lp-Reg from math reasoning. (Part 3)