# FIXING THE BROKEN COMPASS: DIAGNOSING AND IMPROVING INFERENCE-TIME REWARD MODELING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Inference-time scaling techniques have shown promise in enhancing the reasoning capabilities of large language models (LLMs). While recent research has primarily focused on training-time optimization, our work highlights inference-time reward model (RM)-based reasoning as a critical yet overlooked avenue. In this paper, we conduct a systematic analysis of RM behavior across downstream reasoning tasks, revealing three key limitations: (1) RM can impair performance on simple questions, (2) its discriminative ability declines with increased sampling, and (3) high search diversity undermines RM performance. To address these issues, we propose **CRISP** (Clustered Reward Integration with Stepwise Prefixing), a novel inference-time algorithm that clusters generated reasoning paths by final answers, aggregates reward signals at the cluster level, and adaptively updates prefix prompts to guide generation. Experimental results demonstrate that CRISP significantly enhances LLM reasoning performance, achieving up to **5%** accuracy improvement over other RM-based inference methods and an average of **10%** gain over advanced reasoning models.

## 1 INTRODUCTION

The remarkable achievements of OpenAI's o1 have sparked a wave of research into inference-time scaling techniques in reasoning tasks (OpenAI, 2024; DeepSeek-AI et al., 2025; Zeng et al., 2024). Some works aim to enhance models during the training phase, employing reinforcement learning (RL) (Xie et al., 2025; Qu et al., 2025) or supervised fine-tuning (SFT) (Ye et al., 2025; Muennighoff et al., 2025) on high-quality data to equip models with the ability to generate long chains of thought (CoT). Other approaches focus on inference-time optimization, using reward model (RM)-based search strategies such as Monte Carlo Tree Search (MCTS) to guide the model toward more efficient solution paths (Wang et al., 2024b; Setlur et al., 2024; Zhang et al., 2024).

Driven by the great success of the DeepSeek-R1 series (DeepSeek-AI et al., 2025), recent efforts have predominantly focused on reproducing its performance from a training-centric perspective (Muennighoff et al., 2025; Ye et al., 2025; Xie et al., 2025), while largely overlooking inference optimization methods. Although R1-style works achieve strong performance on tasks such as math reasoning, they have been shown to suffer from serious issues such as overthinking (Chen et al., 2024; Sui et al., 2025) and limited task generalization (Zhang et al., 2025a; Zheng et al., 2025). These issues, however, can be mitigated through RM-based inference techniques. For example, on the commonsense reasoning task CSQA (Talmor et al., 2019), DeepSeek-R1-7B (DeepSeek-AI et al., 2025) achieves 64.8 accuracy with an average of 3,613 tokens. In contrast, our RM-based inference method, applied to its base model Qwen2.5-Math-7B (Yang et al., 2024b), reaches a higher accuracy of **72.0** using only **1,100 tokens** on average. Therefore, optimizing inference-time reasoning remains a critical direction, particularly for smaller models.

How can we further improve the reasoning performance of LLMs at inference time? Revisiting R1-style work, one key insight is their identification of the reward hacking issue during RL training, which they address using rule-based reward functions, ultimately improving performances (Liu et al., 2024b; DeepSeek-AI et al., 2025; Gao et al., 2023). This raises a natural question: **Can we similarly analyze the issues of the reward model at inference time and mitigate them to enhance the LLM's reasoning ability?**

In this work, we investigate the factors affecting reward model performance at inference time and propose methods to mitigate its limitations. Specifically, we begin by mathematically modeling the RM-based inference process to identify its key influencing factors: the input questions, the number of sampled responses, and the search parameters. Then, we conduct targeted experiments to analyze the impact of each factor on RM performance: **(1) Input question:** We test the performance of BoN and MCTS across different question difficulty levels and demonstrate that RM-based inference significantly impairs performance on simple questions. **(2) Sampling number:** We analyze the RM's discriminative ability under different numbers $n$ and observe that its performance deteriorates as $n$ increases. The statistical analysis attributes this degradation to an inverse long-tail phenomenon, wherein the RM tends to assign higher scores to low-frequency, incorrect responses. **(3) Search parameters:** We focus on parameters controlling search diversity, such as sampling temperature and MCTS tree structure. Our results show that RM performs best under moderate diversity, while excessive diversity undermines reasoning accuracy.

To mitigate the former issues in RM-based inference, we design a novel algorithm called **CRISP** (**C**lustered **R**eward **I**ntegration with **S**tepwise **P**refixing). CRISP operates in an iterative fashion, where each round begins by sampling reasoning paths conditioned on a dynamic prefix set. These paths are then clustered by their final answers, allowing the algorithm to aggregate reward signals at the cluster level and thereby attenuate the RM's tendency to mis-rank rare but incorrect outputs. We further incorporate an early termination mechanism based on cluster cardinality, which enables efficient inference on simple questions and alleviates RM instability in such cases. Finally, high-scoring paths from dominant clusters inform the construction of stepwise prefixes for the next sampling round, enabling tighter control over search diversity by limiting the number of intermediate states explored. We conduct extensive experiments to compare our method with other baselines. The results not only indicate that our method is effective in improving RM-based reasoning abilities, with accuracy gains of up to **5%**, but also validate the soundness of our earlier findings. Moreover, compared to DeepSeek-R1 models of the same scale, our method reduces average token usage by up to **90%**, while achieving an average accuracy improvement of **10%** on non-mathematical tasks.

Our main contributions are as follows: (1) We draw three critical findings based on a systematic analysis of RM behavior during inference: RM degrades performance on simple questions, fails to effectively distinguish low-frequency incorrect samples, and performs suboptimally under excessive search diversity. (2) We propose CRISP, a novel inference-time algorithm that clusters generated reasoning paths by final answers, aggregates reward signals at the cluster level, and adaptively updates prefix prompts to guide generation, effectively mitigating the shortcomings of reward models at inference time. (3) Extensive experiments demonstrate that CRISP consistently outperforms both inference-time and training-time baselines, with accuracy improvements of up to **5%** compared to other RM-based inference methods, and an average of **10%** over R1 models in non-mathematical reasoning tasks.

## 2 Overall Performance of Reward Models in Inference-Time

In this section, we evaluate the inference-time performance of the current reward model as a preliminary experiment. Specifically, we compare the accuracy of Best-of-N (BoN), which generates multiple responses and selects the best one based on the reward score.

**Experimental Setup** For the policy model, we select representative open-source models: Gemma2-9B (Rivière et al., 2024a), Llama3.1-8B (Rivière et al., 2024b), Qwen2.5-3B and Qwen2.5-14B (Yang et al., 2024a). For the reward models, we select two outcome reward models (ORMs): ArmoRM (Wang et al., 2024a) and Skywork-Llama-3.1-8B (Liu et al., 2024a), and two process reward models (PRMs): Shepherd-Mistral-7B-PRM (Wang et al., 2024b) and Skywork-o1-PRM-Qwen-2.5-7B (o1 Team, 2024). These models demonstrate commendable performance on related benchmarks (see Appendix C for details). As for the evaluation data, following previous works (Snell et al., 2024; Brown et al., 2024; Qi et al., 2024), we select MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), which consists of high-school competition-level math problems. In addition to BoN, we also set two baselines: **SC** and **Oracle**. For the former, we select the major voting answer from $n$ responses. For the latter, we directly recall the existing correct answer from the generated samples, which serves as the performance ceiling.

**Main Results** Figure 1 shows the main results of the evaluation (see Appendix D for more results). We can conclude that: **Advanced reward models have limited performance on the downstream**

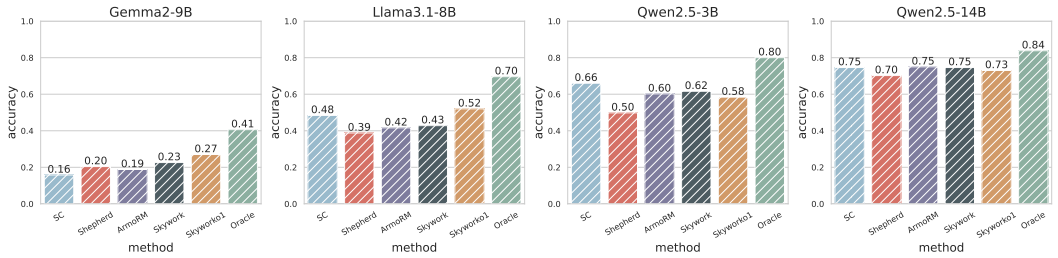



Figure 1: The performance of different policy models using various reward models for BoN inference on the MATH dataset ($n$ = 10).

**math reasoning task.** For most LLMs, BoN only provides minor improvements over SC ($< 5\%$). Specifically, on Qwen2.5-3B, the BoN for all reward models exhibits lower accuracy than SC, indicating that the BoN inference method has limited reasoning performance. Besides, Oracle significantly outpaces other baselines, suggesting that the performance bottleneck lies in the RM's discriminative ability rather than the LLM's generative capability. Therefore, **identifying and mitigating the factors that impair the RM's performance during inference are crucial for enhancing LLM's reasoning ability**.

## 3 Probing RM-based Inference Issues

### 3.1 Mathematical Modeling

During the inference phase, the first step is to input the question $q$ and generate multiple responses $\mathcal{R}$:

$$\mathcal{R} = \mathcal{S}(\mathcal{M}(q), n; \Phi) \tag{1}$$

where $\mathcal{M}(q)$ denotes the output distribution of the policy model after inputting the question, $n$ denotes the number of samples and $\Phi$ denotes the parameters of the search strategy $\mathcal{S}$ (such as sampling temperature). After that, we use a scoring function $f$ to select the best response $\hat{r}$ from $\mathcal{R}$:

$$\hat{r} = \arg\max_{r \in \mathcal{R}} f(r) \tag{2}$$

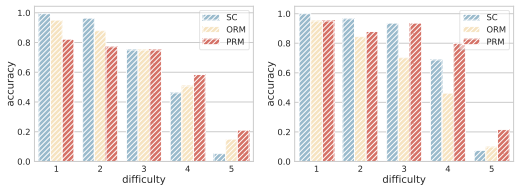
To analyze the performance of the reward model, we define $f$ as the score predicted by the RM. Our work focuses on identifying key factors that influence RM performance. To this end, we vary the components in Eq.1 to observe the accuracy of predicted $\hat{r}$ under different $\mathcal{R}$. Specifically, we study three main factors through probing experiments: the input question $q$, the sampling number $n$, and the search parameters $\Phi$.

### 3.2 Experimental Setup

For reward models, based on results in Figure 1, we select the best-performing Skywork and Skywork-o1 as the ORM and PRM for our subsequent experiments. Regarding policy models, we use Qwen2.5-3B and Llama3.1-8B throughout our experiments. To ensure that our findings are not specific to a particular strategy, we conduct all experiments using both BoN and MCTS. As for evaluation data, we employ the MATH-500 dataset in our main text, and provide additional results on GSM8K (Cobbe et al., 2021) and OlympiadBench (He et al., 2024) in the appendix.

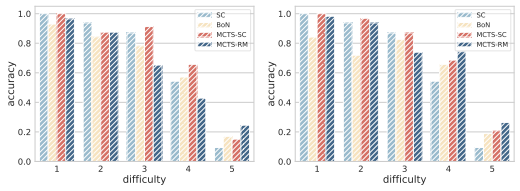
### 3.3 Input Question: Reward Model Underperforms on Easy Questions

**Question Difficulty Modeling** We first investigate how different questions affect the RM's performance. Following former works, we use question difficulty as a metric to classify different questions (Lightman et al., 2024; Snell et al., 2024). We bin the policy model's pass@1 rate (estimated from 10 samples) on each question into five quantiles, each corresponding to increasing difficulty levels. For example, If the model answers correctly 0 or 1 time, the question is level 5 (hardest). If it answers correctly more than 8 times, the question is level 1 (easiest). Besides, we also study the difficulty approximation without the ground truth and report results in Appendix E.




(a) Qwen2.5-3B (b) Llama3.1-8B

Figure 2: Performance of BoN inference across different question difficulty levels.




(a) ORM (b) PRM

Figure 3: Performance of MCTS inference across different question difficulty levels.

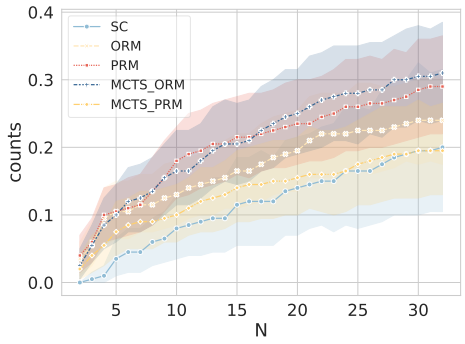



Figure 4: The number of times the model's selection changes from correct to incorrect.

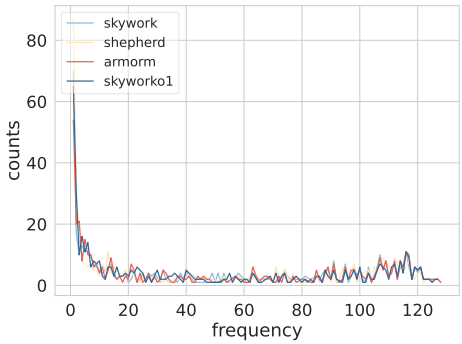



Figure 5: Frequency statistics of the highest-scored negative responses in BoN.

**BoN Performance** After categorizing the data by difficulty, we analyze the BoN performance across different levels. We sample 32 examples from each question and illustrate the accuracy in Figure 2, from which we can conclude that: **Compared to SC, BoN performs worse on simple but better on difficult questions.** From the easiest level 1 to the hardest level 5, the accuracy of SC gradually declines, while BoN transitions from lagging behind SC to surpassing it. We also repeat the experiment on two more math reasoning benchmarks and present the results in Appendix F, further confirming our conclusion.

**MCTS Performance** In MCTS, we use two different scoring functions $f$ to select the final response for comparison: MCTS-SC and MCTS-RM (more functions in Appendix D). For the former, we employ a majority voting method for selection. For the latter, we choose the path with the highest reward score. We perform 32 rollouts over 200 questions, demonstrating the results in Figure 3. Although MCTS provides improvement over BoN, the accuracy of MCTS-RM still lags behind that of SC for low-difficulty problems (see levels 1 and 2 in Figure 3). Besides, MCTS-SC achieves higher accuracy on easy questions but performs worse on harder questions compared to MCTS-RM. These indicate that: **(Cl.1) The introduction of the RM can hinder the LLM's reasoning performance on simple problems.** This pattern is not limited to specific inference strategies.

### 3.4 Sampling Number: RM struggles to distinguish low-frequency negatives

**Performance Gap between Accuracy and Coverage** Recent studies (Brown et al., 2024) show that the coverage of correct answers by LLMs increases with the number of samples, while accuracy plateaus after a small $n$ (see Appendix G for experimental details). Given that recall steadily improves, we suggest that the accuracy bottleneck is likely a result of the RM making more misclassifications as $n$ increases. To investigate this, we first conduct a case study in which we randomly select questions and examine the RM's selection accuracy at different $n$ (see Appendix H for details). The results indicate that, in some cases, the RM assigns the highest score to incorrect responses generated at higher $n$, replacing originally correct answers with incorrect ones. Based on the observation, we further record the number of instances in which the selected answer transitions from correct to incorrect and present the results in Figure 4. All methods exhibit a tendency for more incorrect

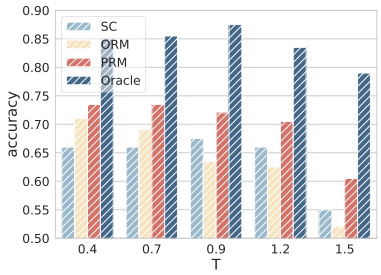
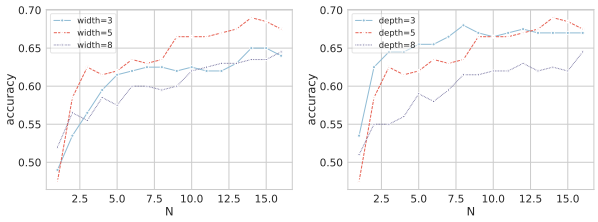



Figure 6: BoN performance across different temperatures (Qwen2.5-3B).




(a) Tree width (b) Tree depth

Figure 7: MCTS performance under different tree structures (ORM).

transitions as $n$ increases. **This indicates that the model increasingly makes erroneous distinctions as the sampling size grows**. Moreover, compared to SC, RM-based inference methods show higher transition counts in Figure 4, which suggests that incorporating reward models introduces more incorrect selections.

**Inverse Long-tail Phenomenon** Why does the reward model perform worse as the sampling number grows? Reflecting on its training process (Wang et al., 2024a; Liu et al., 2024a; Wang et al., 2024b), the training data primarily consists of paired responses (i.e., a correct one and an incorrect one). These pairs represent a constrained subset of the response space. We hypothesize that as $n$ grows, more low-frequency responses (those outside the training distribution) are sampled. The reward model struggles to generalize to these unfamiliar inputs, leading to incorrect responses occasionally receiving higher scores. To validate this hypothesis, we perform a statistical analysis of negative responses. For each question, we select the incorrect response with the highest RM score and compute the frequency of its answer across all samples. As shown in Figures 5 and 23, the RM displays an **inverse long-tail phenomenon** when scoring incorrect responses. For most questions, the top-scoring incorrect answers tend to have very low frequencies (frequency $< 5$ in Figure 5). Conversely, incorrect answers with high occurrence frequencies rarely achieved the highest scores. These findings support our hypothesis: **(Cl.2) RMs struggle to correctly score incorrect responses with low occurrence frequencies, making it difficult to distinguish incorrect responses from correct ones as $n$ grows.**

### 3.5 Search Parameters: RM performs worse on high-diversity distributions

**Search Diversity in BoN** The final influencing factor we investigate is the search parameters $\Phi$, which are primarily utilized to control the diversity of the policy model's search. For the BoN method, the temperature $T$ is the key parameter controlling the search diversity. We sweep $T$ and analyze its influence on the performance, as shown in Figure 6 and 24. For both policy models, BoN performance consistently degrades with increasing $T$, while SC and Oracle (i.e., coverage) remain stable except at high temperatures ($T > 0.9$ in Figure 6). These results indicate that RM is more sensitive to sampling diversity than the policy model. **Higher diversity makes it challenging for the RM to distinguish between positive and negative responses.** To better understand this issue, we perform additional statistical analyses in Appendix I, which suggest that higher sampling temperatures cause the policy model to produce more low-frequency incorrect responses, thereby degrading discriminative accuracy.

**Search Diversity in MCTS** In the MCTS algorithm, search diversity is primarily governed by the tree structure, determined by two key parameters: width and depth. The width refers to the number of child nodes at each node, whereas the depth denotes the length of the longest path from the root to a leaf node. A larger width indicates a broader search space during exploration, while a greater depth implies the model can traverse more intermediate states along a single trajectory. We evaluate MCTS performance under varying settings and present the results in Figure 7 and 27 . The findings reveal: (1) For width, the best performance is observed at intermediate values (width = 5), too high widths lead to a decline in performance. (2) For depth, the best performance is achieved under settings with a lower value (e.g., depth = 3 or 5). These suggest that in MCTS, exploring too many intermediate states can harm performance. Notably, the optimal number of intermediate steps in search does not

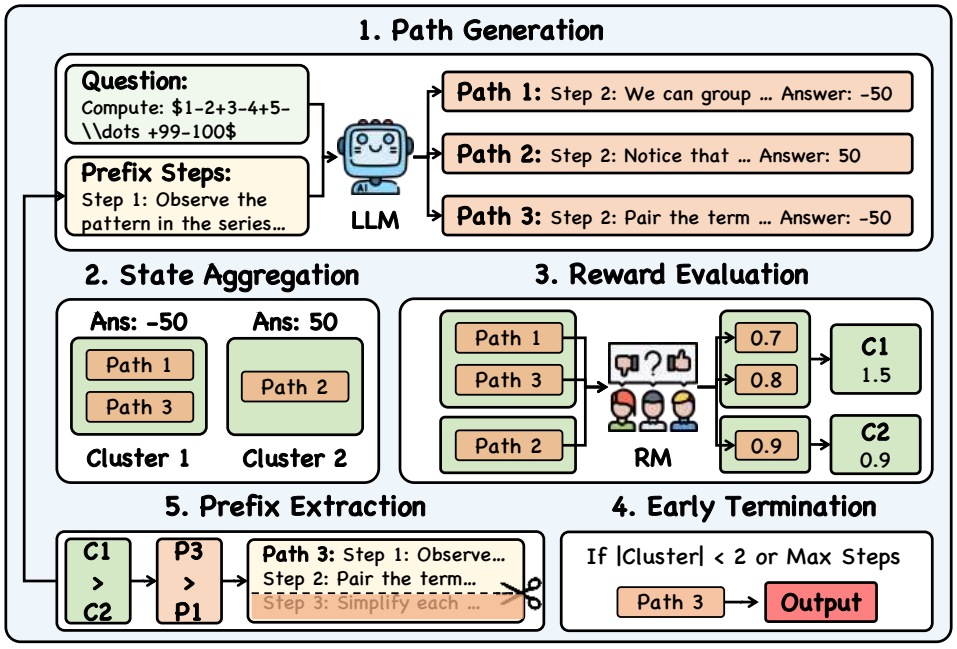



Figure 8: Main process of our CRISP method.

necessarily align with the number of steps a human would take to solve the same problem. We also analyze the impact of exploration weight on the diversity of MCTS, with consistent findings (see Appendix J). In summary, excessive diversity, such as width, depth, or temperature, can impair the performance of the reward model. Thus, we conclude: **(Cl.3) During inference, it is essential to constrain the diversity of the sampling distribution to maintain the optimal performance of the RM.**

# 4 Mitigating RM-based Inference Issues

## 4.1 Our Methodology

In the preceding sections, we uncover key patterns that affect the RM's performance and identify serval issues in RM-based reasoning. To mitigate these issues, we propose a novel RM-based inference algorithm called **Clustered Reward Integration with Stepwise Prefixing (CRISP)**. Figure 8 and Algorithm 1 demonstrate the main process of our method, which comprises five modules:

**Path Generation** Given a question $q$, during each iteration, we generate new reasoning paths based on the existing prefix set $\mathcal{P}$:

$$\mathcal{R} = \mathcal{R} \cup \mathcal{M}(q, n, \mathcal{P}) \tag{3}$$

In the generation process, the policy model generates $n$ complete sequences of remaining reasoning steps conditioned on $\mathcal{P}$ ($\mathcal{P} = \emptyset$ in the init iteration), rather than generating intermediate nodes step by step as in approaches like MCTS. This helps control the diversity of the search space and reduces the negative impact of excessive diversity on the reward model, as discussed in Cl.3.

**State Aggregation** To further reduce the complexity of the state space and mitigate the impact of low-frequency negative examples on the reward model's performance (as discussed in Cl.2), we define a final-answer-based state aggregation function $\psi$:

$$\psi : \mathcal{R} \to \mathcal{C} \tag{4}$$

where $\mathcal{C}$ is the set of final answer clusters (i.e., all responses leading to the same answer), and for any path $r_1, r_2 \in \mathcal{R}$, we have:

$$\psi(r_1) = \psi(r_2) \iff Answer(r_1) = Answer(r_2) \tag{5}$$

Table 1: Accuracy comparison in main experiments, the best results are highlighted in **bold**.

| Methods | | Qwen2.5-3B | | | Llama3.1-8B | | |
|---|---|---|---|---|---|---|---|
| | | *GSM8K* | *MATH* | *Olympiad* | *GSM8K* | *MATH* | *Olympiad* |
| CoT | | 0.78 | 0.46 | 0.24 | 0.85 | 0.38 | 0.11 |
| Self-Consistency | | 0.83 | 0.64 | 0.31 | 0.91 | 0.57 | 0.16 |
| Best-of-N | + ORM | 0.83 | 0.65 | 0.31 | 0.91 | 0.47 | 0.18 |
| | + PRM | 0.87 | 0.61 | 0.34 | 0.95 | 0.62 | 0.23 |
| BoN Weighted | + ORM | 0.83 | 0.67 | 0.31 | 0.89 | 0.53 | 0.20 |
| | + PRM | 0.86 | 0.60 | 0.36 | 0.94 | 0.62 | 0.24 |
| MCTS | + ORM | 0.92 | 0.67 | 0.34 | 0.90 | 0.43 | 0.13 |
| | + PRM | 0.95 | 0.71 | 0.31 | 0.95 | 0.57 | 0.19 |
| Beam Search | | 0.95 | 0.73 | 0.34 | 0.94 | 0.56 | 0.15 |
| **Ours** | + ORM | 0.91 | 0.70 | 0.36 | 0.89 | 0.49 | 0.18 |
| | + PRM | **0.96** | **0.76** | **0.39** | **0.95** | **0.67** | **0.26** |

All paths that produce the same final answer are mapped to the same cluster $\mathcal{C}_j \in \mathcal{C}$. As an example, in Module 2 of Figure 8, paths 1 and 3, both with the answer of -50, are assigned to the same cluster.

**Reward Evaluation** After clustering the responses, we can convert the reward scores $f$ for each path into scores $\mathcal{F}$ for the corresponding clusters $\mathcal{C}_j$ (i.e., lines 17-20 in Algorithm 1):

$$\mathcal{F}(\mathcal{C}_j) = \sum_{x \in \mathcal{C}_j} f(x) \tag{6}$$

In the implementation, we normalize $f(x)$ before summing. By additionally considering the frequency of the answers associated with each path during scoring, we can prevent the reward model from assigning excessively high scores to low-frequency responses, thereby mitigating the issue identified in Cl.2. We will later demonstrate the effectiveness of this clustering strategy through both ablation experiments (see §4.5) and theoretical analysis (see Appendix K).

**Early Termination** This module controls when to exit the loop and return the final response. In addition to the standard exit condition of reaching the maximum number of iterations, we also control early termination by monitoring the number of clusters. If the number falls below a certain threshold (set to 2 in our work), it indicates that the question is relatively simple (as evidenced and discussed in Appendix E). In this case, the algorithm terminates, returning the answer corresponding to the most populated cluster, which is equivalent to SC. This not only reduces inference costs but also mitigates the issue of the reward model underperforming on simple questions (see Cl.1).

**Prefix Extraction** In this module, we extract the top multiple prefixes as the new prefix set $\mathcal{P}$ for the next iteration, based on the scores of the paths and clusters. As illustrated in Module 5 of Figure 8, we first select the top-$k$ clusters with the highest scores (here, $k$=1, so we select Cluster 1). Then, from the selected cluster(s), we choose the path with the highest score (in this case, 0.8 ¿ 0.7, so we select Path 3) to extract the prefix. Specifically, at the $i$-th generation, we extract the first $i$ steps of all paths as $\mathcal{P}$, and repeat the process until termination.

### 4.2 Main Experiments

**Experimental Setup** We compare the reasoning performance of our method with other advanced baselines, including: **CoT** (Wei et al., 2022), **Self-Consistency** (Wang et al., 2023), **Best-of-N**, **BoN Weighted** (Snell et al., 2024), **MCTS** (Hao et al., 2023) and **Beam Search** (Snell et al., 2024). For datasets, in addition to MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), we also validate our methods on GSM8K (Cobbe et al., 2021) and OlympiadBench (He et al., 2024). For models, we continue to select Qwen2.5-3B and Llama3.1-8B as the policy model, while using Skywork-Llama-3.1-8B (ORM) and Skywork-o1-PRM-Qwen-2.5-7B (PRM) as the reward model. We present more details in Appendix L.

Table 2: Comparison between R1 models and our method, the best accuracy are highlighted in **bold**.

| Base Models | Methods | *Math* | | *Commonsense* | | *Social* | | *Logical* | |
|---|---|---|---|---|---|---|---|---|---|
| | | Acc | Length | Acc | Length | Acc | Length | Acc | Length |
| Qwen2.5-Math-1.5B | Chat | 0.52 | 1470 | 0.40 | 1400 | 0.46 | 1204 | 0.40 | 2790 |
| | R1-Distill | **0.79** | 13421 | 0.47 | 6066 | 0.52 | 6407 | 0.35 | 12352 |
| | Ours | 0.59 | 943 | **0.58** | 1004 | **0.61** | 1144 | **0.44** | 1143 |
| Qwen2.5-Math-7B | Chat | 0.74 | 1855 | 0.58 | 1479 | 0.58 | 1388 | 0.49 | 2133 |
| | R1-Distill | **0.88** | 9626 | 0.65 | 3612 | 0.66 | 2920 | 0.50 | 6492 |
| | Ours | 0.79 | 987 | **0.72** | 1100 | **0.66** | 1059 | **0.59** | 2058 |

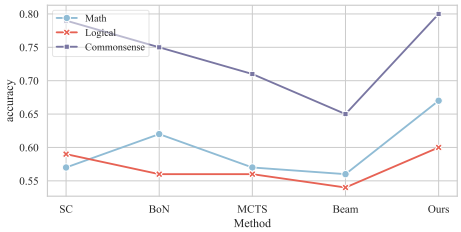



Figure 9: Performance comparison on other reasoning tasks (Llama3.1-8B + Skyworko1).

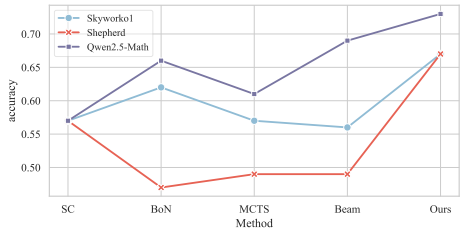



Figure 10: Performance comparison on other reward models (Llama3.1-8B on MATH).

**Main Results** We demonstrate the result in Table 1, from which we can get the following conclusions: **(1) Our proposed CRISP method significantly improves RM's performance in reasoning tasks.** Across all benchmarks and both model backbones, CRISP consistently outperforms existing RM-based inference approaches. Notably, on the Llama3.1-8B model, CRISP achieves a performance gain of up to **5.0%** on the MATH dataset over the best-competing method. **(2) The findings from the preceding analysis are reasonable.** CRISP is specifically crafted to overcome the key issues of reward modeling revealed in §3. Its consistent and significant performance improvements provide strong empirical evidence that CRISP effectively mitigates these limitations, which are critical bottlenecks affecting the model's reasoning performance.

### 4.3 Training-Time vs. Inference-Time Optimization

To demonstrate the continued necessity of our inference-time optimization approach amid the rising dominance of RL and SFT techniques represented by the DeepSeek-R1 series, we compare our method against the R1 model across different reasoning tasks, including math reasoning (MATH-500), commonsense reasoning (CSQA (Talmor et al., 2019)), social reasoning (SIQA (Sap et al., 2019)) and logical reasoning (LogiQA (Liu et al., 2020)). Specifically, given the same base model, we evaluate the accuracy and token consumption among its chat version (using CoT), the R1 distilled version, and our proposed method. From the results in Table 2, we can observe that: **(1) Our method enables more efficient reasoning across all tasks.** It achieves comparable reasoning tokens to the CoT method, while reducing output length by over **90%** compared to the R1 model in the best case. **(2) Our method exhibits stronger generalization capabilities.** Although it underperforms the R1 model on math tasks, it consistently outperforms R1 on other reasoning benchmarks, with average gains of **10%** and **5%** accuracy across two backbones. This highlights the advantage of our inference-time optimization in generalizing across diverse scenarios.

### 4.4 Generalization Capability Evaluation

**Results on More Tasks.** To ensure our method applies to tasks beyond mathematical reasoning, we introduce two additional tasks: logical reasoning (LogiQA (Liu et al., 2020)) and commonsense reasoning (CSQA (Talmor et al., 2019)), and compare the accuracy with other baselines on them. As shown in Figure 9, when using Llama3.1-8B as the policy model and Skyworko1 as the reward model, our method consistently outperforms all baselines across tasks, highlighting its versatility.

**Results on More Reward Models.** To demonstrate the robustness of our method across different RMs, we further evaluate it using two additional RMs: Shepherd-Mistral-7B-PRM (Wang et al.,

Table 3: Time cost comparison (s).

| Dataset | BoN | MCTS | Beam | **Ours** |
|---|---|---|---|---|
| GSM8K | 33.6 | 89.7 | 99.7 | 53.3 |
| MATH | 58.6 | 211.3 | 268.7 | 91.0 |

Table 4: Token consumption comparison.

| Dataset | BoN | MCTS | Beam | **Ours** |
|---|---|---|---|---|
| GSM8K | 6,340 | 9,282 | 8,828 | 3,499 |
| MATH | 11,550 | 18,014 | 27,012 | 11,535 |

2024b) and Qwen2.5-Math-PRM-7B (Zhang et al., 2025b). We replicate the main experiment on the MATH dataset (200 samples) and report the result in Figure 10. The results show that our method still significantly outperforms other baselines when using other reward models. Even with a relatively weak reward model like Shepherd (achieving only 0.47 BoN performance), our method is able to maintain a high level of accuracy.

### 4.5 Other Discussions

**Cost Analysis** As an inference-time method, in addition to accuracy, reasoning cost is also an important factor to consider. We evaluate computational cost (token consumption and inference time) under consistent rollout numbers and device settings, with results demonstrated in Table 3 and Table 4. Our approach outperforms advanced RM-integrated methods such as MCTS and Beam Search in both time and token consumption across two datasets. Despite having a slightly higher inference time than BoN, our method offers an effective balance between efficiency and overall performance.

**Ablation Study** We perform ablation experiments to validate the contribution of each module in the CRISP framework, with results summarized in Appendix M. The results show that removing any single module leads to a decline in performance. As our design is informed by the analysis presented in §3 (i.e., Cl.1-Cl.3), the results provide further empirical support for our findings.

## 5 Related Work

**Inference-time Optimization Technique in LLM's Reasoning** Recent studies have demonstrated that large language models (LLMs) can be effectively enhanced through search-based optimization at inference time (OpenAI, 2024; Zeng et al., 2024; Zhao et al., 2024). These works primarily follow two approaches: optimizing the strategy for LLMs to search for answers (Hao et al., 2023; Snell et al., 2024; Bi et al., 2024; Qi et al., 2024) or improving the reward model's ability to evaluate response quality (Wang et al., 2024b; Zhang et al., 2024; Setlur et al., 2024). However, most studies explore these two approaches separately, with limited research analyzing the impact of search factors on RM performance. Our work addresses this gap and proposes a new search strategy to mitigate RM's deficiencies.

**Reward Model in LLM's Reasoning** The reward model plays a crucial role in complex reasoning tasks of LLMs (Zeng et al., 2024; Setlur et al., 2024; Wang et al., 2024b). Existing works mainly investigate the RM from two perspectives: evaluation and optimization. For the former, researchers design various datasets to evaluate the RM's ability to distinguish between positive and negative responses (Lambert et al., 2024; Liu et al., 2024c; Zheng et al., 2024). For the latter, researchers focus on the training phase, improving the RM's ability by synthesizing high-quality data (Wang et al., 2024b; Liu et al., 2024a) or optimizing the training algorithm (Zhang et al., 2024; Ankner et al., 2024; Lou et al., 2024). There is a lack of in-depth analysis of the potential issues RM faces during inference, as well as methods to optimize RM's performance in the inference stage. Our work addresses the gaps left by these related studies.

## 6 Conclusion

In this work, we focus on analyzing key factors that influence the reward model's performance in reasoning tasks. We find that low question difficulty, large sampling number, and high search diversity can lead to issues in RM-based inference, with in-depth explanations provided. To address these issues, we propose CRISP, a cluster-based, prefix-guided inference algorithm that enhances the robustness and efficiency of the reward model. Experimental results demonstrate that our method is effective in enhancing LLM reasoning capabilities.

## Reproducibility Statement

We have taken several steps to improve the reproducibility of our research. We offer a detailed account of the parameter settings and prompts used in the experiments, which are outlined in Appendix L. The full experimental code is also uploaded in the supplementary materials. We commit to making all code open source if the paper is accepted.

## References

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. Critique-out-loud reward models. *CoRR*, abs/2408.11791, 2024. doi: 10.48550/ARXIV.2408.11791. URL https://doi.org/10.48550/arXiv.2408.11791.

Zhenni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning. *CoRR*, abs/2412.09078, 2024. doi: 10.48550/ARXIV.2412.09078. URL https://doi.org/10.48550/arXiv.2412.09078.

Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *CoRR*, abs/2407.21787, 2024. doi: 10.48550/ARXIV.2407.21787. URL https://doi.org/10.48550/arXiv.2407.21787.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. *CoRR*, abs/2412.21187, 2024. doi: 10.48550/ARXIV.2412.21187. URL https://doi.org/10.48550/arXiv.2412.21187.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168, 2021. URL https://arxiv.org/abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen

Zhang. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Leo Gao, John Schulman, and Jacob Hilton. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, pp. 10835–10866. PMLR, 2023.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 8154–8173. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.507. URL https://doi.org/10.18653/v1/2023.emnlp-main.507.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3828–3850. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.211. URL https://doi.org/10.18653/v1/2024.acl-long.211.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In Joaquin Vanschoren and Sai-Kit Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper/2021/hash/be83ab3ecd0db773eb2dc1b0a17836a1-Abstract-round2.html.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Raghavi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. Rewardbench: Evaluating reward models for language modeling. *CoRR*, abs/2403.13787, 2024. doi: 10.48550/ARXIV.2403.13787. URL https://doi.org/10.48550/arXiv.2403.13787.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. Program induction by rationale generation: Learning to solve and explain algebraic word problems. In Regina Barzilay and Min-Yen Kan (eds.), *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pp. 158–167. Association for Computational Linguistics, 2017. doi: 10.18653/V1/P17-1015. URL https://doi.org/10.18653/v1/P17-1015.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. Skywork-reward: Bag of tricks for reward modeling in llms. *CoRR*, abs/2410.18451, 2024a. doi: 10.48550/ARXIV.2410.18451. URL https://doi.org/10.48550/arXiv.2410.18451.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In Christian Bessiere (ed.), *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pp. 3622–3628. ijcai.org, 2020. doi: 10.24963/IJCAI.2020/501. URL https://doi.org/10.24963/ijcai.2020/501.

Tianqi Liu, Wei Xiong, Jie Ren, Lichang Chen, Junru Wu, Rishabh Joshi, Yang Gao, Jiaming Shen, Zhen Qin, Tianhe Yu, et al. Rrm: Robust reward model training mitigates reward hacking. *arXiv preprint arXiv:2409.13156*, 2024b.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. Rm-bench: Benchmarking reward models of language models with subtlety and style. *CoRR*, abs/2410.16184, 2024c. doi: 10.48550/ARXIV.2410.16184. URL https://doi.org/10.48550/arXiv.2410.16184.

Xingzhou Lou, Dong Yan, Wei Shen, Yuzi Yan, Jian Xie, and Junge Zhang. Uncertainty-aware reward model: Teaching reward models to know what is unknown. *CoRR*, abs/2410.00847, 2024. doi: 10.48550/ARXIV.2410.00847. URL https://doi.org/10.48550/arXiv.2410.00847.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *CoRR*, abs/2501.19393, 2025. doi: 10.48550/ARXIV.2501.19393. URL https://doi.org/10.48550/arXiv.2501.19393.

Skywork o1 Team. Skywork-o1 open series. https://huggingface.co/Skywork, November 2024. URL https://huggingface.co/Skywork.

OpenAI. Introducing openai o1 preview., 2024. URL https://openai.com/index/introducing-openai-o1-preview/. Accessed: 2025-01-24.

Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. Mutual reasoning makes smaller llms stronger problem-solvers. *CoRR*, abs/2408.06195, 2024. doi: 10.48550/ARXIV.2408.06195. URL https://doi.org/10.48550/arXiv.2408.06195.

Yuxiao Qu, Matthew Y. R. Yang, Amrith Setlur, Lewis Tunstall, Edward Emanuel Beeching, Ruslan Salakhutdinov, and Aviral Kumar. Optimizing test-time compute via meta reinforcement fine-tuning. *CoRR*, abs/2503.07572, 2025. doi: 10.48550/ARXIV.2503.07572. URL https://doi.org/10.48550/arXiv.2503.07572.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024a. doi: 10.48550/ARXIV.2408.00118. URL https://doi.org/10.48550/arXiv.2408.00118.

Morgane Rivière, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, Johan Ferret, Peter Liu, Pouya Tafti, Abe Friesen, Michelle Casbon, Sabela Ramos, Ravin Kumar, Charline Le Lan, Sammy Jerome, Anton Tsitsulin, Nino Vieillard, Piotr Stanczyk, Sertan Girgin, Nikola Momchev, Matt Hoffman, Shantanu Thakoor, Jean-Bastien Grill, Behnam Neyshabur, Olivier Bachem, Alanna Walton, Aliaksei Severyn, Alicia Parrish, Aliya Ahmad, Allen Hutchison, Alvin Abdagic, Amanda Carl, Amy Shen, Andy Brock, Andy Coenen, Anthony Laforge, Antonia Paterson, Ben Bastian, Bilal Piot, Bo Wu, Brandon Royal, Charlie Chen, Chintu Kumar, Chris Perry, Chris Welty, Christopher A. Choquette-Choo, Danila Sinopalnikov, David Weinberger, Dimple Vijaykumar, Dominika Rogozinska, Dustin Herbison, Elisa Bandy, Emma Wang, Eric Noland, Erica Moreira, Evan Senter, Evgenii Eltyshev, Francesco Visin, Gabriel Rasskin, Gary Wei, Glenn Cameron, Gus Martins, Hadi Hashemi, Hanna Klimczak-Plucinska, Harleen Batra, Harsh Dhand, Ivan Nardini, Jacinda Mein, Jack Zhou, James Svensson, Jeff Stanway, Jetha Chan, Jin Peng Zhou, Joana

Carrasqueira, Joana Iljazi, Jocelyn Becker, Joe Fernandez, Joost van Amersfoort, Josh Gordon, Josh Lipschultz, Josh Newlan, Ju-yeong Ji, Kareem Mohamed, Kartikeya Badola, Kat Black, Katie Millican, Keelin McDonell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sjösund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. Gemma 2: Improving open language models at a practical size. *CoRR*, abs/2408.00118, 2024b. doi: 10.48550/ARXIV.2408.00118. URL https://doi.org/10.48550/arXiv.2408.00118.

Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. Winogrande: An adversarial winograd schema challenge at scale. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020*, pp. 8732–8740. AAAI Press, 2020. doi: 10.1609/AAAI.V34I05.6399. URL https://doi.org/10.1609/aaai.v34i05.6399.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. Socialiqa: Commonsense reasoning about social interactions. *CoRR*, abs/1904.09728, 2019. URL http://arxiv.org/abs/1904.09728.

Abulhair Saparov and He He. Language models are greedy reasoners: A systematic formal analysis of chain-of-thought. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/pdf?id=qFVVBzXxR2V.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for LLM reasoning. *CoRR*, abs/2410.08146, 2024. doi: 10.48550/ARXIV.2410.08146. URL https://doi.org/10.48550/arXiv.2410.08146.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *CoRR*, abs/2408.03314, 2024. doi: 10.48550/ARXIV.2408.03314. URL https://doi.org/10.48550/arXiv.2408.03314.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, and Xia Ben Hu. Stop overthinking: A survey on efficient reasoning for large language models. *CoRR*, abs/2503.16419, 2025. doi: 10.48550/ARXIV.2503.16419. URL https://doi.org/10.48550/arXiv.2503.16419.

Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. Proofwriter: Generating implications, proofs, and abductive statements over natural language. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pp. 3621–3634. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.FINDINGS-ACL.317. URL https://doi.org/10.18653/v1/2021.findings-acl.317.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pp. 4149–4158. Association for Computational Linguistics, 2019. doi: 10.18653/V1/N19-1421. URL https://doi.org/10.18653/v1/n19-1421.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pp. 10582–10592. Association for Computational Linguistics, 2024a. URL https://aclanthology.org/2024.findings-emnlp.620.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of*

*the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 9426–9439. Association for Computational Linguistics, 2024b. doi: 10.18653/V1/2024.ACL-LONG.510. URL https://doi.org/10.18653/v1/2024.acl-long.510.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=1PL1NIMMrw.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022. URL http://papers.nips.cc/paper_files/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html.

Tian Xie, Zitian Gao, Qingnan Ren, Haoming Luo, Yuqian Hong, Bryan Dai, Joey Zhou, Kai Qiu, Zhirong Wu, and Chong Luo. Logic-rl: Unleashing LLM reasoning with rule-based reinforcement learning. *CoRR*, abs/2502.14768, 2025. doi: 10.48550/ARXIV.2502.14768. URL https://doi.org/10.48550/arXiv.2502.14768.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

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. *CoRR*, abs/2409.12122, 2024b. doi: 10.48550/ARXIV.2409.12122. URL https://doi.org/10.48550/arXiv.2409.12122.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. LIMO: less is more for reasoning. *CoRR*, abs/2502.03387, 2025. doi: 10.48550/ARXIV.2502.03387. URL https://doi.org/10.48550/arXiv.2502.03387.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective. *CoRR*, abs/2412.14135, 2024. doi: 10.48550/ARXIV.2412.14135. URL https://doi.org/10.48550/arXiv.2412.14135.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. Generative verifiers: Reward modeling as next-token prediction. *CoRR*, abs/2408.15240, 2024. doi: 10.48550/ARXIV.2408.15240. URL https://doi.org/10.48550/arXiv.2408.15240.

Wenyuan Zhang, Shuaiyi Nie, Xinghua Zhang, Zefeng Zhang, and Tingwen Liu. S1-bench: A simple benchmark for evaluating system 1 thinking capability of large reasoning models. *arXiv preprint arXiv:2504.10368*, 2025a.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pp. 10495–10516. Association for Computational Linguistics, 2025b. URL https://aclanthology.org/2025.findings-acl.547/.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. Marco-o1: Towards open reasoning models for open-ended solutions. *CoRR*, abs/2411.14405, 2024. doi: 10.48550/ARXIV.2411.14405. URL https://doi.org/10.48550/arXiv.2411.14405.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. Processbench: Identifying process errors in mathematical reasoning. *CoRR*, abs/2412.06559, 2024. doi: 10.48550/ARXIV.2412.06559. URL https://doi.org/10.48550/arXiv.2412.06559.

Tianshi Zheng, Yixiang Chen, Chengxi Li, Chunyang Li, Qing Zong, Haochen Shi, Baixuan Xu, Yangqiu Song, Ginny Y Wong, and Simon See. The curse of cot: On the limitations of chain-of-thought in in-context learning. *arXiv preprint arXiv:2504.05081*, 2025.

## A THE USE OF LARGE LANGUAGE MODELS

Throughout the preparation of this manuscript, a large language model (LLM) was employed to assist exclusively with language refinement. Specifically, the LLM was used for:

- **Grammar and Syntax Improvements:** Correcting errors and optimizing sentence structures.
- **Conciseness and Precision:** Providing alternative phrasings for brevity and accuracy.

All research concepts, analyses, and conclusions were developed independently by the authors. The LLM's contributions were limited to linguistic enhancement and did not influence the study's conceptual content.

## B LIMITATIONS & FUTURE WORK

While our work provides a thorough investigation of RM behavior during inference, it does not address potential issues that may arise during the training of models. In future work, we aim to extend our study to the training phase of reward models. Understanding how training dynamics (such as reward signal design and data sampling strategies) impact downstream reasoning performance could offer deeper insights and help improve the overall reliability of LLM.

## C PERFORMANCE OF SELECTED RMS

To demonstrate that the RM issues identified in our experiments in Section §2 are not due to the selected RM's inherently low discriminative abilities, here we present the performance of our RM. For the two ORMs (e.g. ArmoRM-Llama3-8B and Skywork-Reward-Llama-3.1-8B), we report their performance on RewardBench (Lambert et al., 2024) compared to other baselines in Table 5. For the two PRMs (e.g. Math-Shepherd-Mistral-7B-PRM and Skywork-o1-Open-PRM-Qwen-2.5-7B), we report their performance on ProcessBench (Lambert et al., 2024) compared to other baselines in Table 6. From them, we can get that the performance of these models on relevant benchmarks is comparable to the advanced LLMs (e.g., GPT -4), hence they are representative.

## D ADDITIONAL OVERALL EXPERIMENTS

In addition to the experiments in the main text, we also conduct the experiments in other settings.

Firstly, while the main text compares different RMs using BoN methods, we now replicate this comparison using the MCTS approach. Our settings are as follows:

- **SC:** Using the self-consistency method for comparison;
- **Reward:** Using the reward score as $f$ in MCTS (e.g. MCTS-Reward in §3.3);
- **Maj_vote:** Using the major voting as $f$ in MCTS (e.g. MCTS-SC in §3.3);
- **Q_value:** Using the sum of Q-value in each path as $f$ in MCTS;
- **N_greedy:** At each step, select the node with the most frequent visits N and perform a top-down greedy search on the tree to obtain the final selected path;
- **Q_greedy:** At each step, select the node with the highest Q-value and perform a top-down greedy search on the tree to obtain the final selected path;
- **Oracle:** The coverage of the MCTS method.

In addition, we also use the consistency of the final answer output by the policy model itself as the source of the reward, denoted as 'Self'. The results are demonstrated in Figure 11. We can conclude that: (1) Even with the MCTS framework, the improvement in model reasoning brought by the RM is still minimal, further validating our conclusions in §2. (2) In Skywork and Skyworko1, the average performance of Reward is the best among all scoring functions. Therefore, in the MCTS-related experiments presented in the main text, we default to using it as the scoring function $f$.

Secondly, we focus on math reasoning in the main text, here we repeat our experiments on other types of reasoning tasks. Specifically, for math reasoning, we select another dataset: AQuA (Ling et al., 2017). For commonsense reasoning, we select WinoGrande (WINO) (Sakaguchi et al., 2020) and CSQA (Talmor et al., 2019); For logical reasoning, we select ProofWriter (Tafjord et al., 2021) and ProntoQA (Saparov & He, 2023) The results are demonstrated in Figure 12, 13, 14, 15 and 16. Lastly, we only use discriminative RM in the main text. All of these results are consistent with the conclusion in the main text.

## E Additional Experiments on Question Difficulty Approximation

In the main text, we calculate the question difficulty with assuming oracle access to a ground truth. However, in real-world applications, we are only given access to test prompts and do not know the true answers. Thus, we need to find a function that effectively estimates the problem difficulty without requiring ground truth. Specifically, we propose the following functions:

- **Length:** The average length of all responses to the question;
- **Count:** The count of different answers to the question;
- **Null:** The number of responses that fail to correctly generate the answer.

We classify the problems according to the difficulty levels as outlined in the main text and calculate the above three metrics across different levels of problem difficulty to compare the degree of correlation. The results are illustrated in Figure 17, 18 and 19. We can observe that, comparatively, the Count function is most directly proportional to difficulty. Therefore, we use this function to estimate difficulty when designing the CRISP method in §4.1.

## F Additional Experiments across Different Difficulty Levels

In the main text, we only analyze the impact of question difficulty on the MATH dataset. To demonstrate the generalizability of our conclusions, we repeat this experiment on GSM8K (Cobbe et al., 2021) and Olympiadbench (He et al., 2024). The former dataset contains 8.5K linguistically diverse elementary school math problems designed to evaluate arithmetic reasoning consistency, while the latter is an Olympiad-level bilingual multimodal scientific benchmark. Compared to MATH, the former is simpler, while the latter is more challenging. The results are illustrated in Table 7, 8 and 9. We can observe that the issues identified in Cl.1 are prevalent across various reasoning datasets.

## G Comparison Between Coverage and Accuracy

The changes in accuracy and coverage are shown in Figure 20,21. The results demonstrate that: **Regardless of the inference strategy used, the model's accuracy does not improve as $n$ increases.** The accuracy in plateaus beyond a relatively small number of samples (approximately 30). In contrast, the Oracle setting consistently increases, leading to a persistently widening gap between accuracy and coverage.

## H Case Analysis of Sampling Numbers Experiment

We start with a case analysis to uncover the issues inherent in the reward model. In the analysis, we randomly select five questions from different methods and examine the correctness of answers as $n$ scales. If a question is answered correctly, it indicates that the RM can accurately distinguish the positive examples from the negative ones, otherwise, it cannot. The results of this experiment are demonstrated in Figure 22, from which we can deduce that: **As $n$ increases, LLMs can generate incorrect responses that become increasingly challenging for the reward model to differentiate.** For some cases (like index 3 and 4 in Figure 22), RM assigns the highest score to newly generated incorrect responses, transforming the originally correct answers into incorrect ones.

## I Cause Analysis of Temperature-Induced Accuracy Drop

We further conduct statistical analyses to uncover the reasons for this issue. For each $T$, we calculate the information entropy of incorrect answers across 16 samplings and report the distribution over 200 questions in Figure 6, 24. As the temperature rises, the entropy for both models shows a gradually increasing trend, hence, the distribution of these negative samples becomes more random. This indicates that the policy model generates a greater number of low-frequency incorrect answers at higher temperatures. According to Cl.2, RM struggles to differentiate these negative examples from correct ones, leading to lower inference accuracy. This result not only elucidates the reasons behind the subpar performance of BoN under high diversity conditions but also further corroborates the inverse long-tail phenomenon of the RM.

## J Diversity Experiment on Exploration Constant

In MCTS, apart from the tree structure, the explore weight $c$ also plays a crucial role in balancing the trade-off between exploitation (i.e. choosing actions that are known to yield high rewards) and exploration. A higher value of $c$ encourages more exploration, increasing the weight of the uncertain actions in the UCB formula. A lower value of $c$ favors exploitation, as it prioritizes actions with known higher rewards. We compare the MCTS performance under different $c$ and present the result in Figure 25. We can observe that an excessively large $c$ reduces performance (e.g. $c = 10.0$), indicating that overly high sampling diversity impairs reasoning accuracy, which is consistent with Cl.3 in our main text.

## K Theoretical Analysis of CRISP Method

In this section, we present a theoretical analysis of the clustering strategy (i.e., State Aggregation module + Reward Evaluation module) within the CRISP method, as it serves as the core component of the entire approach.

Assume we have sampled $n$ paths, where each answer $a_i$ corresponds to a reward $r_i$, and $f_i$ is the frequency of $a_i$. In Cl.2, we observe that RM tends to assign a higher $r_i$ to an incorrect $a_i$ with lower $f_i$, sometimes even exceeding the score of the highest-scoring correct example, leading to an incorrect final answer. Our CRISP's clustering method incorporates frequency $f_i$ as a factor into the new reward scores $r'_i$ to mitigate this issue:

$$r'_i = \sum_{a_k = a_i} r_k = f_i \cdot \overline{r_i} \tag{7}$$

where $\overline{r_i}$ represents the average score of the cluster to which $a_i$ belongs. Suppose $a_j$ is the top-scored negative answer, we have:

$$\frac{r'_i}{r'_j} = \frac{f_i}{f_j} \cdot \frac{\overline{r_i}}{\overline{r_j}} \tag{8}$$

where $\overline{r_i}$ represents the average score of the cluster to which $a_i$ belongs. Although $\overline{r_i} < \overline{r_j}$, as long as $\frac{f_i}{f_j} > \frac{\overline{r_j}}{\overline{r_i}}$, we have $r'_j > r'_i$. According to Figure 4, when n=128, in most cases, $f_j < 3$, which is a very small value. Therefore, in most cases, there exists $f_i \gg f_j$, such that $r'_j > r'_i$, reducing the score ranking of these negative examples.

In summary, our CRISP method reduces the tendency of the RM to assign excessively high scores to low-frequency negative examples, thereby increasing the probability of selecting the correct path. It performs better when the generative model samples the correct answer more frequently (i.e., $f_i \gg f_j$).

## L Implementation Details in the Main Experiments

Here, we provide a detailed account of the implementation specifics from the main experiments:

For Self-Consistency, we generate 32 samples and choose the major voting answer as the final prediction. For BoN, we set the temperature to 0.7 to control the diversity and choose the best answer

from 32 samples. For BoN Weighted, we normalize the RM's scoring and use this score as a weight to conduct a weighted vote among different answers, selecting the final prediction. For MCTS, we set the rollout number to 16, the width to 5, the max depth to 5, and the explore weight to 0.1. For Beam Search, we set the Beam numbers to 8, the beam width to 5, and the max depth to 5.

For our method, we generate 16 samples with a temperature setting of 0.7 in the first iteration. In subsequent iterations, we set the sampling numbers to 8 for ORM, 4 for PRM, and the max depth to 3. In prefix extraction, for ORM, we select the top-1 path, for PRM, we select the top-2 paths. For the evaluation data, we sample 500 questions from GSM8K and MATH-500, while sampling 200 questions from OlympiadBench.

We release the prompts we use in Table 11, 12, 13, 14, 15 and 16. All experiments were conducted on NVIDIA A100 GPUs.

## M Ablation Study

To verify the effectiveness of each module of CRSIP, we conduct ablation experiments on different modules in it. The experimental settings are as follows:

- **w/o Termination:** Disable the early termination condition based on the number of clusters;
- **w/o Aggregation:** Eliminate the clustering operation and use the score of each path instead of cluster scores for selection (similar to MCTS);
- **w/o Prefixing:** Cancel the operation of directly generating the remaining steps according to the prefix set, and instead generate intermediate nodes layer by layer (similar to MCTS and Beam).

Figure 28 and Table 10 show the result of the ablation study. Removing each component leads to a decline in performance. Specifically, although w/o termination causes only a small drop, its inclusion not only improves performance but also reduces inference time.

| Reward Model | Score | Chat | Chat Hard | Safety | Reasoning |
|---|---|---|---|---|---|
| Skywork-Reward-Llama-3.1-8B | 93.1 | 94.7 | 88.4 | 92.7 | 96.7 |
| ArmoRM-Llama3-8B-v0.1 | 89.0 | 96.9 | 76.8 | 92.2 | 97.3 |
| Gemini-1.5-pro-0514 | 88.1 | 92.3 | 80.6 | 87.5 | 92.0 |
| gpt-4-0125-preview | 84.3 | 95.3 | 74.3 | 87.2 | 86.9 |
| Meta-Llama-3-70B-Instruct | 75.4 | 97.6 | 58.9 | 69.2 | 78.5 |

Table 5: Comparison of RM's performance on RewardBench.

| Model | GSM8K | MATH | OlympiadBench | OmniMATH | Average |
|---|---|---|---|---|---|
| Shepherd-PRM-7B | 47.9 | 29.5 | 24.8 | 23.8 | 31.5 |
| Skyworko1-PRM-7B | 70.8 | 53.6 | 22.9 | 21.0 | 42.1 |
| Meta-Llama-3-70B-Instruct | 52.2 | 22.8 | 21.2 | 20.0 | 29.1 |
| Llama-3.1-70B-Instruct | 74.9 | 48.2 | 46.7 | 41.0 | 52.7 |
| Qwen2-72B-Instruct | 67.6 | 49.2 | 42.1 | 40.2 | 49.8 |

Table 6: Comparison of RM's performance on ProcessBench.

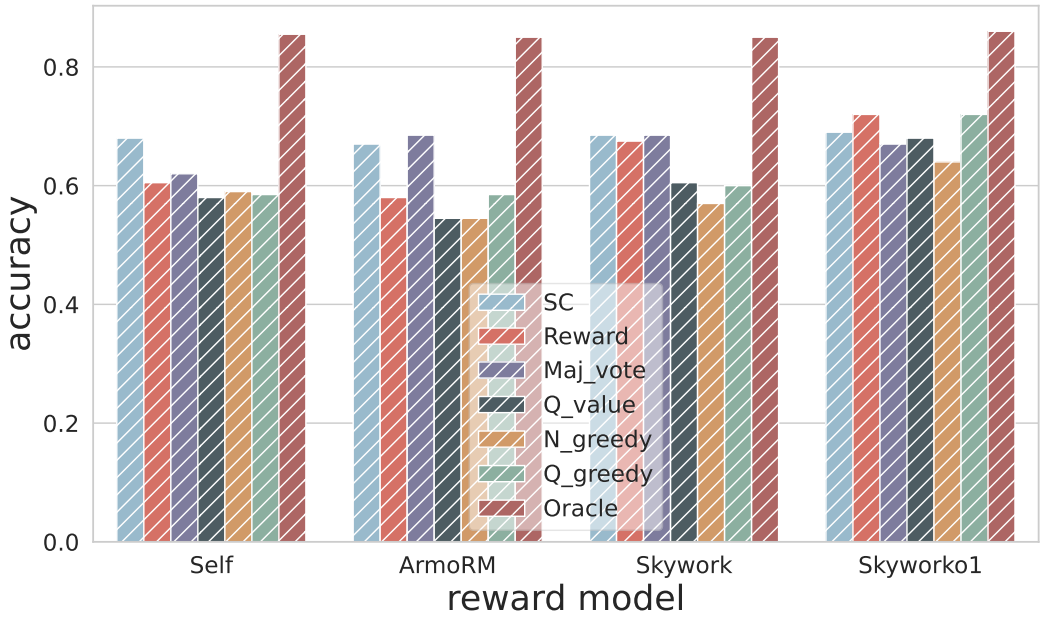



Figure 11: The performance of different reward models using the MCTS inference on the MATH dataset ($n$ = 16, Qwen-2.5-3B).

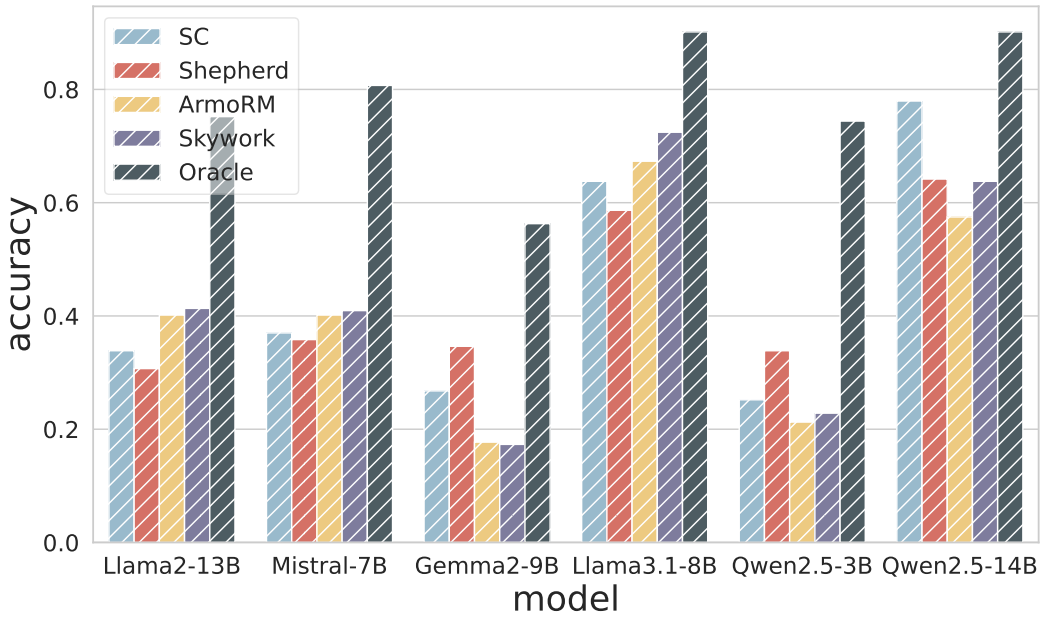



Figure 12: The performance of different policy models using various reward models for BoN inference on the AQuA dataset ($n$ = 10).

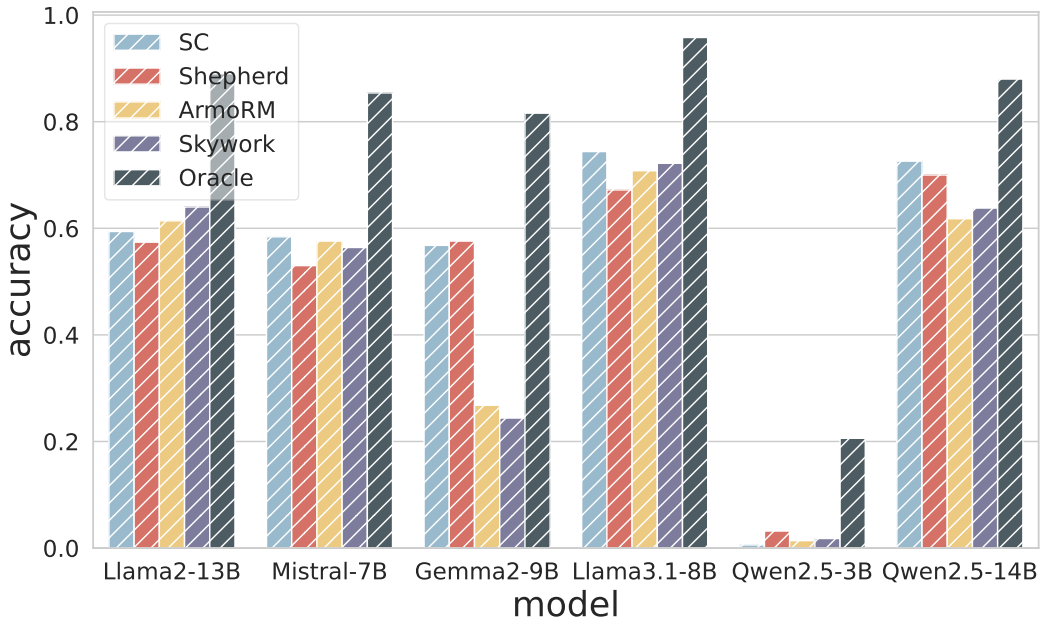



Figure 13: The performance of different policy models using various reward models for BoN inference on the WinoGrande dataset ($n = 10$).

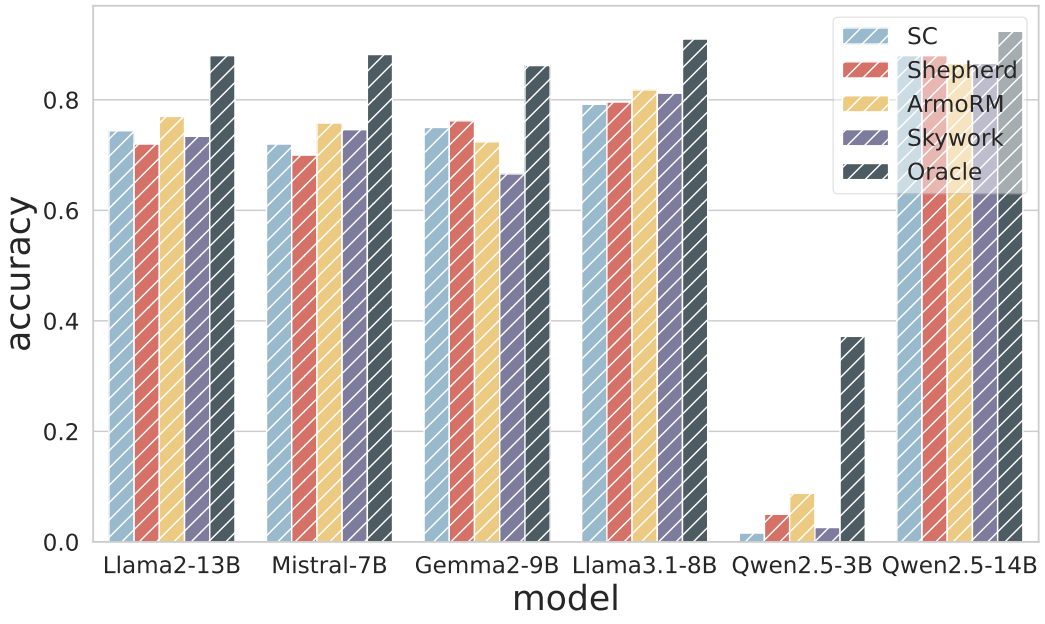



Figure 14: The performance of different policy models using various reward models for BoN inference on the CSQA dataset ($n = 10$).

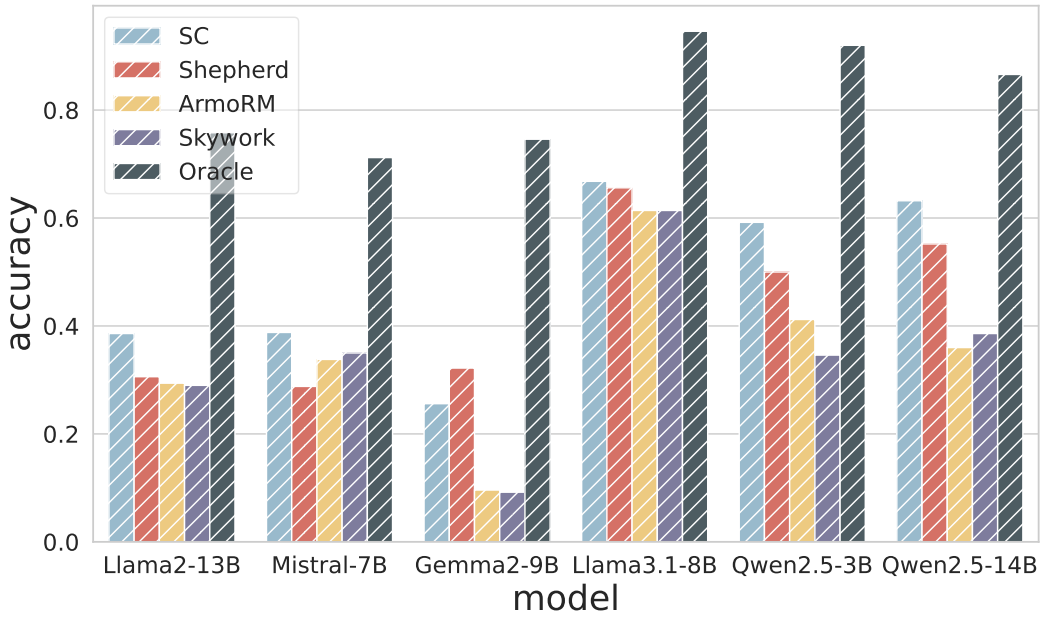



Figure 15: The performance of different policy models using various reward models for BoN inference on the ProofWriter dataset ($n = 10$).

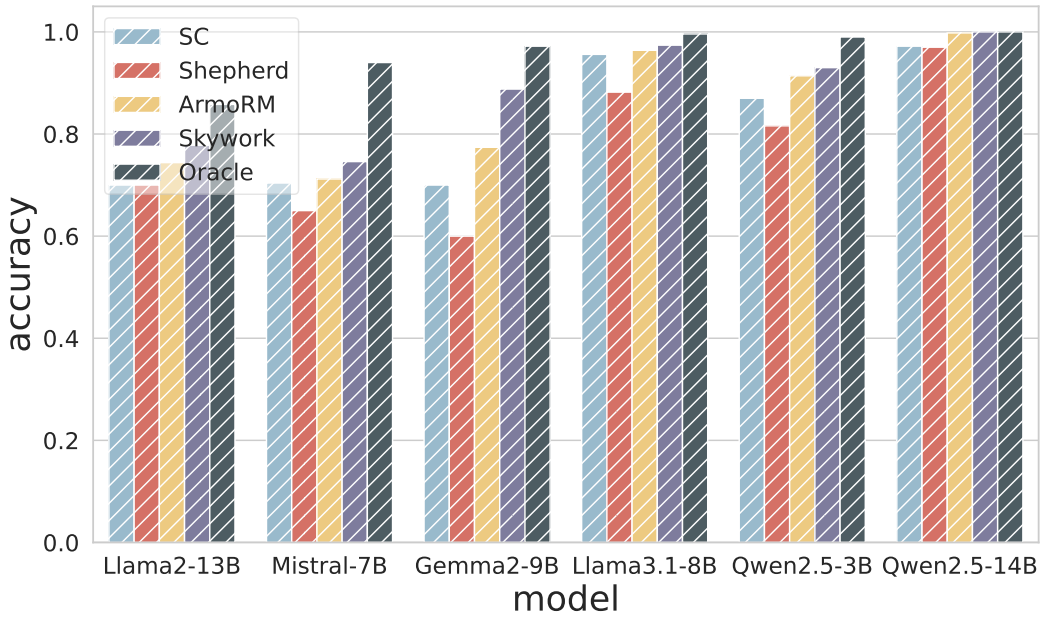



Figure 16: The performance of different policy models using various reward models for BoN inference on the ProntoQA dataset ($n = 10$).

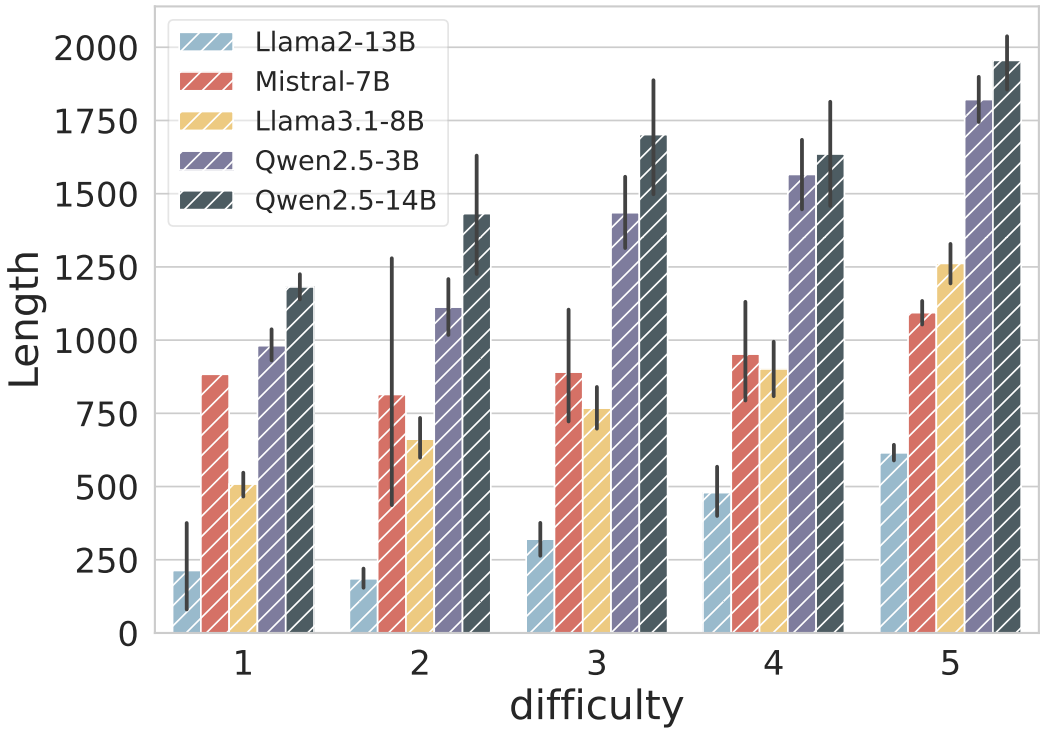



Figure 17: The correlation between output length and the question difficulty.

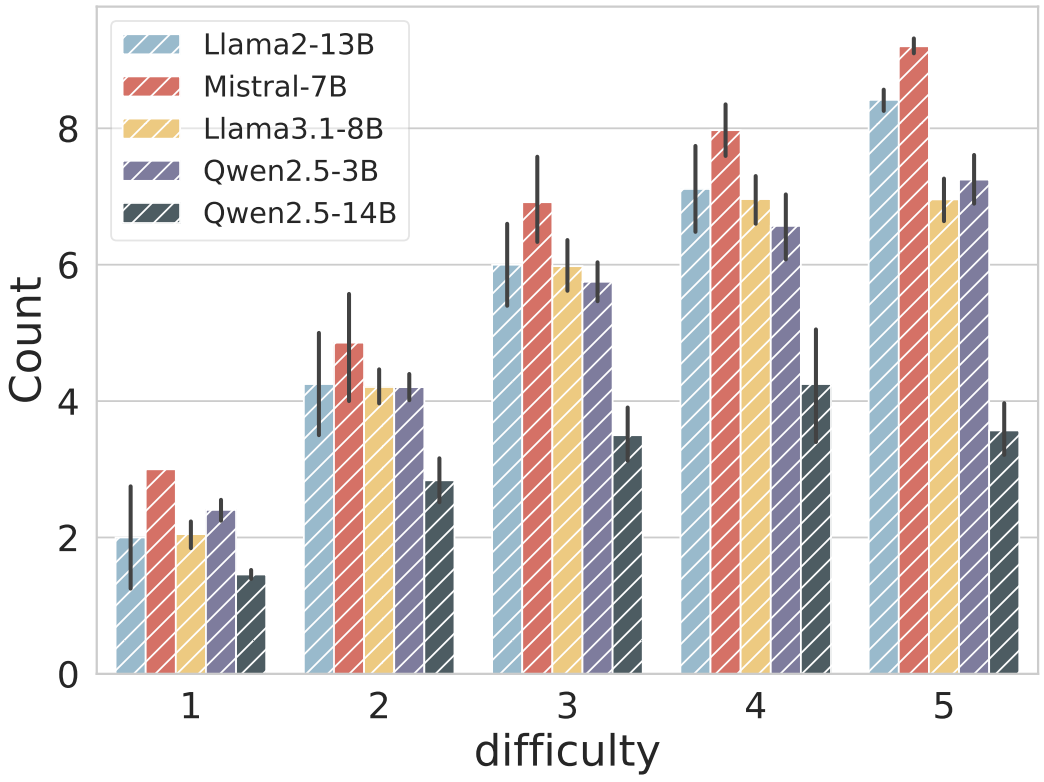



Figure 18: The correlation between the count of answers and the question difficulty.

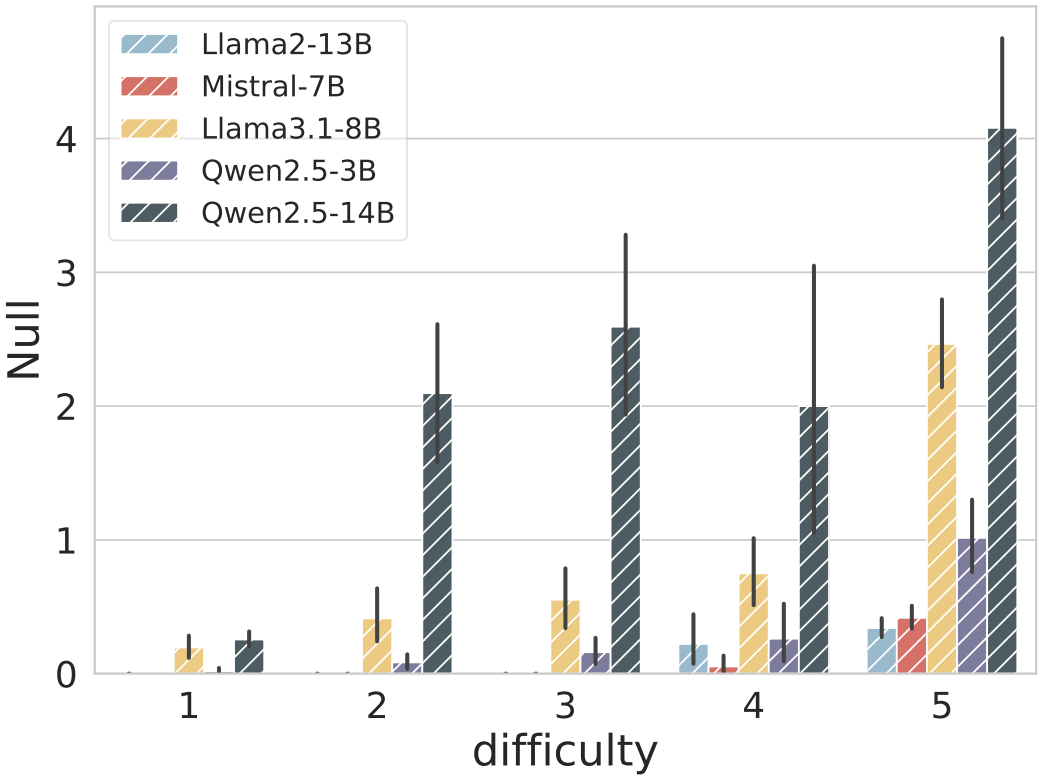



Figure 19: The correlation between the count of no answers and the question difficulty.

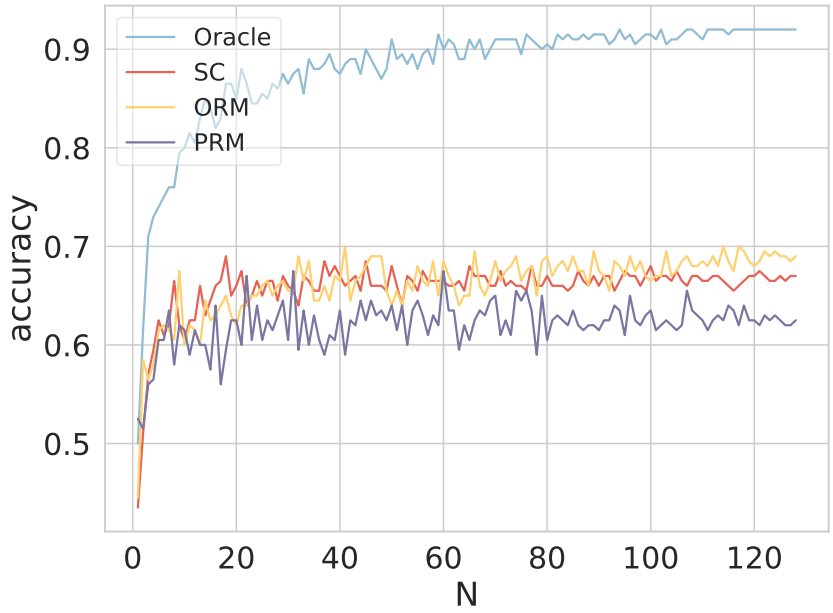



Figure 20: BoN performance across different sampling numbers.

Table 7: Comparison of performance across different difficulty levels on 500 samples of GSM8K (Qwen2.5-3B).

| Method | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | All |
|---|---|---|---|---|---|---|
| Self-Consistency(@128) | **99.7** | 96.8 | 80.0 | 34.6 | 3.2 | 83.2 |
| Best-of-128 + ORM | 98.0 | 87.1 | 72.0 | **65.4** | 12.9 | 83.8 |
| - SC | -1.7 | -9.7 | -8.0 | 30.8 | 9.7 | 0.6 |
| Best-of-128 + PRM | 98.3 | **100.0** | **96.0** | 57.7 | **30.6** | **87.8** |
| - SC | -1.4 | 3.2 | 16.0 | 23.1 | 27.4 | 4.6 |
| Count | 356 | 31 | 25 | 26 | 62 | 500 |

Table 8: Comparison of performance across different difficulty levels on MATH-500 (Qwen2.5-3B).

| Method | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | All |
|---|---|---|---|---|---|---|
| Self-Consistency(@128) | 98.8 | **98.8** | **80.4** | 49.2 | 5.3 | 65.4 |
| Best-of-128 + ORM | **99.4** | 92.8 | 69.6 | **58.5** | 17.3 | **67.8** |
| - SC | 0.6 | -6.0 | -9.8 | 9.3 | 12.0 | 2.4 |
| Best-of-128 + PRM | 88.3 | 71.1 | 78.6 | 53.8 | **21.8** | 62.2 |
| - SC | -10.5 | -27.7 | -1.8 | 4.6 | 16.5 | -3.2 |
| Count | 163 | 83 | 56 | 65 | 133 | 500 |

Table 9: Comparison of performance across different difficulty levels on 200 samples of OlympiadBench (Qwen2.5-3B).

| Method | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | All |
|---|---|---|---|---|---|---|
| Self-Consistency(@32) | **100.0** | **100.0** | 64.3 | 50.0 | 0.8 | 30.5 |
| Best-of-32 + ORM | 100.0 | 80.0 | **78.6** | 40.0 | 3.8 | 31.5 |
| - SC | 0.0 | -20.0 | 14.3 | -10.0 | 3.0 | 1.0 |
| Best-of-32 + PRM | 100.0 | 100.0 | **78.6** | **50.0** | **6.9** | **34.0** |
| - SC | 0.0 | 0.0 | 14.3 | 0.0 | 6.1 | 3.5 |
| Count | 31 | 15 | 14 | 10 | 130 | 200 |

Table 10: Results of our ablation study on different reward models.

| Method | ORM | PRM |
|---|---|---|
| Ours | 0.73 | 0.78 |
| -w/o Termination | 0.72 | 0.76 |
| -w/o Aggregation | 0.71 | 0.75 |
| -w/o Prefixing | 0.64 | 0.72 |

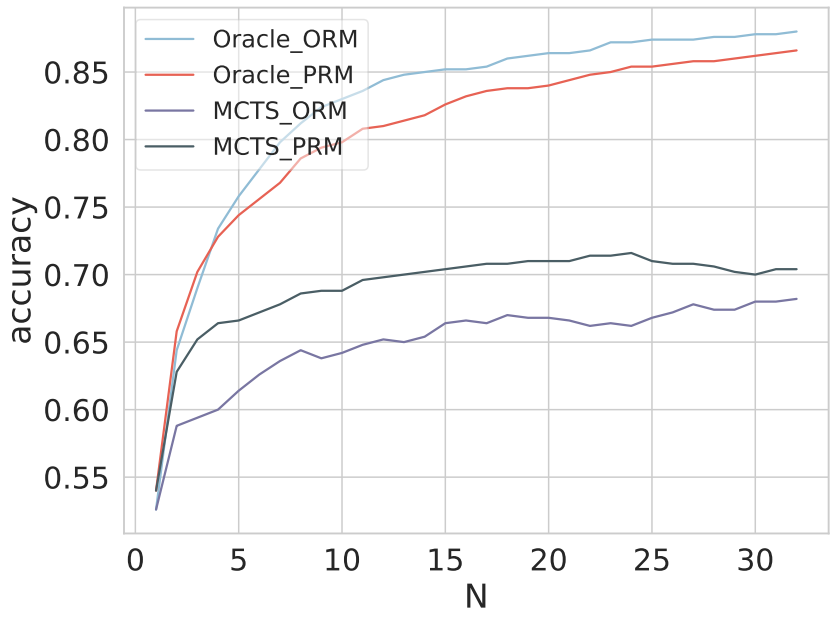
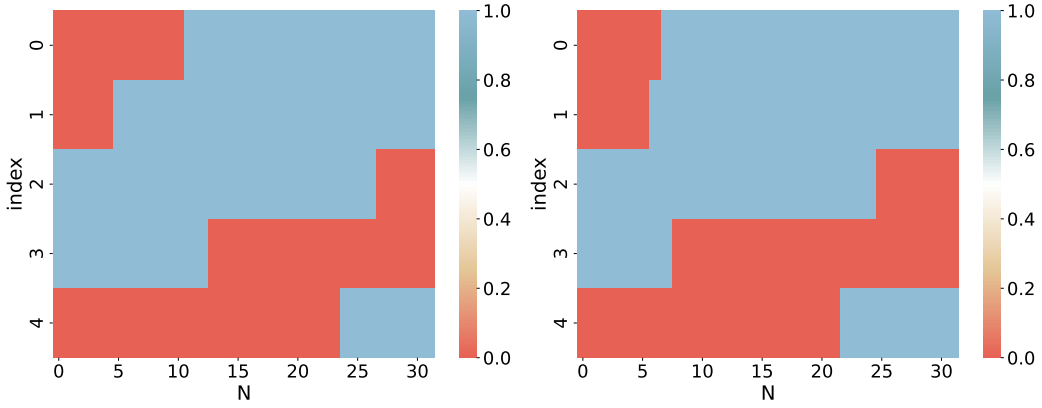



Figure 21: MCTS performance across different sampling numbers.




(a) BoN (b) MCTS

Figure 22: The variation in question answering correctness as the sampling number changes. Blue indicates a correct answer, while red indicates an incorrect answer.

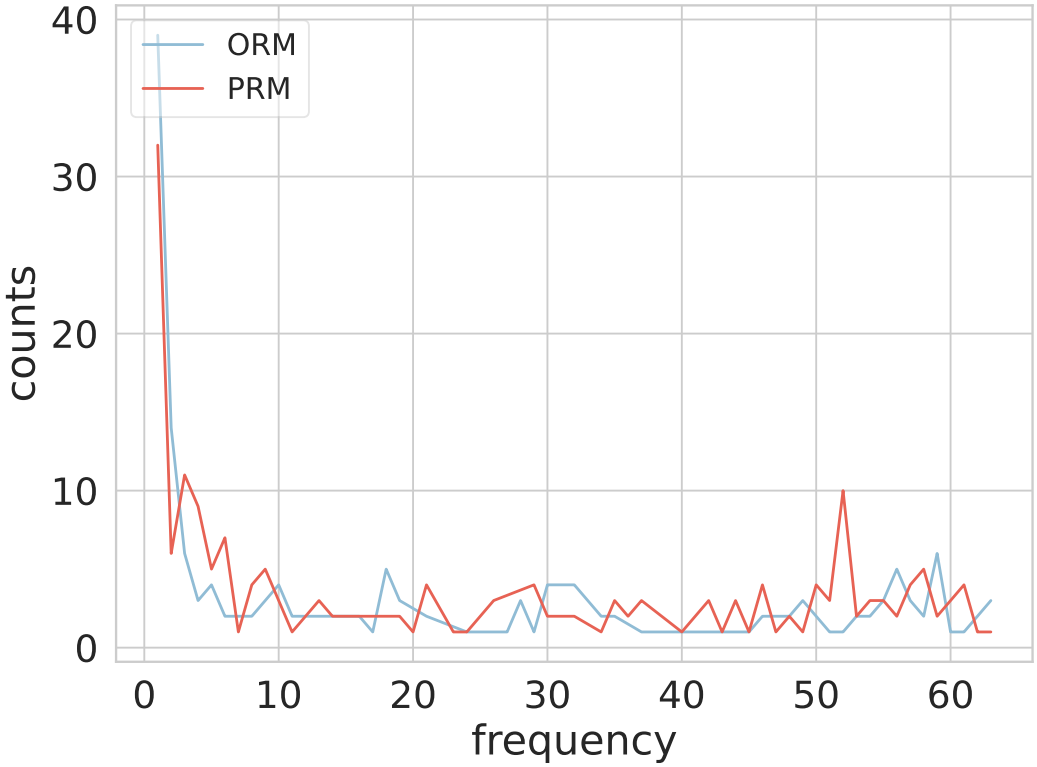



Figure 23: Frequency statistics of the highest-scored negative responses in MCTS.

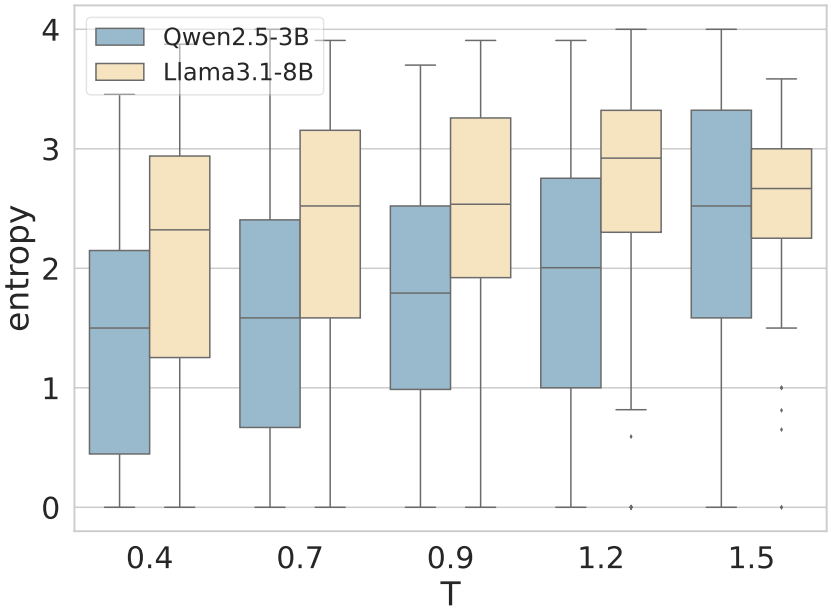
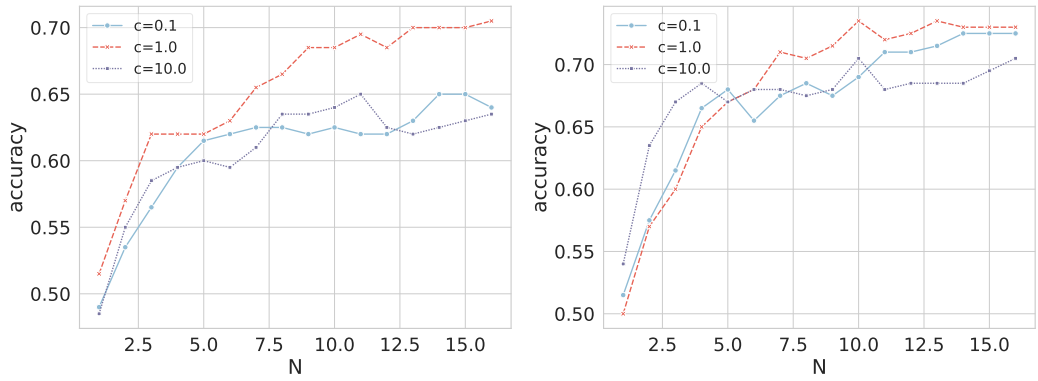



Figure 24: Information entropy of incorrect answers under different sampling temperatures.




(a) BoN (b) MCTS

Figure 25: Performance comparison across different explore weight $c$ on Qwen2.5-3B.

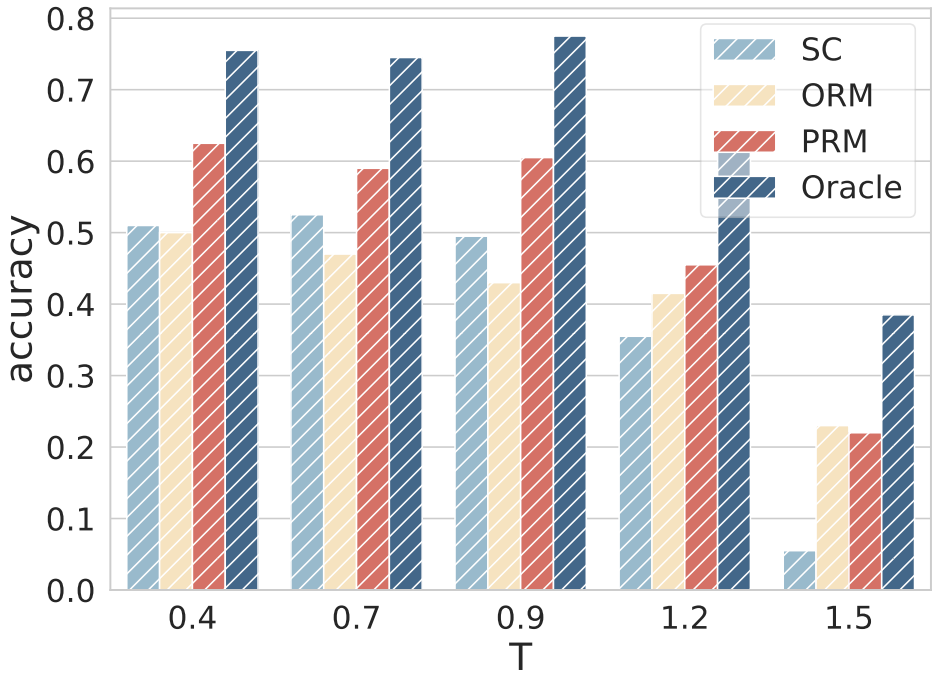
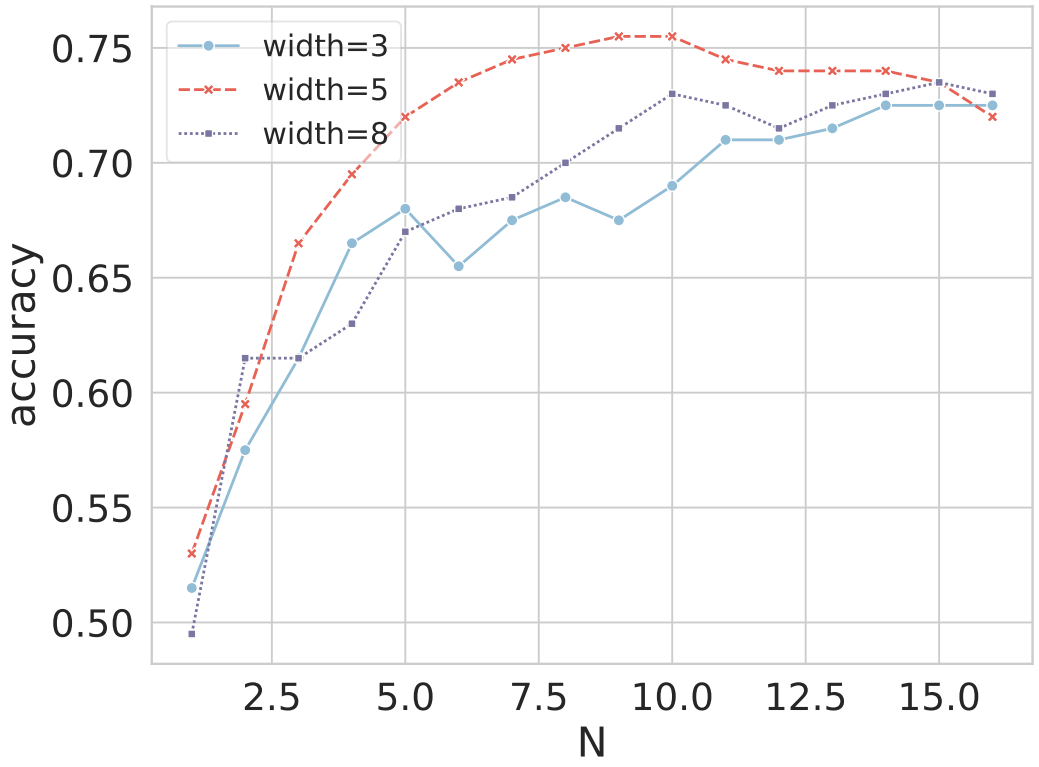
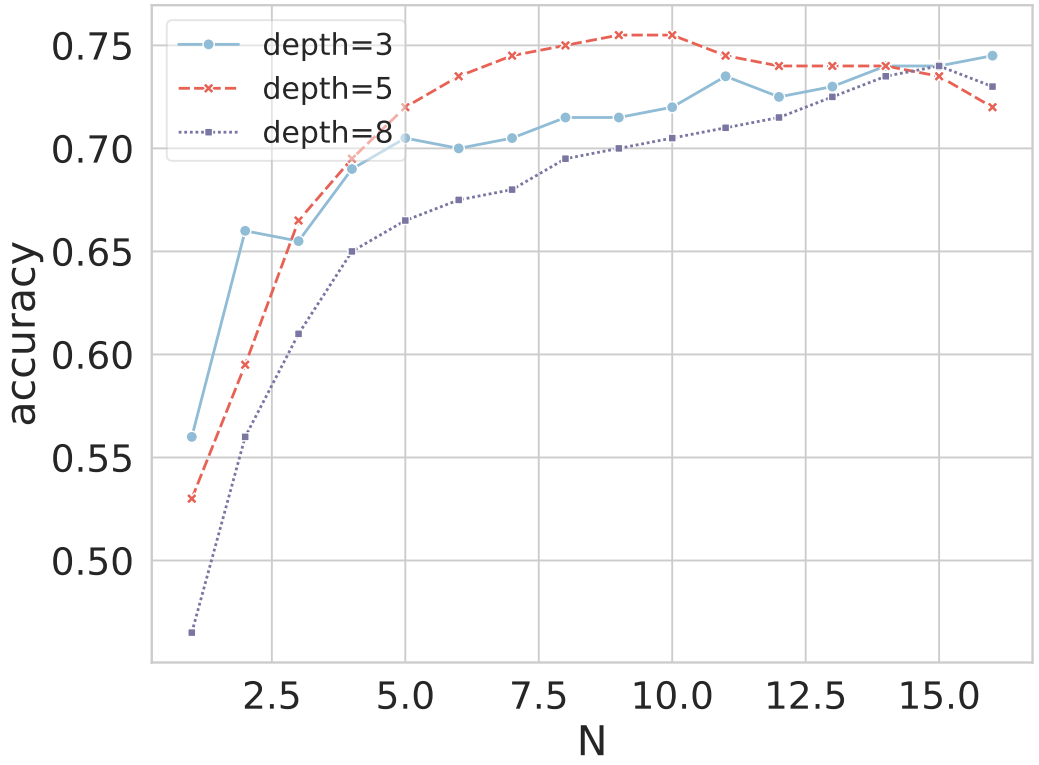



Figure 26: Performance of BoN inference across different sampling temperatures (Llama3.1-8B).




(a) Tree width




(b) Tree depth

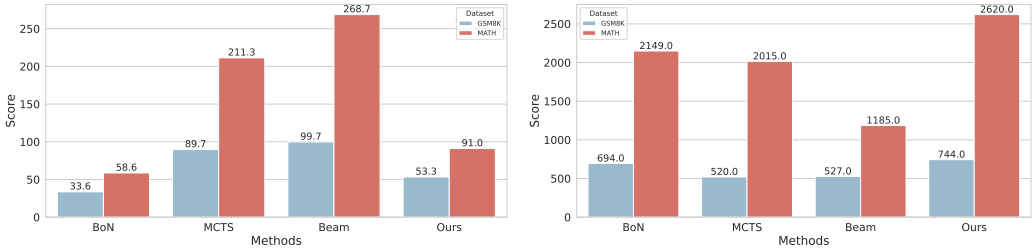
Figure 27: MCTS inference performance under different tree structures (PRM).

**Algorithm 1** Clustered Reward Integration with Stepwise Prefixing

**Require:** Policy model $\mathcal{M}$, reward score $f$, question $q$, max steps $m$, sampling numbers $n$, top-$k$ parameter $k$

$i \leftarrow 0$
$\mathcal{R} \leftarrow \emptyset$ ▷ All responses
$\mathcal{P} \leftarrow \emptyset$ ▷ Response prefixes
$\mathcal{F} \leftarrow \emptyset$ ▷ Score map
$\mathcal{C} \leftarrow \emptyset$ ▷ Clusters
**while** $i < n$ **do**
  **if** $i = 0$ **then**
    $\mathcal{R} \leftarrow \mathcal{M}(q, n)$ ▷ Generate $n$ initial responses
    **if** $|\,\mathrm{Cluster}(\mathcal{R})| = 1$ **then**
      **return** $\mathcal{R}[0]$ ▷ Early exit if only one cluster
    **end if**
  **else**
    $\mathcal{R}_{\text{top}} \leftarrow \left\{\arg\max_{r \in \mathcal{C}_j} f(r) \,\middle|\, \mathcal{C}_j \in \mathcal{C}_{\text{top}}\right\}$
    $\mathcal{P} \leftarrow \{r[:i{+}1] \mid r \in \mathcal{R}_{\text{top}}\}$ ▷ Truncate top responses
    $\mathcal{R} \leftarrow \mathcal{R} \cup \mathcal{M}(q, n, \mathcal{P})$ ▷ Decode more based on prefixes
  **end if**
  $\mathcal{C} \leftarrow \mathrm{Cluster}(\mathcal{R})$ ▷ Cluster current responses
  **for all** $\mathcal{C}_j \in \mathcal{C}$ **do**
    $\mathcal{F}(\mathcal{C}_j) \leftarrow \sum_{x \in \mathcal{C}_j} f(x)$ ▷ Assign cluster-wise reward
  **end for**
  $\mathcal{C}_{\text{top}} \leftarrow$ top-$k$ responses in $\mathcal{C}$ by $\mathcal{F}$
  $i \leftarrow i + 1$
**end while**
**return** $\mathcal{R}_{\text{top}}[0]$


Figure 28: Results of our ablation study on different datasets.




(a) Time Consumption Comparison (s)

(b) Token Consumption Comparison

Figure 29: Results of our cost analysis.

Table 11: Prompts used to sample reasoning paths on the GSM8K dataset.

**Prompt**

Please act as a math teacher and solve the math problem step by step. At the final step, a conclusive answer is given in the format of "The answer is: \boxed{¡ANSWER¿}.", where ¡ANSWER¿ should be a numeric answer.

**# Question:**
Mr. Ruther sold $\frac{3}{5}$ of his land and had 12.8 hectares left. How much land did he have at first?
**# Reasoning:**
Step 1: Mr. Ruther is left with $1 - \frac{3}{5} = \frac{2}{5}$ of his land.
Step 2: Since $\frac{2}{5}$ equals 12.8 hectares, then $\frac{1}{5} = \frac{12.8}{2} = 6.4$ hectares.
Step 3: Total land = $6.4 \times 5 = 32$ hectares.
Step 4: The answer is: $\boxed{32}$

**# Question:**
The Doubtfire sisters are driving home with 7 kittens adopted from the local animal shelter when their mother calls to inform them that their two house cats have just had kittens. She says that Patchy, the first cat, has had thrice the number of adopted kittens, while Trixie, the other cat, has had 12. How many kittens does the Doubtfire family now have?
**# Reasoning:**
Step 1: Patchy has had $3 \times 7 = 21$ kittens.
Step 2: Trixie has had 12 kittens. Total from both cats = $21 + 12 = 33$.
Step 3: Total kittens including adopted = $7 + 33 = 40$.
Step 4: The answer is: $\boxed{40}$

**# Question:**
After transferring to a new school, Amy made 20 more friends than Lily. If Lily made 50 friends, how many friends do Lily and Amy have together?
**# Reasoning:**
Step 1: Amy made $50 + 20 = 70$ friends.
Step 2: Total friends = $70 + 50 = 120$.
Step 3: The answer is: $\boxed{120}$
**# Question:**
{current question}
**# Reasoning:**

Table 12: Prompts used to sample reasoning paths on the MATH dataset.

**Prompt**

Please act as a math teacher and give step-by-step solutions to the user's questions. At the final step, a conclusive answer is given in the format of "The answer is: <ANSWER>.", where <ANSWER> should be a numeric answer.

**# Question:**

How many 3-letter words can we make from the letters A, B, C, and D, if we are allowed to repeat letters, and we must use the letter A at least once? (Here, a word is an arbitrary sequence of letters.)

**# Reasoning:**

Step 1: There are $4^3$ three-letter words from A, B, C, and D, and there are $3^3$ three-letter words from just B, C, and D.

Step 2: There must, then, be $4^3 - 3^3 = 64 - 27 = \boxed{37}$ words from A, B, C, and D containing at least one A.

Step 3: The answer is: $\boxed{37}$

**# Question:**

In the diagram, square $ABCD$ has sides of length 4, and $\triangle ABE$ is equilateral. Line segments $BE$ and $AC$ intersect at $P$. Point $Q$ is on $BC$ so that $PQ$ is perpendicular to $BC$ and $PQ = x$.

**# Reasoning:**

Step 1: Since $\triangle ABE$ is equilateral, we know that $\angle ABE = 60^\circ$.

Step 2: Therefore,

$$\begin{aligned}\angle PBC &= \angle ABC - \angle ABE \\ &= 90^\circ - 60^\circ = 30^\circ.\end{aligned}$$

Step 3: Since $AB = BC$, we know that $\triangle ABC$ is a right isosceles triangle and $\angle BAC = \angle BCA = 45^\circ$.

Step 4: Then, $\angle BCP = \angle BCA = 45^\circ$ and

$$\begin{aligned}\angle BPC &= 180^\circ - \angle PBC - \angle BCP \\ &= 180^\circ - 30^\circ - 45^\circ = \boxed{105^\circ}.\end{aligned}$$

Step 5: The answer is: $\boxed{105}$

**# Question:**

Find the *positive* real number(s) $x$ such that

$$\frac{1}{2}(3x^2 - 1) = (x^2 - 50x - 10)(x^2 + 25x + 5).$$

**# Reasoning:**

Step 1: Write $a = x^2 - 50x - 10$ and $b = x^2 + 25x + 5$.

Step 2: Then the equation given becomes

$$\frac{a + 2b - 1}{2} = ab,$$

so $0 = 2ab - a - 2b + 1 = (a - 1)(2b - 1)$.

Step 3: Then $a - 1 = x^2 - 50x - 11 = 0$ or $2b - 1 = 2x^2 + 50x + 9 = 0$.

Step 4: The former has a positive root, $x = \boxed{25 + 2\sqrt{159}}$, while the latter does not.

Step 5: The answer is: $\boxed{25 + 2\sqrt{159}}$

**# Question:**

{current question}

**# Reasoning:**

Table 13: Prompts used to sample reasoning paths on the Olympiadbench dataset.

**Prompt**

Please act as a math teacher and give step-by-step solutions to the user's questions. At the final step, a conclusive answer is given in the format of "The answer is: \boxed{¡ANSWER¿}.", where ¡ANSWER¿ should be a numeric answer.

**# Question:**
Let $T$ be a rational number. Compute $\sin^2 \frac{T\pi}{2} + \sin^2 \frac{(5-T)\pi}{2}$.
**# Reasoning:**
Step 1: Note that $\sin \frac{(5-T)\pi}{2} = \cos\left(\frac{\pi}{2} - \frac{(5-T)\pi}{2}\right) = \cos\left(\frac{T\pi}{2} - 2\pi\right) = \cos \frac{T\pi}{2}$.
Step 2: Thus the desired quantity is $\sin^2 \frac{T\pi}{2} + \cos^2 \frac{T\pi}{2} = \boxed{1}$.
Step 3: The answer is: $\boxed{1}$

**# Question:**
Let $T = 11$. Compute the value of $x$ that satisfies $\sqrt{20 + \sqrt{T + x}} = 5$.
**# Reasoning:**
Step 1: Squaring both sides gives $20 + \sqrt{T + x} = 25$, so $\sqrt{T + x} = 5$.
Step 2: Squaring again gives $T + x = 25$, so $x = 25 - T = 14$.
Step 3: The answer is: $\boxed{14}$

**# Question:**
The sum of the interior angles of an $n$-gon equals the sum of the interior angles of a pentagon plus the sum of the interior angles of an octagon. Compute $n$.
**# Reasoning:**
Step 1: The sum of interior angles of an $n$-gon is $180^\circ(n - 2)$.
Step 2: A pentagon has sum $180^\circ(5 - 2) = 540^\circ$, and an octagon has sum $180^\circ(8 - 2) = 1080^\circ$.
Step 3: So $180(n - 2) = 540 + 1080 = 1620$, hence $n - 2 = 9$, so $n = 11$.
Step 4: The answer is: $\boxed{11}$
**# Question:**
{current question}
**# Reasoning:**

Table 14: Prompts used to sample reasoning paths on the CSQA dataset.

**Prompt**

Please act as a commonsense teacher and solve the commonsense reasoning problem step by step.

**# Question:**
Google Maps and other highway and street GPS services have replaced what?
**# Options:**
(A) atlas (B) mexico (C) countryside (D) united states (E) oceans
**# Reasoning:**
Step 1: Electronic maps and GPS services are the modern version of paper atlas.
Step 2: In that case, the atlas have been replaced by Google Maps and other highway and street GPS services.
Step 3: The answer is: **A**

**# Question:**
You can share files with someone if you have a connection to a what?
**# Options:**
(A) freeway (B) radio (C) wires (D) computer network (E) electrical circuit
**# Reasoning:**
Step 1: Files usually can be stored in the computers.
Step 2: In that case, we can share them over the Internet.
Step 3: Thus, if we connect to a computer network, we can share the file with others.
Step 4: The answer is: **D**

**# Question:**
The fox walked from the city into the forest, what was it looking for?
**# Options:**
(A) pretty flowers (B) hen house (C) natural habitat (D) storybook (E) dense forest
**# Reasoning:**
Step 1: Since the fox walk from the city into the forest, he may looks for something in the forest but not in the city.
Step 2: From all of the options, the natural habitat are usually away from cities.
Step 3: The answer is: **C**
**# Question:**
{current question}
**# Options:**
{current options}
**# Reasoning:**

Table 15: Prompts used to sample reasoning paths on the SIQA dataset.

**Prompt**

Please act as a commonsense teacher and solve the commonsense reasoning problem step by step.

**# Question:**
Quinn wanted to help me clean my room up because it was so messy. What will Quinn want to do next?
**# Options:**
(A) Eat messy snacks (B) help out a friend (C) Pick up the dirty clothes
**# Reasoning:**
Step 1: Quinn wants to clean the room up.
Step 2: Picking up the dirty clothes is one way to clean the room.
Step 3: Thus, Quinn will want to pick up the dirty clothes next.
Step 4: The answer is: C

**# Question:**
Sydney had so much pent up emotion, they burst into tears at work. How would Sydney feel afterwards?
**# Options:**
(A) affected (B) like they released their tension (C) worse
**# Reasoning:**
Step 1: Crying is often a way to release tension.
Step 2: Sydney burst into tears at work.
Step 3: Thus, she would release the tension.
Step 4: The answer is: B

**# Question:**
Their cat kept trying to escape out of the window, so Jan placed an obstacle in the way. How would Jan feel afterwards?
**# Options:**
(A) scared of losing the cat (B) normal (C) relieved for fixing the problem
**# Reasoning:**
Step 1: The cat tried to escape so Jan needed to stop it to avoid losing the cat.
Step 2: Jan placed an obstacle in the way so the cat could not escape.
Step 3: The problem has been solved.
Step 4: Thus, Jan will feel relieved for fixing the problem.
Step 5: The answer is: C
**# Question:**
{current question}
**# Options:**
{current options}
**# Reasoning:**

Table 16: Prompts used to sample reasoning paths on the LogiQA dataset.

**Prompt**

Please act as a logical teacher and reason step by step to solve the logical reasoning problem.

**# Context:**
There are 90 patients with a disease T that is very difficult to treat and has taken the same routine drug. The patients were divided into two equal groups. The first group was given an experimental drug W, which is used to treat T, and the second group was given a placebo without W. Statistics ten years later showed that 44 people died in both groups, so the experimental drug was ineffective.
**# Question:**
Based on the above information, which of the following options, if correct, will best weaken the above argument?
**# Options:**
(A) Among the patients who died above, the average year of death in the second group was two years earlier than that in the first group. (B) Among the patients who died, the average life span of the second group was two years younger than that of the first group. (C) Among the above-mentioned living patients, the condition of the second group was more serious than that of the first group. (D) Among the above-mentioned living patients, those in the second group were older than those in the first group.
**# Reasoning:**
Step 1: Analyzing each option: A suggests drug W might extend life since the average death year in the drug W group is later than the placebo, directly challenging the drug's perceived ineffectiveness.
Step 2: B, similar to A, implies longer life in the drug W group but doesn't directly link to post-treatment lifespan.
Step 3: C indicates drug W may reduce disease severity but doesn't address lifespan or mortality, the main focus.
Step 4: D, about age differences, lacks direct relevance to drug effectiveness.
Step 5: Therefore, A most effectively weakens the argument against drug W's effectiveness.
Step 6: The answer is: A
**# Question:**
{current question}
**# Options:**
{current options}
**# Reasoning:**