

Rewarding Curse: Analyze and Mitigate Reward Modeling Issues for LLM Reasoning

Jiachun Li^{1,2}, Pengfei Cao^{1,2}, Yubo Chen^{1,2}, Jiexin Xu³, Huaijun Li³,
Xiaojian Jiang³, Kang Liu^{1,2}, Jun Zhao^{1,2}

¹School of Artificial Intelligence, University of Chinese Academy of Sciences

²The Key Laboratory of Cognition and Decision Intelligence for Complex Systems,
Institute of Automation, Chinese Academy of Sciences

³China Merchants Bank

{jiachun.li, pengfei.cao, yubo.chen, kliu, jzhao}@nlpr.ia.ac.cn

Abstract

Inference-time scaling techniques enable large language models (LLMs) to solve complex reasoning problems, in which the reward model (RM) plays an important role in selecting correct paths. In this paper, we present a systematic analysis of the RM’s performance in downstream reasoning tasks, which are overlooked in related works. We explore the discriminative abilities of RM from three perspectives and draw some meaningful conclusions: (1) Question difficulty: RM can impair LLM’s performance when solving simple problems; (2) Sampling number: RM struggles to distinguish between correct and incorrect responses as sampling increases and encounters low-frequency negatives; (3) Search diversity: RM requires a response distribution with moderate diversity to achieve optimal performance. Based on these findings, we design a new algorithm called Optimal Clustering Tree Search (OCTS) to mitigate these RM’s issues in the inference phase simultaneously. Experimental results demonstrate our method effectively enhances LLM’s reasoning capabilities, achieving up to a 3.2% improvement in accuracy.

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). In these works, the reward model (RM) plays a pivotal role in guiding large language models (LLMs) to search for the correct reasoning path in the inference phase (Wang et al., 2024b; Setlur et al., 2024; Zhang et al., 2024). When combined with strategies like Best-of-N (BoN) or Monte Carlo Tree Search (MCTS), the RM can recall the best response based on the LLM’s thorough exploration of the solution space, thereby improving the reasoning performance.

Due to the significance of reward models, a series of works have been proposed to study their performances, which provide mixed results (Lambert et al., 2024; Liu et al., 2024b; Zheng et al., 2024). On one hand, some works show that RM combined with advanced search strategies (e.g. MCTS) can effectively enhance the LLM’s reasoning capabilities (Setlur et al., 2024; Jiang et al., 2024; Wan et al., 2024). On the other hand, some works like DeepSeek-R1 point out that neural reward models suffer from reward hacking, which limits their effectiveness in enhancing LLM reasoning capabilities (DeepSeek-AI et al., 2024; DeepSeek-AI et al., 2025). These conflicting findings motivate the need for a systematic analysis of the reward model’s performance. However, related works mainly focus on evaluating the overall performance of the RM, without delving into how various factors influence these models in downstream reasoning tasks (Wang et al., 2024b; Zhang et al., 2024).

In this work, we focus on investigating the factors that influence the RM’s performance in math reasoning tasks and analyzing its existing issues. Specifically, we begin by mathematically modeling the RM-based inference process to identify the key factors in it, including questions, sampling number, and search parameters. Then, we conduct experiments centered around these factors to analyze their impact on the RM’s performance: (1) For the questions, we test the performance of BoN and MCTS across different question difficulty levels, demonstrating that the introduction of the RM primarily enhances the model’s reasoning ability on difficult problems, while it can impair the performance on simpler problems. (2) For the sampling number, we track the RM’s discriminative ability under different numbers and find that more samples make it increasingly difficult for the model to distinguish between positive and negative responses. Through statistical analysis, we attribute this issue to the RM’s inverse

long-tail phenomenon, where they tend to assign higher scores to incorrect responses that occur with low frequency. (3) For the search parameters, we primarily study the impact of certain parameters that control search diversity, including sampling temperature, tree structure, etc. We find that RM requires a response distribution with moderate diversity to achieve optimal performance. Excessive diversity generates high-quality negative examples, which lead to performance degradation.

Based on these findings, we design a novel RM-based inference algorithm called Optimal Clustering Tree Search (OCTS), which consists of three steps: exploration, selection and expansion. In exploration, we directly generate multiple responses like BoN, stopping the algorithm and returning the majority-voted answer if the question difficulty is low. This step mitigates the performance drop caused by the RM on simple problems. In selection, we cluster responses based on their answers and select the current optimal path by considering both frequency and reward scores, which mitigates the inverse long-tail issue. In expansion, we use the early steps of the selected path as prefixes and directly generate all of the remaining steps as new paths. This reduces the intermediate states of the tree, alleviating the performance loss caused by excessive diversity. We conduct extensive experiments to compare our method with other baselines. The results not only indicate our method is effective in improving RM-based reasoning abilities but also demonstrate the rationality of our former findings.

Our main contributions are as follows: (1) We systematically analyze the key factors influencing RM’s performance in downstream reasoning tasks under the guidance of mathematical modeling. (2) Through comprehensive experiments, we have three meaningful findings, including RM being harmful when facing simple questions, RM struggling to distinguish low-frequency negative samples, and RM performing worse on high-diversity distributions. (3) We propose the OCTS algorithm, which effectively enhances the RM-based reasoning performance of LLMs. Experimental results demonstrate that we can improve up to **3.2%** accuracy compared to other baselines.

2 Preliminaries

In this section, we first evaluate the performance of various RMs on the math reasoning task as a foundation for our subsequent analysis (§2.1). Then,

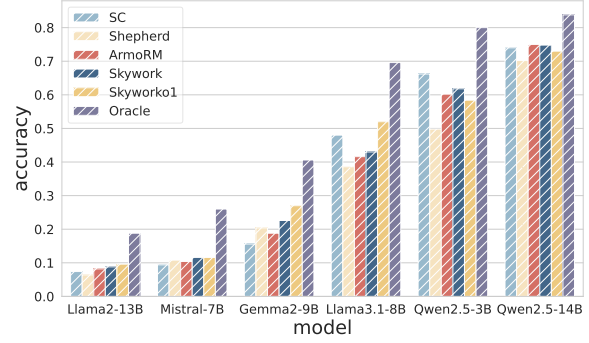


Figure 1: The performance of different policy models using various reward models for BoN inference on the MATH dataset ($N = 10$).

we mathematically model the RM-based inference process to help identify the key factors we study in this work (§2.2).

2.1 Overall Performance Experiment

A commonly used method to evaluate reward models is BoN, which generates multiple responses and selects the best one based on the reward score. The settings in this experiment are as follows:

Baselines In addition to BoN, we also set two baselines: SC and Oracle. For SC, we sample N responses and select the major voting answer. For Oracle, we assume the existence of an oracle selector, which directly selects the correct answer from the N samples generated. We regard SC as a measure of the LLM’s own reasoning capabilities, with Oracle serving as the ceiling of the performance.

Models For the policy model, we select some representative open-source models: Mistral-7B (Jiang et al., 2023), Gemma2-9B (Rivière et al., 2024a), Llama2-13B (Touvron et al., 2023), Llama3.1-8B (Rivière et al., 2024b), Qwen2.5-3B and Qwen2.5-14B (Yang et al., 2024). For the reward model, we evaluate some advanced models: Shepherd-Mistral-7B-PRM (Wang et al., 2024b), ArmoRM-Llama3-8B (Wang et al., 2024a), Skywork-Llama-3.1-8B (Liu et al., 2024a), Skywork-o1-PRM-Qwen-2.5-7B (o1 Team, 2024). These models encompass a variety of base models and demonstrate commendable performance (see Appendix A for details).

Datasets 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) as the dataset for our experiments, which consists of high-school competition-level

math problems. Unless otherwise specified, all experiments in this paper are conducted on it.

Main Results Figure 1 shows the main results of the evaluation (more results, including more datasets and the MCTS in Appendix B). We can conclude that: **Advanced reward models have limited performance on the downstream math reasoning task.** For most LLMs (except for Gemma2-9B), BoN only provides minor improvements over SC (<5%). On Qwen-2.5-3B, the BoN for all reward models exhibits lower accuracy than SC, indicating that the RM can even undermine the model’s inherent 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, **investigating factors that influence the RM’s performance and analyzing existing issues are crucial for enhancing LLM’s reasoning performance**, which is also the focal point of this paper.

2.2 Mathematical Modeling

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

$$\mathbb{R} = S(M(q), N; \Phi) \quad (1)$$

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

$$\hat{r} = \arg \max_{r \in \mathbb{R}} f(r) \quad (2)$$

There are various ways to define f . Since we focus on studying the RM’s performance, we set f as the score output by the reward model.

In this work, we primarily investigate factors that affect the performance of the reward model (i.e. f in Eq. 2). To this end, we modify the variables in Eq. 1 to observe the accuracy of predicted \hat{r} under different \mathbb{R} . Specifically, we conduct experiments on two representative search strategies S (i.e. BoN and MCTS), analyzing from three perspectives: the question q , the sampling number N , and the search parameters Φ .

3 Question Difficulty: RM is harmful when facing simple questions

In this section, we primarily investigate how different questions affect the reward model’s perfor-

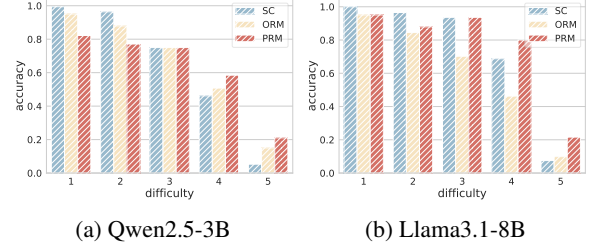


Figure 2: Performance of BoN inference across different question difficulty levels.

mance. To answer this question, we first model different questions using difficulty as a metric (§3.1). Then, we conduct experiments on the BoN (§3.2) and MCTS (§3.3) paradigms to analyze the influence of question difficulty.

3.1 Question Difficulty Modeling

Following the approach of former works, we use question difficulty as a metric to classify different questions (Lightman et al., 2024; Snell et al., 2024). Specifically, 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 only 0 or 1 time, the question is classified as the hardest level 5. Conversely, if the model answers correctly more than 8 times, the question is classified as the easiest level 1. Besides, we also study the difficulty approximation without the ground truth and report results in Appendix C.

3.2 BoN Performance across Difficulty

Experimental Setup We employ the BoN strategy to evaluate the RM’s reasoning performance across varying difficulty levels. Based on results in Figure 1, we select the best-performing Skywork and Skywork-o1 as the ORM (Outcome Reward Model) and PRM (Process Reward Model) for our subsequent experiments. We sample 32 examples from Qwen2.5-3B and Llama3.1-8B models for each question and compare the accuracy.¹

Main Results We illustrate the experimental results in Figure 2, from which we conclude that: Compared to SC, **BoN performs worse on simple questions but better on difficult questions.** From the easiest level 1 to the hardest level 5, the performance of SC gradually declines, while BoN’s

¹Unless otherwise specified, all subsequent experiments utilize Qwen2.5-3B as the primary policy model and Llama3.1-8B as the supplementary model.

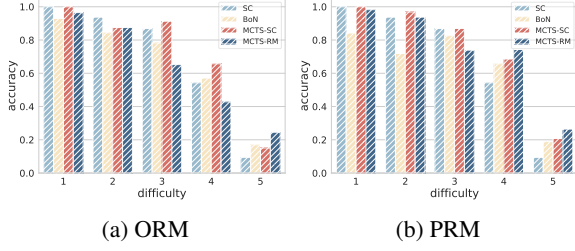


Figure 3: Performance of MCTS inference across different question difficulty levels.

accuracy transitions from lagging behind SC to surpassing it.

Experiments on More Datasets To validate the generalizability of our conclusion, we also repeat the experiment on two more math reasoning benchmarks: GSM8K (Cobbe et al., 2021) and Olympiad-Bench (He et al., 2024). We present the details and results of this experiment in Appendix D, which further confirm our above conclusion.

3.3 MCTS Performance across Difficulty

Can more advanced search strategies help to mitigate the above performance decrease on simple questions? To study this problem, we use the MCTS as the search strategy and compare performance across different difficulty levels.

Experimental Setup In MCTS, we use two different scoring functions f to select the final response: MCTS-SC and MCTS-RM (more functions in Appendix B). For the former, we employ a majority voting method to select the final answer. For the latter, we choose the path with the highest reward score as the final answer. We use ORM and PRM to evaluate the Q-value of final states and perform 32 rollouts over 200 questions.

Main Results Figure 3 shows the main results of this experiment. We can observe that, although MCTS provides a certain improvement over BoN, the accuracy of MCTS-RM still lags behind that of SC for low-difficulty problems (see level 1 and 2 in Figure 3a). In contrast, MCTS-SC achieves higher accuracy on easy questions but performs worse on harder questions compared to MCTS-RM. This indicates that: **(Cl.1) The introduction of the RM can hinder the LLM’s performance on simple problems but improve its effectiveness when addressing complex tasks.** This pattern is not limited to specific search strategies.

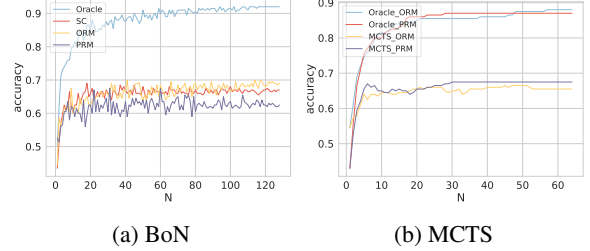


Figure 4: Two inference methods performance across difference sampling numbers.

4 Sampling Numbers: RM struggles to distinguish low-frequency negatives

In this section, we study the impact of the sampling numbers on the RM’s performance. To this end, we first investigate the overall performance of BoN and MCTS as the sampling numbers scales (§4.1). Next, we evaluate the reward model’s capability to differentiate between correct and incorrect responses under different N (§4.2). Finally, we conduct a statistical analysis of the negative samples to explain the reasons behind our findings (§4.3).

4.1 Overall Performance

Experimental Setup Recent works (Brown et al., 2024) demonstrate the LLM’s coverage of correct answers (i.e. the Oracle in §2.1) increases as the sampling number grows, whereas the accuracy does not fully scale with N . Based on it, we further investigate whether introducing better RMs and the MCTS strategy can reduce the gap between coverage and accuracy. Specifically, for BoN, we vary N from 1 to 128, while for MCTS, we set N from 1 to 64. For the implementation of MCTS, we utilize the RM’s score as the scoring function f (i.e. MCTS-RM in §3.3), which will serve as the default in the subsequent experiments.

Main Results The changes in accuracy and coverage are shown in Figure 4. From the results, we can conclude that: **(1) The performance of BoN inference can not be scaled with N .** For both PRM and ORM, the accuracy of BoN inference plateaus beyond approximately 50 samples in Figure 7a. As N increases, the gap between their accuracy and coverage becomes progressively larger. **(2) The accuracy of MCTS also cannot scale continuously with the sampling number.** As shown in Figure 7b, the accuracy of MCTS also plateaus before reaching a relatively small N . In contrast, the Oracle setting consistently increases, leading to

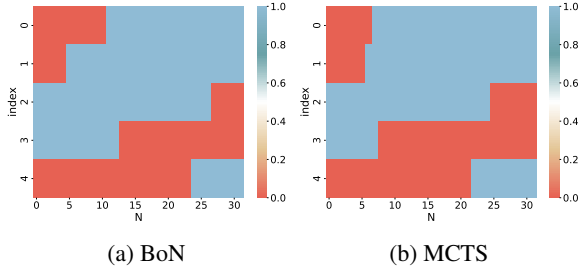


Figure 5: The variation in question answering correctness as the sampling number changes. Blue indicates a correct answer, while red indicates an incorrect answer.

a persistently widening gap between accuracy and coverage.

4.2 Discriminative Experiments on RM

In the context of increasing coverage, whether the accuracy improves or not is predominantly contingent upon the discriminative capacity of the reward model. Thus, we design experiments to observe the performance of the RM in distinguishing positive and negative instances as N increases.

Case Analysis 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 5, 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 5), RM assigns the highest score to newly generated incorrect responses, transforming the originally correct answers into incorrect ones.

Statistical Analysis We further record the number of times the answer changes from correct to incorrect and report the results in Figure 6. All of the methods tend to make more incorrect transitions as N increases. **This demonstrates that the model tends to make more wrong differentiation under a higher sampling number.** Besides, compared to SC, RM-based methods perform worse (have higher counts in Figure 6), which highlights the introduction of reward models leads to more wrong selection.

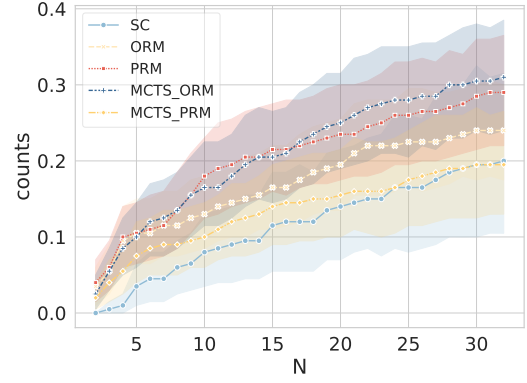


Figure 6: The number of times the model's responses change from correct to incorrect as N changes.

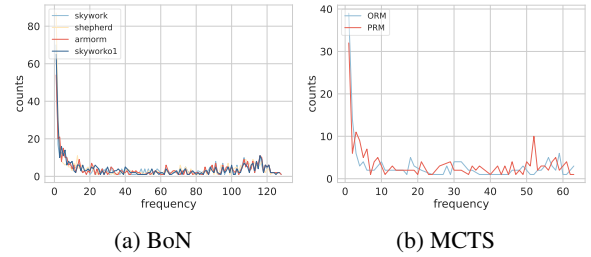


Figure 7: Frequency statistics of the highest-scored negative samples.

4.3 Negative Examples Analysis

Hypothesis on the Issue 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). The limited number of samples covers only a restricted response distribution. Therefore, we hypothesize that: As N increases, a significant number of low-frequency samples, which deviate from the distribution of the training data, are generated. The reward model struggles to generalize to these unseen samples, resulting in higher scores being assigned to incorrect responses within them.

Experimental Setup To validate our hypothesis, we conduct a statistical analysis of negative examples. For each question, We select the incorrect response with the highest RM score and then calculate the frequency of its answer across all samples. Then, we analyze the distribution of this frequency across 500 questions.

Main Results The results depicted in Figure 7 illustrate that the RM exhibits an **inverse long-tail phenomenon** during scoring wrong responses. On

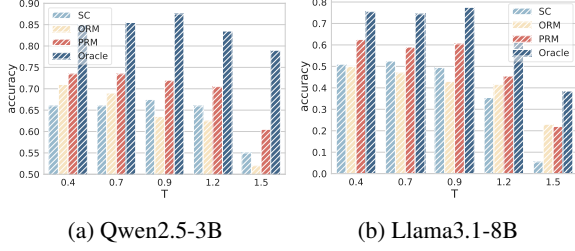


Figure 8: Performance of BoN inference across different sampling temperatures.

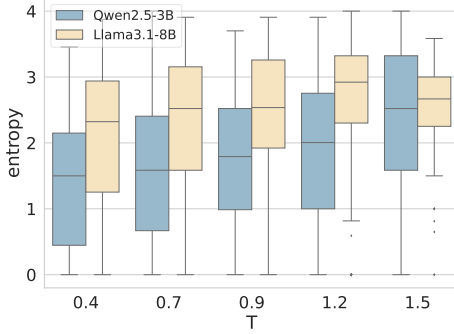


Figure 9: Information entropy of incorrect answers under different sampling temperatures.

the vast majority of questions, incorrect answers with low occurrence frequencies (*frequency* < 5 in Figure 7) receive the highest scores among all negative examples. Conversely, incorrect answers with high occurrence frequencies rarely achieved the highest scores (*counts* < 10 in Figure 7). The results effectively substantiate our earlier hypothesis: **(C1.2) Due to the limited distribution covered by the RM’s paired training data, it struggles to correctly generalize to incorrect responses with low occurrence frequencies, making it increasingly difficult to distinguish incorrect responses from correct ones as N grows.**

5 Search Diversity: RM performs worse on high-diversity distributions

The final influencing factor we investigate is the search parameters Φ . It encompasses various parameters within the search strategy, which are primarily utilized to control the diversity of the policy model’s search. In BoN inference, we study the influence of the temperature T on RM’s performance (§5.1). Then, in MCTS, we mainly investigate the impact of tree structures on RM (§5.2).

5.1 Search Diversity in BoN

Overall Performance When we directly sample responses from LLMs (like SC and BoN), the tem-

perature T is a key parameter to control the search diversity. Here, we traverse different temperatures to study the performance variations of BoN and demonstrate the results in Figure 8. For both policy models, as the temperature increases, the performance of BoN continues to decline. In contrast, SC and Oracle (i.e. coverage) exhibit more stable performance, showing a declining trend only when the sampling temperature exceeds a relatively high threshold ($T > 0.9$ in Figure 8). This indicates that RM is more sensitive to sampling diversity compared to the policy model. **Higher diversity makes it challenging for the RM to distinguish between positive and negative responses.**

Statistical Analysis 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 9. 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 C1.2 in §4.3, 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.

5.2 Search Diversity in MCTS

Experimental Setup In the MCTS algorithm, the main parameter governing search diversity is the tree structure, which is defined by two key parameters: width and maximum depth. The width refers to the number of child nodes at each node. A larger width corresponds to a broader search space in the exploration process. The depth represents the length of the longest path from the root node to the leaf node. A greater depth suggests that the model can explore more intermediate states along a single branch. We set the default width and maximum depth to 5 and compare the reasoning performance of MCTS under different parameters.

Main Results The results of the comparison are illustrated in Figure 10. From the figure, we can conclude that: (1) For width, the best performance is observed at intermediate values (width = 5), too

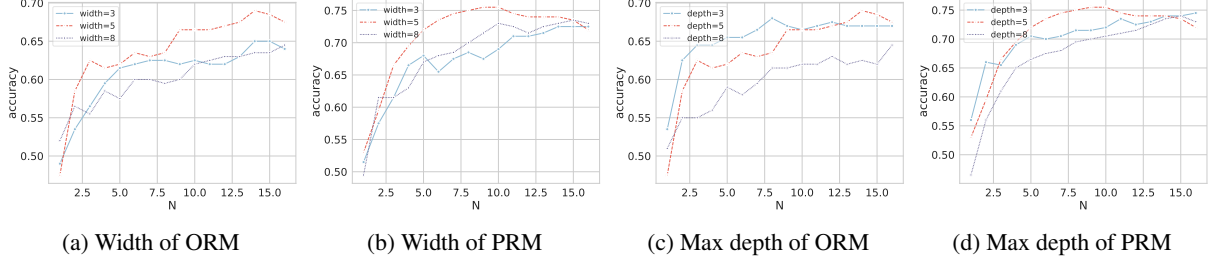


Figure 10: MCTS inference performance under different tree structures.

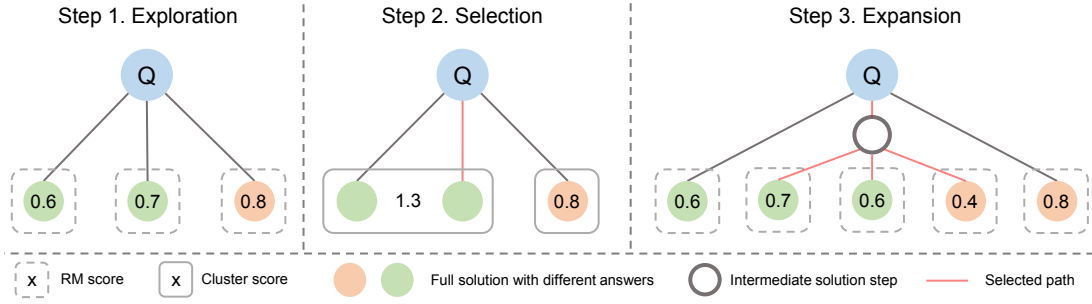


Figure 11: Main process of our OCTS algorithm.

high widths lead to a decline in performance. This indicates that MCTS should prevent the majority of rollouts from being used solely for exploration. (2) For max depth, the best performance is achieved under settings with a lower value (i.e., max depth = 3 or 5). This suggests that, for MCTS, an excessive number of intermediate states (max depth > 5) can impair performance. In practical searches, the optimal number of intermediate states does not align with the steps a human would take to solve the problem. To sum up, excessive diversity from both width and depth can degrade the RM’s performance. Thus, we have: **(CI.3) Regardless of the search strategy, it is essential to limit the diversity of the sampling distribution to maintain the optimal performance of the RM.** We also study the diversity of MCTS based on the explore weight, and the results are consistent (see Appendix E).

6 Application: OCTS boosts RM-based reasoning performance

6.1 Our Method

In the preceding sections, we uncover key patterns that affect the RM’s performance and identify serial issues in RM-based reasoning. To mitigate these issues, we propose a novel RM-based inference algorithm called **Optimal Clustering Tree Search (OCTS)**, as an application of our findings. Figure 11 demonstrates the main process of our algorithm, which comprises three key steps:

Exploration We initially employ the BoN strategy for exploration, directly generating complete reasoning paths. If the number of answers is below a certain threshold, it indicates that the question is simple (see Appendix C for details). In such case, the algorithm stops directly, selecting the path with the highest score from the majority-voted answers as the final output. This not only reduces the additional inference costs but also avoids the RM’s detrimental impact on the reasoning performance for simple questions. Thus, this early stopping mechanism can mitigate the issue in CI.1.

Selection In this step, we cluster all paths based on their answers and sum the reward scores of all paths within the same cluster to derive a cluster score. For example, in Step 2 of Figure 11, we have two clusters, one with a score of 1.3 ($0.6 + 0.7 = 1.3$) and the other with a score of 0.8. Then, using these cluster scores as the criterion, we select the top-k clusters and choose the best path from each of them. Therefore, we can use answer frequency as another key factor in path scoring to reduce the scores of low-frequency negative examples, mitigating the issue in CI.2.

Expansion After selecting k paths, we expand the tree based on them. Specifically, in the n-th round, we use the first n solution steps of each selected path as the intermediate states (see intermediate solution step in Step 3 of Figure 11) to

Methods	GSM8k		MATH	
	ORM	PRM	ORM	PRM
CoT	78.2	78.2	46.2	46.2
Self-Consistency	82.8	82.8	64.2	64.2
Best-of-N	83.0	86.8	64.6	61.2
BoN Weighted	83.4	86.2	66.6	59.8
MCTS	91.8	94.8	66.6	70.6
Beam Search	-	94.6	-	72.6
Ours	90.8	95.8	69.8	75.6

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

generate new children nodes. All remaining steps are generated directly to reduce the intermediate nodes in the tree. By controlling the complexity of the tree, we can reduce the negative impact of excessive diversity on the RM, as described in C1.3.

We repeat steps 2 and 3 until the tree reaches the maximum depth and select the best path in the last round as the final output.

6.2 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). For models, we 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 of the implementation in Appendix F.

Main Results We demonstrate the result on Qwen-2.5-3B in Table 1 (more results, including more datasets, more models, and the ablation study are presented in Appendix G), from which we can get the following conclusions: **(1) Our OCTS method enables more effective utilization of the reward model in reasoning tasks.** Compared to other RM-based inference methods, it effectively improves reasoning accuracy across different datasets and models, with an improvement of up to **3.2%**. **(2) The findings drawn from the former analysis are reasonable.** Given that our method is an application derived from the analytical conclusions, its superior performance can further substan-

tiate the correctness of our earlier conclusions.

7 Related Works

Inference-time Scaling Technique The emergence of o1-like models has demonstrated that inference-time scaling effectively enhances LLM’s reasoning abilities (OpenAI, 2024; DeepSeek-AI et al., 2025; Zeng et al., 2024; Zhao et al., 2024). Existing work primarily follows 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 (e.g. sampling numbers, search diversity) 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., 2024b; 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.

8 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. After analyzing these issues, we design a new inference algorithm called OCTS to mitigate them. Experimental results demonstrate that our method is effective in enhancing RM-based reasoning capabilities.

Limitations

Although our work conducts an in-depth analysis and mitigating in RM-based reasoning, it has several limitations. Firstly, we refrain from analyzing the problem of reward models in more reasoning tasks such as commonsense and logic because the LLM’s performance on these tasks is already sufficiently strong. Secondly, in this work, we focus solely on investigating the issues present during the inference phase, while potential problems that may arise during the training of RM are not explored. This is because the training process involves significant randomness and a large number of parameters, making it challenging to pinpoint the key factors that affect RM’s performance. We leave these limitations as our future work to explore.

References

- Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. [Critique-out-loud reward models](#). *CoRR*, abs/2408.11791.
- Zheni Bi, Kai Han, Chuanjian Liu, Yehui Tang, and Yunhe Wang. 2024. [Forest-of-thought: Scaling test-time compute for enhancing LLM reasoning](#). *CoRR*, abs/2412.09078.
- Bradley C. A. Brown, Jordan Juravsky, Ryan Saul Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. 2024. [Large language monkeys: Scaling inference compute with repeated sampling](#). *CoRR*, abs/2407.21787.
- 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. 2021. [Training verifiers to solve math word problems](#). *CoRR*, 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, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaoqun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. [Deepseek-v3 technical report](#). *CoRR*, abs/2412.19437.
- Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Meng Li, Miaoqun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *Preprint*, arXiv:2501.12948.
- DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, 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, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaoqun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. 2024. [Deepseek-v3 technical report](#). *CoRR*, abs/2412.19437.

- Shibo Hao, Yi Gu, Haodi Ma, Joshua Jiahua Hong, Zhen Wang, Daisy Zhe Wang, and Zhiting Hu. 2023. [Reasoning with language model is planning with world model](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 8154–8173. Association for Computational Linguistics.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, ACL 2024, Bangkok, Thailand, August 11-16, 2024, pages 3828–3850. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. 2023. [Mistral 7b](#). *CoRR*, abs/2310.06825.
- Jinhao Jiang, Zhipeng Chen, Yingqian Min, Jie Chen, Xiaoxue Cheng, Jiapeng Wang, Yiru Tang, Haoxiang Sun, Jia Deng, Wayne Xin Zhao, Zheng Liu, Dong Yan, Jian Xie, Zhongyuan Wang, and Ji-Rong Wen. 2024. [Technical report: Enhancing LLM reasoning with reward-guided tree search](#). *CoRR*, abs/2411.11694.
- 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. 2024. [Rewardbench: Evaluating reward models for language modeling](#). *CoRR*, abs/2403.13787.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program induction by rationale generation: Learning to solve and explain algebraic word problems](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 158–167. Association for Computational Linguistics.
- Chris Yuhao Liu, Liang Zeng, Jiakai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024a. [Skywork-reward: Bag of tricks for reward modeling in llms](#). *CoRR*, abs/2410.18451.
- Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024b. [Rm-bench: Benchmarking reward models of language models with subtlety and style](#). *CoRR*, abs/2410.16184.
- Xingzhou Lou, Dong Yan, Wei Shen, Yuzi Yan, Jian Xie, and Junge Zhang. 2024. [Uncertainty-aware reward model: Teaching reward models to know what is unknown](#). *CoRR*, abs/2410.00847.
- Skywork o1 Team. 2024. [Skywork-o1 open series](#). <https://huggingface.co/Skywork>.
- OpenAI. 2024. [Introducing openai o1 preview](#). Accessed: 2025-01-24.
- Zhenting Qi, Mingyuan Ma, Jiahang Xu, Li Lyna Zhang, Fan Yang, and Mao Yang. 2024. [Mutual reasoning makes smaller llms stronger problem-solvers](#). *CoRR*, abs/2408.06195.
- Morgane Rivi  re, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L  onard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram  , 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 McDonnell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sj  sund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024a. [Gemma 2: Improving open language models at a practical size](#). *CoRR*, abs/2408.00118.

- Morgane Rivi re, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, L onard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ram , 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 Eltyshov, 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 McDonnell, Kelvin Nguyen, Kiranbir Sodhia, Kish Greene, Lars Lowe Sj sund, Lauren Usui, Laurent Sifre, Lena Heuermann, Leticia Lago, and Lilly McNealus. 2024b. [Gemma 2: Improving open language models at a practical size](#). *CoRR*, abs/2408.00118.
- Keisuke Sakaguchi, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2020. [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*, pages 8732–8740. AAAI Press.
- Abulhair Saparov and He He. 2023. [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.
- Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. 2024. [Rewarding progress: Scaling automated process verifiers for LLM reasoning](#). *CoRR*, abs/2410.08146.
- Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. [Scaling LLM test-time compute optimally can be more effective than scaling model parameters](#). *CoRR*, abs/2408.03314.
- Oyvind Tafjord, Bhavana Dalvi, and Peter Clark. 2021. [Proofwriter: Generating implications, proofs, and abductive statements over natural language](#). In *Findings of the Association for Computational Linguistics: ACL/IJCNLP 2021, Online Event, August 1-6, 2021*, volume ACL/IJCNLP 2021 of *Findings of ACL*, pages 3621–3634. Association for Computational Linguistics.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [Commonsenseqa: A question answering challenge targeting commonsense knowledge](#). In *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)*, pages 4149–4158. Association for Computational Linguistics.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aur lien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023. [Llama 2: Open foundation and fine-tuned chat models](#). *CoRR*, abs/2307.09288.
- Ziyu Wan, Xidong Feng, Muning Wen, Stephen Marcus McAleer, Ying Wen, Weinan Zhang, and Jun Wang. 2024. [Alphazero-like tree-search can guide large language model decoding and training](#). In *Forty-first International Conference on Machine Learning, ICLR 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.
- Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. 2024a. [Interpretable preferences via multi-objective reward modeling and mixture-of-experts](#). In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 10582–10592. Association for Computational Linguistics.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024b. [Math-shepherd: Verify and reinforce llms step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16,*

2024, pages 9426–9439. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. [Self-consistency improves chain of thought reasoning in language models](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *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*.

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. 2024. [Qwen2.5 technical report](#). *CoRR*, abs/2412.15115.

Zhiyuan Zeng, Qinyuan Cheng, Zhangyue Yin, Bo Wang, Shimin Li, Yunhua Zhou, Qipeng Guo, Xuanjing Huang, and Xipeng Qiu. 2024. [Scaling of search and learning: A roadmap to reproduce o1 from reinforcement learning perspective](#). *CoRR*, abs/2412.14135.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. [Generative verifiers: Reward modeling as next-token prediction](#). *CoRR*, abs/2408.15240.

Yu Zhao, Huifeng Yin, Bo Zeng, Hao Wang, Tianqi Shi, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2024. [Marco-o1: Towards open reasoning models for open-ended solutions](#). *CoRR*, abs/2411.14405.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. [Processbench: Identifying process errors in mathematical reasoning](#). *CoRR*, abs/2412.06559.

A Performance of Selected RMs

To demonstrate that the RM issues identified in our experiments in Section §2.1 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 2. 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 3. From them, we can get that the performance of these models on relevant benchmarks is comparable to the advanced LLMs (e.g. gpt4), hence they are representative.

B 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 12. 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 §2.1. (2) In Skywork and Skywork01, 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 and He, 2023). The results are demonstrated in Figure 13, 14, 15, 16 and 17. Lastly, we only use discriminative RM in the main text. All of these results are consistent with the conclusion in the main text.

C 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 18, 19 and 20. 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 OCTS method in §5.1.

D 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 4, 5 and 6. We can observe that the issues identified in C1.1 are prevalent across various reasoning datasets.

E 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 21. 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 C1.3 in our main text.

F Implementation Details in the Main Experiments

Here we provide a detailed account of the implementation specifics from the main experiments in §6. 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, in Step 1, we generate 16 samples with a temperature setting of 0.7 and stop the algorithm if the answer count is less than 2. In Step 2, for ORM, we select the top-1 path, for PRM, we select the top-2 paths. In Step 3, we set the width (i.e. the children numbers for each node) to 8 for ORM, 4 for PRM, and the max depth to 3. We release all of the prompts we use in the accompanying software package.

G Additional Results of the Main Experiments

Results on More Datasets To test whether our method remains effective on more challenging tasks, we further compared the performance of our method with other baselines on the Olympiad-Bench dataset (He et al., 2024) in Figure 22. The results demonstrate the effectiveness of OCTS.

Results on More Models We repeat the main experiments on the Llama3.1-8B model and report the result in Table 7 (here we sample 200 questions from each dataset). Our method also demonstrates the best performance on this model across three different tasks.

Ablation Study To verify the effectiveness of each step, here we conducted ablation experiments on a combination of Qwen2.5-3B + ORM, using 200 samples each from GSM8k and MATH. The experimental settings are as follows:

- **-Exploration:** Disable the early stopping mechanism in Step 1;
- **-Selection:** Eliminate the clustering operation and use the score of each path instead of cluster scores for selection (similar to MCTS);
- **-Expansion:** Cancel the operation of directly generating the remaining steps, and instead generate intermediate nodes layer by layer (similar to MCTS and Beam).

Table 8 shows the result of the ablation study. Removing each component leads to a decline in performance. Specifically, although removing exploration 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 2: 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 3: Comparison of RM’s performance on ProcessBench.

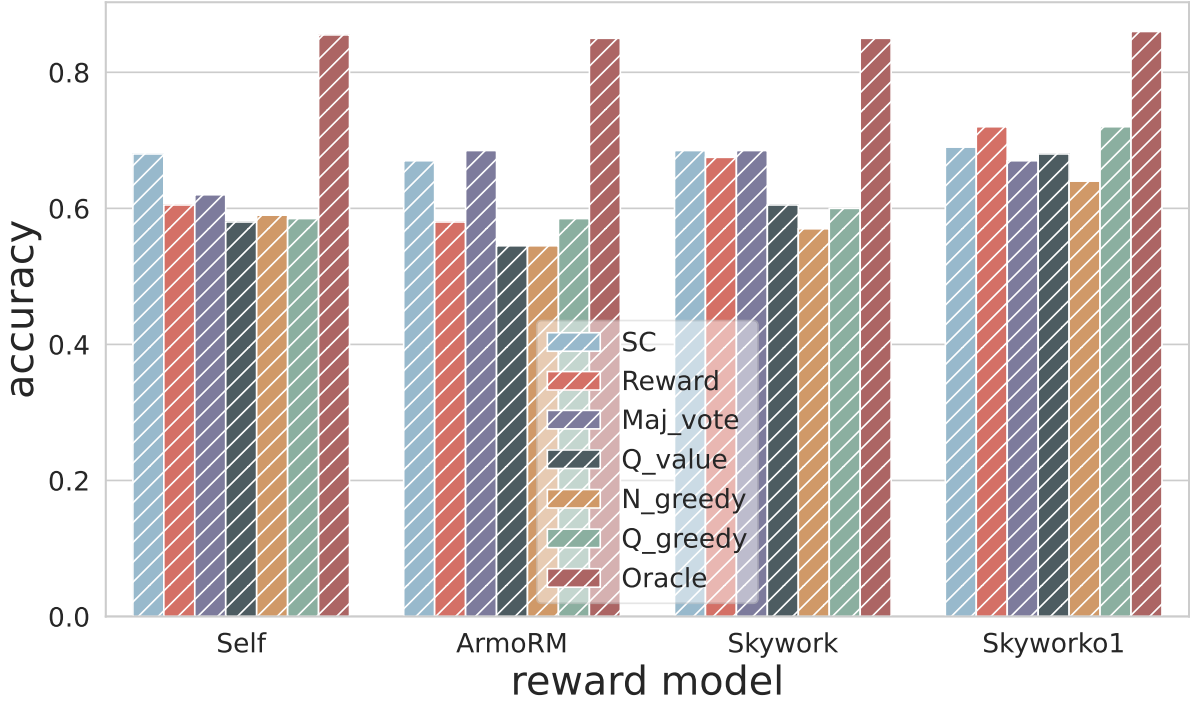


Figure 12: The performance of different reward models using the MCTS inference on the MATH dataset ($N = 16$, Qwen-2.5-3B).

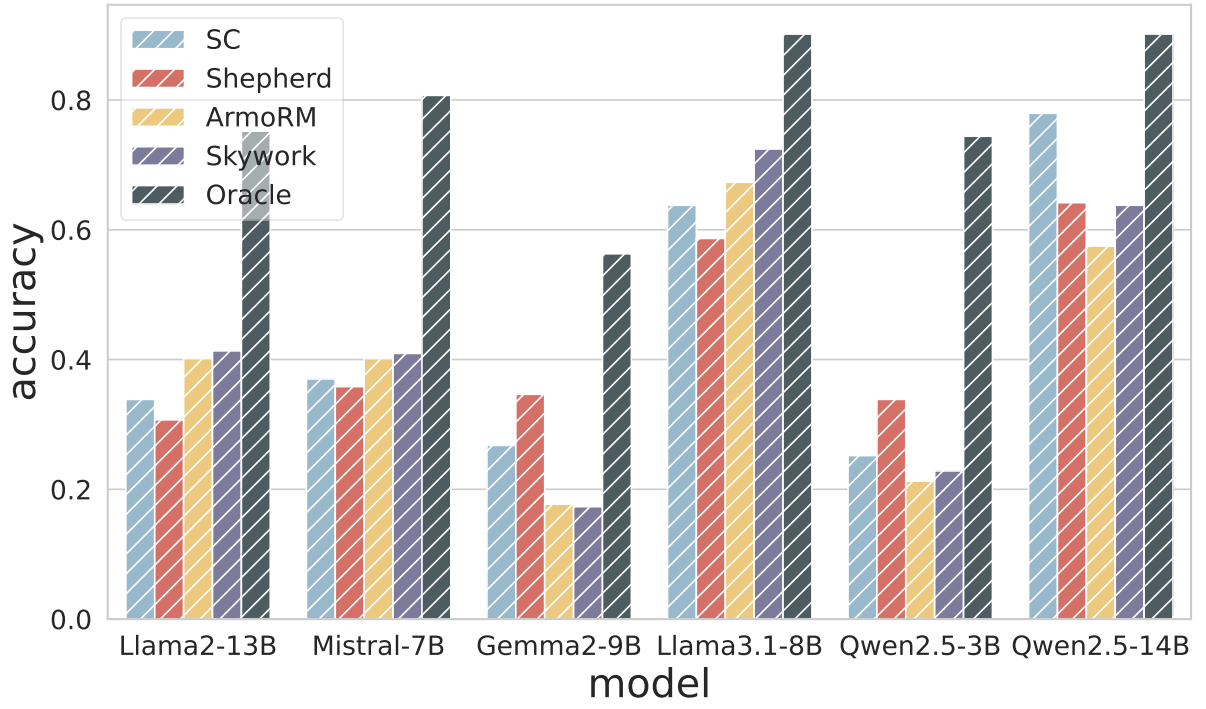


Figure 13: The performance of different policy models using various reward models for BoN inference on the AQuA dataset ($N = 10$).

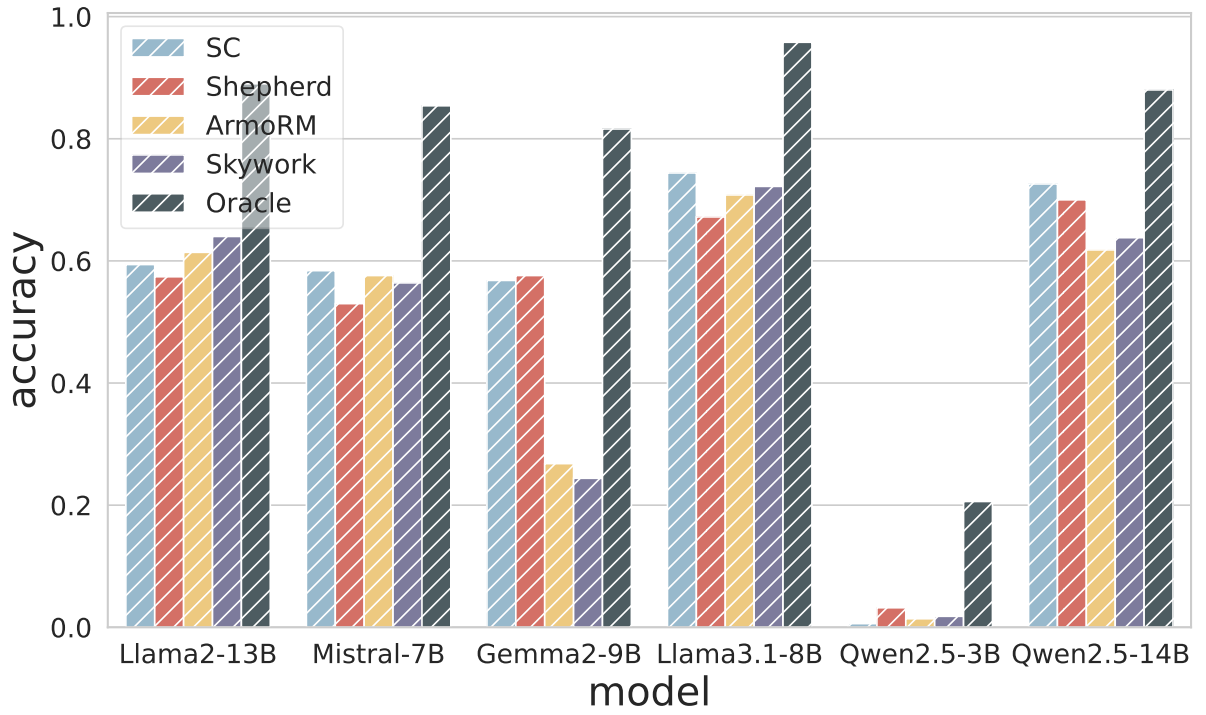


Figure 14: The performance of different policy models using various reward models for BoN inference on the WinoGrande dataset ($N = 10$).

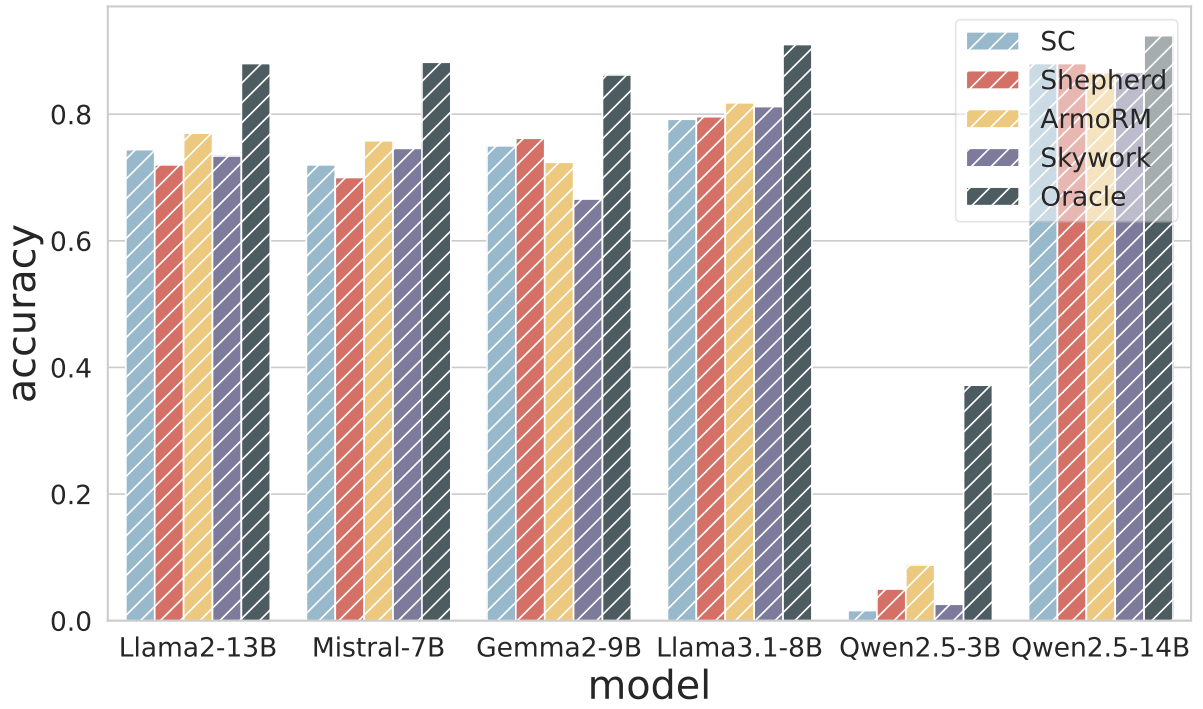


Figure 15: The performance of different policy models using various reward models for BoN inference on the CSQA dataset ($N = 10$).

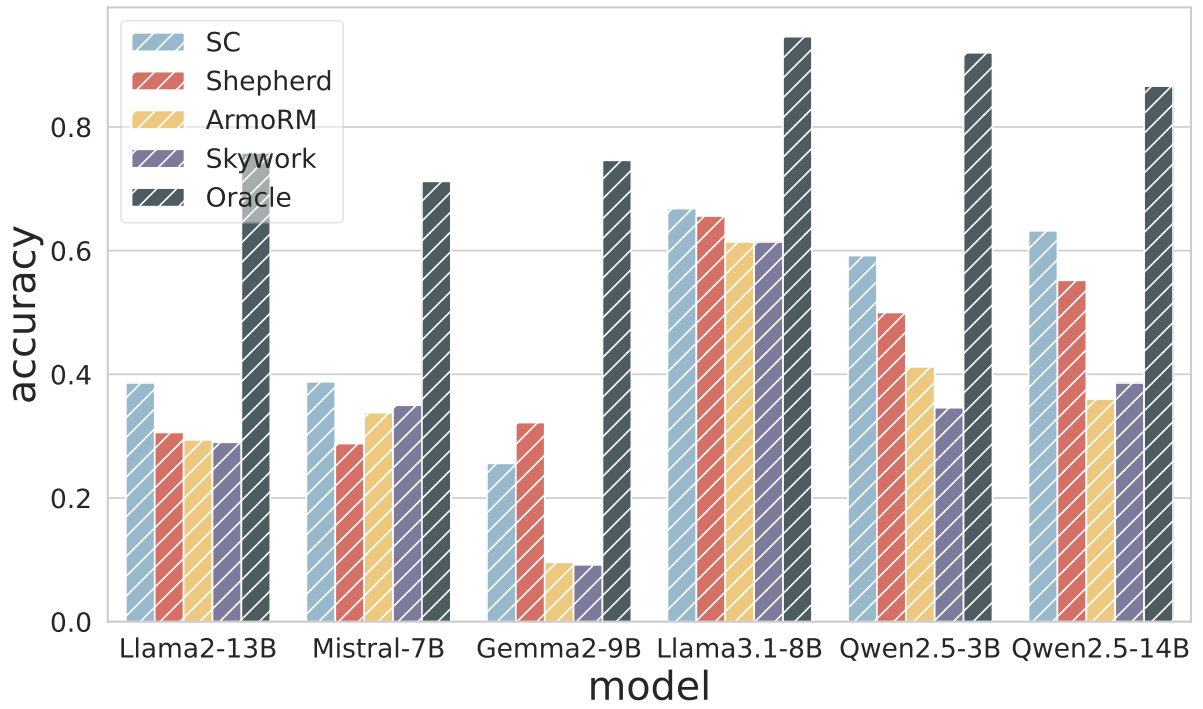


Figure 16: The performance of different policy models using various reward models for BoN inference on the ProofWriter dataset ($N = 10$).

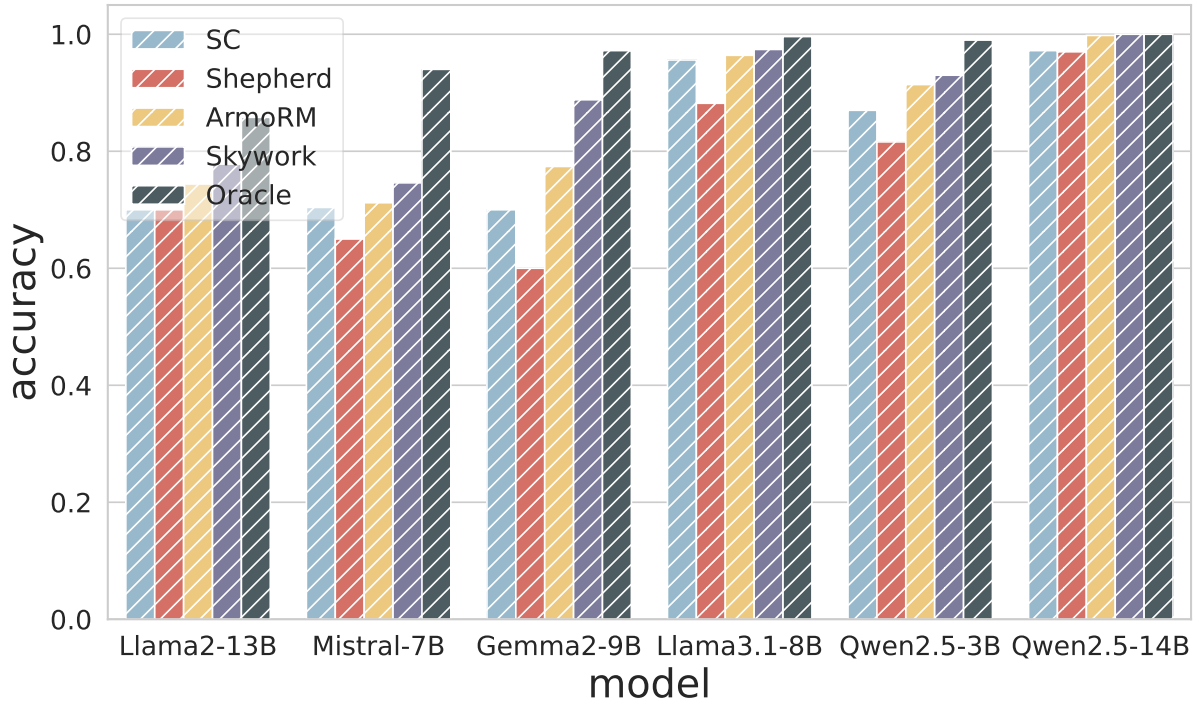


Figure 17: The performance of different policy models using various reward models for BoN inference on the ProntoQA dataset ($N = 10$).

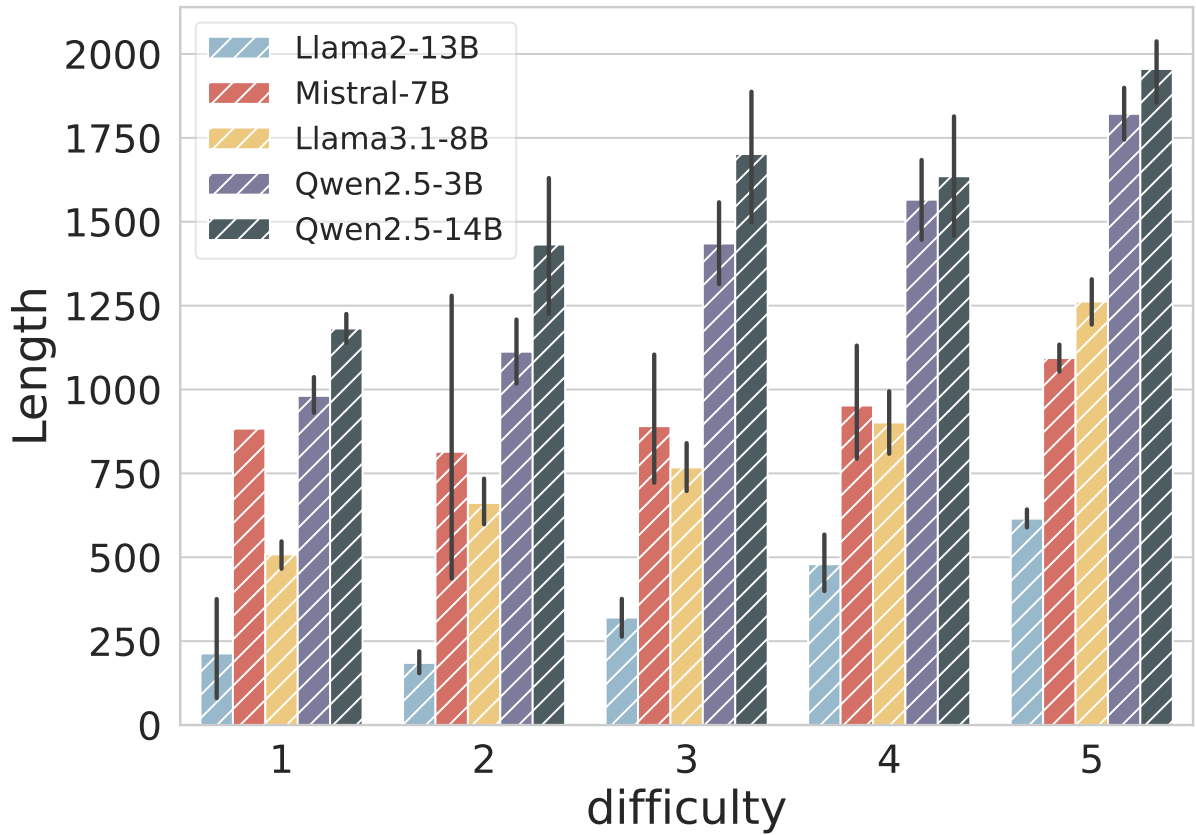


Figure 18: The performance of different policy models using various reward models for BoN inference on the ProntoQA dataset ($N = 10$).

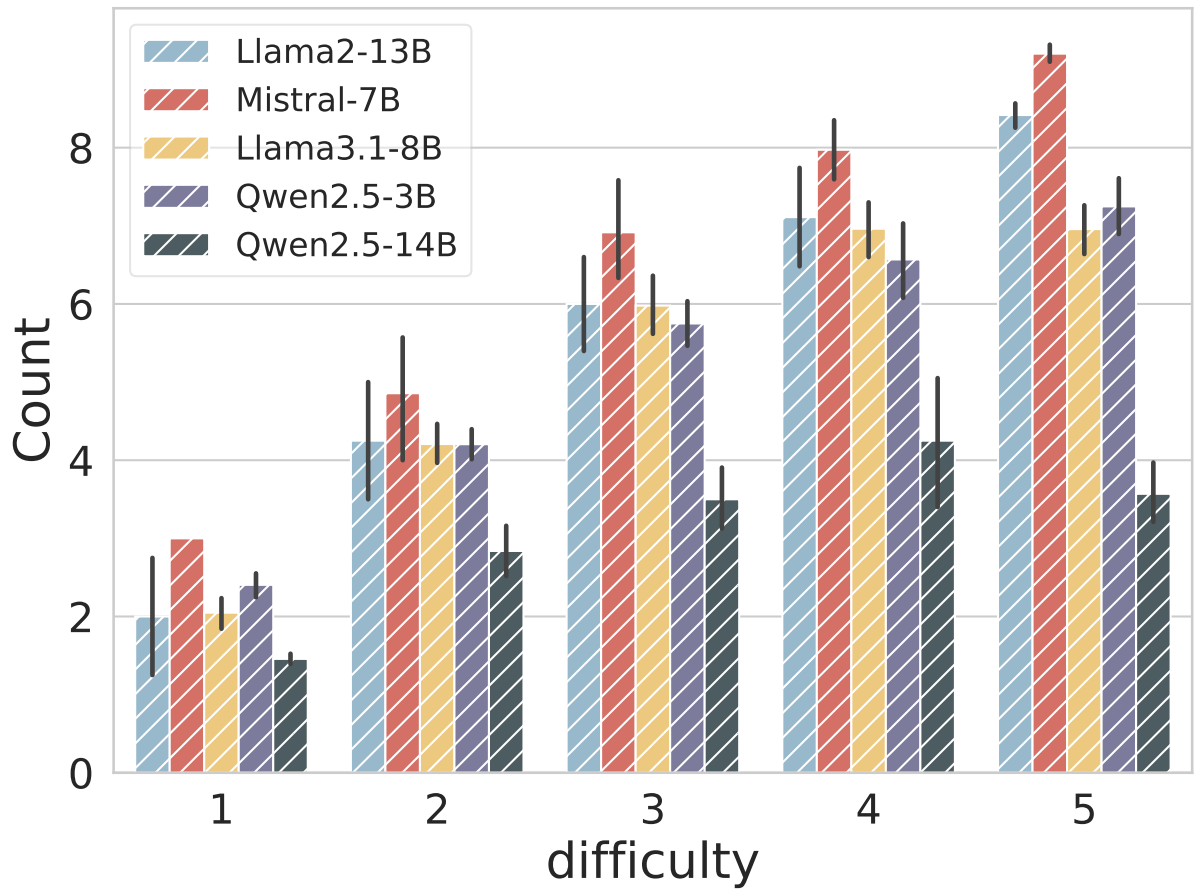


Figure 19: The performance of different policy models using various reward models for BoN inference on the ProntoQA dataset ($N = 10$).

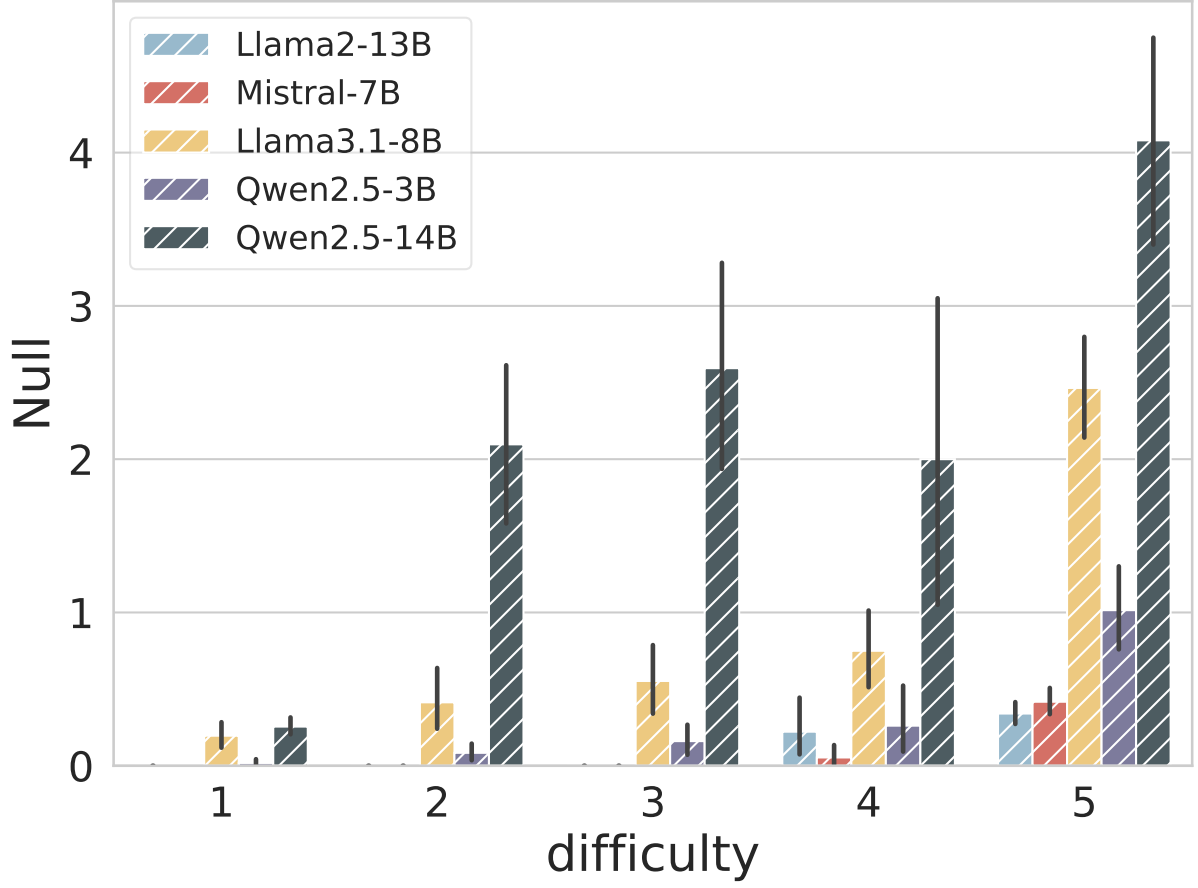


Figure 20: The performance of different policy models using various reward models for BoN inference on the ProntoQA dataset ($N = 10$).

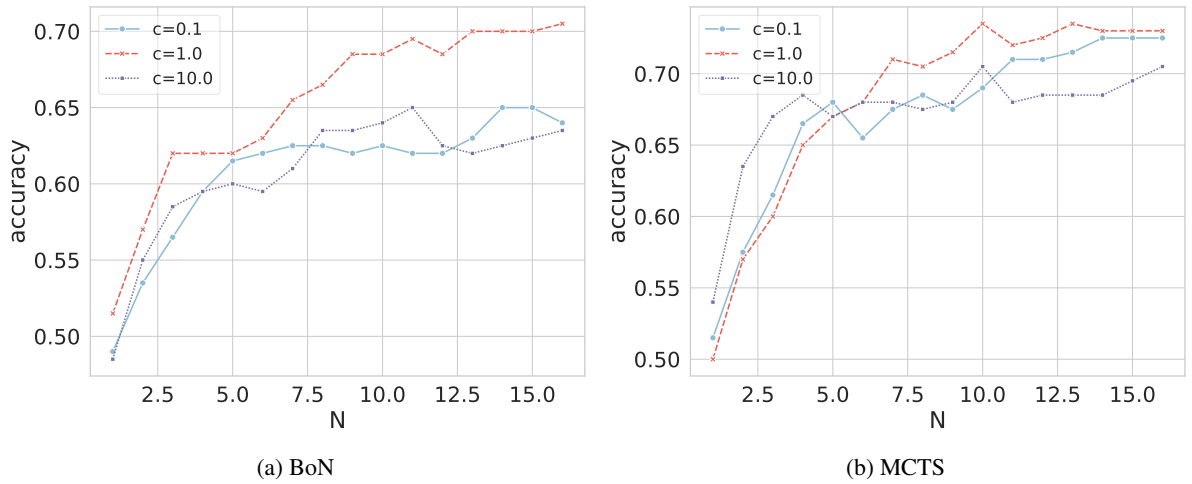


Figure 21: Performance comparison across different explore weight c on Qwen2.5-3B.

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 4: 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)	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 5: 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(@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 6: Comparison of performance across different difficulty levels on 200 samples of OlympiadBench (Qwen2.5-3B).

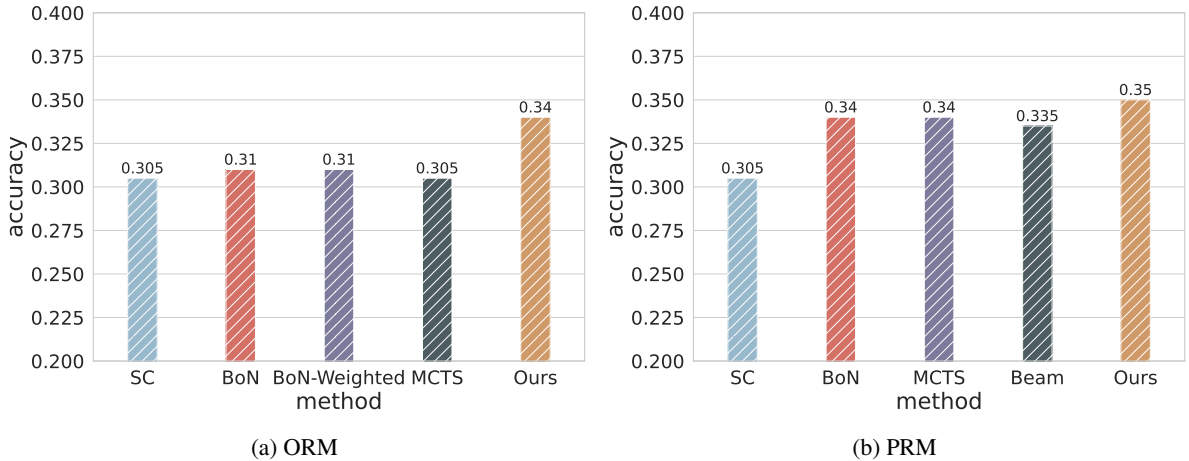


Figure 22: Performance comparison on OlympiadBench (200 samples).

Methods		GSM8k	MATH	OlympiadBench
CoT		84.5	38.0	10.5
Self-Consistency		91.0	56.5	16.0
Best-of-N	+ ORM	90.5	47.0	17.6
	+ PRM	95.0	62.0	23.0
BoN Weighted	+ ORM	88.5	52.5	19.8
	+ PRM	94.0	61.5	24.0
MCTS	+ ORM	90.0	43.0	12.5
	+ PRM	95.0	57.0	19.0
Beam Search		94.0	55.5	14.5
Ours	+ ORM	88.5	49.0	18.0
	+ PRM	95.0	67.0	26.0

Table 7: Performance comparison on Llama3.1-8B, the best results are highlighted in **bold**.

Methods	GSM8k	MATH
Ours	91.5	72.5
-Exploration	91.0	72.5
-Selection	89.5	65.5
-Expansion	87.5	64.0

Table 8: Results of the ablation Study.