

RETHINKING ENTROPY REGULARIZATION IN LARGE REASONING MODELS

Yuxian Jiang^{1,2*}, Yafu Li^{1†}, Guanxu Chen^{1,3}, Dongrui Liu^{1†}, Yu Cheng⁴, Jing Shao^{1†},

¹Shanghai Artificial Intelligence Laboratory ²Fudan University

³Shanghai Jiao Tong University ⁴Chinese University of Hong Kong

Contact: yuxianjiang7@gmail.com, yafuly@gmail.com, liudongrui@pjlab.org.cn, shaojing@pjlab.org.cn

ABSTRACT

Reinforcement learning with verifiable rewards (RLVR) has shown great promise in enhancing the reasoning abilities of large reasoning models (LRMs). However, it suffers from a critical issue: entropy collapse and premature convergence. Naive entropy regularization, a common approach for encouraging exploration in the traditional RL literature, fails to address this problem in the context of LRM. Our analysis reveals that this failure stems from the vast action space and long trajectories in LRMs, which easily trigger a global entropy explosion as the model indiscriminately explores all possible actions and states. To address this, we propose **SIREN** (SelectIve entRopy rEgularizationN), a method that confines exploration to a meaningful subset of actions and states. **SIREN** achieves this through a two-step entropy masking mechanism, consisting of a top-p mask and a peak-entropy mask. In addition, regularization is transformed into a self-anchored form to stabilize training. Across five mathematical benchmarks, **SIREN** attains superior average performance over previous entropy-related RLVR approaches, exemplified by a +6.6 maj@k improvement on AIME24/25 with Qwen2.5-Math-7B. Further analysis confirms that **SIREN** promotes greater response diversity and maintains entropy at an appropriate level, which helps to preserve the validation pass@k throughout training. This effectively mitigates the premature convergence problem common in RLVR for LRMs. Our code is available at [GitHub](#).

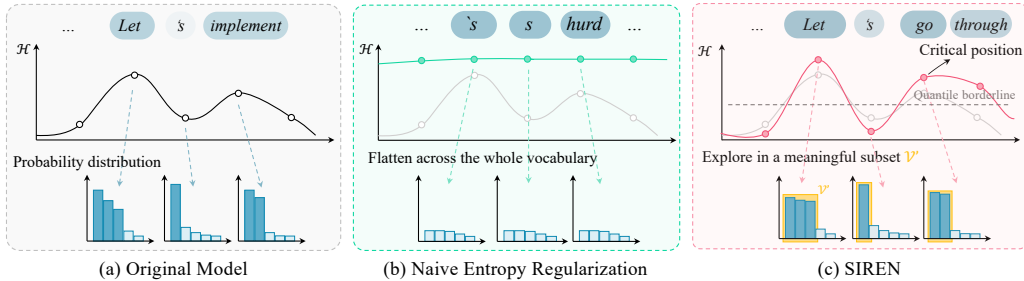


Figure 1: Comparison of entropy regularization strategies for large reasoning models (LRMs). (a) The original model produces highly concentrated probability distributions, where different tokens exhibit uneven entropy levels. (b) Naive entropy regularization flattens the distribution indiscriminately across the entire vocabulary, resulting in uninformative exploration and global entropy explosion. (c) Our proposed method, **SIREN**, selectively regularizes entropy within meaningful token subsets, focusing on critical positions identified by entropy quantiles. This enables controlled and effective exploration.

*This work was done during an internship at Shanghai Artificial Intelligence Laboratory, supervised by Dongrui Liu.

†Corresponding author.

1 INTRODUCTION

Reinforcement learning with verifier rewards (RLVR) has become a promising method for enhancing large reasoning models (LRMs) by integrating reward modeling with automated verification (Zhang et al., 2025). This technique has been shown to substantially improve complex reasoning abilities in domains such as mathematics and code (DeepSeek-AI, 2025; Team et al., 2025; Yang et al., 2025). Despite its success, a critical limitation has emerged in practice: entropy collapse and premature convergence (Yu et al., 2025; Cui et al., 2025). This issue causes the policy to become deterministic at an early stage, leading responses to collapse into near-identical outputs, and ultimately reducing both training efficiency in later stages and overall performance.

A common and intuitive solution to mitigate this issue is to incorporate entropy regularization to encourage exploration (Ziebart, 2010; Haarnoja et al., 2018b;c). In the context of LRMs, naive entropy regularization exhibits strong hyperparameter sensitivity and limited efficiency: a small coefficient yields minimal performance gains, while a large coefficient triggers a rapid entropy explosion (Cui et al., 2025). To better understand this behavior, we conduct a preliminary experiment. By analyzing probability distributions and the generated tokens from both an original model and one that experienced an entropy explosion under strong entropy regularization, we observe two key phenomena: (1) In the entropy-exploded model, the probability distribution becomes nearly uniform over the whole vocabulary, whereas in the original model, probability mass concentrates on a small, semantically meaningful subset. (2) In the entropy-exploded model, entropy remains uniformly high across almost all token positions, whereas in the original model, only a small fraction of tokens exhibit high entropy. Taken together, these phenomena suggest that the vast action space of LRMs makes it easy to increase the entropy by diffusing the probability mass across many low-utility tokens. The autoregressive generation process then propagates this uncertainty forward, causing the accumulated entropy bonus to scale with sequence length and ultimately trigger an explosion.

Based on these findings, we propose **SelectIve entRopy rEgularizationN** (**SIREN**), a novel method that restricts exploration to a more effective subset of actions and states. **SIREN** consists of a two-step masking mechanism to identify exploration targets precisely: (1) a *top-p mask*, which limits exploration to semantically meaningful subsets within the large action space of LRMs; and (2) a *peak-entropy mask*, which limits exploration on several logically critical tokens along long trajectories. Furthermore, to calibrate the regularization strength and stabilize training, we reformulate the naive regularization as *self-anchored regularization*, which helps control the overall entropy level during training.

To evaluate **SIREN**’s performance, we conduct a comprehensive evaluation across multiple backbones and mathematical reasoning benchmarks. Our method achieves significantly improved performance on Qwen2.5-Math-7B, with a maj@k of **54.6** and an avg@k of **46.1**. Compared to other baselines that also incorporate entropy, **SIREN** improves the maj@k by **+4.8** and the avg@k by **+1.6**. **SIREN** also attains higher maj@k and avg@k on the smaller model Qwen2.5-Math-1.5B and the comparatively weaker model LLaMa3.1-8B. Through an analysis of pass@k and perplexity, which are commonly used to estimate the upper bound of reasoning and the uncertainty of generation (Holtzman et al., 2019; Yue et al., 2025; Cheng et al., 2025), we observe that **SIREN** yields more diverse responses and explores a wider scope, pushing the boundary of reasoning. In addition, the joint dynamics of entropy and validation pass@k during training indicate that **SIREN** performs effective, progressive exploration: it maintains high entropy in early stages and gradually converges later, ensuring sustained exploration while preventing degradation in validation pass@k.

In summary, the key contributions of this work are as follows.

- We analyze the limitations of naive entropy regularization in RLVR, showing that *the vast action space and long trajectories* flatten probability distributions across most positions. This highlights the need to control the effective scope of regularization.(Sec. 2)
- We propose **SIREN**, a novel method that selects exploration scopes at both the action and the trajectory levels for effective entropy regularization, while transforming the naive regularization into a self-anchored form to stabilize training.(Sec. 3)
- We achieve competitive results on mathematical benchmarks, showing consistent improvements over other entropy-related baselines on maj@k and avg@k.(Sec. 4)

Questions: Let p be the least prime number for which there exists a positive integer n such that $n^4 + 1$ is divisible by p^2 . Find the least positive integer m such that $m^4 + 1$ is divisible by p^2 .

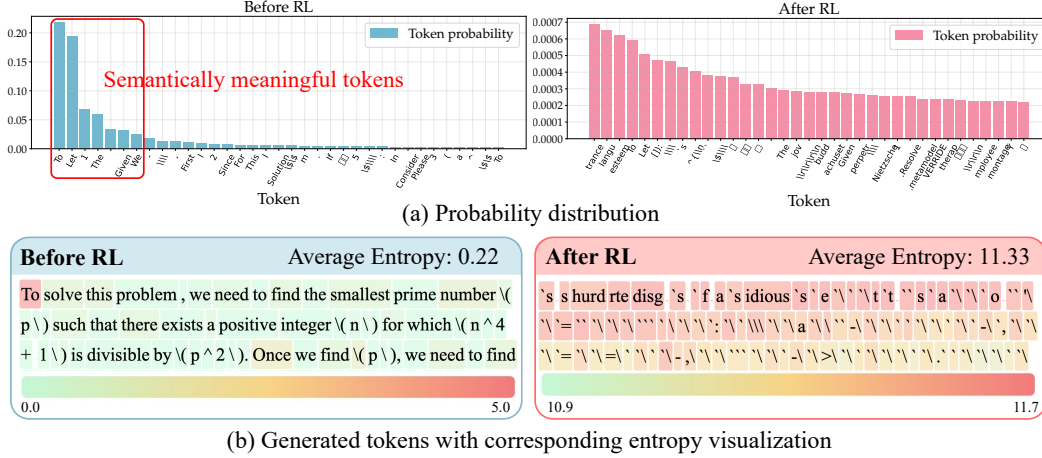


Figure 2: Comparison of the first token’s probability distributions and generated tokens before and after RL with naive entropy regularization (entropy coefficient set to 0.005).

2 PRELIMINARY ANALYSIS

As effective exploration remains a central challenge in traditional RL, entropy regularization is commonly employed to encourage policy diversity, thus facilitating broader state exploration (Haarnoja et al., 2018a;c; Liu et al., 2020). Mathematically, given a query q , the entropy of the current policy π_θ for each token v_j is defined in Eq. 1, where $\mathcal{V}$ denotes the vocabulary.

$$\mathcal{H}_j = - \sum_{v \in \mathcal{V}} \pi_\theta(v | q, v_{<j}) \log \pi_\theta(v | q, v_{<j}). \quad (1)$$

The training objective is given by Eq 2, where J_{PO} is the policy optimization objective and β is the entropy coefficient controlling the strength of regularization. o denotes a trajectory.

$$J = J_{PO} + \beta \frac{1}{|o|} \sum_{j=1}^{|o|} \mathcal{H}_j, \quad (2)$$

However, such naive entropy regularization exhibits notable limitations in the context of RLVR for LRMs. Cui et al. (2025) provides empirical evidence that it is highly sensitive to the entropy coefficient and prone to entropy collapse. The intuitive rationale behind this ineffectiveness lies in the large action spaces and long trajectories in LRMs, two characteristics that are markedly distinct from those of traditional RL tasks. In tasks like Hopper, Walker, or HalfCheetah, agents act in a 3–6 dimensional continuous space for a few hundred to about a thousand steps (Todorov et al., 2012). By contrast, LRMs select from hundreds of thousands of tokens over several thousand steps.

To better understand the phenomenon and validate our hypothesis, we conduct a preliminary experiment. Specifically, we train Qwen2.5-Math-7B with an entropy coefficient of 0.005, which is slightly higher than a suitable value for entropy regularization (as a reference, training with 0.001 does not lead to entropy explosion). All other datasets and hyperparameter configurations are kept consistent with the main experiment (Sec. 4.1). During training, we observe an entropy explosion at step 40 and use this checkpoint for comparison. We use a representative question from AIME24 (Li et al., 2024) and examine the probability distribution together with the generated tokens, qualitatively illustrating how the model behaves in the action space and over the trajectory, as shown in Figure 2. More cases can be found in Appendix F.

2.1 ANALYSIS OF TOKEN PROBABILITY DISTRIBUTIONS OVER ACTION SPACE

We first examine how RL with naive entropy regularization affects the action probabilities, i.e., the token probability distribution in LRMs. Figure 2(a) shows the distribution for the first token gener-

ated by the model before and after RL. Before RL, semantically meaningful tokens are concentrated at the top of the distribution, accounting for the vast majority of probability mass (red box in the left part of Figure 2(a)), while most other tokens are meaningless, with probabilities close to zero. After RL, however, the distribution becomes nearly uniform, with many semantically meaningless tokens rising to the top of the ranking. This phenomenon verifies our hypothesis: in LRMs, the extremely large action space and initially concentrated probabilities cause flattening of the distribution to yield disproportionately large entropy gains, reaching the maximum value of Eq. 1 and biasing the model toward increasing entropy.

Policy Nucleus. Motivated by this analysis, we introduce the concept of a policy nucleus to constrain exploration. The policy nucleus is defined as a subset of the vocabulary containing the most semantically important tokens, whose generation is reasonable and unlikely to cause incoherent responses. Since these tokens consistently occupy the top ranks in the original model’s probability distribution, we adopt the terminology of Top-p sampling (Holtzman et al., 2019) and refer to this subset as the nucleus. Our key insight is that **exploration should be concentrated within the policy nucleus rather than uniformly across the entire vocabulary**, offering an effective solution to the challenges posed by the vast action space in LRMs.

2.2 ANALYSIS OF TOKEN ALONG LONG TRAJECTORIES

Next, we analyze how strong entropy regularization affects states along a long trajectory, i.e., the generated tokens in LRMs. As shown in Figure 2(b), before RL, the model produces coherent and meaningful solution steps with relatively low average token-level entropy. After RL, however, the outputs degenerate into largely meaningless token sequences, accompanied by a substantial increase in average entropy. To further illustrate this effect, we use heatmaps to visualize token-level policy entropy. In the model after RL, nearly every token exhibits uniformly high entropy, indicating that the model explores extensively across almost all positions in the sequence. This may be because the autoregressive process turns early entropy spikes into a “chain reaction”, i.e., uncertainty begets uncertainty, so as sequence length increases, the chance of such cascades grows, making explosions more likely.

In contrast, the original model shows variation in the token entropy across positions. Consistent with previous studies (Cui et al., 2025; Cheng et al., 2025; Wang et al., 2025), we observe that a small fraction of high-entropy tokens serve logical roles in guiding the reasoning process, such as *To*, *Once*, *Then*, while most tokens, which primarily fill sentences, exhibit much lower entropy. This observation suggests that **token-level entropy can be used to identify critical tokens, allowing regularization to be applied selectively**. Such targeted intervention may help prevent cascade spikes in a long trajectory of LRMs.

3 SELECTIVE ENTROPY REGULARIZATION

Building on our preliminary analysis (Sec. 2), we introduce *SIREN* (SelectIve entRopy rEgularization), a method that selectively applies entropy regularization to prevent the entropy explosion caused by large action spaces and long trajectories. *SIREN* employs a two-step entropy masking mechanism: (1) a *top-p mask* that restricts exploration to the policy nucleus, and (2) a *peak-entropy mask* that confines exploration to critical tokens (Sec. 3.2). In addition, *SIREN* replaces the naive regularization with a *self-anchored regularization*, calibrating the regularization strength and improving training stability (Sec. 3.3). The overall framework is illustrated in Figure 3.

3.1 PRELIMINARY

Verifiable rewards remove the traditional reward model used in reinforcement learning and instead assign binary 0/1 rewards by directly comparing the model’s extracted answer with a predefined ground truth. Such rewards are widely adopted in tasks like mathematical reasoning and code generation, where correctness can be explicitly verified.

GRPO. Building on PPO (Schulman et al., 2017), Group Relative Policy Optimization (GRPO, Shao et al., 2024) models the entire response as a single action and computes the KL divergence

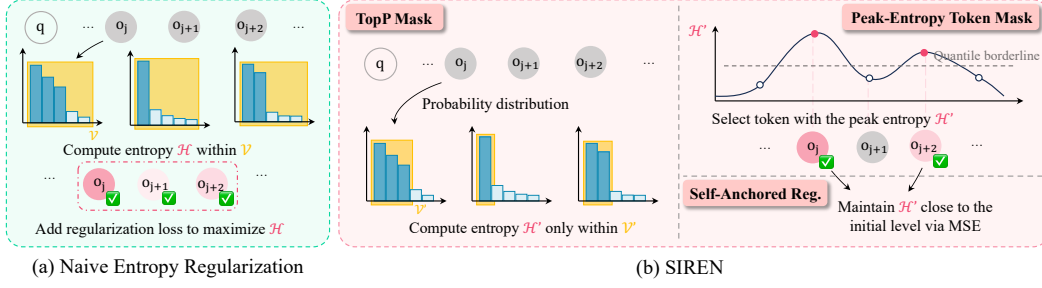


Figure 3: The pipeline of **SIREN** compared to naive entropy regularization. (a) Naive entropy regularization computes entropy over the entire vocabulary and maximizes it uniformly across all tokens, which often leads to excessive exploration in LRMs. (b) **SIREN** introduces selective entropy regularization, including: (i) a *top-p mask* that restricts entropy computation to the policy nucleus, (ii) a *peak-entropy token mask* that identifies the most informative tokens, and (iii) a *self-anchored form of regularization* that maintains the entropy magnitude close to its initial level. Together, these components enable more targeted exploration and improved training stability.

over the full sequence rather than token by token. For each query q , GRPO samples a group of outputs $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$. Given the binary reward of output o_i as $R(o_i)$, the group-normalized advantage $\hat{A}_i$ is defined as $\hat{A}_i = \frac{R(o_i) - \text{mean}(\{R(o_j)\}_{j=1}^G)}{\text{std}(\{R(o_j)\}_{j=1}^G)}$.

Then, inheriting from PPO, the training objective is:

$$J_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{P}(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min(\text{CLIP}(r_{i,t}(\theta), \epsilon) \hat{A}_{i,t}, r_{i,t}(\theta) \hat{A}_{i,t}) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right], \quad (3)$$

where $r_{i,t}(\theta) = \pi_{\theta}(o_{i,t}|q, o_{i,<t}) / \pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})$ is an importance sampling term and D_{KL} is the KL divergence.

Dr.GRPO. Dr.GRPO (Liu et al., 2025) modifies the original GRPO objective by removing the standard deviation term in the denominator of the advantage. The modified advantage is defined as $\hat{A}_i^{\text{Dr}} = R(o_i) - \text{mean}(\{R(o_j)\}_{j=1}^G)$, and correspondingly redefining the group-normalized advantage as:

$$J_{\text{Dr.GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{P}(Q), \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(O|q)} \left[\frac{1}{G} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min(\text{CLIP}(r_{i,t}(\theta), \epsilon) \hat{A}_{i,t}^{\text{Dr}}, r_{i,t}(\theta) \hat{A}_{i,t}^{\text{Dr}}) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right], \quad (4)$$

Our method builds on Dr.GRPO, as experiments demonstrate that Dr.GRPO achieves higher accuracy while producing shorter responses compared to GRPO (Liu et al., 2025). We further enhance it with our proposed regularization techniques.

3.2 SELECTIVE TWO-STEP ENTROPY MASKING

We first apply top-p masking to recompute entropy within the policy nucleus for each token, and then apply peak-entropy token masking to identify critical tokens. For regularization, the entropy and gradients are computed only on these selected candidates and tokens.

Top-p Mask within a Token. Since LRMs have an extremely large action space, exploration over the entire space often assigns high probability to meaningless tokens, leading to incoherent outputs. To address this, we introduce a *top-p mask* to constrain exploration within the policy nucleus, i.e., the subset of meaningful tokens that are worth exploring. A visualization is shown in the yellow

region on the left side of Figure 3(b). Formally, the policy nucleus is defined as:

$$\mathcal{V}_j^{(p)} = \arg \min_{\mathcal{S} \subseteq \mathcal{V}} |\mathcal{S}| \text{ s.t. } \sum_{v \in \mathcal{S}} P(v | v_{<j}) \geq p, \quad (5)$$

where $\mathcal{V}$ denotes the original vocabulary, and $P(v | v_{<j})$ is the probability of token v_j .

The top-p mask and the corresponding recomputed entropy are then defined as:

$$M_j^{\text{top-p}}(v) = \begin{cases} 1, & v \in \mathcal{V}_j^{(p)} \\ 0, & \text{otherwise,} \end{cases} \quad P'(v | v_{<j}) = \frac{M_j^{\text{top-p}}(v) P(v | v_{<j})}{\sum_{u \in \mathcal{V}} M_j^{\text{top-p}}(u) P(u | v_{<j})}, \quad (6)$$

$$\mathcal{H}'_j = - \sum_{v \in \mathcal{V}} P'(v | v_{<j}) \log P'(v | v_{<j}). \quad (7)$$

Peak-Entropy Mask for a Trajectory. Similarly, as analyzed in Sec.2, for long trajectories in LRMs, uncertainty accumulates exponentially, eventually causing entropy to explode at every token position, as exemplified in the right part of Figure 2(b). Only a small subset of tokens plays a disproportionately important role in exploration, typically corresponding to sentence heads or logical connectors that determine the reasoning direction, while most other tokens serve as semantic fillers. Consistent with prior observations (Cheng et al., 2025; Wang et al., 2025; Cui et al., 2025), we find that such critical tokens tend to exhibit relatively high entropy.

To focus on regularization on these tokens, we introduce a *peak-entropy mask*, which selects tokens whose entropy lies in the top τ -quantile within a trajectory. Concretely, for the i -th trajectory o_i in a batch, we define the τ -quantile of the token-level entropy, computed from the recomputed entropy $\mathcal{H}'$ in Eq. 7, as:

$$q_\tau^{(i)} = \text{Quantile}_\tau \left\{ \mathcal{H}'(v_1^{(i)}), \mathcal{H}'(v_2^{(i)}), \dots, \mathcal{H}'(v_{|o_i|}^{(i)}) \right\}, \quad (8)$$

and construct the peak-entropy mask $\mathbf{m}_j^{\text{peak}}$ by retaining tokens above the τ -quantile of entropy:

$$\mathbf{m}_j^{\text{peak}} = \begin{cases} 1, & \text{if } \mathcal{H}'(v_j) \geq q_\tau^{(i)}, \\ 0, & \text{otherwise.} \end{cases} \quad (9)$$

Finally, the entropy $\overline{\mathcal{H}}$ is aggregated over all trajectories within a batch as shown in Eq. 10, where G is the number of trajectories in the batch.

$$\overline{\mathcal{H}} = \frac{1}{\sum_{i=1}^G \sum_{j=1}^{|o_i|} \mathbf{m}_j^{\text{peak}}} \sum_{i=1}^G \sum_{j=1}^{|o_i|} \mathbf{m}_j^{\text{peak}} \cdot \mathcal{H}'(v_j^{(i)}). \quad (10)$$

3.3 SELF-ANCHORED REGULARIZATION

The naive objective of entropy regularization, which generally aims to maximize entropy (Haarnoja et al., 2018a; Liu et al., 2020), can easily fall into two extremes in the context of LRMs: either excessively high, leading to global explosion and incoherent outputs, or too low, failing to mitigate premature convergence of the policy. To address this issue and keep the policy’s entropy at an appropriate level, we modify the objective into a self-anchored regularization. It is defined as the mean squared error (MSE) between the aggregated entropy $\overline{\mathcal{H}}$ (computed as in Eq. 10) and an entropy anchor $\mathcal{H}_a$:

$$L_{sa} = (\overline{\mathcal{H}} - \mathcal{H}_a)^2. \quad (11)$$

Accordingly, the new training objective is formulated as $J' = J_{PO} - \beta L_{sa}$.

Instead of introducing an external hyperparameter for the entropy anchor $\mathcal{H}_a$, we initialize it using the aggregated entropy $\overline{\mathcal{H}}$ at the initial step (Eq. 10), which is why we term it *self-anchored*. This initialization is model- and dataset-specific, reflecting the inherent uncertainty of the pre-trained policy before accuracy-driven RL fine-tuning. At this stage, the model typically produces more diverse outputs with higher entropy, providing a natural and exploration-friendly starting point.

Table 1: Overall performance of Qwen2.5-Math-7B compared with Dr.GRPO, naive entropy regularization, and other RLVR methods that also incorporate entropy. All models are evaluated under a unified setting. Bold indicates the best results.

	AIME24		AIME25		AMC22		MATH500		Olympiad Bench		Average	
	Maj@32	Avg@32	Maj@32	Avg@32	Maj@32	Avg@32	Maj@8	Avg@32	Maj@8	Avg@32	Maj.	Avg.
Qwen2.5-Math-7B	30.0	11.0	13.3	3.8	60.2	29.8	70.8	46.4	25.3	15.4	39.9	21.3
Dr.GRPO	30.0	21.5	16.7	15.4	67.5	61.1	84.2	81.2	45.3	42.0	48.7	44.2
Naive Entropy Reg.	33.3	28.0	20.0	14.1	63.9	58.9	84.2	80.4	44.4	40.9	49.2	44.5
Clip-Cov	33.3	22.5	20.0	18.1	67.5	59.3	78.4	76.6	41.6	38.3	48.2	43.0
KL-Cov	36.7	26.7	16.7	15.1	66.3	60.2	84.0	79.6	44.0	40.9	49.5	44.5
Entropy Adv.	30.0	27.4	13.3	12.9	65.1	58.7	80.8	78.4	43.0	40.0	46.4	43.5
On Forking Tokens	36.7	24.0	20.0	13.9	62.7	57.2	84.6	81.1	45.0	40.9	49.8	43.3
SIREN (ours)	43.3	28.4	26.7	17.9	71.1	60.1	85.0	81.9	46.8	42.5	54.6	46.1

4 EXPERIMENT

4.1 SETUP

Models. The main experiments and analysis are conducted on Qwen2.5-Math-7B (Yang et al., 2024). We also extend our method to the smaller model Qwen2.5-Math-1.5B (Yang et al., 2024), and the weaker model Llama3.1-8B (Dubey et al., 2024), with results analyzed in Sec. 4.2.

Training Configurations. For the Qwen series models, following the setup in LUFFY (Yan et al., 2025), we train on the OpenR1-Math-46k-8192 dataset¹, a subset of OpenR1-Math-220k (Face). Consistent with DAPO (Yu et al., 2025), we remove the KL loss term and increase the clip ratio to 0.28. The rollout batch size is set to 128, and the update batch size is 8. We perform 8 rollouts per prompt and set the temperature to 1.0 for rollout generation. For Llama3.1-8B, given the relative weakness of the model, we use a simpler dataset constructed from the OpenR1-Math-46k-8192, GSM8K (Cobbe et al., 2021), and MATH500 (Hendrycks et al., 2021) training sets. When training Llama3.1-8b, we use 16 rollouts per prompt, with a rollout batch size of 512 and an update batch size of 32. Additional details on dataset construction and training can be found in Appendix B.

Evaluation. For evaluation, we focus on five widely used mathematical reasoning benchmarks, namely AIME24, AIME25, AMC (Li et al., 2024), OlympiadBench (He et al., 2024), and MATH500 (Hendrycks et al., 2021). We report maj@k and avg@k. Maj@k (majority voting, Wang et al., 2022) selects the most frequent answer among k samples and verifies its correctness. This metric does not require an external verifier and can simultaneously capture both the model’s exploration ability and its inherent confidence. Avg@k, which averages scores over k responses, is a commonly used metric that captures the model’s overall performance (Yu et al., 2025; Yan et al., 2025). For AIME24, AIME25, and AMC, we set n to 32 due to the relatively small test sets; for the remaining benchmarks, we set n to 8. The temperature is fixed at 0.6 for all evaluations. For each model, we select the checkpoint with the highest validation accuracy for evaluation, where validation is performed every 10 steps; detailed settings are provided in the Appendix B.

Baseline methods. We implement several prior approaches that incorporate entropy during reinforcement learning as baselines for comparison, including: (1) *Dr.GRPO* (Liu et al., 2025); (2) *Naive Entropy Regularization* with an entropy coefficient of 0.001; (3) *Clip-Cov / KL-Cov* (Cui et al., 2025): mitigating entropy collapse by either clipping tokens that exhibit high covariance between action probability and advantage, or by applying KL penalties to those tokens; (4) *Entropy Adv.* (Cheng et al., 2025): augmenting the advantage function with an entropy-based term; (5) *RL on forking tokens* (Wang et al., 2025): applying policy gradient updates only to high-entropy tokens.

4.2 MAIN RESULTS

SOTA performance on RLVR with Qwen2.5-Math-7B. Table 1 presents a comparative evaluation of the original Qwen2.5-Math-7B model, several baseline methods, and our proposed approach. **SIREN** yields substantial gains over the base model and achieves competitive or superior performance compared to strong RL baselines across benchmarks. Specifically, it achieves an average maj@k of **54.6**, outperforming the strongest baseline by **+4.8** points. In AIME24 and AIME25,

¹<https://huggingface.co/datasets/Elliott/Openr1-Math-46k-8192>

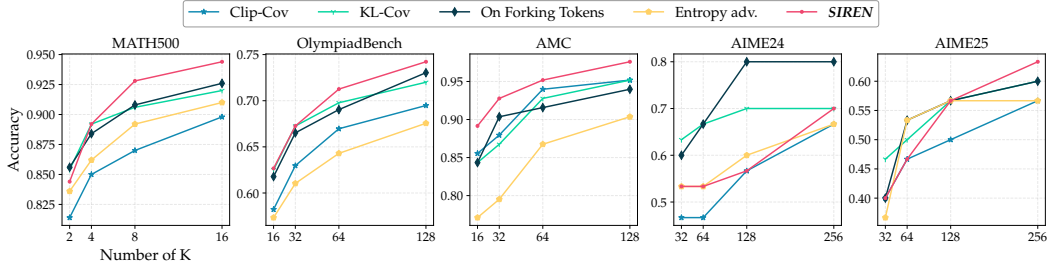


Figure 5: Pass@k performance of *SIREN* in comparison with various RLVR baselines, scaling with k .

the most challenging benchmarks, our method achieves a maj@32 score of **43.3** and **26.7**, respectively, achieving consistent improvements of **+6.6** points over the best-performing baseline on both datasets. For the overall average avg@n, our method reaches **46.1**, surpassing previous approaches and setting a new record.

Extending to Other Models. To further verify the generalizability of our method, we extend *SIREN* to smaller and weaker models. The evaluation results are presented in Figure 4. On the smaller model Qwen2.5-math-1.5B and the weaker model LLaMA3.1-8B, *SIREN* consistently achieves the best maj@k and avg@k across five mathematical benchmarks, outperforming both Dr.GRPO and naive entropy regularization. *SIREN* improves maj@k by +2.4 on Qwen2.5-math-1.5B and +2.8 on LLaMA3.1-8B.

5 ANALYSIS

We conduct a detailed analysis to investigate the relationship between *SIREN* and the exploration capability of LRMs. Specifically, we first evaluate the pass@k performance of *SIREN* (Figure 5), which illustrates its effectiveness in promoting exploration and expanding the reasoning boundary. We then compute the perplexity of responses generated by *SIREN* (Figure 6), showing that it preserves greater diversity than naive methods, thereby facilitating exploration. Next, we examine the training dynamics, including entropy variations and validation pass@16 (Figure 7), which further confirm *SIREN*’s ability to control exploration. Finally, we conduct an ablation study to assess the contribution of each module, highlighting the impact of individual components on overall performance.

5.1 *SIREN* PROMOTES EXPLORATION

Pass@k Analysis. We adopt pass@k to examine the exploration boundary of models trained with *SIREN*. Following prior work (Yue et al., 2025; Cheng et al., 2025), pass@k is widely recognized as an indicator of the upper bound of reasoning ability. Specifically, we set k to 256 for the difficult yet small-scale benchmarks (AIME, AIME 2025), k to 128 for the medium-level and relatively larger benchmarks (AMC, OlympiadBench), and k to 16 for the simpler benchmark (MATH500). The results are presented in Figure 5. *SIREN* achieves strong performance even at small values of k , with its advantage increasing as k grows, particularly on relatively easy benchmarks such as MATH500, AMC, and OlympiadBench. On the hardest benchmark, AIME25, *SIREN* achieves the best pass@k at the maximum k . On AIME24, however, it performs slightly worse than some baselines in pass@k while achieving higher maj@k and avg@k, as shown in Table 1. These results

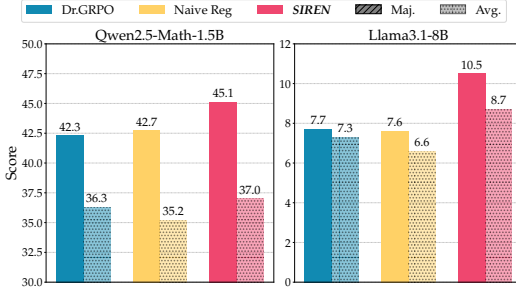


Figure 4: Average maj@k and avg@k across five mathematical benchmarks for *SIREN* on different backbones, including Qwen2.5-Math-1.5B and LLaMA3.1-8B (see Appendix D.1 for detailed results).

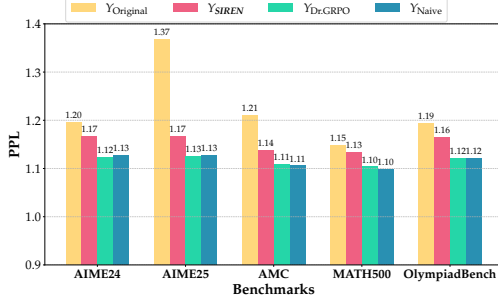


Figure 6: Average perplexity of generated responses across all benchmarks, computed using the base model Qwen2.5-Math-7B.

demonstrate that *SIREN* not only explores a wider range of possible responses, but also effectively balances exploration with exploitation, producing higher-quality and more consistent answers across multiple trials.

Perplexity Analysis. Besides pass@k, the exploratory effects promoted by *SIREN* can also be verified using perplexity (PPL). Following Yue et al. (2025), PPL measures a model’s ability to predict a response, with lower PPL indicating higher probability to generate response Y . Previous work (Yue et al., 2025) shows that RL reduces PPL, revealing a collapse in a smaller subset of responses and thus harming exploration. We quantitatively evaluated the original model’s PPL of responses generated by the original model, Dr.GRPO, naive entropy regularization, and *SIREN*. For responses, we reused the outputs obtained during the evaluation phase. As shown in Fig. 6, *SIREN* consistently achieves higher PPL compared to baseline RL methods, suggesting that it effectively mitigates the convergence of responses and encourages exploration over a broader range of possible outputs.

5.2 *SIREN* MAKES EXPLORATION MORE EFFECTIVE

We further study the training dynamics to understand why *SIREN* can promote exploration. We focus on two metrics closely related to exploration, namely entropy and validation pass@16, with the results shown in Figure 7. Dr.GRPO, Naive Reg (0.001), and *SIREN* (0.001) all exert little to no control over entropy, causing the policy to converge prematurely during the early stages of training. This lack of entropy regulation limits exploration of new states, leading to a decrease in validation pass@k. In contrast, Naive Reg (0.005) uses an excessively large entropy coefficient, resulting in rapid entropy explosion and poor performance. *SIREN* (0.005), however, maintains a substantial level of entropy during the early training stage, with a slight increase between steps 50–80, which encourages the generation of more diverse responses. As training progresses, the entropy gradually converges and eventually stabilizes at a higher level than the other baselines. By keeping entropy at an appropriate level throughout training, *SIREN* ensures continuous exploration, which is effectively converted into more correct responses and leads to improved validation pass@k. Comparison of *SIREN* with other baselines is provided in the Appendix E.2.

5.3 ABLATION STUDY

Our method consists of three components: *the peak-entropy mask, the top-p mask, and the self-anchored regularization*. We perform ablations by removing each component in turn. The average results are shown in Table 2.

In these benchmarks, removing any single component from our method leads to noticeable drops in both maj@k and avg@k. In particular, removing the *self-anchored regularization* causes the most severe degradation, with maj@k and avg@k decreasing by **+10.3** and **+15.5** points, respectively. This is because the self-anchored regularization balances performance and entropy: it sustains high

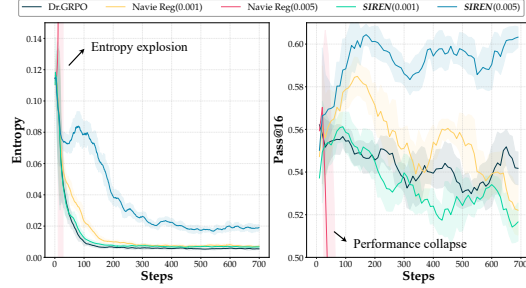


Figure 7: Training dynamics of entropy (left) and pass@16 performance (right) under different regularization strategies.

Table 2: Ablation results of *SIREN* on Qwen2.5-Math-7B.

	Maj.	Avg.
<i>SIREN</i>	54.6	46.1
w/o TopP Mask	49.2	43.1
w/o Peak-Entropy Mask	48.9	44.0
w/o Self-Anchored Reg.	44.3	30.6
w/ only Self-Anchored Reg.	52.3	43.6

entropy for exploration while maintaining stability. Without it, even when we set the coefficient to a very small value 0.0001, the entropy still rises to an excessively high level, leading to a severe drop in performance. For the *peak-entropy mask* and the *Top-p mask*, removing either leads to a drop in performance, indicating that each mask alone is insufficient to fully regulate the model’s exploration, while their combined usage yields the best results. Given the severe performance drop, we conduct an ablation study using *only self-anchored regularization*. This demonstrates that the stabilization technique can improve the performance of naive entropy regularization. However, its effectiveness remains inferior to that of *SIREN*. Hyperparameters and full experimental results are provided in Appendix D.2.

6 RELATED WORK

Reinforcement Learning for LLMs. Reinforcement learning has been a major technique for post-training large language models (LLMs) (Ouyang et al., 2022; Touvron et al., 2023; Rafailov et al., 2023). Recently, RLVR, which replaces the reward model with a verification function that compares the model’s boxed answer against the ground truth, has demonstrated substantial potential for improving reasoning capabilities (Shao et al., 2024; DeepSeek-AI, 2025; Jaech et al., 2024; Team et al., 2025). This has attracted increasing attention from the research community, leading to a variety of subsequent improvements (Zeng et al., 2025; Hu et al., 2025; Yeo et al., 2025; Liu et al., 2025; Yu et al., 2025). Yet, despite its promise, RLVR still faces significant challenges such as entropy collapse and premature convergence, which severely restrict exploration and limit the performance.

Exploration in reinforcement learning. Effective exploration remains an important topic in reinforcement learning. One line of research incorporates intrinsic motivation or reward bonuses to construct auxiliary signals that incentivize agents to explore new states (Bellemare et al., 2016; Ostrovski et al., 2017; Pathak et al., 2017; Burda et al., 2018; Fortunato et al., 2019). Another line of work improves exploration by maximizing entropy, thereby introducing uncertainty into the policy to encourage more diverse behaviors (Ziebart, 2010; Haarnoja et al., 2017; 2018a;c). As an important signal for policy optimization, entropy has also been utilized in RLVR. Cui et al. (2025) investigates the empirical correlation between entropy and performance, and proposes clipping or KL-based mechanisms to indirectly control entropy. Meanwhile, Cheng et al. (2025); Zheng et al. (2025); Wang et al. (2025) treat entropy as a heuristic signal, using it for advantage shaping, enhancing the rollout phase, or loss masking. Unlike the aforementioned methods, which exploit entropy only indirectly or as an auxiliary signal, our method analyzes the failure of naive entropy regularization and directly extends it from traditional RL to RLVR for LRMs. This enables explicit and effective control of entropy, thereby directly promoting policy exploration.

7 CONCLUSION

This work revealed that entropy collapse in LRMs arises from uncontrolled exploration, amplified by vast action spaces and long trajectories. To address this, we introduced *SIREN*, which increases naive regularization with a two-step entropy mask and self-anchored regularization. Experiments and analysis demonstrated that *SIREN* not only stabilizes training but also drives richer reasoning, surpassing previous entropy-based methods. Our results underlined a central insight: LRMs require targeted and effective exploration, providing a foundation for future advances in enhancing LRM exploration.

REFERENCES

- Marc Bellemare, Sriram Srinivasan, Georg Ostrovski, Tom Schaul, David Saxton, and Remi Munos. Unifying count-based exploration and intrinsic motivation. *Advances in neural information processing systems*, 29, 2016.
- Yuri Burda, Harrison Edwards, Amos Storkey, and Oleg Klimov. Exploration by random network distillation. *arXiv preprint arXiv:1810.12894*, 2018.

- Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*, 2025.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.
- Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, et al. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.
- DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL <https://arxiv.org/abs/2501.12948>.
- Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv-2407, 2024.
- Hugging Face. Open r1: A fully open reproduction of deepseek-r1, january 2025. URL <https://github.com/huggingface/open-r1>, pp. 9.
- Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, Charles Blundell, and Shane Legg. Noisy networks for exploration, 2019. URL <https://arxiv.org/abs/1706.10295>.
- Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018a.
- Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pp. 1861–1870. Pmlr, 2018b.
- Tuomas Haarnoja, Aurick Zhou, Kristian Hartikainen, George Tucker, Sehoon Ha, Jie Tan, Vikash Kumar, Henry Zhu, Abhishek Gupta, Pieter Abbeel, et al. Soft actor-critic algorithms and applications. *arXiv preprint arXiv:1812.05905*, 2018c.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.
- Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model. *arXiv preprint arXiv:2503.24290*, 2025.
- Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

- Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Huang, Kashif Rasul, Longhui Yu, Albert Q Jiang, Ziju Shen, et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.
- Zhuang Liu, Xuanlin Li, Bingyi Kang, and Trevor Darrell. Regularization matters in policy optimization—an empirical study on continuous control. In *International Conference on Learning Representations*, 2020. URL <https://openreview.net/forum?id=yrlmzrH3IC>.
- Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.
- Georg Ostrovski, Marc G Bellemare, Aäron Oord, and Rémi Munos. Count-based exploration with neural density models. In *International conference on machine learning*, pp. 2721–2730. PMLR, 2017.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.
- Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *International conference on machine learning*, pp. 2778–2787. PMLR, 2017.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in neural information processing systems*, 36:53728–53741, 2023.
- John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, pp. 1279–1297, 2025.
- Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.
- Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pp. 5026–5033. IEEE, 2012.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.
- Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xionghui Chen, Jianxin Yang, Zhenru Zhang, et al. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *arXiv preprint arXiv:2506.01939*, 2025.
- Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.
- Jianhao Yan, Yafu Li, Zican Hu, Zhi Wang, Ganqu Cui, Xiaoye Qu, Yu Cheng, and Yue Zhang. Learning to reason under off-policy guidance. *arXiv preprint arXiv:2504.14945*, 2025.

- 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. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.
- 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, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.
- Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning in llms. *arXiv preprint arXiv:2502.03373*, 2025.
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.
- Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*, 2025.
- Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun Ma, and Junxian He. Simplerl-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. *arXiv preprint arXiv:2503.18892*, 2025.
- Kaiyan Zhang, Yuxin Zuo, Bingxiang He, Youbang Sun, Runze Liu, Che Jiang, Yuchen Fan, Kai Tian, Guoli Jia, Pengfei Li, Yu Fu, Xingtai Lv, Yuchen Zhang, Sihang Zeng, Shang Qu, Haozhan Li, Shijie Wang, Yuru Wang, Xinwei Long, Fangfu Liu, Xiang Xu, Jiase Ma, Xuekai Zhu, Ermo Hua, Yihao Liu, Zonglin Li, Huayu Chen, Xiaoye Qu, Yafu Li, Weize Chen, Zhenzhao Yuan, Junqi Gao, Dong Li, Zhiyuan Ma, Ganqu Cui, Zhiyuan Liu, Biqing Qi, Ning Ding, and Bowen Zhou. A survey of reinforcement learning for large reasoning models, 2025. URL <https://arxiv.org/abs/2509.08827>.
- Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*, 2025.
- Brian D Ziebart. *Modeling purposeful adaptive behavior with the principle of maximum causal entropy*. Carnegie Mellon University, 2010.

A LIMITATIONS

We believe there is still room for improvement in our work. First, due to limited computational resources, we restrict our experiments to models with up to 8B parameters. Previous work (Wang et al., 2025) suggests that the benefits of enhanced exploration may be more pronounced in larger models, so evaluating *SIREN* at larger scales remains an important direction for future work. Second, the range of models considered in this study is relatively narrow. Although we included both weaker and stronger baselines and extended our analysis to the Llama series, other categories are not covered, such as distilled models and instruction-tuned models. Finally, our training and evaluation are limited to the mathematical reasoning domain. We did not examine other RLVR scenarios, such as code generation or scientific reasoning, which may further highlight the generality of *SIREN*.

Future Directions Future directions involve exploring additional approaches for controlling entropy. For instance, annealing strategies may provide a more precise mechanism to regulate the exploration process. Moreover, richer signals could be leveraged for token selection, such as correctness feedback or importance sampling scores. Extending our study beyond the mathematical reasoning domain represents an important direction. Applying *SIREN* to other domains and modalities, such as code generation or multimodal reasoning, may further demonstrate its generality and effectiveness.

B EXPERIMENTAL DETAILS

Construction Method of Dataset for Llama3.1-8B. We manually construct an easier dataset for Llama3.1-8B. Specifically, for prompts drawn from OpenR1-Math-46K-8192, the GSM8K training set, and the MATH500 training set, we employ LLaMa3.1-8B-Instruct, LLaMa3.1-72B, and LLaMa3.1-72B-Instruct to generate 8 responses per prompt. We assume that stronger models from the same series can reflect the potential performance of weaker ones. During sampling, the temperature is set to 0.6 and the maximum response length is set to 8192. We filter out prompts for which none of the models can produce a correct answer. The resulting dataset contains 35K examples. The system prompt is shown in Appendix C.

Training Qwen2.5-Series Models. For the Qwen2.5 series models, we sample 8 responses per prompt. The learning rate is fixed at $1e-6$ with a warm-up of 5 steps. The rollout batch size is set to 128, and the mini-batch size for updates is 8 prompts, meaning that the policy is updated 16 times per rollout. The maximum response length is 3072 tokens, and the maximum prompt length is 1024 tokens; prompts exceeding this limit are filtered out before training. We train the models for 2 epochs, corresponding to a total of 714 steps.

Most experiments on Qwen2.5-Math-7B are conducted on 16 NVIDIA A800 GPUs across 2 nodes (8 GPUs per node), with two exceptions: (1) *SIREN* on Qwen2.5-Math-7B with an entropy coefficient of 0.001, and (2) the ablation experiment using only self-anchored regularization. Both are conducted on 8 NVIDIA H200 GPUs. For Qwen2.5-Math-1.5B, all experiments are conducted on 4 NVIDIA H200 GPUs.

Our implementation is based on verl (Sheng et al., 2025)². We employ vLLM³ as the rollout generator and math-verify⁴ for answer extraction and verification. We gratefully acknowledge the contributions of these open-source repositories.

Training Llama3.1-8B. For Llama3.1-8B, due to the relative weakness of the model, we sample 16 responses per prompt. The rollout batch size is set to 512, and the mini-batch size for updates is 32, resulting in 16 policy updates per rollout. The maximum response length is 2048 tokens and the maximum prompt length is 1024 tokens. We train the model for 2 epochs on a dataset of 35K examples, corresponding to a total of 136 steps. We directly adopt the chat template of Llama3.1-8B-instruct without any modification. All experiments are conducted on 8 NVIDIA H200 GPUs.

²<https://github.com/volcengine/verl>

³<https://github.com/vllm-project/vllm>

⁴<https://github.com/huggingface/Math-Verify>

Table 3: Overall performance on five mathematical benchmarks, extending to Qwen2.5-Math-1.5B and LLaMa3.1-8B.

	AIME24		AIME25		AMC22		MATH500		Olympiad Bench		Average	
	<i>Maj@32</i>	<i>Avg.</i>	<i>Maj@32</i>	<i>Avg.</i>	<i>Maj@32</i>	<i>Avg.</i>	<i>Maj@8</i>	<i>Avg.</i>	<i>Maj@8</i>	<i>Avg.</i>	<i>Maj.</i>	<i>Avg.</i>
<i>Qwen2.5-Math-1.5B</i>												
Qwen2.5-Math-1.5B	23.3	8.2	13.3	4.2	51.8	27.5	59.4	35.3	31.0	21.6	35.8	19.4
Dr.GRPO	16.7	14.1	16.7	8.4	55.4	44.1	81.2	77.3	41.5	37.5	42.3	36.3
Naive Entropy Reg	23.3	15.3	20.0	9.8	57.8	48.1	75.4	70.8	36.9	32.1	42.7	35.2
<i>SIREN</i>	20.0	13.8	23.3	12.0	60.2	47.9	80.2	75.7	41.6	36.0	45.1	37.0
<i>LLaMa3.1-8B</i>												
LLaMa3.1-8B	3.3	0.1	0.0	0.0	2.4	1.5	8.8	5.1	1.5	1.4	2.5	1.6
Dr.GRPO	0.0	0.0	0.0	0.0	6.0	7.4	25.8	22.5	6.8	6.1	7.7	7.3
Naive Entropy Reg	3.3	0.1	3.3	0.5	8.4	7.6	20.6	19.5	5.6	5.2	7.6	6.6
<i>SIREN</i>	0.0	0.0	3.3	1.4	13.3	9.7	28.4	25.1	7.7	7.1	10.5	8.7

Validation. For validation, we use the full test sets of the five datasets. Validation is performed every 10 steps. For each question, 16 responses are generated, with the sampling temperature set to 0.6 and the maximum response length kept consistent with training.

Evaluation. For evaluation, we set the temperature to 0.6 and the maximum number of new tokens to 32,768 to avoid artificially truncating responses. We use vLLM to generate the responses and math-verify to evaluate them.

C SYSTEM PROMPT

System prompt for Qwen-series

Your task is to follow a systematic, thorough reasoning process before providing the final solution. This involves analyzing, summarizing, exploring, reassessing, and refining your thought process through multiple iterations. Structure your response into two sections: Thought and Solution. In the Thought section, present your reasoning using the format: “<think>\n thoughts </think>\n”. Each thought should include detailed analysis, brainstorming, verification, and refinement of ideas. After “</think>\n” in the Solution section, provide the final, logical, and accurate answer, clearly derived from the exploration in the Thought section. If applicable, include the answer in `\boxed{ }` for closed-form results like multiple choices or mathematical solutions.

System prompt for LLaMa3.1-8B

Your task is to follow a systematic, thorough reasoning process before providing the final solution. This involves analyzing, summarizing, exploring, reassessing, and refining your thought process through multiple iterations. Structure your response into two sections: Thought and Solution. In the Thought section, each thought should include detailed analysis, brainstorming, verification, and refinement of ideas. In the Solution section, provide the final, logical, and accurate answer, clearly derived from the exploration in the Thought section. If applicable, include the answer in `\boxed{ }` for closed-form results like multiple choices or mathematical solutions. Let’s think step by step.

System prompt for validation and evaluation

Please reason step by step, and put your final answer within `\boxed{ }`.

Table 4: Overall performance of Qwen2.5-Math-7B compared with Dr.GRPO, naïve entropy regularization, and other RLVR methods that also incorporate entropy. All models are evaluated under a unified setting. Bold indicates the best results.

	AIME24		AIME25		AMC22		MATH500		Olympiad Bench		Average	
	Maj@32	Avg@32	Maj@32	Avg@32	Maj@32	Avg@32	Maj@8	Avg@32	Maj@8	Avg@32	Maj.	Avg.
SIREN	43.3	28.4	26.7	17.9	71.1	60.1	85.0	81.9	46.8	42.5	54.6	46.1
w/o TopP Mask	40.0	26.4	16.7	14.4	68.7	60.4	81.6	77.9	39.3	36.7	49.2	43.1
w/o Peak-Entropy Mask	33.3	29.0	26.7	16.8	65.1	59.2	80.0	77.9	39.4	37.3	48.9	44.0
w/o Self-Anchored Reg.	23.3	15.7	13.3	6.1	60.2	43.9	81.8	59.9	43.0	27.6	44.3	30.6
w/ only Self-Anchored Reg.	30.0	22.3	26.7	13.5	67.5	58.2	87.4	80.7	49.8	43.2	52.3	43.6

D ADDITIONAL RESULTS

D.1 EXTENDING TO OTHER MODELS

Table 3 reports detailed results on five mathematical benchmarks. On Qwen2.5-Math-1.5B, **SIREN** achieves an average maj@k of 45.1 and avg@k of 37.0, outperforming naive baselines by +2.4 on average maj@k, respectively. On Llama3.1-8B, **SIREN** attains 10.5 maj@k and 8.7 avg@k, also surpassing all other baselines.

D.2 ABLATION STUDY

Hyperparameters For the ablation experiment with the self-anchored regularization, we set the entropy coefficient to 0.005, consistent with the main experiment. For the experiment without self-anchored regularization, we set the entropy coefficient to 0.0001 to avoid entropy explosion as much as possible.

Results The detailed performance of the ablation study is reported in Table 4. As shown, removing each component (top-p mask, peak-entropy mask, or self-anchored regularization) leads to a performance drop, with the removal of the self-anchored regularization causing the most severe decline: -10.3 maj@k and -15.5 avg@k. This highlights the critical role of self-anchored regularization in maintaining training stability and enabling the effectiveness of other modules.

Since the performance degradation caused by removing the self-anchored regularization is too significant, we also conduct an ablation experiment using only this component. The results show that self-anchored regularization can also improve the performance of naive entropy regularization, yielding notable gains, especially on relatively easier benchmarks such as MATH500 and Olympiad-Bench. However, the overall average performance still lags behind **SIREN**, indicating that each component in **SIREN** contributes effectively.

E ANALYSIS

E.1 **SIREN** SELECTS EFFECTIVE TOKENS

To provide a more intuitive illustration of what tokens are selected by our method, we construct a word cloud. Specifically, we randomly sample 8 questions from MATH500 and use Qwen2.5-7B-Math to generate 16 responses for each question, with the hyperparameters kept consistent with those used in the evaluation. We then collect the selected tokens and draw a word cloud with a maximum of 200 words (see Fig. 8). As shown in the figure, the high-frequency words are concentrated on core terms closely related to mathematical problem statements and reasoning, such as Given, The, To, Answer, and Consider, along with typical mathematical concepts such as product, sum, value, and equation. This demonstrates that our method effectively captures the key information in problem descriptions and solution processes rather than being dominated by redundant or noisy tokens, thereby further validating the rationality and effectiveness of the proposed mask strategy.



Figure 8: Word cloud of tokens selected by *SIREN*.

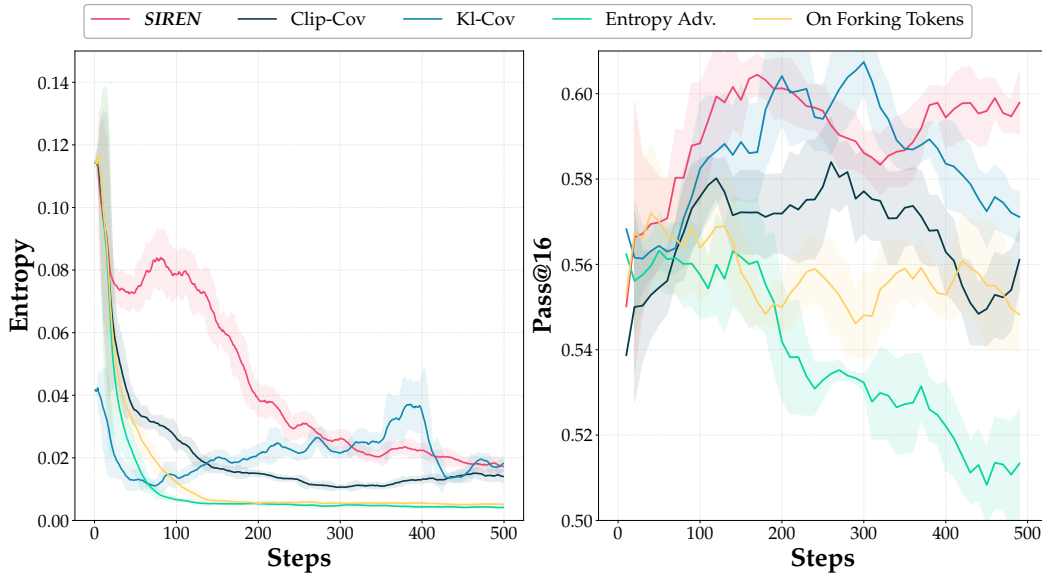


Figure 9: Training dynamics of entropy and validation pass@16, comparing *SIREN* with other RLVR baselines.

E.2 *SIREN* MAKES EXPLORATION MORE EFFECTIVE (PART 2)

We study the training dynamics of the RLVR baseline, with results shown in Figure 9. Consistent with the main text, *SIREN* maintains higher entropy and pass@k, indicating more sustained

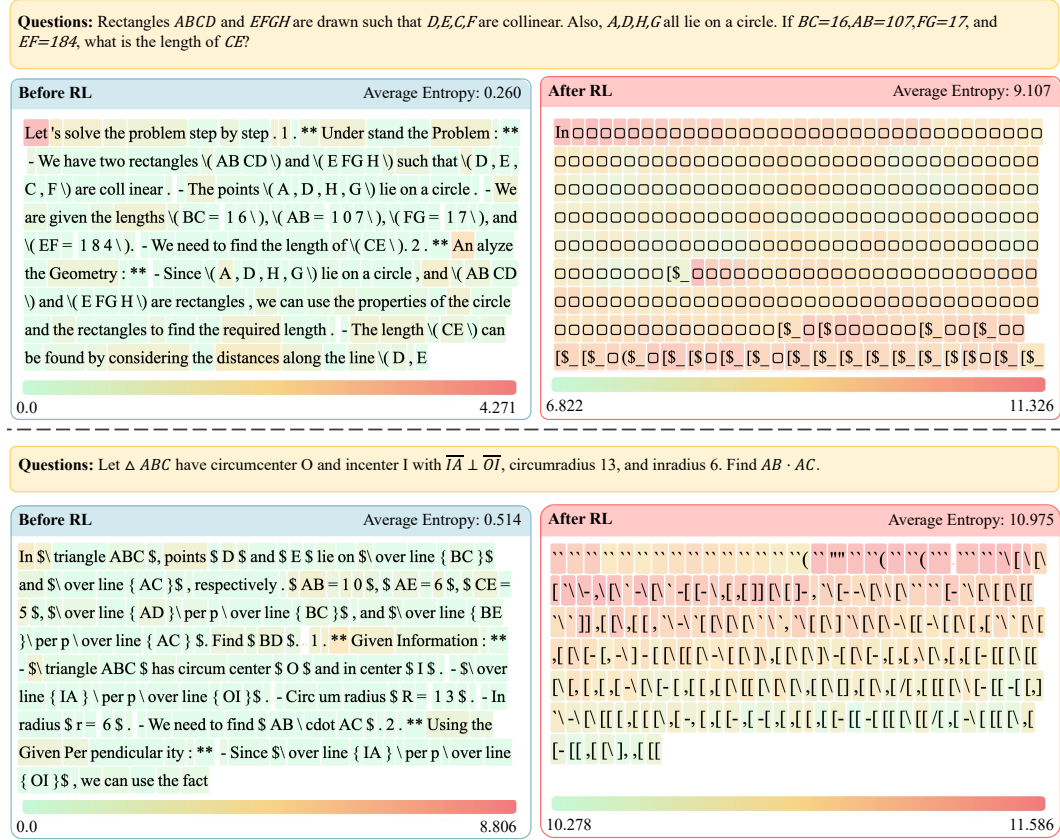


Figure 10: More comparison cases between the original model and the entropy-exploded model, with questions sampled from AIME25 (first 200 tokens).

exploration. Interestingly, $KI-Cov$ also preserves entropy and pass@k well; however, it introduces additional computational overhead during training due to the need to compute the KL divergence with the original model.

F CASE STUDY

F.1 MORE CASES OF PRELIMINARY ANALYSIS

We provide additional cases comparing generated tokens from the original model and the entropy-exploded model, as shown in Fig. 10. The observations are consistent with the main text: (1) after RL training with entropy explosion, the generated content becomes meaningless; (2) in the original model, tokens with higher entropy tend to correspond to logically critical positions, such as initiating or bridging a line of reasoning and elaborating individual reasoning steps, and so on.

F.2 COMPARISON OF RESPONSES GENERATED BY *SIREN* AND RL BASELINES

Compared with the baseline *Dr.GRPO*, *SIREN* emphasizes structured reasoning and step-by-step verification, providing more accurate solutions.

Question and Ground Truth

Question: Alice chooses a set A of positive integers. Then Bob lists all finite nonempty sets B of positive integers with the property that the maximum element of B belongs to A . Bob's list has 2024 sets. Find the sum of the elements of A .

Ground Truth: 55

Response from RL Baseline

Total Length: 1043 Tokens

To solve this problem, let's start by understanding the relationship between the set A and the sets B that Bob lists. The key point is that the maximum element of each set B must be an element of A . If A has n elements, then the number of possible sets B is related to the number of ways we can choose a maximum element from A and then choose any subset of the elements less than the maximum element to be the remaining elements of B .

If A has n elements, then for each element $a \in A$, we can form sets B where a is the maximum element. The number of such sets B is $2^{a-1} - 1$ (we subtract 1 to exclude the empty set). Summing this over all elements of A gives the total number of sets B . If $A = \{a_1, a_2, \dots, a_n\}$, then the total number of sets B is:

$$\sum_{i=1}^n (2^{a_i-1} - 1) = \sum_{i=1}^n 2^{a_i-1} - n$$

We know that this sum is equal to 2024. So we need to find a set A such that:

$$\sum_{i=1}^n 2^{a_i-1} - n = 2024$$

This can be rearranged to:

$$\sum_{i=1}^n 2^{a_i-1} = 2024 + n$$

We need to find a set A that satisfies this equation. One way to approach this is to start by finding the smallest possible set A and then iterating to find the correct set. Let's start by assuming A has the smallest possible elements, i.e., $A = \{1, 2, 3, \dots, n\}$. Then the sum becomes:

$$\sum_{i=1}^n 2^{i-1} = 2^0 + 2^1 + 2^2 + \dots + 2^{n-1} = 2^n - 1$$

We need to find n such that:

$$2^n - 1 = 2024 + n$$

We can solve this equation by trial and error. Let's start by checking small values of n .

For $n = 11$:

$$2^{11} - 1 = 2047 \quad \text{and} \quad 2024 + 11 = 2035$$

This is close but not quite right. Let's try $n = 10$:

$$2^{10} - 1 = 1023 \quad \text{and} \quad 2024 + 10 = 2034$$

This is too small. So $n = 11$ is the correct value. Now we need to find the set A that satisfies the equation. The set A must be $\{1, 2, 3, \dots, 11\}$, but we need to check if there is a smaller set that works.

We can see that the sum $2^{11} - 1 = 2047$ is too large, so we need to find a smaller set. The correct set A must be a subset of $\{1, 2, 3, \dots, 11\}$ that satisfies the equation. The correct set is $A = \{11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$, but we need to find a smaller set.

The correct set A is $\{11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$, but we need to find a smaller set. The correct set is $A = \{11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1\}$, but we need to find a smaller set.

The sum of the elements of A is:

$$11 + 10 + 9 + 8 + 7 + 6 + 5 + 4 + 3 + 2 + 1 = 66$$

So the final answer is: 66

Response from SIREN**Total Length: 1987 Tokens**

Let's begin by understanding the problem. We are given that Bob lists all finite nonempty sets B of positive integers such that the maximum element of B belongs to a set A chosen by Alice. The number of such sets B is 2024, and we need to find the sum of the elements of A .

Let's denote the set A as $A = \{a_1, a_2, \dots, a_n\}$ where $a_1 < a_2 < \dots < a_n$. For each element a_i in A , the sets B that have a_i as their maximum element are all the nonempty subsets of $\{1, 2, \dots, a_i - 1\}$. The number of nonempty subsets of a set with $a_i - 1$ elements is $2^{a_i - 1} - 1$. Therefore, the total number of sets B is the sum of these for all a_i in A :

$$(2^{a_1 - 1} - 1) + (2^{a_2 - 1} - 1) + \dots + (2^{a_n - 1} - 1) = 2024$$

This simplifies to:

$$2^{a_1 - 1} + 2^{a_2 - 1} + \dots + 2^{a_n - 1} - n = 2024$$

$$2^{a_1 - 1} + 2^{a_2 - 1} + \dots + 2^{a_n - 1} = 2024 + n$$

We need to find a set of integers $\{a_1, a_2, \dots, a_n\}$ such that the sum of powers of 2 is $2024 + n$. To find the correct set, we first note that $2024 + n$ must be expressible as a sum of distinct powers of 2. The largest power of 2 less than or equal to 2024 is $2^{10} = 1024$. We can start by expressing 2024 in binary:

$$2024_{10} = 11111101000_2 = 1024 + 512 + 256 + 128 + 64 + 32 + 8$$

This means:

$$2024 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3$$

If we add n to 2024, we need to check if the resulting sum can be expressed as a sum of distinct powers of 2. To find the correct n , we can check the next smallest values. If $n = 1$, then:

$$2024 + 1 = 2025$$

The binary representation of 2025 is:

$$2025_{10} = 11111101001_2 = 1024 + 512 + 256 + 128 + 64 + 32 + 8 + 1$$

This means:

$$2025 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3 + 2^0$$

This is a sum of distinct powers of 2, so it is valid. Therefore, the set A could be $\{11, 10, 9, 8, 7, 6, 3, 1\}$. The sum of the elements of A is:

$$11 + 10 + 9 + 8 + 7 + 6 + 3 + 1 = 55$$

To verify, the number of sets B is:

$$(2^{11-1}-1)+(2^{10-1}-1)+(2^{9-1}-1)+(2^{8-1}-1)+(2^{7-1}-1)+(2^{6-1}-1)+(2^{3-1}-1)+(2^{1-1}-1)$$

$$= (2^{10} - 1) + (2^9 - 1) + (2^8 - 1) + (2^7 - 1) + (2^6 - 1) + (2^5 - 1) + (2^2 - 1) + (2^0 - 1)$$

$$= 1024 - 1 + 512 - 1 + 256 - 1 + 128 - 1 + 64 - 1 + 32 - 1 + 4 - 1 + 1 - 1$$

$$= 1024 + 512 + 256 + 128 + 64 + 32 + 4 + 1 - 8$$

$$= 2025 - 8 = 2017$$

This is not 2024, so we need to check the next value. If $n = 2$, then:

$$2024 + 2 = 2026$$

The binary representation of 2026 is:

$$2026_{10} = 11111101010_2 = 1024 + 512 + 256 + 128 + 64 + 32 + 8 + 2$$

This means:

$$2026 = 2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^5 + 2^3 + 2^1$$

This is a sum of distinct powers of 2, so it is valid. Therefore, the set A could be $\{11, 10, 9, 8, 7, 6, 3, 1\}$ with an additional element, say 2. The sum of the elements of A is:

$$11 + 10 + 9 + 8 + 7 + 6 + 3 + 1 + 2 = 57$$

But, the number of sets B is:

$$(2^{11-1}-1)+(2^{10-1}-1)+(2^{9-1}-1)+(2^{8-1}-1)+(2^{7-1}-1)+(2^{6-1}-1)+(2^{3-1}-1)+(2^{1-1}-1) \\ + (2^{2-1}-1)$$

$$= (2^{10}-1) + (2^9-1) + (2^8-1) + (2^7-1) + (2^6-1) + (2^5-1) + (2^2-1) + (2^0-1) + (2^1-1)$$

$$= 1024 - 1 + 512 - 1 + 256 - 1 + 128 - 1 + 64 - 1 + 32 - 1 + 4 - 1 + 1 - 1 + 2 - 1$$

$$= 1024 + 512 + 256 + 128 + 64 + 32 + 4 + 1 + 2 - 9$$

$$= 2026 - 9 = 2017$$

This is not 2024, so the correct set A must be $\{11, 10, 9, 8, 7, 6, 3, 1\}$ with $n = 1$. The sum of the elements of A is 55. So, the final answer is:

$$\boxed{55}$$