I^{2}B-LPO: Latent Policy Optimization via Iterative Information Bottleneck

Huilin Deng[1,2], Hongchen Luo [1*], Yue Zhu[5], Long Li[2],
Zhuoyue Chen[2,3], Xinghao Zhao[2], Ming Li[1], Chuyang Zhao[5],
Jihai Zhang[2], Mengchang Wang[2], Yang Cao[1,7*], Yu Kang[1,6],
[1]University of Science and Technology of China, [2]Alibaba DAMO Academy,
[3]Zhejiang University, [4]Shanghai Jiao Tong University,
[5]Beihang University, [6]Hefei University of Technology, [7]Hefei Comprehensive National Science Center
First Author: huilin_deng@mail.ustc.edu.cn
Correspondence: lhc12@mail.ustc.edu.cn forrest@ustc.edu.cn

Abstract

Despite recent advances in Reinforcement learning with verifiable rewards (RLVR) for large language model (LLM) reasoning, most methods suffer from exploration collapse, as the semantic homogeneity of random rollouts traps models in narrow, over-optimized behaviors. Existing methods leverage policy entropy to encourage exploration, but face inherent limitations: global entropy regularization is susceptible to reward hacking, inducing meaningless verbosity, whereas local token-selective updates struggle with the strong inductive bias of pre-trained models. To this end, we propose Latent Policy Optimization via Iterative Information Bottleneck (I^{2}B-LPO), which shifts from *statistical perturbation of token distributions* to *topological branching of reasoning trajectories*. I^{2}B-LPO triggers latent branching at high-entropy states to diversify reasoning trajectories and applies the Information Bottleneck as a trajectory filter and self-reward to ensure concise and informative exploration. Empirical results on four mathematical benchmarks demonstrate that I^{2}B-LPO achieves state-of-the-art performance, with margins of up to 5.3% in accuracy and 7.4% in diversity metrics. Code is available at https://github.com/denghuilin-cyber/IIB-LPO.

1 Introduction

Reinforcement learning with verifiable rewards (RLVR) has emerged as a key method for LLM reasoning, particularly in tasks with deterministic verification such as mathematics (DeepSeek-AI et al., 2025; Yang et al., 2024). This paradigm trains models to rollout and differentiate between multiple reasoning paths, reinforcing trajectories that lead to correct solutions while penalizing incorrect paths (Cheng et al., 2025). This contrastive approach has driven remarkable success across mathematical and broader reasoning tasks.

Despite this success, RLVR methods can unintentionally cause **exploration collapse** (Jiang et al., 2025), where models converge on narrow, over-optimized behaviors and lose their incentive to explore alternative strategies. This pathology stems from the semantic homogeneity of random rollouts (Ju et al., 2025). While these rollouts exhibit variations in surface phrasing, the underlying reasoning rapidly degenerates into a few high-probability *reasoning templates* (Zhang et al., 2025a), i.e., they share nearly identical reasoning patterns. Consequently, comparing such homogeneous paths yields vanishing advantage differentials and thus uninformative learning signals for policy optimization.

To mitigate this, policy entropy has been strategically leveraged to encourage exploration (Wang et al., 2025a). Existing approaches typically fall into two paradigms. Entropy-regularization methods (Chao et al., 2024) directly maximize the entropy of generated instances by globally smoothing the token-wise probability distribution (Fig. 1(a)). However, this global smoothing is susceptible to **reward hacking**—incentivizing the model to generate semantically vacuous verbosity (Yao et al., 2024). As shown in Fig. 1(a), the model tends to generate excessive meta-discourse irrelevant to the problem. Conversely, token-selective methods (Cui et al., 2025) amplify policy updates on high-entropy tokens, locally sharpening the next-token distribution at critical positions (Fig. 1(b)). Yet, such local sharpening struggles against the **inductive biases**. High-entropy positions frequently correspond to lexical ambiguity (e.g., synonym choice);

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 23647–23664
July 2-7, 2026

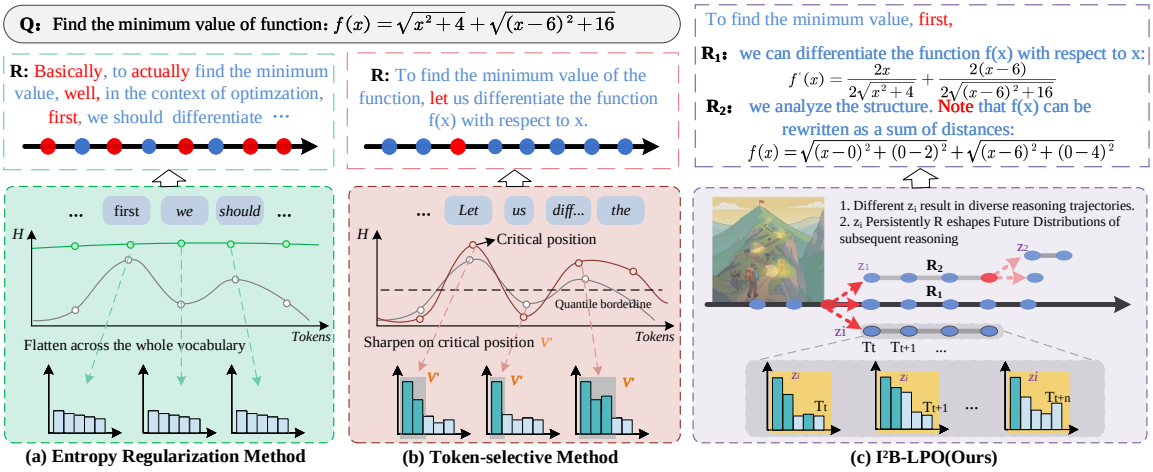


Figure 1: **Comparison of exploration paradigms in RLVR.** (a) Entropy Regularization globally smooths the probability distribution, leading to high-entropy yet meaningless verbosity. (b) Token-selective Methods locally sharpen the distribution; synonym replacement at these isolated points cannot overcome inductive biases. (c) I^{2}B-LPO introduces topological branching via latent variables z, resulting in distinct reasoning trajectories (e.g., Differentiation-based R_1 vs. Geometry-based R_2).

merely sharpening the distribution at these points is insufficient to overcome the strong prior of pre-trained models (Casper et al., 2023). Consequently, both paradigms are confined to probabilistic perturbations, lacking the capacity to induce structural diversification in the reasoning process itself.

To address these limitations, this paper proposes a fundamental paradigm shift: moving from *statistical perturbation* of token distributions to *topological bifurcation of reasoning trajectories* (Fig. 1(c)). We introduce Latent Policy Optimization via Iterative Information Bottleneck (I^{2}B-LPO), which triggers latent branching at high-entropy states to shatter inductive biases and utilizes information-theoretic constraints to curb reward hacking. As illustrated in Fig. 4, the method operates through two key mechanisms: (1) Entropy-driven Latent Branching utilizes a Conditional Variational Autoencoder (CVAE) (Fang et al., 2021) to sample diverse latent variables z_i at detected bifurcation (high-entropy states). Each z_i serves as a structural prompt, injected into the LLM's attention layers, continually steering the trajectory of subsequent reasoning. (2) Information Bottleneck Regularization functions as a dual-purpose filter and self-reward. By quantifying the trade-off between rationale compression and predictive power (Lei et al., 2025a), the IB objective identifies compact, informative paths for policy updates while simultaneously penalizing semantically vacuous verbosity.

Empirical results demonstrate I^{2}B-LPO's SOTA performance in both reasoning accuracy and semantic diversity, without excessive generation length. The significant margins in accuracy (5.3%) and diversity (7.4%) validate its efficacy in prompting exploration. The main contributions are summarized as follows: (1) Paradigm shifts from statistical token perturbation to topological trajectory branching. (2) Entropy-Driven Latent Branching explicitly restructures the reasoning topology to induce trajectory diversity. (3) Dual-Purpose IB uses it both as a trajectory filter and self-reward, favoring concise, informative reasoning. (4) SOTA performance in both accuracy and diversity without excessive length.

2 Related Work

Entropy in RL Entropy quantifies the dispersion of vocabulary distribution and reflects the predictive uncertainty of LLMs. Thus, it strategically guides the exploratory behavior of LLMs (Chao et al., 2024; Deng et al., 2025). Existing methods typically fall into two categories. Token-selective methods target high-entropy tokens for selective updates (Wang et al., 2025a). Regularization-based methods, such as concurrent works by (Wang et al., 2025b) and (Cheng et al., 2025), incorporate an entropy regularizer directly into the loss or advantage function. Most priors rely on indiscriminate regularizers on token distribution. Instead, we propose a paradigm shift——conditional branching for effective exploration.

3 Preliminary Analysis

Pivotal Decision Points. To validate whether entropy signals pivotal reasoning steps, we sampled

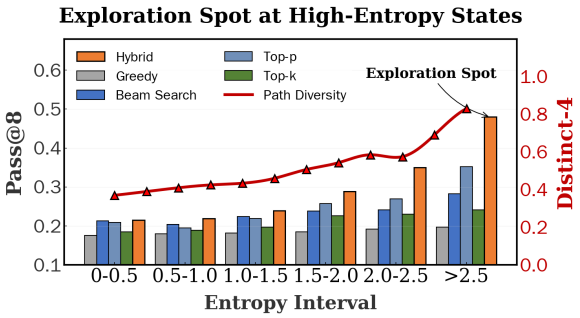


Figure 2: Performance of various decoding strategies trained on DeepMath. For each problem, we sample truncation points across entropy intervals to simulate varied exploration behaviors.

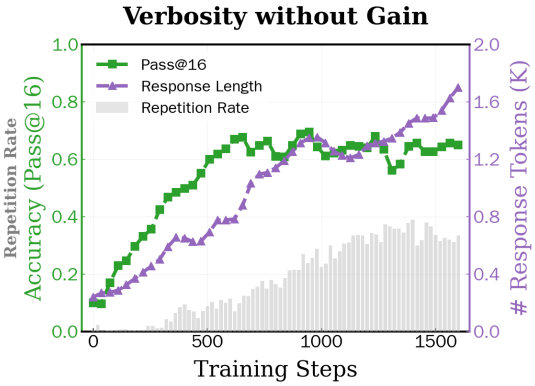


Figure 3: Accuracy and Response Length under GRPO. Notably, gray bars denote 4-gram repetition rate.

prefixes conditioned on token-level entropy to simulate different exploration strategies via various decoding methods. Fig. 2 shows that in high-entropy intervals (>2.5) hybrid strategy substantially outperforms baselines, a performance gap far larger than in low-entropy regions (<1.0). This confirms high-entropy states as decision points; branching here unlocks exploration potential.

Verbosity without Gain. As shown in Fig. 3, when applying standard GRPO, the accuracy plateaus early while the response length continues to rise, accompanied by a growing 4-gram repetition rate (gray bars). This indicates that without additional guidance, the model tends to produce verbose and repetitive reasoning without genuine performance gains. Therefore, we introduce the IB self-reward to suppress such unproductive expansion.

4 Methodology

4.1 Problem Formulation

Formally, let **q** denote input prompt, **r** the reasoning trajectory. We initialize the policy optimization with GRPO, starting with M initial rollouts $\mathcal{R}_o = \{r_1, \ldots, r_M\}$. I^{2}B-LPO comprises two phases:

- **Entropy-driven Branching**: For each base rollout r_i, we generate K distinct branches while retaining the original one. This yields a branching set $\mathcal{R} = \{r_{i,j} \mid 1 \le i \le M, 1 \le j \le K+1\}$, containing $M \times (K+1)$ paths. Each path $r_{i,j}$ consists of a sequence of tokens $(o_1, \ldots, o_{T_{i,j}})$.
- **IB-Pruning**: we eliminate samples with low IB-scores from $\mathcal{R}$, retaining a high-quality subset $\mathcal{R}^* \in R$ such that $|\mathcal{R}^*| = N$.

4.2 Entropy-driven Latent Branching

4.2.1 Entropy-Driven Bifurcation Detection

For a given trajectory $r = (o_1, \ldots, o_T)$, we identify a "bifurcation point" $t^* \in \{1, \ldots, T\}$ corresponding to a state of high uncertainty. The token-level entropy H_t at step t is computed as:

$$H_t = -\sum_{v \in \mathcal{V}} P(v \mid q, o_{<t}) \log P(v \mid q, o_{<t}), \quad (1)$$

where $\mathcal{V}$ is the vocabulary. Following, we select the top 5% highest-entropy steps as candidate Ω (τ: the 95$^{\text{th}}$ percentile of entropy history $\mathcal{H}$):

$$\Omega = \{t \mid H_t \ge \tau\}, \quad t^* \sim \text{Uniform}(\Omega). \quad (2)$$

Finally, we extract the prefix context c_{t^*} by concatenating the query q with the partial path preceding the split $c_{t^*} = [q, o_1, \ldots, o_{t^*-1}]$.

4.2.2 Latent Sampling via CVAE

Given the prefix c_{t^*}, we employ a separately trained **Conditional Variational Autoencoder (CVAE)** to sample latent variables z. Implementation of CVAE is detailed in Appendix B. Formally, the CVAE models the conditional distribution of a solution trajectory y given context via a latent variable:

$$p(y \mid x) = \int_z p(y \mid z, x)\, p(z \mid x)\, dz, \quad (3)$$

where the **prior network** $p(z \mid x)$ captures the distribution of z_i. During training, for each r_i, we sample K independent latent codes from prior:

$$z^{(j)} \sim p(z \mid c_{t^*}), \quad j = 1, \ldots, K. \quad (4)$$

4.2.3 Latent-Guided Reasoning via PSA

This section explores ***how latent variables $\mathbf{z}_j$, generated by a CVAE, influence subsequent reasoning in LLMs.*** We employ **Pseudo Self-Attention (PSA)**, incorporating the latent code $z \in \mathbb{R}^d$ into the self-attention mechanism at a per-layer basis. The detailed injection is as follows:

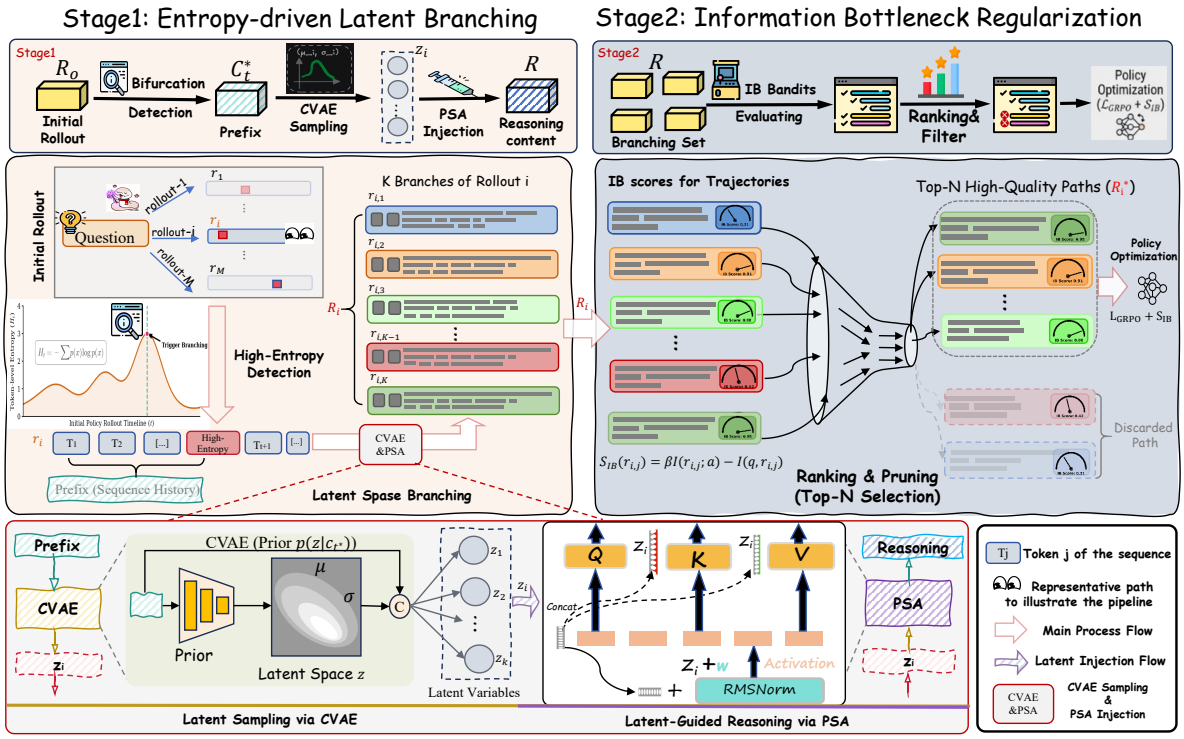


Figure 4: **Pipeline of the I^{2}B-LPO.** We use a representative path r_i from the initial set R_o to illustrate the workflow, which operates in two phases. (1) **Entropy-driven Latent Branching** expands r_i into the branching set R_i via Latent Sampling and PSA Injection, which are depicted in the bottom section. (2) **Information Bottleneck Regularization** applies IB as a dual-purpose filter and self-reward to ensure concise and informative exploration.

- *Phase 1: Adaptive Norm Modulation.* The latent code is projected and added to the learnable scaling parameter of RMSNorm:

$$w_l' = w_l + \gamma(t) \cdot \text{Proj}_\phi(z_j), \tag{5}$$

where $\gamma(t)$ is a dynamic decay factor that anneals the latent influence over time.

- *Phase 2: Augmented Self-Attention.* The latent code is further projected into modulation vectors and concatenated with original keys and values:

$$K_j' = \begin{bmatrix} z_j K \\ K \end{bmatrix}, \quad V_j' = \begin{bmatrix} z_j V \\ V \end{bmatrix} \in \mathbb{R}^{(1+l)\times d} \tag{6}$$

where the notation $(\cdot)$ indicates row-wise concatenation. The enhanced attention is computed as:

$$\text{PSA}(Q, K_j', V_j') = \text{softmax}\left(\frac{QK_j'^T}{\sqrt{d_k}}\right) V_j'. \tag{7}$$

Thus, z_i implicitly shifts features via RMSNorm and explicitly guides attention as a structural prompt, thereby steering the reasoning trajectory.

4.3 Information Bottleneck Regularization

4.3.1 IB Score: Theory and Computation

This section formulates the IB score and its approximation. We formulate LLM reasoning as an optimization trade-off between **rationale compression** and its **predictive power**. Specifically, an optimal policy minimizes prompt retention while maximizing the informativeness of final answers:

$$\min \mathcal{L}_{\text{IB}}(r) = I(q; r) - \beta I(r; a), \tag{8}$$

where $I(\cdot;\cdot)$ denotes Mutual Information (MI). The MI between r and a can be decomposed as:

$$I(r; a) = H(r) - H(r \mid a), \tag{9}$$

where $H(\cdot)$ denotes entropy. Eq. 9 implies that maximizing $I(r; a)$ balances exploration (diversity $H(r)$) and precision (low ambiguity $H(r|a)$).

Formally, we define $S_{\text{IB}}(r) \equiv -\mathcal{L}_{\text{IB}}(r)$ at the trajectory-level, where higher values indicate a better balance between compression and informativeness. Based on the derivation in Appendix. C.2, we approximate the $S_{\textbf{IB}}(r)$ as:

$$\begin{aligned} S_{\text{IB}}(r) &= \frac{1}{T}\sum_{t=1}^{T}\Big(\log \pi(o_t \mid o_{<t}, q) + \lambda \cdot H(o_t \mid o_{<t}, q)\Big) \\ &= \frac{1}{T}\sum_{t=1}^{T} \mathcal{A}_t H(o_t \mid o_{<t}, q), \end{aligned} \tag{10}$$

where T is the response length. $H(o_t \mid o_{<t}, q)$ and $\mathcal{A}_t$ denote the policy entropy and advantage function at step t, respectively.

Table 1: **Overall Performance Comparison across 4 Mathematical Benchmarks.** We evaluate I^{2}B-LPO on two different backbones: **Qwen2.5-7B** and **Qwen3-14B**. We report **Pass@n** accuracy (%) alongside average response length (**#Tok**). **Bold** denotes the highest accuracy, while *colored italics* indicate the longest average response length.

Method	AIME2025			AIME2024			MATH-500			Olympiad		
	P@1	P@256	#Tok	P@1	P@256	#Tok	P@1	P@16	#Tok	P@1	P@16	#Tok
Qwen2.5-7B												
+GRPO (Standard)	8.2	50.0	1292	10.3	46.7	776	54.4	58.4	661	44.9	61.6	762
Entropy-Reg.	9.3	47.7	*3198*	12.7	55.6	*2358*	57.4	70.4	*1802*	48.9	63.4	1823
Entropy-Adv	11.8	53.3	2187	13.6	56.7	1424	58.5	74.0	1223	51.8	64.6	*1894*
KL-Cov	11.3	52.1	1878	15.2	72.3	1283	68.2	82.1	1488	53.0	65.4	1190
80/20	10.2	47.9	1898	16.2	70.5	1315	61.8	79.1	1190	49.6	62.2	1479
SPINE	11.2	52.7	1634	14.4	68.2	1179	76.2	86.5	1104	52.5	67.8	1389
SRLM	9.7	48.6	1574	15.1	65.1	1334	73.4	84.7	1058	46.5	61.0	1167
I^{2}B-LPO (Ours)	**13.6**	**55.0**	1465	**18.6**	**79.7**	1245	**81.5**	**90.5**	1080	**58.0**	**69.5**	1172
Qwen3-14B												
+GRPO (Standard)	27.0	62.3	2045	34.4	67.8	1437	89.2	92.9	825	55.7	67.8	894
Entropy-Reg.	28.8	59.8	*4285*	37.3	58.9	*2963*	90.2	93.0	*2387*	58.7	68.6	1880
Entropy-Adv	27.8	61.3	2287	42.8	63.1	2265	89.5	93.5	1626	57.8	70.6	1709
KL-Cov	34.6	62.8	2390	45.4	80.6	2018	91.7	93.6	1798	62.0	76.8	1801
80/20	33.5	66.2	2168	43.9	78.5	1989	91.6	94.1	1488	60.8	73.2	*2290*
SPINE	28.4	59.6	2034	39.2	70.6	1966	89.9	93.2	1392	65.5	78.3	1643
SRLM	29.7	58.3	2201	37.2	69.4	1876	91.1	94.6	1892	62.5	71.7	1686
I^{2}B-LPO (Ours)	**38.3**	**68.1**	2465	**46.6**	**82.5**	1830	**93.5**	**95.7**	1432	**68.0**	**82.5**	1581

4.3.2 IB-Guided Pruning and Optimization

Given the candidate set $\mathcal{R}$ derived from entropy-driven branching, we refine the pool of reasoning paths by retaining the top-N trajectories based on their IB scores. Formally, obtaining this optimal subset $\mathcal{R}^*$ is equivalent to solving:

$$\mathcal{R}^* = \arg\max_{\substack{\mathcal{S}\subseteq\mathcal{R} \\ |\mathcal{S}|=N}} \sum_{r\in\mathcal{S}} S_{\text{IB}}(r), \tag{11}$$

where $\mathcal{S}$ denotes any subset of $\mathcal{R}$ with $|\mathcal{S}| = N$. In practice, we first compute $S_{\text{IB}}(r)$ for each $r \in \mathcal{R}$ using Eq. (10), sort the results, and retain only the top N paths. The pruned set $\mathcal{R}^*$ then serves as the training data for policy optimization. We also incorporate the IB score as an **auxiliary maximization objective** during training:

$$\mathcal{S}_{\text{IB}}(\theta;\mathcal{R}^*) = \frac{1}{N}\sum_{r\in\mathcal{R}^*} S_{\text{IB}}(r). \tag{12}$$

$$\mathcal{J} = \mathcal{J}_{\text{GRPO}} + \gamma\cdot\mathcal{S}_{\text{IB}}(\theta;\mathcal{R}^*), \tag{13}$$

where γ is a coefficient balancing task correctness and reasoning efficiency.

5 Experiment Settings

Training Configuration. We conduct experiments on GRPO using the veRL framework. To build strong baselines, we adopt several techniques from DAPO, including Clip-Higher and Group-Sampling. Detailed settings are in Appendix E.

Datasets. Our training data are sourced from DAPO and MATH. To ensure training efficiency, we filter samples that are either too trivial or intractable (see Appendix E for details). For evaluation, four benchmarks, including MATH-500 (Hendrycks et al., 2021), AIME2025, AIME24, Olympiadbench (He et al., 2024), are selected.

Table 2: **Diversity Analysis on GSM8K Benchmark. Accuracy** is measured by Pass@1. **Diversity** is evaluated via Distinct-1, Distinct-4, 1−Self-BLEU, and 1−Self-ROUGE (higher is better for all metrics).

Method	Pass@1	Distinct-1	Distinct-4	Self-BLEU	Self-ROUGE
Qwen2.5-3B					
Base	85.7	0.20	0.44	0.25	0.28
80/20	89.1	0.26	0.48	0.30	0.36
Entropy.Reg	86.8	0.32	**0.78**	0.82	0.78
I^{2}B-LPO	**92.8**	**0.35**	0.76	**0.85**	**0.81**
Qwen2.5-7B					
Base	91.6	0.28	0.51	0.34	0.39
80/20	93.5	0.35	0.65	0.38	0.45
Entropy.Reg	93.8	0.57	0.73	0.47	**0.81**
I^{2}B-LPO	**95.6**	**0.69**	**0.87**	**0.56**	0.76
Qwen3-14B					
Base	92.5	0.33	0.59	0.38	0.43
80/20	94.2	0.37	0.62	0.41	0.45
Entropy.Reg	93.4	0.56	0.71	0.51	0.57
I^{2}B-LPO	**97.8**	**0.59**	**0.73**	**0.57**	**0.63**

Baselines. We conduct experiments on Qwen2.5-37B and Qwen3-14B. Baselines span three categories: (1)entropy regularization (Entropy-Reg (Chao et al., 2024) and Entropy-Adv (Cheng et al., 2025)) (2) token-selective methods (KL-Cov (Cui et al., 2025) and 80/20 (Wang et al., 2025a)), and (3) self-reward methods (SPINE (Wu et al., 2025) and SRLM (Yuan et al., 2024)).

Metrics. ***Avg. # Tokens*** reports the average token length. For diversity, we report Distinct-n (Li et al., 2016) and Self-BLEU (Papineni et al., 2002). (1) ***Distinct-n*** counts the ratio of unique n-grams. (2) ***Self-BLEU***. We report 1 − Score, with higher values indicating greater diversity.***Perplexity***

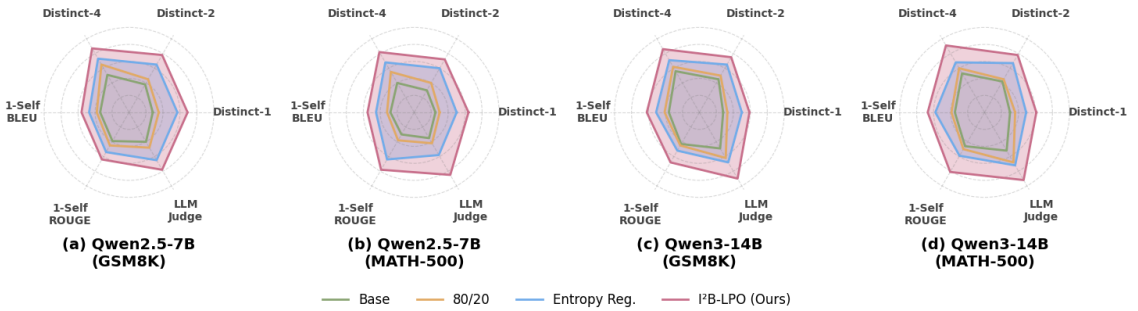
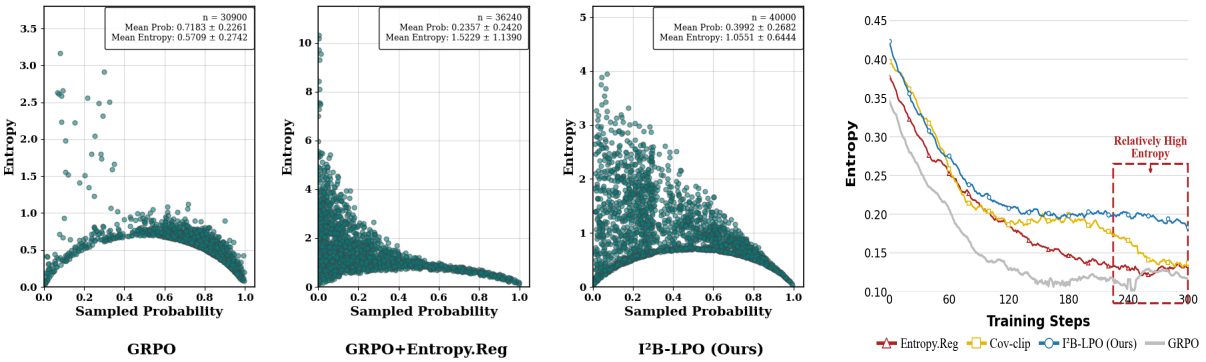


Figure 5: **The performance of the trained model on six diversity metrics.** We evaluate I^{2}B-LPO using Qwen2.5-7B and Qwen3-14B models across the GSM8K and MATH-500 datasets. For each metric, a higher value indicates greater diversity. And the diversity metrics are calculated across 10 generated responses per prompt.



(a) Probability-Entropy scatter plots (b) Entropy Dynamic comparison

Figure 6: **Joint Analysis of Entropy Dynamics.** (a) Probability-Entropy scatter plots of five exploratory tokens from training samples at training step 500 on Qwen2.5-7B-Base, displaying a random sample of 5% of all data points. (b) Average entropy dynamics across training steps on the MATH dataset.

(PPL) measures uncertainty of the generated sequence(Appendix.E). Low-PPL responses are generally more fluent and semantically coherent.

6 Experiment

This section evaluates I^{2}B-LPO's ability to generate diverse, high-quality responses. We aim to answer the following research questions:

- **RQ1 (Overall Performance):** Does I²B-LPO outperform baseline methods in terms of both reasoning accuracy and semantic diversity?
- **RQ2 (Exploration Behavior Analysis):** How does the entropy-triggered mechanism reshape exploration dynamics? And how does latent injection steer subsequent reasoning—introducing structured exploration or random noise?
- **RQ3 (IB Pruning Efficacy):** How does the IB-based self-reward compare to other self-reward methods, and what specific reasoning characteristics correspond to high and low IB scores?
- **RQ4 (Ablation Study):** How does each component of I²B-LPO contribute to performance?

6.1 Quality-diversity balance (RQ1)

In this section, we present fine-grained results on the diversity and quality.

Quality. Tab. 1 reports Pass@n ($n \in [1, 256]$) and average response length. I^{2}B-LPO consistently outperforms baselines across both backbones, with advantages widening as n increases. Unlike Entropy-Reg., which artificially inflates entropy via excessive verbosity, I^{2}B-LPO maintains a better balance between quality and efficiency.

Diversity. We present diversity metrics in Fig. 5 and Tab. 2. I^{2}B-LPO consistently outperforms baseline models, demonstrating a clear advantage in diversity. While entropy-based methods primarily boost lexical variation (Distinct-n), our approach achieves superior performance on semantic metrics (LLM-as-Judge and Self-BLEU).

6.2 Exploration Behavior Analysis (RQ2)

This section analyzes the distribution of exploratory tokens and entropy dynamics.

Distributions of Exploratory Tokens. As categorized in Appendix A, high-entropy tokens fulfill

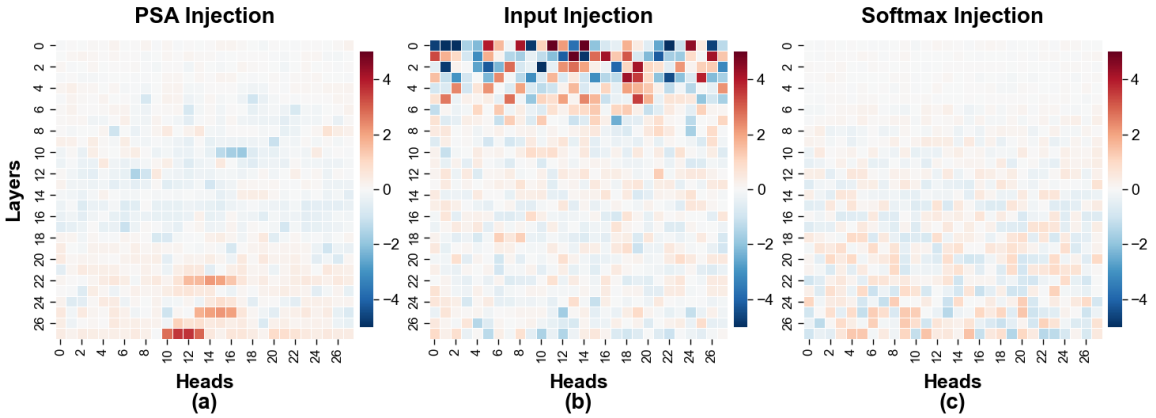
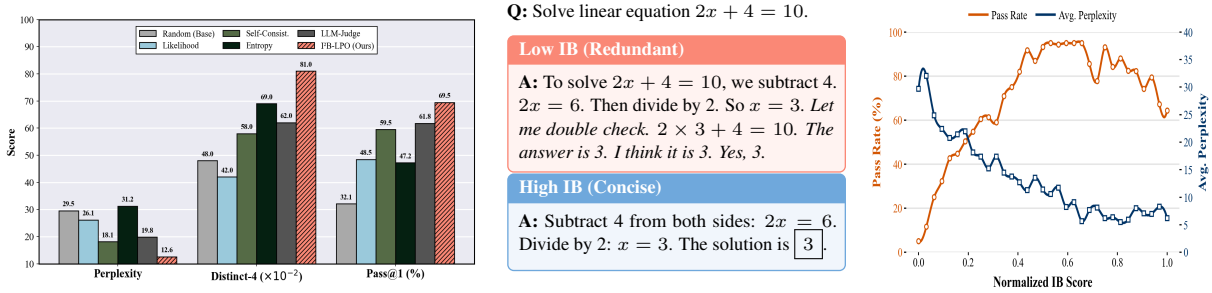


Figure 7: Attention head patterns contrasted between high (Level 9) and low (Level 3) difficulty on Deepmath. Red indicates heads activated by complex problems, while blue denotes heads responsive to simpler ones.



(a) Performance Comparison (b) Case Study (c) Accuracy & Perplexity Trends

Figure 8: **Overview of Information Bottleneck (IB) Analysis.** (a) Performance comparison on OlympiadBench. (b) Case study illustrating how higher IB scores correlate with more concise reasoning. (c) The impact of IB scores on model accuracy and perplexity.

distinct functional roles. Fig. 6 (a) compares the probability–entropy distributions of exploratory tokens ("but", "however", "thus", "wait", "let"). The key observations are summarized as follows:

- In GRPO, tokens concentrate in high-probability, low-entropy regions, similar to (Ju et al., 2026).
- Adding an entropy loss does shift this distribution, but often leads to anomalously high entropy (>8) for some tokens.
- I²B-LPO maintains a balanced distribution by activating exploratory tokens across a wider entropy spectrum, preserving their reasoning utility without overfitting into deterministic patterns.

Mitigation of Entropy Collapse. Fig. 6 (b) tracks entropy dynamics. While all methods show an initial drop, GRPO suffers from continuous decay, indicating severe entropy collapse. In contrast, I²B-LPO stabilizes entropy levels after the initial phase. This confirms our method effectively preserves exploration capacity, preventing convergence toward deterministic patterns.

Ablation Study of Latent Injection To verify whether latent injection fosters structured exploration rather than introducing random noise. We investigate three injection strategies: Early Fusion at input-level, Deep Fusion via PSA (detailed in Sec . 4.2.3), and Late Fusion at Softmax-layer. The formulations are described below:

① *Early Fusion at input-level:* We inject the latent code z_i by adding it element-wise to the input embedding $h(x_t)$ of each token x_t. This broadcasts the prior across the sequence dimension:

$$h'_i(x_t) = h(x_t) + z_i, \quad \forall t = 1, \ldots, T. \tag{14}$$

② *Late Fusion at Softmax-layer:* This strategy directly utilizes the projection of z_i to influence the distribution of LLM's vocabulary. We first map the latent vector z_i to a logit adjustment vector $p_{z_i} \in \mathbb{R}^V$. This p_{z_i} is then superimposed onto the original logits p to derive the final distribution:

$$p'_i = p + p_{z_i}, \quad \hat{y} = \text{softmax}(p'_i), \tag{15}$$

where p_{z_i} biases the token selection probability $\hat{y}$.

Analysis. Tab. 3.A presents ablation results that isolate the contribution of the structured semantic prior learned by the CVAE: random noise increases diversity (Dist-4) but lowers accuracy and raises

perplexity, while the CVAE provides both diversity and semantic coherence. Fig. 7 contrasts attention patterns for high- (Level 9) and low-difficulty (Level 3) problems on DeepMath (details in Appendix D). Key observations are:

- Input Injection: chaotic early activations diminish in deeper layers, diluted before reasoning.
- Softmax Injection: modifying logits externally hinders gradients from shaping internal attention, leading to scattered activations.
- PSA induces "Structured Activation": it mobilizes difficulty-sensitive heads in final layers (24–27), consistent with (Shojaee et al., 2025; Deng et al., 2026).

This confirms that PSA effectively engages deep reasoning without aimless randomness.

6.3 Self-Reward Effectiveness (RQ3)

In this section, we benchmark IB metric against other self-rewards methods: Likelihood (Wu et al., 2016; Deng et al., 2024), Self-Consistency (Wang et al., 2023), Entropy.Reg (Chao et al., 2024), and LLM-Judge (Zheng et al., 2024). As shown in Fig. 8(a), IB self-reward outperforms baselines across accuracy, diversity, and confidence (perplexity). Notably, it resolves the diversity-confidence trade-off observed in Entropy.Reg, which sacrifices confidence for diversity. Fig. 8(b) shows that higher IB scores correspond to concise reasoning chains, while lower scores identify redundant content. Fig. 8(c) further demonstrates that IB metric balances exploration and confidence by reducing perplexity while improving quality.

6.4 Ablation of Main Components (RQ4)

Tab. 3 presents ablation results on the core components. The results indicate that (1) entropy-based branching outperforms other strategies by accurately targeting high-uncertainty nodes; (2) the full IB mechanism (loss + pruning) forms an explore–converge loop, significantly reducing PPL by 11.7. (3) Overall, the complete framework elevates MATH Pass@1 from 54.4% to 81.5%, maintaining high diversity and low perplexity. Tab. 4 shows ablation results of key hyperparameters. PSA injection depth follows an inverted U-curve, peaking at the last 12 layers. The max_length of 8192 tokens provides sufficient capacity for complex reasoning.

6.5 Qualitative Analysis

This section analyzes IB-related failure cases with representative examples. As shown in Appendix C.4 Tab.6, typical low-IB cases are categorized into three distinct failure modes: (1) Vacuous Verbosity, (2) Repetitive Loops, and (3) Logical Drift.

Example 1: Log Equation Minimum

Q: From the equation $\log_4(x+2y)+\log_4(x-2y)=1$, find k such that the minimum of $x-|y|$ is $\sqrt{k}$.

(0) High-IB Trajectory (Success, $S_{IB}=0.85$, Reward=1.0):

A: To solve, combine logs: $\log_4((x+2y)(x-2y))=1$, so $x^2-4y^2=4$. Thus $x=\sqrt{4+4y^2}$ (positive for min). Minimize $f(y)=\sqrt{4+4y^2}-|y|$ for $y\geq 0$: $f'(y)=\frac{4y}{\sqrt{4+4y^2}}-1=0$, yields $y=\frac{1}{\sqrt{3}}$, $f(y)=\sqrt{3}$. So $k=3$. $\boxed{3}$

(1) Failure Mode A: Vacuous Verbosity ($S_{IB}=0.15$, Reward=0.0)

A: It is important to note that we start by considering the equation carefully. Basically, combining the logs seems like a good idea, let me think step by step. As we all know, log addition means multiplication, so perhaps $\log_4((x+2y)(x-2y))=1$, which implies something. It is crucial to emphasize that $x^2-4y^2=4$, and we should minimize $x-|y|$. Let me reflect: for positive x, $x=\sqrt{4+4y^2}$. Now, to be thorough, the derivative is key, but first, let's recall that minimization requires setting derivative to zero. After some thought, $y=1/\sqrt{3}$, and $\min=\sqrt{3}$, so $k=3$. $\boxed{3}$

Analysis: High $I(q;r)$ due to meta-phrases ("It is important", "Let me think") adding length without info.

(2) Failure Mode B: Repetitive Loops ($S_{IB}=-0.85$, Reward=0.0)

A: The equation is $\log_4(x+2y)+\log_4(x-2y)=1$. The equation is $\log_4(x+2y)+\log_4(x-2y)=1$. Combine: $x^2-4y^2=4$. Combine again: $x^2-4y^2=4$. $x=\sqrt{4+4y^2}$. $x=\sqrt{4+4y^2}$. Minimize $f(y)=\sqrt{4+4y^2}-y$. Minimize $f(y)=\sqrt{4+4y^2}-y$. Derivative: $4y/\sqrt{4+4y^2}-1=0$. Derivative: $4y/\sqrt{4+4y^2}-1=0$. $y=1/\sqrt{3}$, $\min=\sqrt{3}$, [...repeats expansion-verification loop for 1024+ tokens...] $\boxed{3}$

Analysis: Extreme $I(q;r)$ from repetitions (equation restated 2x, steps looped) → No new info gain, pruned.

(3) Failure Mode C: Logical Drift ($S_{IB}=0.20$, Reward=0.0)

Table 3: **Ablation Study of I²B-LPO Components on Qwen2.5-7B.** We evaluate the contribution of core components: Branching Triggers, Latent Injection, and IB self-rewards. To isolate the impact of topological structures, Blocks A and B are conducted without the IB self-reward mechanism. **Note:** non-ablated components are fixed to optimal settings($K = 7$, $N = 8$). † denotes the PSA Fusion without the injection weight decay.

Method	AIME25			AIME24			MATH			OlympiadBench		
	Pass@1	Dist-4	PPL	Pass@1	Dist-4	PPL	Pass@1	Dist-4	PPL	Pass@1	Dist-4	PPL
GRPO Baseline	8.2	0.16	32.2	10.3	0.19	27.8	54.4	0.40	17.6	44.9	0.32	19.8
A: Latent Injection Ablation (Fixed: Entropy Branching+ w/o IB)												
w/o Latent Injection	10.2	0.34	32.2	15.5	0.39	27.8	59.7	0.64	18.4	48.3	0.50	19.5
Noise Injection (=1.0)	8.0	0.31	36.1	10.0	0.36	33.1	56.1	0.48	22.0	41.8	0.45	25.1
Noise Injection (=5.0)	6.4	0.41 | 42.5	8.9	0.43	40.2	45.2	0.63	28.0	38.0	0.56	31.0	
+ Input-Level Fusion	10.6	0.38	30.5	16.2	0.48	26.5	61.2	0.68	16.2	49.5	0.52	19.8
+ Softmax-Level Fusion	10.8	0.41	38.6	16.4	0.45	33.4	63.5	0.66	21.5	50.8	0.55	25.4
+ Norm Modulation	11.0	0.43	36.5	17.0	0.46	31.8	64.2	0.72	20.8	51.2	0.56	24.5
+ KV-Augmentation	11.2	0.44	35.8	17.2	0.47	31.0	65.1	0.74	20.1	51.6	0.57	23.8
+ PSA Fusion†	11.6	0.45	35.1	17.2	0.48	30.5	65.4	0.75	19.2	51.9	0.57	23.0
+ PSA Fusion (Ours)	11.8	0.47	34.2	17.5	0.49	29.8	66.7	0.78	19.5	52.3	0.59	22.1
Δ *vs. w/o Latent Injection*	**+1.6**	**+0.13**	+2.0	**+2.0**	**+0.1**	+2.0	**+0.1**	**+0.14**	+1.1	**+4.0**	**+0.09**	+2.6
B: Branching Trigger Ablation (Fixed: PSA + w/o IB)												
w/o Branching ($K = 0$)	8.6	0.21	30.1	13.4	0.23	26.7	54.5	0.43	17.3	45.3	0.35	19.1
+ Random Branching	9.1	0.24	31.9	13.6	0.34	25.9	55.4	0.46	17.8	46.7	0.40	20.5
+ Likelihood Branching	9.4	0.26	32.0	14.2	0.37	27.3	56.7	0.51	18.1	47.9	0.45	19.8
+ Entropy Branching	10.2	0.34	32.2	15.7	0.39	27.8	59.7	0.64	18.4	48.3	0.50	19.5
Δ *vs. w/o Branching*	**+1.6**	**+0.13**	+2.1	**+2.3**	**+0.16**	+1.1	**+5.2**	**+0.21**	+1.1	**+3.0**	**+0.15**	+0.4
C: IB Mechanism Ablation (Fixed: Entropy + PSA)												
w/o IB	11.8	0.47	34.2	17.5	0.49	29.8	66.7	0.78	19.5	52.3	0.59	22.1
+ IB Loss	12.2	0.49	24.1	18.1	0.52	20.4	78.3	0.76	14.2	54.6	0.60	14.5
+ IB Pruning	12.9	0.50	25.0	18.3	0.51	25.4	79.6	0.80	15.6	56.8	0.65	16.5
I²B-LPO (Full Method)	**13.6**	**0.51**	22.5	**18.6**	**0.54**	19.2	**81.5**	**0.82**	13.1	**58.0**	**0.65**	12.6
Δ *vs. w/o IB*	**+1.8**	**+0.04**	**-11.7**	**+1.1**	**+0.05**	**-10.6**	**+14.8**	**+0.04**	**-6.4**	**+5.7**	**+0.06**	**-9.5**

Table 4: **Hyperparameter Sensitivity Analysis.** Blue indicates Injection Depth; Green indicates Branching Factor (K); Yellow indicates Max Response Length. The bold rows denote the chosen settings.

Configuration	MATH		OlympiadBench	
	Pass@1	Dist-4	Pass@1	Dist-4
PSA Injection Depth				
Last 4 Layers	76.5	0.75	53.2	0.58
Last 8 Layers	79.8	0.79	56.4	0.62
Last 12 Layers	**81.5**	**0.82**	**58.0**	**0.65**
Last 16 Layers	81.1	0.81	57.6	0.64
Last 20 Layers	80.4	0.80	56.9	0.63
Last 28 Layers	78.2	0.78	55.1	0.61
Branching Factor				
K=1	75.5	0.75	52.8	0.54
K=3	78.7	0.78	55.8	0.60
K=7	**81.5**	**0.82**	**58.0**	**0.65**
K=15	82.8	0.86	58.4	0.69
Max Response Length				
2048 Tokens	68.2	0.70	45.5	0.55
4096 Tokens	79.5	0.80	56.2	0.63
8192 Tokens	**81.5**	**0.82**	**58.0**	**0.65**

A: Combine logs: $\log_4(x^2 + 4y^2) = 1$ (error: wrong product). So $x^2 + 4y^2 = 4$, $x = \sqrt{4 - 4y^2}$. Minimize $x - |y| = \sqrt{4 - 4y^2} - y$. Derivative: $-4y/\sqrt{4 - 4y^2} - 1 = 0 \Rightarrow y = 0$, min $= 2$. So $\sqrt{k} = 2$, $k = 4$. $\boxed{4}$

Analysis: Low $I(r; a)$ (wrong product, leads to invalid min) despite conciseness → Pruned as irrelevant.

7 Conclusion

This paper addresses the exploration collapse caused by the semantic homogeneity of reasoning paths. Our I²B-LPO framework enables structured exploration via entropy-driven latent branching and a dual-purpose IB for simultaneous filtering and reward shaping. Extensive experiments confirm that I²B-LPO achieves SOTA accuracy and diversity, effectively overcoming the exploration bottleneck. This work provides a principled and extensible framework for exploration-aware policy optimization in LLM reasoning.

Limitations and Potential Risk

Limitations. First, entropy may not consistently reflect semantic diversity in open-ended tasks. Second, despite IB-based pruning, the branching mechanism remains more computationally intensive than single-rollout approaches. We aim to explore more efficient algorithms in the future.

Potential Risk. Our method focuses on mathematical reasoning tasks using the public benchmarks. It does not involve sensitive data or content generation, so it poses minimal ethical risks.

Acknowledgments

This work is supported in part by the National Natural Science Foundation of China (U25A20454, 92467302), the Key Science & Technology Project of Anhui Province (No.202523j08050018), and the Central Guidance for Local Science and Technology Development Fund Project (No. 2025ZY01119).

References

Daniel Adiwardana, Minh-Thang Luong, David R So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V Le. 2020. Towards a human-like open-domain chatbot. *arXiv preprint arXiv:2001.09977*.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Stephen Casper, Xander Davies, Thomas Shi, Claudia andz Benton, Michael Mozer, and 1 others. 2023. Open problems and fundamental limitations of reinforcement learning from human feedback. *Transactions on Machine Learning Research*. Also available as arXiv:2307.15217.

Chen-Hao Chao, Chien Feng, Wei-Fang Sun, Cheng-Kuang Lee, Simon See, and Chun-Yi Lee. 2024. Maximum entropy reinforcement learning via energy-based normalizing flow. *Preprint*, arXiv:2405.13629.

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

Huilin Deng, Hongchen Luo, Wei Zhai, Yang Cao, and Yu Kang. 2026. Vmad: Visual-enhanced multimodal large language model for zero-shot anomaly detection. *Preprint*, arXiv:2409.20146.

Huilin Deng, Hongchen Luo, Wei Zhai, Yanming Guo, Yang Cao, and Yu Kang. 2024. Prioritized local matching network for cross-category few-shot anomaly detection. *IEEE Transactions on Artificial Intelligence*, 5(9):4550–4561.

Huilin Deng, Ding Zou, Rui Ma, Hongchen Luo, Yang Cao, and Yu Kang. 2025. Boosting the generalization and reasoning of vision language models with curriculum reinforcement learning. *Preprint*, arXiv:2503.07065.

Le Fang, Tao Zeng, Chaochun Liu, Liefeng Bo, Wen Dong, and Changyou Chen. 2021. Transformer-based conditional variational autoencoder for controllable story generation. *Preprint*, arXiv:2101.00828.

Jan-Philipp Fränken, Elad Zelikman, Rafael Rafailov, Kashyap Gandhi, Tobias Gerstenberg, and Noah D. Goodman. 2024. Self-supervised alignment with mutual information: Learning to follow principles without preference labels. In *Advances in Neural Information Processing Systems*, volume 37. Curran Associates, Inc.

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. 2024. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3921–3940. Association for Computational Linguistics.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. In *International Conference on Learning Representations*.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *Preprint*, arXiv:2504.11456.

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 *Advances in Neural Information Processing Systems Datasets and Benchmarks Track*.

Yuxian Jiang, Yafu Li, Guanxu Chen, Dongrui Liu, Yu Cheng, and Jing Shao. 2025. Rethinking entropy regularization in large reasoning models. *arXiv preprint arXiv:2509.25133*.

Feng Ju, Zeyu Qin, Rui Min, Zhitao He, Lingpeng Kong, and Yi R. Fung. 2025. Reasoning path divergence: A new metric and curation strategy to unlock llm diverse thinking. *Preprint*, arXiv:2510.26122.

Feng Ju, Zeyu Qin, Rui Min, Zhitao He, Lingpeng Kong, and Yi R. Fung. 2026. Reasoning path divergence: A new metric and curation strategy to unlock llm diverse thinking. *Preprint*, arXiv:2510.26122.

Shiye Lei, Zhihao Cheng, Kai Jia, and Dacheng Tao. 2025a. Revisiting llm reasoning via information bottleneck. *Preprint*, arXiv:2507.18391.

Shiye Lei, Zhihao Cheng, Kai Jia, and Dacheng Tao. 2025b. Revisiting llm reasoning via information bottleneck. *Preprint*, arXiv:2507.18391.

Jiwei Li, Dan Jurafsky, and Eduard Hovy. 2016. A diversity-promoting objective function for neural conversation models. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 110–119, San Diego, California. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, Pennsylvania, USA.

Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv*, abs/2506.06941.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025b. Reinforcement learning for reasoning in large language models with one training example. *Preprint*, arXiv:2504.20571.

Jianghao Wu, Yasmeen George, Jin Ye, Yicheng Wu, Daniel F. Schmidt, and Jianfei Cai. 2025. Spine: Token-selective test-time reinforcement learning with entropy-band regularization. *Preprint*, arXiv:2511.17938.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, and 1 others. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *Forty-first International Conference on Machine Learning*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2025. Self-rewarding language models. *Preprint*, arXiv:2401.10020.

Jiayi Zhang, Simon Yu, Derek Chong, Anthony Sicilia, Michael R. Tomz, Christopher D. Manning, and Weiyan Shi. 2025a. Verbalized sampling: How to mitigate mode collapse and unlock llm diversity. *Preprint*, arXiv:2510.01171.

Qingyang Zhang, Haitao Wu, Changqing Zhang, Peilin Zhao, and Yatao Bian. 2025b. Right question is already half the answer: Fully unsupervised llm reasoning incentivization. *Preprint*, arXiv:2504.05812.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Appendix

A Categorizing High-Entropy Tokens

In Fig. 9, Top20% impactful updates primarily target high-entropy tokens. These tokens tend to produce larger gradients during backpropagation. This indicates that progress in this stage is mainly driven by resolving uncertainty at critical "forks" in reasoning paths (Wang et al., 2025a). In RLVR, tokens generated by models exhibit different functional roles that collectively drive the reasoning process. Based on their operational characteristics, we categorize tokens into three roles:

- **Logical Structuring Tokens:** Govern reasoning flow (*e.g.* causal, contrastive, progressive, and parallel connectors). They help structure multi-step argumentation or explanations.
- **Metacognitive Tokens:** Reflect meta-cognitive functions, especially self-monitoring behaviors (*e.g.* verifying, summarizing, and revising). These tokens actively guide the reasoning process through reflective adjustment and solution refinement.

- **Semantic Support Tokens:** Provide linguistic elements that ensure fluency, coherence, and informativeness (*e.g.* core grammatical elements, domain-specific entities).

We provide examples of each category in Table 5.

Category	Examples
Logical Structuring	Causal (*e.g.* 'therefore', 'because'), contrastive (*e.g.* 'however', 'but'), progressive (*e.g.* 'first', 'next', 'finally'), and parallel (*e.g.* 'and', 'also').
Metacognitive	Verifying (*e.g.* 'Let's check'), revising (*e.g.* 'Correction', 'Wait'), summarizing (*e.g.* 'In summary'), and planning (*e.g.* 'First, I will...').
Semantic Support	Grammatical elements (*e.g.* 'the', 'is', 'of'), domain entities (*e.g.* 'problem', 'solution'), and adjectives (*e.g.* 'correct', 'final').

Table 5: Examples of Token Categories in RLVR.

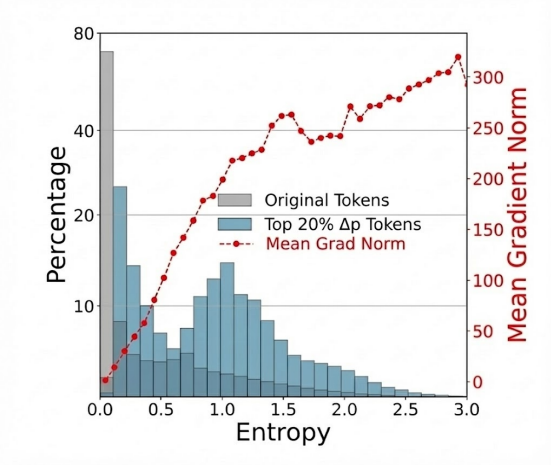


Figure 9: Token entropy and gradient distribution.

B Implementation Details of the CVAE

In this section, we provide the architectural specifications and the training objective of our CVAE.

Network Architecture: A typical CVAE consists of three parts: (1) The **Encoder** $q_\phi(z \mid x, c)$ approximates the posterior distribution of the latent variable z. (2) The **Prior Network** $p_\theta(z \mid c)$ models the prior distribution of z, which is conditioned only on the context c. (3) The **Decoder Network** $p_\theta(y \mid z, x)$ takes z and x to decode the output y.

Specifically, we employ **DeBERTa-v2-large** (He et al., 2020) as the Encoder $q_\phi(z|x, c)$. DeBERTa is chosen for its disentangled attention mechanism, which effectively encodes both content and relative positions. Formally, given the input sequence x, we first extract the contextualized hidden states via the encoder. To obtain a fixed-size vector representation $h_x \in \mathbb{R}^{d_{model}}$, we apply mean pooling over the token embeddings:

$$h_x = \text{MeanPool}(\text{DeBERTa}(x)), \qquad (16)$$

where $d_{model} = 1024$ for the large architecture. To construct the probabilistic latent space, h_x is projected into the variational parameters—mean μ and log-variance $\log \sigma^2$—via two separate linear transformations:

$$\mu = W_\mu h_x + b_\mu, \quad \log \sigma^2 = W_\sigma h_x + b_\sigma, \quad (17)$$

where $W_{(\cdot)} \in \mathbb{R}^{d_z \times d_{model}}$ are learnable projection matrices and d_z is the dimension of the latent bottleneck (e.g., $d_z = 128$). Finally, the latent variable z is sampled using the standard reparameterization trick:

$$z = \mu + \sigma \odot \epsilon, \quad \epsilon \sim \mathcal{N}(0, I). \qquad (18)$$

This latent code z serves as a compact semantic anchor, which is subsequently fused into the decoder (LLaMA) to guide the generation of output y.

Training Objective. Formally, the CVAE is trained by maximizing the Evidence Lower Bound (ELBO):

$$\begin{aligned} L_{\text{ELBO}} &= L_{\text{REC}} - L_{\text{KL}} \\ &= \mathbb{E}_{q_\phi(z|x,y)}[\log p_\theta(y|z,x)] \\ &\quad - \text{KL}(q_\phi(z|x,y) \| p_\theta(z|x)) \\ &\leq \log p(y|x), \end{aligned} \qquad (19)$$

where L_{REC} denotes the reconstruction loss, and L_{KL} represents the Kullback-Leibler divergence between the posterior and prior distributions. The CVAE is trained on MATH and GSM8K training sets.

C IB-Aware Scoring Metric

C.1 General Theory

Information Bottleneck (IB) formalizes the trade-off between compressing input X into a compact representation $\hat{X}$ while maintaining predictive

power for target Y, via the Lagrangian objective:

$$\mathcal{L}_{\text{IB\_naive}} = I(X;\hat{X}) - \beta I(\hat{X};Y), \tag{20}$$

where $I(\cdot;\cdot)$ denotes Mutual Information (MI) and β controls the trade-off. The MI between X and Y is defined as:

$$I(X;Y) = H(X) - H(X|Y), \tag{21}$$

where $H(\cdot)$ denotes entropy. A high value of $I(X;Y)$ indicates that knowledge of one variable reduces uncertainty about the other.

C.2 Derivation of the IB-Score

Following the *IB-Aware Reasoning Optimization*, the optimization objective in LLM reasoning is formulated as:

$$\mathcal{L}_{\text{IB}} = I(q;r) - \beta I(r;a). \tag{22}$$

To derive a tractable metric, we decompose these two terms respectively.

Minimizing Complexity $I(q;r)$. We first expand $I(q;r)$ using the definition of mutual information and the assumption that the marginal entropy of reasoning $H(r)$ is invariant under the policy π (Assumption 1 in (Lei et al., 2025b)). Applying the chain rule of entropy to the autoregressive trajectory $r = (o_1, \ldots, o_T)$, we obtain:

$$\begin{aligned} I(q;r) &= H(r) - H(r \mid q) \\ &= H(r) - \sum_{t=1}^{T} H(o_t \mid o_{<t}, q). \end{aligned} \tag{23}$$

Therefore, minimizing the mutual information $I(q;r)$ is equivalent to maximizing the conditional entropy sum $\sum_t H(o_t \mid o_{<t}, q)$. This term encourages the model to maintain high entropy during generation, preventing it from collapsing into memorized or over-deterministic patterns dependent solely on the specific prompt phrasing.

Maximizing Informativeness $I(r;a)$. For the second term, we have $I(r;a) = H(r) - H(r \mid a)$. Maximizing this quantity requires minimizing the conditional entropy $H(r \mid a)$. In our specific selection phase, we restrict our candidate pool to the validity-verified subset $\mathcal{R}_{correct} = \{r \in \mathcal{R} \mid \hat{a}(r) = a\}$. For any path $r \in \mathcal{R}_{correct}$, the reasoning r implies the answer a with certainty (i.e., $p(a|r) \approx 1$). Under this condition, the posterior probability of the reasoning path answered can be approximated by the likelihood of the path itself:

$$p(r \mid q, a) = \frac{p(a \mid r, q)p(r \mid q)}{p(a \mid q)} \propto p(r \mid q). \tag{24}$$

Consequently, minimizing the uncertainty of the reasoning answered ($H(r \mid q, a)$) corresponds to selecting trajectories that maximize the log-likelihood $\log \pi(r \mid q)$. High-likelihood paths within $\mathcal{R}_{correct}$ represent the most confident reasoning traces that lead to correct solution.

The IB-Score formulation. Combining the derivations above, we transform the trajectory-level IB objective into a tractable token-level utility function. By substituting the log-likelihood for the informativeness term and the token-level entropy for the complexity term, we propose the **IB-Score** for a reasoning path r:

$$\begin{aligned} \mathcal{S}_{\text{IB}}(r) &= \sum_{t=1}^{T} \left(\underbrace{H(o_t \mid o_{<t}, q)}_{\substack{\text{Exploration Proxy} \\ (\text{Max } I(q;r))}} - \underbrace{\beta H(o_t \mid o_{<t}, q, a)}_{\substack{\text{Relevance Proxy} \\ (\text{Min } I(r;a))}} \right) \\ &= \sum_{t=1}^{T} \lambda_t H(o_t \mid o_{<t}, q) \\ &\cong \sum_{t=1}^{T} \mathcal{A}_t H(o_t \mid o_{<t}, q) \end{aligned} \tag{25}$$

We formulate the Information Bottleneck objective as maximizing a score $\mathcal{S}_{\text{IB}}$. This score balances exploration (Standard Entropy) against relevance (Negative Conditional Entropy):

$$\begin{aligned} \mathcal{S}_{\text{IB}}(r) &= \sum_{t=1}^{T} \underbrace{(H(o_t \mid o_{<t}, q) - \beta H(o_t \mid o_{<t}, q, a))}_{s_t^{\text{IB}}\text{: Per-token IB Score}} \\ &= \sum_{t=1}^{T} s_t^{\text{IB}} \end{aligned} \tag{26}$$

Let $H_t = H(o_t \mid o_{<t}, q)$ and $H_{t|a} = H(o_t \mid o_{<t}, q, a)$. We set $\beta = 2$ as per the theoretical analysis. The per-token score becomes:

$$s_t^{\text{IB}} = H_t - 2H_{t|a} \tag{27}$$

Using the inequality $0 \leq H_{t|a} \leq H_t$, we analyze the bounds of s_t^{IB}:

$$\text{Best:} \quad H_{t|a} = 0 \implies s_t^{\text{IB}} = \mathbf{H_t} \tag{28}$$

$$\text{Worst:} \quad H_{t|a} = H_t \implies s_t^{\text{IB}} = -\mathbf{H_t} \tag{29}$$

Thus, the score term lies in the symmetric interval:

$$s_t^{\text{IB}} \in [-H_t, H_t] \tag{30}$$

We can therefore represent s_t^{IB} as a modulated entropy term:

$$s_t^{\text{IB}} = \lambda_t H_t, \quad \text{where } \lambda_t \in [-1, 1] \tag{31}$$

Now we map the coefficient λ_t to the Reinforcement Learning context:

- **High Score** ($\lambda_t \to 1$): Occurs when the token is perfectly predictive. This corresponds to a "Good" action.
- **Low Score** ($\lambda_t \to -1$): Occurs when the token provides no information about the answer. This corresponds to a "Bad" action.

The Advantage function $\mathcal{A}_t$ naturally captures this property: high positive $\mathcal{A}_t$ for good actions, negative $\mathcal{A}_t$ for bad actions. Thus, we can directly approximate λ_t with the advantage (without any negative sign):

$$\lambda_t \approx \mathcal{A}_t \tag{32}$$

Substituting $\lambda_t \approx \mathcal{A}_t$ back into the summation:

$$\begin{aligned} \mathcal{S}_{\text{IB}}(r) &= \sum_{t=1}^{T} s_t^{\text{IB}} \\ &= \sum_{t=1}^{T} \lambda_t H_t \\ &\cong \sum_{t=1}^{T} \mathcal{A}_t H(o_t \mid o_{<t}, q) \end{aligned} \tag{33}$$

C.3 Theoretical Analysis on Verbosity and Pruning Optimality

To further enhance the theoretical rigor, we supplement two analyses that directly address: (1) why IB suppresses vacuous verbosity, and (2) why the Top-N IB pruning strategy is optimal.

Proposition 1 (Verbosity Suppression) *Given a fixed answer a. Let a reasoning trajectory be r, and append a token x to obtain $r' = (r, x)$. If this token is conditionally independent of the answer given the preceding context (i.e., $I(x; a \mid r) = 0$, or equivalently $x \perp a \mid r$), then*

$$S_{IB}(r') \le S_{IB}(r), \tag{34}$$

and it strictly decreases when $I(q; x \mid r) > 0$, i.e., $S_{IB}(r') < S_{IB}(r)$.

By definition, the IB loss and score are given by:

$$S_{IB}(r) = \beta I(r; a) - I(q; r). \tag{35}$$

For the trajectory after appending the token $r' = (r, x)$, by the chain rule of mutual information:

$$I(q; r') = I(q; r, x) = I(q; r) + I(q; x \mid r). \tag{36}$$

Therefore, the change in the IB loss is:

$$\begin{aligned} L_{IB}(r') - L_{IB}(r) &= \big(I(q; r') - I(q; r)\big) \\ &\quad -\beta\big(I(r'; a) - I(r; a)\big) \\ &= I(q; x \mid r) - \beta\, I(x; a \mid r). \end{aligned} \tag{37}$$

Under the condition of the proposition where $I(x; a \mid r) = 0$, we have:

$$L_{IB}(r') - L_{IB}(r) = I(q; x \mid r) \ge 0. \tag{38}$$

Taking the negative of both sides yields:

$$S_{IB}(r') - S_{IB}(r) = -\big(L_{IB}(r') - L_{IB}(r)\big) \le 0, \tag{39}$$

hence $S_{IB}(r') \le S_{IB}(r)$. If additionally $I(q; x \mid r) > 0$ (meaning a "filler" token is typically still driven by the prompt/context), then L_{IB} increases strictly and S_{IB} decreases strictly.

Alignment with token-level approximation: Under the token-level IB decomposition $S_{IB}(r) = \sum_{t=1}^{T} s_t^{IB}$, where

$$s_t^{IB} = H_t - \beta H_{t|a}. \tag{40}$$

If a token o_t is a "redundant/vacuous" token such that it is approximately conditionally independent of the answer given the context (i.e., $H_{t|a} \approx H_t$):

$$s_t^{IB} \approx (1 - \beta) H_t \le 0 \quad (\text{for } \beta > 1). \tag{41}$$

Hence, inserting such redundant tokens decreases (or at least does not increase) the total score $S_{IB}(r)$, explaining IB's systematic suppression of vacuous verbosity.

One-sentence interpretation: A "vacuous filler token" does not increase $I(r; a)$ (it carries no information about the answer), but it increases $I(q; r)$ (more dependent on the prompt). Therefore, the IB trade-off becomes worse and the score decreases, making reward hacking ineffective.

Lemma 1 (Optimality of Top-N Pruning) *Given a candidate set $\mathcal{R}$ and scores $S_{IB}(r)$, consider the optimization problem:*

$$\max_{S \subseteq \mathcal{R},\, |S| = N} \sum_{r \in S} S_{IB}(r). \tag{42}$$

Its optimal solution $S^\star$ must be the Top-N set: selecting the N trajectories with the highest scores.

Let T denote the set formed by sorting trajectories by $S_{IB}(r)$ in descending order and taking the top N trajectories (the Top-N set). Assume there exists an optimal solution $S^\star \ne T$. Then there must exist:

- $r \in S^\star \setminus T$ (the optimal set contains a non-Top-N trajectory), and

Table 6: Summary of IB-related Failure Modes

Failure Mode	Observed Behavior	IB Diagnosis (Theoretical Basis)	Filter Effect
Vacuous Verbosity	Fluent but empty meta-talk.	High cost ($I(q;r)$ ↑): length↑, no gain in $I(r;a)$.	Suppresses reward hacking.
Repetitive Loops	Repeats previous steps or statements.	Infinite cost ($I(q;r)$ ↑↑): entropy↑, zero benefit.	Cuts infinite loops.
Logical Drift	Concise but contains factual errors / hallucinations.	Low benefit ($I(r;a)$ ↓): fails to predict correct a.	Filters hallucinations.

- $r' \in T \backslash S^\star$ (a higher-scoring Top-N trajectory is not selected in the optimal set).

By the definition of Top-N, we know that $S_{IB}(r') \geq S_{IB}(r)$. When $S^\star \neq T$, we can take the strict inequality $S_{IB}(r') > S_{IB}(r)$.

Construct the exchanged set $\tilde{S}$:

$$\tilde{S} = (S^\star \setminus \{r\}) \cup \{r'\}. \tag{43}$$

Clearly $|\tilde{S}| = N$, and the sum of scores is:

$$\begin{aligned} \sum_{u \in \tilde{S}} S_{IB}(u) &= \sum_{u \in S^\star} S_{IB}(u) - S_{IB}(r) + S_{IB}(r') \\ &\geq \sum_{u \in S^\star} S_{IB}(u). \end{aligned} \tag{44}$$

If $S_{IB}(r') > S_{IB}(r)$, then the inequality is strict, contradicting the optimality of $S^\star$. Therefore, the optimal solution must equal the Top-N set T.

One-sentence interpretation: Under the objective to select N trajectories to maximize the total IB score, sorting and taking Top-N is the mathematically optimal solution.

C.4 Qualitative Analysis

Example 1: Logarithmic equation

Q: Solve for x: $\log_2(x-1) + \log_2(x+3) = 5$.

(0) High-IB Trajectory ($S_{IB} = 0.92$, Reward=1.0)

A: Combine logarithms: $\log_2((x-1)(x+3)) = 5$. Exponentiate: $x^2+2x-3 = 2^5 = 32$. Solve $x^2+2x-35 = 0 \Rightarrow (x+7)(x-5) = 0$. Since domain requires $x > 1$, reject $x = -7$. Thus, $x = 5$. $\boxed{5}$

(1) Failure Mode A: Vacuous Verbosity ($S_{IB} = 0.39$, Reward=1.0)

A: To solve this logarithmic equation, we must first consider the domain. The arguments must be positive. Basically, we use the property $\log a + \log b = \log(ab)$. Let me think... applying this gives $\log_2((x-1)(x+3)) = 5$. It is important to proceed carefully. Now we turn it into exponential form, $x^2 + 2x - 3 = 32$. Solving for x, we get 5 and -7. Checking domain, it is 5. $\boxed{5}$

Analysis: [Low Score]. Although the answer is correct (Reward=1.0), the model dilutes valid reasoning with excessive meta-discourse ("Basically", "Let me think").

(2) Failure Mode B: Repetitive Loops ($S_{IB} = -0.89$, Reward=1.0)

A: Combine terms to get $\log_2(x^2 + 2x - 3) = 5$. This means $x^2 + 2x - 3 = 32$. So we have $x^2 + 2x - 35 = 0$. Let's check the factors. $(x + 7)$ and $(x - 5)$. So $x^2-5x+7x-35$. This expands to $x^2+2x-35$. Correct. Back to the equation $x^2 + 2x - 35 = 0$. Factors are $(x + 7)$ and $(x - 5)$... [...repeats expansion-verification loop for 2048+ tokens...] $\boxed{5}$

Analysis: [Negative Score]. The model gets stuck verifying its own algebraic expansion repeatedly. The cost ($I(q;r)$) explodes as length increases, while the information gain is zero. This "infinite cost" path is aggressively pruned.

(3) Failure Mode C: Logical Drift ($S_{IB} = 0.28$, Reward=0.0)

A: Combine logarithms: $\log_2((x-1)(x+3)) = 5$. This gives $x^2 + 2x - 3 = 32$. Solving $x^2 + 2x - 35 = 0$ yields $x = -7$ and $x = 5$. The solutions are -7 and 5. $\boxed{5, 7}$

Analysis: [Low Score]. Despite being concise (Low Cost), the trajectory has Low Predictive Power ($I(r;a)$) because it fails to check the domain ($x > 1$) and includes an extraneous root ($x = -7$), drifting to a partially incorrect conclusion.

D Localization of Difficulty-Sensitive Attention Heads

Fig. 7 depicts attention head patterns contrasting high (Level 9) vs. low (Level 3) difficulty on DeepMath (He et al., 2025), a dataset spanning 13 difficulty levels (3.0 to 9.0). Let $H \in \mathbb{R}^{B \times L \times N \times d}$

denote the output tensor of the multi-head attention (MHA) layer, where B is the batch size, L is the sequence length, N is the number of attention heads, and d is the head dimension.

The final contextual representation Z is typically obtained by projecting the concatenated output of all heads via the output projection matrix $W_o \in \mathbb{R}^{(Nd)\times D}$:

$$Z = \text{Reshape}(H)W_o^\top \in \mathbb{R}^{B\times L\times D}, \tag{45}$$

where $D = N \times d$ represents the model's hidden dimension.

To analyze the independent contribution of the i-th attention head ($i \in \{1, \ldots, N\}$), an ablation strategy is employed. An ablated representation $H^{(i)}$ is constructed by retaining the output of the i-th head while zeroing out all other heads:

$$H^{(i)}_{b,\ell,j,:} = \begin{cases} H_{b,\ell,i,:} & \text{if } j = i \\ \mathbf{0} & \text{otherwise} \end{cases}, \quad \forall b \in [B], \ell \in [L], j \in [N] \tag{46}$$

Subsequently, the projected representation $Z^{(i)}$ is computed using W_o. We extract the embedding of the **last token** for each sample b, denoted as $z_b^{(i)} = Z^{(i)}_{b,L-1,:} \in \mathbb{R}^D$.

D.1 Difficulty Scoring

Using the pre-trained probe direction v_{diff}, the difficulty score contribution of the i-th head for sample b is calculated as the normalized projection onto the difficulty direction:

$$s_b^{(i)} = \frac{\langle z_b^{(i)}, v_{diff}\rangle}{\|v_{diff}\|_2}. \tag{47}$$

For a batch of samples sharing the same ground-truth difficulty level, the mean head-wise attribution is aggregated:

$$\bar{s}^{(i)} = \frac{1}{B}\sum_{b=1}^{B} s_b^{(i)}. \tag{48}$$

D.2 Differentiation Score and Localization

To determine whether a specific head is sensitive to hard or easy problems, the *Differentiation Score* is computed as the difference between the mean attributions of high-difficulty (Hard) and low-difficulty (Easy) cohorts:

$$\Delta^{(i)} = \bar{s}^{(i)}_{hard} - \bar{s}^{(i)}_{easy} \tag{49}$$

- If $\Delta^{(i)}$ is significantly **positive**: The head is activated to identify **difficult** problems.
- If $\Delta^{(i)}$ is significantly **negative**: The head is activated to identify **simple** problems.

By computing $\Delta^{(i)}$ for all heads, the specific components responsible for difficulty perception can be precisely located.

E Additional Implementation Details

Other Related Work: Self-Rewarding Reasoning. Recent RL advancements in LLMs rely heavily on outcome-based rewards, yet such sparse supervision neglects intermediate reasoning validity. As step-level labels require substantial human effort, self-rewarding methods have gained increasing attention. (Bai et al., 2022) and (Yuan et al., 2025) use predefined rules to generate preference pairs for Direct Preference Optimization (DPO) training. (Fränken et al., 2024) maximize mutual information between principles and responses, while (Zhang et al., 2025b) is based on semantic entropy clustering. SPINE (Wu et al., 2025) adopts a majority-vote-based reward for self-consistency. However, unlike SPINE, which suppresses high-entropy states via static voting, we actively leverage them as gateways for topological branching and iterative trajectory refinement.

Perplexity (PPL) measures uncertainty of the generated sequence as:

$$\text{PPL}(o^i) = \exp\left(-\frac{1}{N}\sum_{t=1}^{N}\log \pi(o_t^i \mid o_{<t}^i)\right)$$

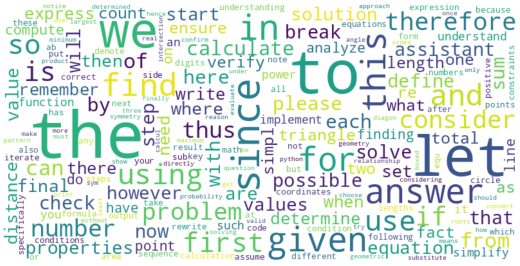
where $\pi(o_t^i \mid o_{<t}^i)$ is next-token probability. Low-PPL responses are generally more fluent and semantically coherent (Adiwardana et al., 2020).

RL Training Configuration For both GRPO and DAPO, we use the hyperparameters in Tab.7, without using entropy or KL losses. Specifically, Branching maxtimes caps trajectory branching at 4, while Samples per prompt denotes the maximum few-shot CoT demonstrations. Experiments on Qwen2.5-7B run on **8×H100 GPUs**, taking ∼50 hours for one epoch. Experiments on Qwen3-14B run on **16×H100 GPUs**, taking ∼64 hours for one epoch.

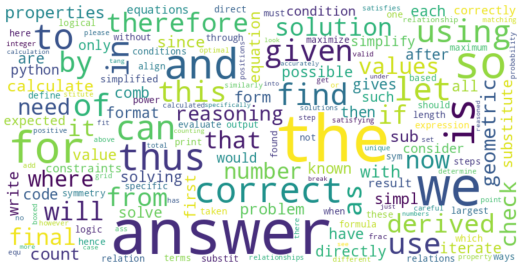
Word Cloud The Entropy Reg. outputs more generic syntactic fillers (e.g., 'is', 'to', 'can'). I^{2}B-LPO produces more reasoning-focused keywords.

Hyperparameter	Value
Optimizer	AdamW
Policy learning rate	$1e^{-6}$ for qwen2_5 series $5e^{-6}$ for qwen3 series
Training batch size	128
Samples per prompt	8
Rolloutn	4
Branching K	7
N	8
γ	0.003
Max response length	8192
Max prompt length	2048
Rollout temperature	1.0
Clip range ϵ_{low}, ϵ_{high}	0.2, 0.28
α for IB loss	0.005

Table 7: Hyperparameters of GRPO and DAPO training



(a) Naive Entropy Regularization.



(b) I^{2}B-LPO (Ours).

Figure 10: **Word cloud comparison on AIME2024 using Qwen2.5-7B. The Entropy Reg. outputs more generic syntactic fillers (e.g., 'is', 'to', 'can'). I^{2}B-LPO produces more reasoning-focused keywords.**

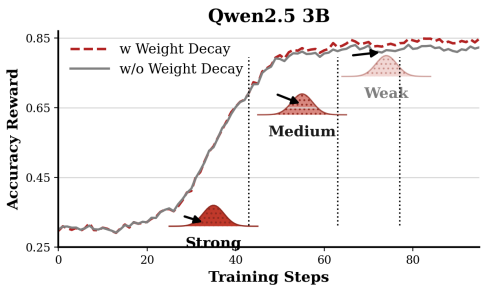


Figure 12: **Ablation study on the dynamic decay of latent injection.**

Response length analysis. I^{2}B-LPO achieves a balanced response length, sitting between GRPO and Entropy. Reg. It avoids the superficial verbosity and non-informative filler observed in Entropy. Reg, while demonstrating more comprehensive reasoning steps compared to GRPO.

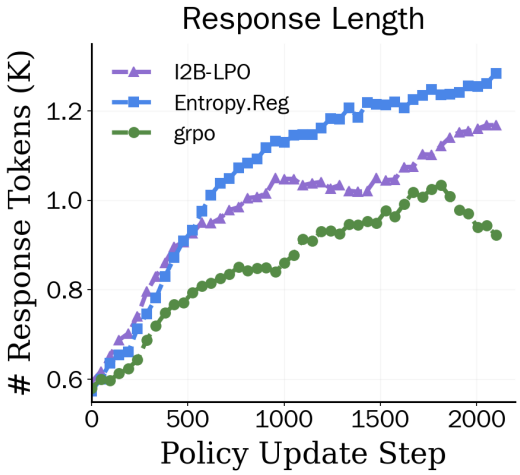
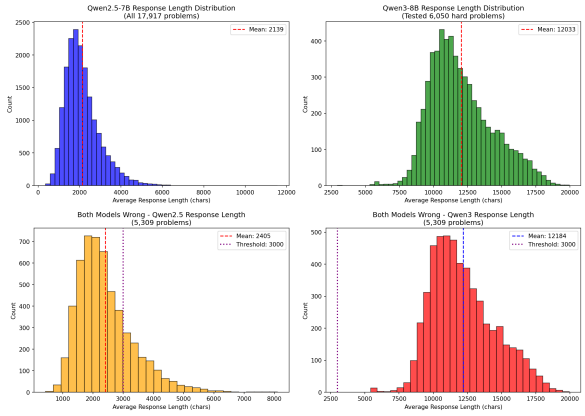
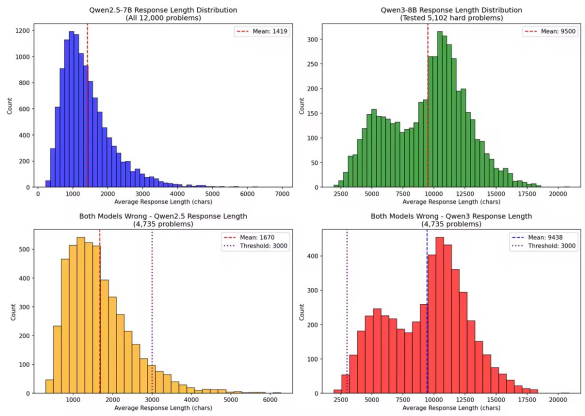


Figure 11: **Response length analysis on Qwen2.5-3B.**



(a) DAPO-Math-17k



(b) MATH

Figure 13: Response length distribution results on DAPO and MATH datasets.

Ablation of weight decay in PSA. We implement an exponential decay schedule for the injection factor $\gamma(t)$, annealing the strength from 5×10^{-2} down to 5×10^{-4}. This experiment is

conducted on MATH datasets with qwen2.5-3B base model. The Gaussian icons (Strong, Medium, Weak) visually represent the diminishing intensity of the latent variable $\mathbf{z}$ over time.

Data Pre-processing To focus training on the "learning frontier", **i.e,** problems neither trivial nor intractable, we filter the dataset using two reference models (Qwen2.5-7B and Qwen3-8B) with $n = 8$ rollouts each. We exclude: (1) **trivial samples** consistently solved by both models, as they provide negligible gradient signals for policy improvement; and (2) **intractable samples** where both models fail, specifically targeting excessively long responses ($>$3,000 tokens for Qwen2.5-7B, $>$10,000 for Qwen3-8B) to mitigate hallucination loops. The response length distributions of the resulting filtered datasets are illustrated in Figure 13. The final training set consists of **6,486** samples from MATH and **13,583** from DAPO.