Reinforced Efficient Reasoning via Semantically Diverse Exploration

Ziqi Zhao[1] **Zhaochun Ren**[2] **Jiahong Zou**[1] **Liu Yang**[1] **Zhiwei Xu**[1] **Xuri Ge**[1]
Zhumin Chen[1] **Xinyu Ma**[3] **Daiting Shi**[3] **Shuaiqiang Wang**[3] **Dawei Yin**[3] **Xin Xin**[1*]
[1]Shandong University [2]Leiden University [3]Baidu Inc.
ziqizhao.work@gmail.com z.ren@liacs.leidenuniv.nl
xinxin@sdu.edu.cn

Abstract

Reinforcement learning with verifiable rewards (RLVR) has proven effective in enhancing the reasoning of large language models (LLMs). Monte Carlo Tree Search (MCTS)-based extensions improve upon vanilla RLVR (e.g., GRPO) by providing tree-based reasoning rollouts that enable fine-grained and segment-level credit assignment. However, existing methods still suffer from limited exploration diversity and inefficient reasoning. To address the above challenges, we propose **r**einforced efficient reas**o**ning via **s**emantically diverse **e**xplorations, i.e., **ROSE**, for LLMs. To encourage more diverse reasoning exploration, our method incorporates a semantic-entropy-based branching strategy and an ε-exploration mechanism. The former operates on already sampled reasoning rollouts to capture semantic uncertainty and select branching points with high semantic divergence to generate new successive reasoning paths, whereas the latter stochastically initiates reasoning rollouts from the root, preventing the search process from becoming overly local. To improve efficiency, we design a length-aware segment-level advantage estimator that rewards concise and correct reasoning while penalizing unnecessarily long reasoning chains. Extensive experiments on various mathematical reasoning benchmarks with Qwen and Llama models validate the effectiveness and efficiency of ROSE. Codes are available at https://github.com/ZiqiZhao1/ROSE-rl.

1 Introduction

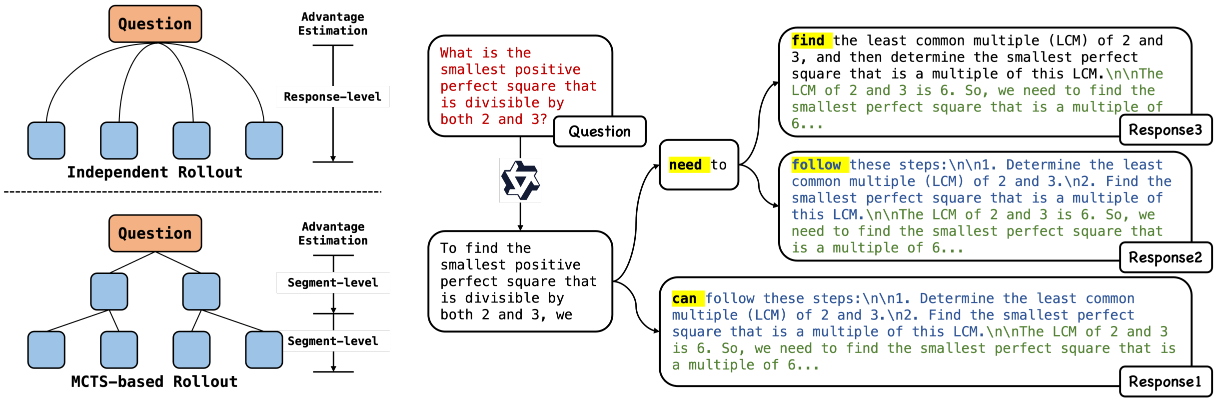
Reinforcement learning with verifiable rewards (RLVR) has recently been proposed to enhance the reasoning of large language models (LLMs) in verifiable settings, including mathematical reasoning and code generation (Guo et al., 2025; Shao et al., 2024; Liu et al., 2025; Yu et al., 2025). Typical RLVR algorithms, such as GRPO (Guo et al., 2025; Shao et al., 2024) and its variants (Yu et al., 2025; Liu et al., 2025), estimate the advantage of an entire rollout response based on the verified reward and uniformly propagate this advantage to all tokens within the response.

While the uniform credit assignment is simple yet effective, it constrains the learning potential of the model and conflicts with human intuition. For example, a reasoning chain that produces an incorrect response may still contain certain correct steps. Moreover, recent studies have indicated that this training paradigm may lead to "overthinking", in which models are engaged in redundant reasoning (Chen et al., 2024; Dai et al., 2025; Nie et al., 2026). To further improve model performance, a more effective credit assignment approach is to employ Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006) during response rollout sampling. Unlike vanilla GRPO, which generates a group of independent responses for a given problem, MCTS enables the model to produce responses in a tree-based structure, as illustrated in Figure 1a, allowing segment-level credit assignment by computing value differences between parent and child nodes.

Despite the progress achieved by MCTS-based RLVR algorithms (Li et al., 2025; Yang et al., 2025b; Zheng et al., 2025b; Dong et al., 2025), **limited exploration diversity** and **inefficient reasoning** still exist. Specifically, most existing work uses generation entropy as the criterion for MCTS branching (Zheng et al., 2025b; Dong et al., 2025). These methods first identify the position with the highest generation entropy. Then, tokens preceding this position are kept fixed, and successive tokens are regenerated. Although generation entropy measures a policy's uncertainty over token selection in the current action space, this metric does not generalize well to the semantic space. Figure 1b shows a case study of generation entropy-based branching. The tokens *can* in response 1 and *need* in response 2 correspond to the positions of high-

*Corresponding author.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 47994–48007



(a) (b)

Figure 1: (a) Comparison between independent rollout (vanilla GRPO) and MCTS-based rollout. (b) Case study of generation-entropy-based branching. The tokens highlighted in yellow indicate the different tokens generated at the positions of highest entropy. Identical text across different responses is marked with the same colour (green or blue).

est entropy in two separate generations, yet their semantic meanings are largely the same, and the subsequent reasoning in both responses is identical (highlighted in blue). Furthermore, although response 2 and response 3 follow different reasoning paths after branching, they are semantically similar, and their subsequent reasoning remains consistent with other responses (highlighted in green). This indicates that current methods fail to generate semantically diverse rollouts. Additionally, existing MCTS-based methods do not address the overthinking problem effectively, and current approaches for efficient reasoning either incur performance degradation or offer trivial performance gains (Dai et al., 2025; Arora and Zanette, 2025). How to achieve both improved performance and efficient reasoning based on MCTS remains an open question.

To address the aforementioned challenges, we propose **r**einforced efficient reas**o**ning via **s**emantically diverse **e**xplorations, i.e., **ROSE**, for LLMs. To address the first challenge, we introduce a semantic-entropy-based branching strategy together with an ε-exploration mechanism. The semantic entropy metric, defined over differences in token semantics, identifies positions along a reasoning path where the model exhibits high uncertainty in the semantic space, thereby guiding the exploration toward more diverse reasoning paths. In addition, to prevent the search process from becoming overly local, the ε-exploration mechanism stochastically regenerates the reasoning rollout from scratch. Together, these methods promote more diverse and effective exploration. To address the second challenge, we integrate credit assignment with the length of the reasoning chain. Leveraging the tree structure, we estimate values for each node and assign credit at the segment level. For different correct reasoning chains originating from the same node, longer chains with deeper depth are penalized to encourage more efficient reasoning. These components make fuller use of MCTS samples, aiming to enhance the model's reasoning ability through more diverse and efficient exploration. In summary, our contributions are:

- We introduce a semantic-entropy guided MCTS-based rollout strategy together with an ϵ-exploration mechanism, which enables more diverse exploration compared with existing approaches.
- We propose a segment-level advantage estimation method that incorporates reasoning length, enabling stronger performance while producing more efficient reasoning.
- Extensive experiments on a wide range of mathematical reasoning tasks (AIME2025, AIME2024, AMC2023, MATH500), using both Qwen and Llama models, validate the effectiveness and efficiency of our approach.

2 Related Work

2.1 Reinforcement Learning for LLMs

Reinforcement learning has been widely adopted to align LLMs with human preferences through reinforcement learning from human feedback (RLHF) (Lee et al., 2024; Ouyang et al., 2022). More recently, reinforcement learning with verifiable rewards (RLVR) has emerged as an effective approach for enhancing the reasoning ability of LLMs (Guo et al., 2025; Shao et al., 2024; Dai et al., 2025; Hao et al., 2025; Meng et al., 2025; Zhou et al., 2026; Wu et al., 2026; Xie et al., 2025; Wu et al., 2025b). By using rule-based binary (0/1) rewards to simplify reward design, GRPO (Guo et al., 2025; Shao et al., 2024) removes the need for training an extra critic model compared with vanilla PPO (Schulman et al., 2017), leading to a substantial reduction in RL training overhead. Recent studies, including DAPO (Yu et al., 2025), Dr.GRPO (Liu et al., 2025), VAPO (Yue et al., 2025), GSPO (Zheng et al., 2025a), and CPG (Chu et al., 2025), have explored improving the GRPO loss function to further enhance its reasoning capability. In contrast to these approaches, our work focuses on improving the rollout process to enable more diverse exploration and credit assignment, without modifying the loss function. As a result, the proposed method is in principle compatible with a wide range of GRPO-based algorithms.

2.2 MCTS for LLM Reasoning

Monte Carlo Tree Search (MCTS) (Kocsis and Szepesvári, 2006; Świechowski et al., 2023) offers a principled framework for exploring structured decision spaces, making it a natural candidate for performing credit assignment based on the intermediate reasoning steps. Recent studies have explored MCTS-based sampling in RL training, showing progress on mathematical reasoning tasks (Li et al., 2025; Yang et al., 2025b; Zheng et al., 2025b) as well as other complex problems (Ji et al., 2025; Dong et al., 2025).

A key challenge in applying MCTS lies in deciding where to branch, as this choice fundamentally determines the exploration trajectory and the quality of the reasoning. Prior approaches rely on random branching (Ji et al., 2025), generation-entropy-based branching (Dong et al., 2025; Zheng et al., 2025b), branching based on fixed-length segments (Li et al., 2025), or performing branching during decoding via beam search (Yang et al., 2025b). However, all these strategies fall short in promoting sufficient diverse exploration. Meanwhile, existing methods do not explicitly account for the impact of reasoning length during advantage estimation, which can lead to overthinking during model inference. In contrast, our approach enhances exploration diversity and enables more efficient reasoning, leading to improved performance on complex reasoning tasks.

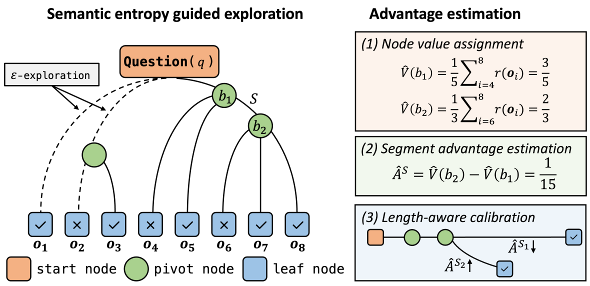


Figure 2: The overview of the ROSE framework. The figure on the left illustrates the structure of the tree-based rollout. Pivot nodes refer to nodes with the highest semantic uncertainty, which are selected according to the semantic entropy. The rollout procedure is detailed in Section 3.1. The figure on the right depicts the advantage estimation pipeline, comprising three stages: (1) node value assignment, (2) segment advantage estimation and (3) length-aware calibration. These stages are described in detail in Section 3.2.

3 Method

This section details ROSE. We first introduce how to achieve effective and diverse exploration in Section 3.1. Section 3.2 then describes how tree-based exploration is leveraged to perform advantage estimation and encourage efficient reasoning. Finally, Section 3.3 presents the overall learning objective. Figure 2 illustrates an overview of ROSE.

3.1 Semantic-Entropy Guided Exploration

Given a question q, vanilla GRPO performs rollouts by sampling a group of independent responses $\{\mathbf{o}_i\}_{i=1}^{G}$. In contrast, MCTS-based methods introduce tree-structured rollouts, allowing different responses to share common prefix tokens. The key to tree-based rollout is identifying appropriate branching positions, which encourages the model to perform more effective exploration. A common practice is to use generation entropy to determine branching positions (Dong et al., 2025; Zheng et al., 2025b). Generation entropy provides a principled measure of the uncertainty of a policy π_θ, i.e., the LLM, over its action space, i.e., the vocabulary $\mathcal{V}$. Given a question q and a generated response $\mathbf{o}_i$, the generation entropy of the policy at position k is defined as:

$$\mathcal{H}_k = -\sum_{v\in\mathcal{V}} p_\theta(v|q, \mathbf{o}_{i,<k}) \log p_\theta(v|q, \mathbf{o}_{i,<k}) \tag{1}$$

where $p_\theta(v|q, \mathbf{o}_{i,<k})$ represents the probability distribution over the vocabulary $\mathcal{V}$ at position k. Such entropy has been widely used in traditional RL (Haarnoja et al., 2018; Wang et al., 2022),

where different actions often exhibit significant differences, such as movement directions in a game environment (Bellemare et al., 2013). However, this assumption does not always hold in language generation. Consider the two words *can* and *need* in Figure 1b. When the LLM is uncertain about which one to select, the generation entropy may be high. From a semantic perspective, however, this choice is actually well-determined, as both words serve the same functional role of indicating modal intent. As a result, the responses branched from this position may exhibit extremely high similarity, and in some cases even follow identical subsequent trajectories, thereby limiting the potential for more diverse exploration.

Based on this observation, we design an additional metric to evaluate the **semantic divergence** among the current candidate tokens. Specifically, given a question q and a generated response $\mathbf{o}_i$, at position k, we first select the top-20 tokens from $\mathcal{V}$ with the highest probabilities to form the set $\mathcal{V}_k$ for efficiency. Then, for each token $v_i \in \mathcal{V}_k$, its corresponding embedding $\mathbf{e}_{v_i}$ is obtained from the LLM. We then compute semantic divergence as the sum of pairwise similarities between all tokens in $\mathcal{V}_k$, weighted by their probabilities:

$$\mathrm{SD}_k = -\sum_{v_i, v_j \in \mathcal{V}_k} p_\theta(v_i|q, \mathbf{o}_{i,<k}) p_\theta(v_j|q, \mathbf{o}_{i,<k}) \cdot \cos\langle \mathbf{e}_{v_i}, \mathbf{e}_{v_j} \rangle \tag{2}$$

where $\cos\langle \mathbf{e}_{v_i}, \mathbf{e}_{v_j} \rangle$ represents the cosine similarity between the embeddings of tokens v_i and v_j. The key idea of semantic divergence is that when the high-probability tokens exhibit large semantic differences, the current position becomes an ideal branching point, leading to more distinct subsequent reasoning paths.

Finally, we define **semantic entropy** as the product of generation entropy and semantic divergence, and use it as the branching indicator:

$$SE_k = SD_k \cdot \mathcal{H}_k \tag{3}$$

This combined measure captures both probabilistic uncertainty and semantic dispersion, allowing ROSE to more accurately identify positions where alternative continuations are more likely to lead to genuinely diverse reasoning paths.

The rollout process based on branching metrics is summarized as follows. Given a question q, a complete response is first generated. For each position in the generated response, the proposed branching metric is computed, and the position with the highest value is selected. Then a new response is regenerated at this position, keeping the preceding part of the response unchanged. The corresponding metric of the newly generated sequence is computed, and the selection is then performed on all existing rollout responses. The whole process is repeated until the number of generated responses reaches the predefined parameter G.

In addition, inspired by the ε-greedy (Sutton et al., 1998) strategy in reinforcement learning, we propose an **ε-exploration** mechanism. Specifically, before generating each response, there is an ε probability of generating the response from scratch, i.e., rolling out an independent response; otherwise, the rollout follows the proposed semantic-entropy-based branching strategy. This mechanism prevents the search from becoming overly focused on local regions and further balances the depth and breadth of exploration. After completing the rollout process for a given query, we can obtain a tree structure, an example of which is shown on the left side of Figure 2. During rollout, we apply dynamic sampling (Yu et al., 2025) to remove groups whose responses receive identical rewards, improving efficiency. The proposed methods offer an effective exploration-exploitation tradeoff to better search the reasoning paths for LLMs.

3.2 Advantage Estimation

Based on the tree-structured exploration, we perform segment-level credit assignment through (1) node value assignment, (2) segment advantage estimation and (3) length-aware calibration.

Node value assignment. After completing tree-structured sampling, a single response may contain multiple branching nodes. Including the start and the terminal positions, these nodes partition a response into several consecutive segments. Formally, for a response $\mathbf{o}_i$ to a given question q, let $b_0, b_1, \ldots, b_k$ denote the node positions, where b_0 is the start position, $b_1, \ldots, b_{k-1}$ correspond to the pivot positions immediately before each branching point, and b_k is the leaf (terminal) position. The response can then be decomposed as

$$\mathbf{o}_i = \bigcup_{j=1}^{k} \mathbf{o}_{i, b_{j-1} < t \le b_j}, \text{with } b_0 = 0, b_k = |\mathbf{o}_i| \tag{4}$$

Under this partition, each segment is initiated at either the start position or a branching position selected based on maximal semantic entropy observed during the rollout stage. For a pivot node b_j

with $0 < j < k$, we define the set of responses that contain this node as:

$$\Omega_{b_j} = \{\mathbf{o}_m | \mathbf{o}_m \text{ traverses the pivot node } b_j\} \quad (5)$$

For the start node b_0, we define Ω_{b_0} as the set of all responses, i.e., $\Omega_{b_0} = \{\mathbf{o}_m\}_{m=1}^{G}$. For the leaf node, we define Ω_{b_k} as the set of $\mathbf{o}_i$, i.e., $\Omega_{b_k} = \{\mathbf{o}_i\}$.

Next, we define the value of node b_j with $0 \leq j \leq k$ as the average reward of responses in Ω_{b_j}.

$$\hat{V}(b_j) = \frac{1}{|\Omega_{b_j}|} \sum_{\mathbf{o}_m \in \Omega_{b_j}} r(\mathbf{o}_m) \quad (6)$$

where $r()$ denotes a rule-based reward function that evaluates the correctness of each response and assigns a binary reward (1 for correct and 0 for incorrect).

Segment advantage estimation. Next, we can compute the segment-level advantage between two nodes based on the values assigned to the nodes. According to the definition of node value, the value of a node can be interpreted as the probability of deriving a correct reasoning chain starting from that node. Therefore, the reasoning contribution of the segment is quantified by the difference between the two node values. Specifically, for any token $\mathbf{o}_{i,t} \in \mathbf{o}_i$ with $b_{j-1} < t \leq b_j$, the advantage of $\mathbf{o}_{i,t}$ is defined as:

$$\hat{A}_{i,t} = \hat{V}(b_j) - \hat{V}(b_{j-1}) \quad (7)$$

Length-aware calibration. Furthermore, although multiple reasoning paths may lead to correct outcomes, we aim to encourage the model to adopt more efficient reasoning and avoid overthinking. To this end, we apply a length-aware calibration to the advantages of responses that are correct but require an excessive number of tokens. Specifically, we first identify the shortest correct response $\mathbf{o}_s$. Then, for every other correct response $\mathbf{o}_c$, we locate the pivot node b_c at which $\mathbf{o}_s$ and $\mathbf{o}_c$ diverge. That is, prior to b_c, the two responses share an identical subsequence $\mathbf{o}_{s,\leq b_c} = \mathbf{o}_{c,\leq b_c}$, whereas after b_c they follow distinct continuations $\mathbf{o}_{s,>b_c}$ and $\mathbf{o}_{c,>b_c}$, respectively. A length-proportional calibration is then applied to the longer response, thereby encouraging the model to produce more efficient reasoning. Specifically, for each token $\mathbf{o}_{c,t} \in \mathbf{o}_c$ with $t > b_c$, its advantage is updated according to the following rule:

$$\hat{A}_{i,t} \leftarrow \hat{A}_{i,t} - |\hat{A}_{i,t}| \cdot (1 - (\frac{|\mathbf{o}_s| - b_c}{|\mathbf{o}_c| - b_c})^{\alpha}) \quad (8)$$

where α is a hyperparameter controlling the extent of the adjustment. The ratio $\frac{|\mathbf{o}_s|-b_c}{|\mathbf{o}_c|-b_c}$ measures the relative lengths of the two reasoning branches after their divergence at b_c. Since the two responses share an identical reasoning prefix before b_c, their post-divergence segments can be directly compared: the more efficient branch receives a higher advantage, while the longer branch incurs a length-proportional adjustment.

3.3 Model Training

We adopt the improved modifications of vanilla GRPO's optimizing objective proposed in Dr.GRPO (Liu et al., 2025) as the training objective, together with a KL penalty term:

$$\mathcal{L}_{\text{ROSE}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\sum_{t=1}^{|\mathbf{o}_i|}\Bigg(\min\Big(r_{i,t}(\theta)\hat{A}_{i,t}, [r_{i,t}(\theta)]_{1-\epsilon'}^{1+\epsilon'}\hat{A}_{i,t}\Big) - \beta D_{\text{KL}}(\pi_\theta \| \pi_{\text{ref}})\Bigg), \quad (9)$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(\mathbf{o}_{i,t} \mid q, \mathbf{o}_{i,<t})}{\pi_{\theta_{\text{old}}}(\mathbf{o}_{i,t} \mid q, \mathbf{o}_{i,<t})}, \quad (10)$$

π_{old} denotes sampling model, π_{ref} denotes reference model and operator $[r_{i,t}(\theta)]_{1-\epsilon'}^{1+\epsilon'}$ clips the ratio to $[1-\epsilon', 1+\epsilon']$.

4 Experimental Setup

4.1 Datasets and Metrics

For the training dataset, following prior studies (Zhu et al., 2025; Liu et al., 2025), we use MATH (Hendrycks et al., 2021), which contains 7,500 problems. For evaluation, four publicly available standard mathematical reasoning benchmarks are considered, including AIME2024, AIME2025, AMC23, and MATH500. MATH500 is a subset of the test split of the MATH dataset, consisting of 500 problems. During validation, we sample 8 responses for each question and adopt pass@8 as the primary metric for assessing the performance of model reasoning. pass@k measures whether at least one of the k sampled responses correctly solves a given problem. Unlike prior work that relies on the mean@k metric (Li et al., 2025; Yang et al., 2025b), which reflects the average accuracy across all samples, pass@k more effectively captures the model's ability to solve previously challenging problems that it might fail to answer.

Model	Dataset	Base Model	GRPO Variants			MCTS-Based		ROSE
			GRPO	DAPO	Dr.GRPO	TreePO	FR3E	
`Qwen3-4B-Base`	AIME2024	13.33	16.67	16.67	16.67	16.67	16.67	**23.33**+6.67
	AIME2025	16.67	20.00	16.67	23.33	13.33	20.00	**23.33**+3.33
	MATH500	74.00	79.8	79.00	78.60	**82.00**	80.00	80.80-1.20
	AMC23	45.00	**77.50**	75.00	70.00	72.50	75.00	**77.50**+0.00
	Average	37.25	48.49	46.83	47.14	46.12	47.92	**51.24**+2.75
`Qwen3-8B-Base`	AIME2024	13.33	23.33	26.67	26.67	23.33	23.33	**33.33**+6.67
	AIME2025	10.00	23.33	23.33	23.33	23.33	23.33	**30.00**+6.67
	MATH500	68.20	79.40	79.40	81.60	**84.20**	80.80	83.00-1.20
	AMC23	47.50	72.50	75.00	72.50	70.00	75.00	**80.00**+5.00
	Average	34.76	49.64	51.10	51.02	50.21	50.62	**55.75**+4.65
`Llama-3.2-3B-Ins.`	AIME2024	10.00	16.67	16.67	13.33	16.67	16.67	**20.00**+3.33
	AIME2025	0.00	3.33	3.33	**6.67**	3.33	**6.67**	**6.67**+0.00
	MATH500	46.00	53.40	54.60	54.40	52.60	54.40	**55.00**+0.40
	AMC23	35.00	40.00	37.50	40.00	35.00	37.50	**45.00**+5.00
	Average	22.75	28.35	28.02	28.60	26.90	28.81	**31.67**+2.86

Table 1: Experimental results with pass@8 metric (%). For each test dataset, we report the best scores achieved during training. Boldface denotes the best results under each dataset. The absolute improvement or degradation compared to the second-best score is also indicated.

4.2 Model and Baselines

To provide a more comprehensive comparison of the proposed method, we evaluate it using backbone models from two model families, Qwen and Llama, with different parameter scales, including `Llama-3.2-3B-Instruct` (Grattafiori et al., 2024), `Qwen3-4B-Base`, and `Qwen3-8B-Base` (Yang et al., 2025a). The `Qwen3` models have two modes (thinking and non-thinking), and the non-thinking mode is adopted for both training and inference.

Comparisons are performed between ROSE and existing approaches, which mainly fall into two categories: GRPO-based variants and MCTS-based methods. The GRPO-based variants include vanilla GRPO (Guo et al., 2025; Shao et al., 2024), Dr.GRPO (Liu et al., 2025), and DAPO (Yu et al., 2025). Dr.GRPO computes advantages as deviations from the group mean, without variance normalization, and removes the length normalization term from the loss function. DAPO improves GRPO by incorporating techniques such as clip-higher and rejection sampling.

The MCTS-based baselines include FR3E (Zheng et al., 2025b) and TreePO (Li et al., 2025). FR3E is a representative MCTS-based method that determines branching positions based on generation entropy and adopts a two-step framework for segment-level advantage computation. Besides, TreePO structures the rollout process as a tree by branching at fixed-length segments and computes advantages over the resulting sub-trees.

4.3 Implementation Details

All experiments are conducted using the VeRL framework (Sheng et al., 2025) in this paper. For RL training, we set the batch size to 512, the number of rollouts per prompt as $G = 8$, the learning rate to 1×10^{-6}, the clipping ratio as $\epsilon' = 0.2$, the KL divergence coefficient as $\beta = 0.001$, and the maximum number of training epochs to 8. For evaluation, the temperature is set to 0.6, top-p sampling is applied with $p = 0.95$, and 8 candidate responses are sampled per prompt. Prompts whose lengths exceed 2048 tokens are filtered out, and the maximum generation length is set to 4096 tokens. The probability of generating the response from scratch ε is set to 0.5 by default, and the coefficient for length-aware calibration α is searched from $\{0.5, 1, 2, 3\}$. Our experiments are conducted on $8 \times$ NVIDIA A800 (80G) GPUs.

5 Experimental Results

5.1 Overall Performance

Accuracy evaluation. Table 1 presents the experimental results of all methods. It can be observed that ROSE achieves substantial improvements over the strongest baseline in most settings. DAPO and Dr.GRPO are variants that modify the GRPO loss function. However, they do not yield consistent or substantial improvements, with performance gains

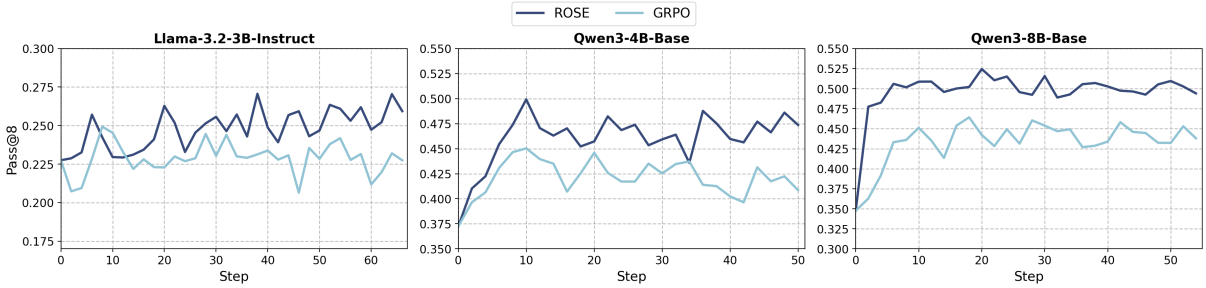


Figure 3: Learning curves. Average performance across four datasets as training progresses.

Metric	AIME2024	AIME2025	MATH500	AMC23	Average
Llama-3.2-3B-Instruct					
Generation Entropy	16.67	**6.67**	55.2	42.5	30.26
Semantic Divergence	**20.00**	**6.67**	54.6	42.5	30.94
Semantic Entropy	**20.00**	**6.67**	**55.0**	**45.00**	**31.67**
Qwen3-4B-Base					
Generation Entropy	20.00	23.33	79.80	72.5	48.91
Semantic Divergence	20.00	**26.67**	**80.80**	75.00	50.62
Semantic Entropy	**23.33**	23.33	**80.80**	**77.50**	**51.24**
Qwen3-8B-Base					
Generation Entropy	20.00	23.33	81.20	72.5	49.26
Semantic Divergence	30.00	26.67	**83.00**	75.00	53.67
Semantic Entropy	**33.33**	**30.00**	**83.00**	**80.00**	**55.75**

Table 2: Experimental results with pass@8 metric (%) for different branching metrics.

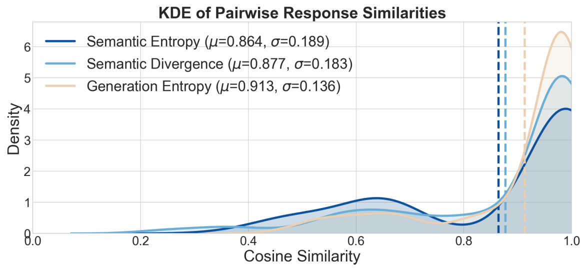


Figure 4: Kernel density estimation (KDE) of pairwise sentence similarities. The dashed line indicates the average cosine similarity.

observed only in certain scenarios, such as models with larger parameter scales. Among MCTS-based approaches, TreePO and FR3E achieve performance comparable to GRPO and its variants. In particular, TreePO yields pronounced improvements on the in-domain dataset MATH500 but performs worse on other benchmarks. This suggests that its fixed-length branching strategy fails to induce more diverse reasoning trajectories during exploration, limiting out-of-domain generalization.

For ROSE, significant performance gains are first observed on more challenging tasks, indicating that the method facilitates more divergent exploration during the rollout phase, which is beneficial for solving difficult problems. In addition, ROSE consistently yields notable improvements across different model scales. Larger models typically encapsulate richer knowledge, and the proposed approach appears to leverage this capacity more effectively, resulting in greater performance gains. Finally, recent studies have suggested that models in the Qwen family may suffer from potential data leakage (Wu et al., 2025a). Nevertheless, comparable performance gains are also observed on Llama models of similar parameter scales, indicating that the improvements are not confined to a specific model family.

Learning dynamics. Figure 3 presents the learning curves of GRPO and ROSE. Across different model scales, ROSE exhibits a clear performance improvement over the vanilla GRPO. Moreover, as the model scale increases, the learning curves of ROSE become noticeably more stable. After convergence, ROSE maintains stable and competitive performance, whereas the vanilla GRPO shows noticeable fluctuations and fails to achieve significant improvements in small-scale (3B and 4B) models.

5.2 Branching Metric Analysis

Performance comparison. Table 2 presents the experimental results for three different branching metrics (entropy, semantic divergence, and semantic entropy). For all branching metrics, the probability ε is fixed to 0.5. The results show that entropy achieves performance comparable to vanilla GRPO, indicating that it fails to effectively distinguish uncertain regions in the model's reasoning trajectories. In contrast, both semantic divergence and semantic entropy yield consistent improvements over entropy, with semantic entropy exhibiting greater robustness across different datasets.

Analysis of reasoning diversity. To further analyze the differences among branching metrics, we conduct a quantitative analysis of the diversity of responses generated during the rollout phase under different branching metrics. Specifically, for a fixed batch of questions, rollouts are per-

Method	Llama-3.2-3B-Instruct		Qwen3-8B-Base	
	Pass@8	Length	Pass@8	Length
Base Model	22.75 +0.0	788.6 +0.0%	34.76 +0.0	932.7 +0.0%
GRPO	28.35 +5.6	804.8 +2.1%	49.64 +14.9	907.3 -2.7%
Dr.GRPO	28.60 +5.9	794.5 +0.7%	51.02 +16.3	930.6 -0.2%
ROSE				
⊢ $\alpha = 0$	30.98 +8.2	733.1 -7.0%	55.17 +20.4	907.2 -2.7%
⊢ $\alpha = 1$	**31.67** +8.9	702.0 -11.0%	**55.75** +21.0	904.4 -3.0%
⊢ $\alpha = 2$	31.29 +8.5	715.4 -9.3%	54.87 +20.1	897.2 -3.8%
⊢ $\alpha = 3$	30.78 +8.0	692.6 -12.2%	55.12 +20.4	885.8 -5.0%
⊢ $\alpha = 10$	29.06 +6.3	634.3 -19.6%	54.29 +19.5	860.4 -7.8%

Table 3: Experimental results with pass@8 (%) and length (token counts) metrics.

formed using different metrics, and the pairwise cosine similarity between embeddings of multiple responses corresponding to the same question is computed. The embeddings are obtained using the `Qwen-text-embedding-v4` model. The resulting similarity distributions are visualized using kernel density estimation (KDE), as shown in Figure 4.

As illustrated in the figure, the distributions induced by our methods exhibit lower peaks and heavier tails. The mean similarities of semantic entropy and semantic divergence are comparable and both are lower than those of entropy, indicating a higher degree of dispersion among generated responses. Such increased reasoning diversity encourages broader exploration of the solution space, which aligns with the observed performance gains on more challenging benchmarks.

5.3 Reasoning Efficiency Analysis

To investigate whether ROSE can achieve more efficient reasoning, we evaluate different values of the hyperparameter α and report the corresponding pass@8 scores and reasoning lengths. The averaged results across the four datasets are presented in Table 3.

The results show a clear trend that increasing α reduces the reasoning length while maintaining strong pass@8 performance. In particular, moderate values of α (e.g., $\alpha = 1$ or $\alpha = 2$) yield the best trade-off between accuracy and efficiency, achieving higher pass@8 scores together with substantial reductions in reasoning length. Even with a relatively large value of α (i.e., $\alpha = 10$), our method still consistently outperforms GRPO variants in terms of pass@8. Overall, these results demonstrate that ROSE enables more efficient reasoning without sacrificing task performance. The evolution of response length across training steps is presented in Appendix 5.6.

Method	L-3B-Ins.	Q-4B-Base	Q-8B-Base
ROSE	**31.67**	**51.24**	**55.75**
⊢ w/o ε-exploration	26.44	48.81	49.27
⊢ w/ random branching	29.94	48.05	49.02
⊢ w/o advantage estimation	30.43	49.21	52.32

Table 4: Experimental results with pass@8 metric (%). `L` denotes `Llama-3.2` and `Q` denotes `Qwen3`.

5.4 Ablation Study

An ablation study is conducted to analyze the contribution of each component, with the results presented in Table 4. **(1) w/o ε-exploration** removes the ε-possibility branching mechanism (i.e., $\varepsilon = 0$), which results in consistent performance drops across all backbone models, indicating that the model tends to fall into overly local exploration and loses exploration diversity. Additional results examining different values of ε are provided in Appendix 5.7. **(2) w/ random branching** randomly determines the branching positions during rollout, which leads to performance degradation, indicating that the semantic entropy metric can effectively identify uncertain points along the reasoning trajectories. **(3) w/o advantage estimation** removes the segment-level advantage estimation and instead directly uses the GRPO advantage formulation and loss function. This modification leads to degraded performance, which suggests that segment-level advantage estimation plays an important role in shaping learning signals during reasoning.

5.5 Case Study

We also conduct case studies and observe that our semantic-entropy-based approach can identify semantical uncertain positions along the reasoning paths, encouraging more diverse reasoning. Detailed examples are provided in the Appendix A.

5.6 Response Length Dynamics

To further investigate the effect of the hyperparameter α, we examine how the average response length per prompt during the rollout stage and the evaluation stage evolves over training steps, with the results presented in Figure 5. We observe that as α increases, the generated response lengths in both the rollout and evaluation stages decrease substantially, and are consistently shorter than those of Dr.GRPO and GRPO. This indicates that our method can effectively regulate the generation length by adjusting α, thereby enabling more efficient reasoning.

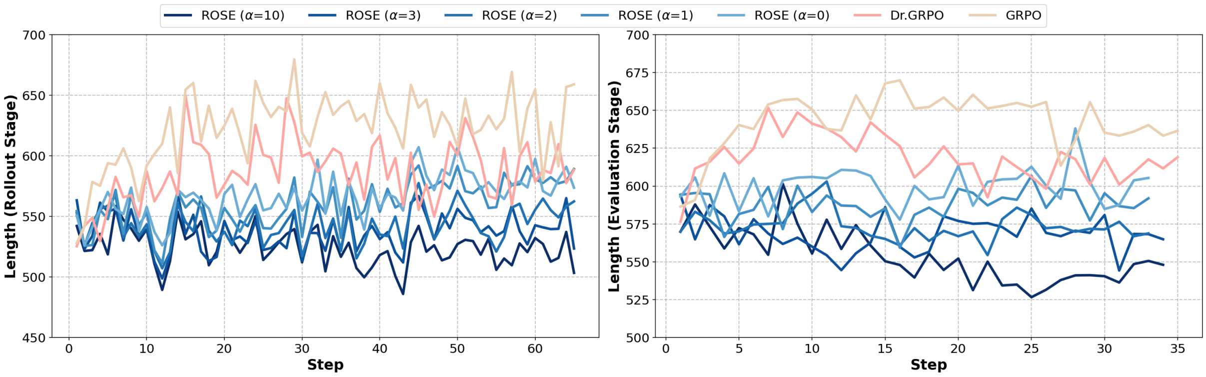


Figure 5: The average response length per prompt during the rollout stage (left) and the evaluation stage (right)

Metric	AIME2024	AIME2025	MATH500	AMC	Average
Llama-3.2-3B-Instruct					
ε=0	13.33	3.33	51.60	37.50	26.44
ε=0.3	**20.00**	3.33	54.60	37.50	28.86
ε=0.5	16.67	**6.67**	**55.60**	**45.00**	**30.98**
ε=0.7	16.67	3.33	54.20	**45.00**	29.80
ε=1 (Dr.GRPO)	13.33	**6.67**	54.40	40.00	28.60
Qwen3-4B-Base					
ε=0	20.00	**23.33**	79.40	72.50	48.81
ε=0.3	20.00	20.00	79.60	75.00	48.65
ε=0.5	**23.33**	**23.33**	**80.80**	**80.00**	**51.87**
ε=0.7	20.00	**23.33**	79.20	75.00	49.38
ε=1 (Dr.GRPO)	16.67	**23.33**	78.60	70.00	47.15
Qwen3-8B-Base					
ε=0	26.67	20.00	80.40	70.00	49.27
ε=0.3	**33.33**	**26.67**	83.20	75.00	54.55
ε=0.5	**33.33**	**26.67**	**83.20**	**77.50**	**55.17**
ε=0.7	30.00	**26.67**	83.00	75.00	53.67
ε=1 (Dr.GRPO)	26.67	23.33	81.60	72.50	51.02

Table 5: Experimental results with pass@8 metric (%) under different ε-exploration possibilities.

5.7 Impact of ε-exploration

We investigate the impact of different values of ε in ε-exploration on model performance, with the results reported in Table 5. When $\varepsilon = 1$, i.e., each rollout is sampled independently, our method degenerates into the Dr.GRPO algorithm. We observe that, across all backbone models, performance first improves and then degrades as ε increases. When $\varepsilon = 0$, exploration becomes overly local, which hinders diversity in the model's exploration. Conversely, when $\varepsilon = 1$, the tree-structured exploration is lost, preventing effective segment-level advantage estimation.

6 Conclusion

In this work, we presented ROSE, a novel reinforcement learning framework designed to enhance both the reasoning accuracy and efficiency of LLMs. Specifically, to encourage more diverse reasoning exploration, our method incorporates a semantic-entropy-based branching strategy alongside an ε-exploration mechanism. Simultaneously, to improve efficiency, we design a length-aware segment-level advantage estimator that promotes concise reasoning paths. Extensive experiments across various mathematical benchmarks validate that ROSE significantly outperforms state-of-the-art baselines in both effectiveness and efficiency.

Limitations

Our work mainly has two limitations. First, our experiments were conducted on models with up to 8B parameters, and we plan to investigate the scalability on larger architectures (e.g., 14B) in future work. Second, we primarily focused on mathematical reasoning tasks. In the future, we plan to extend our approach to other domains, such as code generation and question answering.

Ethical Considerations

This work aims to enhance the reasoning capabilities of LLMs. We acknowledge that advanced reasoning abilities could potentially be misused for malicious purposes, and we advocate for the deployment of these models alongside robust safety alignment protocols to mitigate such risks. Regarding the experimental setup, all datasets utilized in this work are open-source and publicly available. We have strictly adhered to their respective licenses and ensured that our usage is consistent with their intended purposes.

Acknowledgments

This research was (partially) supported by the Natural Science Foundation of China (62202271, 62472261, 62372275, 62272274, T2293773), the National Key R&D Program of China with grant No. 2024YFC3307300 and No. 2022YFC3303004, the Provincial Key R&D Program of Shandong Province with grant No. 2024CXGC010108, the Natural Science Foundation of Shandong Province with grant No. ZR2024QF203, the Technology Innovation Guidance Program of Shandong Province with grant No. YDZX2024088. All content represents the opinion of the authors, which is not necessarily shared or endorsed by their respective employers and/or sponsors.

References

Daman Arora and Andrea Zanette. 2025. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. 2013. The arcade learning environment: An evaluation platform for general agents. *Journal of artificial intelligence research*, 47:253–279.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, and 1 others. 2024. Do not think that much for 2+ 3=? on the overthinking of o1-like llms. *arXiv preprint arXiv:2412.21187*.

Xiangxiang Chu, Hailang Huang, Xiao Zhang, Fei Wei, and Yong Wang. 2025. Gpg: A simple and strong reinforcement learning baseline for model reasoning. *arXiv preprint arXiv:2504.02546*.

Muzhi Dai, Chenxu Yang, and Qingyi Si. 2025. S-grpo: Early exit via reinforcement learning in reasoning models. *arXiv preprint arXiv:2505.07686*.

Guanting Dong, Hangyu Mao, Kai Ma, Licheng Bao, Yifei Chen, Zhongyuan Wang, Zhongxia Chen, Jiazhen Du, Huiyang Wang, Fuzheng Zhang, and 1 others. 2025. Agentic reinforced policy optimization. *arXiv preprint arXiv:2507.19849*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*, pages 1861–1870. Pmlr.

Zhezheng Hao, Hong Wang, Haoyang Liu, Jian Luo, Jiarui Yu, Hande Dong, Qiang Lin, Can Wang, and Jiawei Chen. 2025. Rethinking entropy interventions in rlvr: An entropy change perspective. *arXiv preprint arXiv:2510.10150*.

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. *NeurIPS*.

Yuxiang Ji, Ziyu Ma, Yong Wang, Guanhua Chen, Xiangxiang Chu, and Liaoni Wu. 2025. Tree search for llm agent reinforcement learning. *arXiv preprint arXiv:2509.21240*.

Levente Kocsis and Csaba Szepesvári. 2006. Bandit based monte-carlo planning. In *European conference on machine learning*, pages 282–293. Springer.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Ren Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and 1 others. 2024. Rlaif vs. rlhf: Scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning*, pages 26874–26901. PMLR.

Yizhi Li, Qingshui Gu, Zhoufutu Wen, Ziniu Li, Tianshun Xing, Shuyue Guo, Tianyu Zheng, Xin Zhou, Xingwei Qu, Wangchunshu Zhou, and 1 others. 2025. Treepo: Bridging the gap of policy optimization and efficacy and inference efficiency with heuristic tree-based modeling. *arXiv preprint arXiv:2508.17445*.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, and 1 others. 2025. Mm-eureka: Exploring visual aha moment with rule-based large-scale reinforcement learning. *CoRR*.

Shuaiyi Nie, Siyu Ding, Wenyuan Zhang, Linhao Yu, Tianmeng Yang, Yao Chen, Tingwen Liu, Weichong Yin, Yu Sun, and Hua Wu. 2026. Attnpo: Attention-guided process supervision for efficient reasoning. *arXiv preprint arXiv:2602.09953*.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

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

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

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

Richard S Sutton, Andrew G Barto, and 1 others. 1998. *Reinforcement learning: An introduction*, volume 1. MIT press Cambridge.

Maciej Świechowski, Konrad Godlewski, Bartosz Sawicki, and Jacek Mańdziuk. 2023. Monte carlo tree search: A review of recent modifications and applications. *Artificial Intelligence Review*, 56(3):2497–2562.

Xu Wang, Sen Wang, Xingxing Liang, Dawei Zhao, Jincai Huang, Xin Xu, Bin Dai, and Qiguang Miao. 2022. Deep reinforcement learning: A survey. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):5064–5078.

Fei Wu, Zhenrong Zhang, Qikai Chang, Jianshu Zhang, Quan Liu, and Jun Du. 2026. Step potential advantage estimation: Harnessing intermediate confidence and correctness for efficient mathematical reasoning. *arXiv preprint arXiv:2601.03823*.

Mingqi Wu, Zhihao Zhang, Qiaole Dong, Zhiheng Xi, Jun Zhao, Senjie Jin, Xiaoran Fan, Yuhao Zhou, Huijie Lv, Ming Zhang, and 1 others. 2025a. Reasoning or memorization? unreliable results of reinforcement learning due to data contamination. *arXiv preprint arXiv:2507.10532*.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025b. Table-r1: Region-based reinforcement learning for table understanding. *arXiv preprint arXiv:2505.12415*.

Can Xie, Ruotong Pan, Xiangyu Wu, Yunfei Zhang, Jiayi Fu, Tingting Gao, and Guorui Zhou. 2025. Unlocking exploration in rlvr: Uncertainty-aware advantage shaping for deeper reasoning. *arXiv preprint arXiv:2510.10649*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Zhicheng Yang, Zhijiang Guo, Yinya Huang, Xiaodan Liang, Yiwei Wang, and Jing Tang. 2025b. Treerpo: Tree relative policy optimization. *arXiv preprint arXiv:2506.05183*.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, Jiaze Chen, Chengyi Wang, TianTian Fan, Zhengyin Du, and 1 others. 2025. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, and 1 others. 2025a. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*.

Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhoufutu Wen, Chenghua Lin, Wenhao Huang, and 1 others. 2025b. First return, entropy-eliciting explore. *arXiv preprint arXiv:2507.07017*.

Yixiao Zhou, Yang Li, Dongzhou Cheng, Hehe Fan, and Yu Cheng. 2026. Look inward to explore outward: Learning temperature policy from llm internal states via hierarchical rl. *arXiv preprint arXiv:2602.13035*.

Xinyu Zhu, Mengzhou Xia, Zhepei Wei, Wei-Lin Chen, Danqi Chen, and Yu Meng. 2025. The surprising effectiveness of negative reinforcement in llm reasoning. *arXiv preprint arXiv:2506.01347*.

A Case Study

To more comprehensively investigate the differences between entropy-based branching and semantic-entropy–based branching, we present a case study, with the results of the two methods shown in Figure 6 and Figure 7, respectively. Specifically, given a question, we use `Llama-3.2-3B-Instruct` as the backbone model to generate a complete response. We then apply two different methods to determine the branching positions based on this response and regenerate accordingly, ultimately forming a group of responses.

From Figure 6, we observe that all responses generated by the entropy-based branching method produce incorrect answers. Notably, the shared prefix among these responses already contains an erroneous calculation (highlighted in blue). However, the branching position identified by entropy-based branching occurs after this point, which consequently propagates the incorrect reasoning into the subsequent generations.

In contrast, our proposed semantic-entropy–based branching method performs branching before the erroneous calculation (highlighted in blue), enabling more fine-grained reasoning that avoids this error. Although subsequent reasoning errors may still occur (e.g., response 2 in Figure 7), branching is again triggered prior to the error, ultimately yielding a correct response (e.g., response 3 in Figure 7). This case study demonstrates that our method can more accurately identify regions of higher uncertainty in the model's reasoning trajectory, thereby encouraging more diverse and effective reasoning paths.

```
Find the sum of the first six terms in the geometric sequence $\frac12,\frac14,\frac18,\dots$. Express your
answer as a common fraction.
```

Question

```
Step 1: The problem requires us to find the sum of the first six terms in the given geometric sequence.
Step 2: To solve this, we need to identify the first term (a) and the common ratio (r) in the sequence.
Step 3: In the given sequence, the first term is $\frac12$ and the common ratio between consecutive terms is
$\frac14 \div \frac12 = \frac14$.
Step 4: Now that we know the values of a and r, we can use the formula for the sum of the first n terms in a
geometric sequence: $S_n = \frac{a(1-r^n)}{1-r}$, where $n$ is the number of terms.
Step 5: Plugging in the values of a, r, and n into the formula, we get $S_6 = \frac{\frac12\left(1-
\left(\frac14\right)^6\right)}{1-\frac14}$.
Step 6: Simplifying the equation, we get $S_6 = \frac{\frac12\left(1-\frac{1}{4096}\right)}{\frac34}$.
Step 7: Further simplifying, we get $S_6 = \frac{\frac12 \cdot \frac{4095}{4096}}{\frac34}$.
Step 8: Continuing to simplify, we get $S_6 = \frac{\frac{4095}{8192}}{\frac34}$.
Step 9:
```

Prefix

```
Inverting and multiplying, we get $S_6 = \frac{4095}{8192} \cdot \frac34$.
Step 10: Calculating the final value, we get $S_6 = \frac{4095}{12288}$.
Step 11: Simplifying the fraction, we get $S_6 = \frac{3645}{10224}$.
Step 12: Finally, simplifying the fraction further, we get $S_6 = \frac{1365}{4096}$.

The final answer is: $\boxed{\frac{1365}{4096}}$ ✗
```

Response1

```
Multiplying the numerator by the reciprocal of the denominator, we get $S_6 = \frac{4095}{8192}
\cdot \frac34 = \frac{4095}{12288}$.
Step 10: Finally, we can simplify this fraction by dividing both the numerator and denominator by
their greatest common divisor (GCD), which is 3.
Step 11: Simplifying the fraction gives us $S_6 = \frac{4095 \div 3}{12288 \div 3} =
\frac{1365}{4096}$.

Therefore, the sum of the first six terms in the given geometric sequence is
$\boxed{\frac{1365}{4096}}$.

The final answer is $\boxed{\frac{1365}{4096}}$ ✗
```

Response2

```
Finally, simplifying the fraction, we get $S_6 = \boxed{\frac{2045}{2562}}$.

The final answer is: $\boxed{\frac{2045}{2562}}$ ✗
```

Response3

Figure 6: Case study. An example where entropy is used as the branching metric in the rollout phase.

```
Find the sum of the first six terms in the geometric sequence $\frac12,\frac14,\frac18,\dots$. Express your
answer as a common fraction.
```

Question

```
Step 1: The problem requires us to find the sum of the first six terms in the given geometric sequence.
Step 2: To solve this, we need to identify the first term (a) and the common ratio (r) in the sequence.
Step 3: In the given sequence, the first term is $\frac12$
```

Prefix

```
and the common ratio between consecutive terms is $\frac14 \div \frac12 = \frac14$.
Step 4: Now that we know the values of a and r, we can use the formula for the sum of the first n
terms in a geometric sequence: $S_n = \frac{a(1-r^n)}{1-r}$, where $n$ is the number of terms.
Step 5: Plugging in the values of a, r, and n into the formula, we get $S_6 = \frac{\frac12\left(1-
\left(\frac14\right)^6\right)}{1-\frac14}$.
Step 6: Simplifying the equation, we get $S_6 = \frac{\frac12\left(1-
\frac{1}{4096}\right)}{\frac34}$.
Step 7: Further simplifying, we get $S_6 = \frac{\frac12 \cdot \frac{4095}{4096}}{\frac34}$.
Step 8: Continuing to simplify, we get $S_6 = \frac{\frac{4095}{8192}}{\frac34}$.
Step 9: Inverting and multiplying, we get $S_6 = \frac{4095}{8192} \cdot \frac34$.
Step 10: Calculating the final value, we get $S_6 = \frac{4095}{12288}$.
Step 11: Simplifying the fraction, we get $S_6 = \frac{3645}{10224}$.
Step 12: Finally, simplifying the fraction further, we get $S_6 = \frac{1365}{4096}$.

The final answer is: $\boxed{\frac{1365}{4096}}$ ✗
```

Response1

```
.
Step 4: The common ratio (r) can be found by dividing any term by the preceding term.
Step 5: By looking at the sequence, we can see that each term is obtained by multiplying the
previous term by $\frac{1}{2}$. So, $r = \frac{1}{2}$.
Step 6: The formula for the sum of the first n terms of a geometric sequence is given by:
$S_n = \frac{a(1 - r^n)}{1 - r}$, where a is the first term, r is the common ratio, and n is the
number of terms.
Step 7: Plugging in the values we found in steps 3-5, we get $S_6 = \frac{\frac{1}{2}(1 -
(\frac{1}{2})^6)}{1 - \frac{1}{2}}$
Step 8: We can simplify this expression by evaluating the numerator and denominator separately.
```

Prefix

```
We get $S_6 = \frac{\frac{1}{2}(1 - \frac{1}{64})}{\frac{1}{2}}$
Step 9: Simplifying further, we get $S_6 = \frac{\frac{1}{2} \cdot
\frac{63}{64}}{\frac{1}{2}}$ = $\frac{\frac{63}{64}}{\frac{1}{2}}$
Step 10: Multiplying $\frac{63}{64}$ by the reciprocal of $\frac{1}{2}$ gives us
$\frac{63}{64} \cdot \frac{2}{1}$ = $\frac{63}{32}$

Final Answer: $\boxed{\frac{63}{32}}$ ✗
```

Response2

```
Step 9: Evaluating the numerator: $1 - (\frac{1}{2})^6 = 1 - \frac{1}{64} =
\frac{63}{64}$
Step 10: Evaluating the denominator: $1 - \frac{1}{2} = \frac{1}{2}$
Step 11: Now we can rewrite the sum formula as $S_6 = \frac{\frac{1}{2} *
\frac{63}{64}}{\frac{1}{2}}$.
Step 12: Simplifying the expression gives $S_6 = \frac{\frac{63}{128}}{\frac{1}{2}}$
Step 13: Dividing by a fraction is the same as multiplying by its reciprocal, so $S_6
= \frac{63}{128} * 2 = \frac{63}{64}$
Step 14: Therefore, the sum of the first six terms in the given geometric sequence is
$\frac{63}{64}$.

\boxed{\frac{63}{64}} ✓
```

Response3

Figure 7: Case study. An example where semantic entropy is used as the branching metric in the rollout phase.