The Lessons of Developing Process Reward Models in Mathematical Reasoning

Anonymous ACL submission

Abstract

Process Reward Models (PRMs) aim to identify and mitigate intermediate errors in the reasoning processes in mathematical reasoning of Large Language Models (LLMs). However, the development of effective PRMs faces significant challenges, particularly in data annotation and evaluation methodologies. In this paper, through extensive experiments, we demonstrate that commonly used Monte Carlo (MC) estimation-based data synthesis for PRMs typically yields inferior performance and generalization compared to LLM-as-a-judge and human annotation methods. Furthermore, we identify potential biases in conventional Best-of-N (BoN) evaluation strategies for PRMs. To address these challenges, we develop a consensus filtering mechanism that effectively integrates MC estimation with LLM-as-a-judge and advocates a more comprehensive evaluation framework that combines response-level and step-level metrics. Based on the mechanisms, we significantly improve both model performance and data efficiency in the BoN evaluation and the step-wise error identification task. Finally, we release a new state-of-the-art PRM that outperforms existing open-source alternatives and provides practical guidelines for future research.

1 Introduction

In recent years, Large Language Models (LLMs) have made remarkable advances in mathematical reasoning (OpenAI, 2023; Dubey et al., 2024; Shao et al., 2024; Zhu et al., 2024; Yang et al., 2024a,c,b), yet they can make mistakes, leading to wrong conclusions. Moreover, even when achieving correct final answers, these powerful models can still regularly use flawed reasoning steps, which undermine the reliability and trustworthiness of LLMs' reasoning processes. To address these challenges, Process Reward Models (PRMs; Lightman et al. 2023; Wang et al. 2024b) are proposed to identify and mitigate process errors, thereby enabling finer-grained supervision on the reasoning process.

One critical challenge of developing PRMs lies in the data annotation for the correctness of reasoning processes, which is typically expensive and time-consuming. While Lightman et al. (2023) recruited human annotators with detailed instructions and elaborate procedures to achieve satisfactory annotation quality, the prohibitive cost pushes researchers to explore automated annotation methods. Among them, one commonly used approach is to assess process correctness by estimating the empirical probability of leading to the correct final answers through Monte Carlo (MC) methods, which has attracted great research interests and has also been commonly employed in practice (Xiong et al., 2024; Wang et al., 2024b; Luo et al., 2024). Another challenge lies in evaluating PRM performance, as previous studies (Lightman et al., 2023; Wang et al., 2024b; Luo et al., 2024) have predominantly relied on the Best-of-N (BoN) evaluation, which selects the highest-scored response from N candidates according to a PRM. Recently, PROCESSBENCH (Zheng et al., 2024) have emerged to evaluate the capability of PRMs in identifying step-wise correctness.

Nevertheless, during the training of our own PRM following conventional principles to construct data using MC estimation and evaluate on BoN, we gain several crucial lessons. **In terms of MC estimation**, (1) we observe that the PRM trained via MC estimation demonstrated significantly inferior performance and generalization capabilities compared to LLM-as-a-judge (Zheng et al., 2023) and human annotation. (2) We attribute the suboptimal performance of MC estimation to its fundamental limitation, which attempts to evaluate deterministic current-step correctness based on potential future outcomes. It significantly relies on the performance of the completion model,

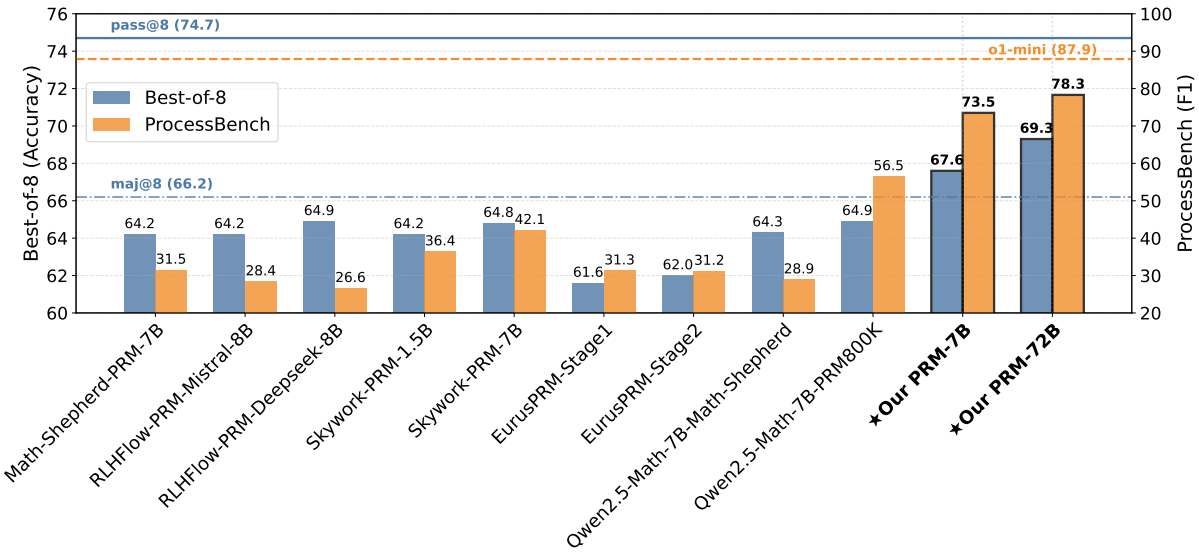

Figure 1: Overview of evaluation results on the Best-of-8 strategy of the policy model Qwen2.5-Math-7B-Instruct and the benchmark PROCESSBENCH (Zheng et al., 2024) across multiple PRMs (see Table 4 and Table 5 for details).

which may generate correct answers based on incorrect steps, or incorrect answers based on correct steps, introducing substantial noise and inaccuracy verification into step-wise correctness estimation. **Regarding the BoN evaluation**, (1) the unreliable policy models generate responses with correct answers but flawed processes, leading to a misalignment between the outcome evaluation criteria of BoN and the PRM objectives of process verification. (2) The PRMs with limited process verification capability demonstrate tolerance for these cases, resulting in inflated BoN performance. (3) We find that in the step scores distribution of existing PRMs, a significant proportion of minimum scores are concentrated on the final answer steps, indicating PRMs have shifted from process to outcome-based assessment in BoN.

To address these challenges, we propose a consensus filtering mechanism that combines MC estimation with LLM-as-a-judge, retaining only instances where both agree on error locations in the solution. Our approach improves both data efficiency and performance over existing PRMs in the conventional BoN evaluation. Furthermore, we advocate for complementing response-level BoN with step-wise evaluation methods. We employ the step-wise benchmark PROCESSBENCH (Zheng et al., 2024) to measure the ability to identify process errors. Our trained PRMs exhibit impressively stronger error identification performance than other open-source models, from PRMs to general language models, confirming that our training approach genuinely teaches PRMs to assess the correctness of intermediate reasoning steps.

2 Preliminary Trials

We initially explored training PRMs through MC estimation-based reasoning step annotation, following the methodology established in Math-Shepherd (Wang et al., 2024b). We collected a large-scale dataset of approximately 500,000 queries with golden answers. For each query, we generate 6-8 diverse responses and split them into individual reasoning steps using the delimiter "\n\n". To assess the correctness of each step, we conduct 8 independent completions starting from this step, estimating the step labels based on the empirical probabilities yielding the correct final answer. We trained PRMs with either hard labels or soft labels. For *hard* labels, we treat a step as correct if any one of the 8 completions yields the correct final answer, and negative otherwise. For *soft* labels, we determined the value (between 0 and 1) as the proportion of 8 completions leading to the correct final answers.

However, when we evaluate our trained PRMs on Best-of-8 and PROCESSBENCH, we found that the MC estimation-based PRMs do not possess noticeable advantages. As shown in Table 1, it reveal two critical limitations: (1) In the average Best-of-8 evaluations across diverse mathematical benchmarks, our trained models could not surpass the performance of simple majority voting, i.e., maj@8. (2) When evaluate on PROCESSBENCH for identi-

Setting	Best-of-8	PROCESSBENCH
maj@8	**66.2**	-
PRM800K	64.9	**56.5**
MC estimated hard labels	65.5	40.2
MC estimated soft labels	64.4	40.2

Table 1: Preliminary trials results on Best-of-8 and PROCESSBENCH using PRMs trained with MC estimated hard labels and soft labels, human-annotated PRM800K. maj@8 represents the majority voting of 8 responses.

Setting	# samples	Avg.
MC Estimation (Math-Shepherd)	440k	64.3
MC Estimation (our data)	860k	**65.9**
LLM-as-a-judge (our data)	860k	65.3
Human Annotation (PRM800K)	264k	64.9

Table 2: PRMs performance comparison on the Best-of-8 strategy. The models are trained on the different data construction methods including MC estimation, LLM-as-a-judge, and human annotation.

Method	# samples	GSM8K	MATH	Olympiad Bench	Omni-MATH	Avg.F1
MC Estimation (Math-Shepherd)	440k	62.5	31.6	13.7	7.7	28.9
MC Estimation (our data)	860k	**74.0**	47.3	19.4	19.8	40.1
LLM-as-a-judge (our data)	860k	60.9	49.5	39.4	36.1	46.5
Human Annotation (PRM800K)	264k	68.2	**62.6**	**50.7**	**44.3**	**56.5**

Table 3: PRMs performance comparison on PROCESSBENCH. The models are trained on the different data construction methods including MC estimation, LLM-as-a-judge, and human annotation.

fying erroneous reasoning steps, our trained models perform significantly worse than their counterparts trained on human-annotated data PRM800K (Lightman et al., 2023). The detailed experimental configurations and comprehensive results are demonstrated in Appendix A.

These undesirable evaluation performances push us to reflect on the currently prevalent data synthesis approach and evaluation strategy. Through the subsequent optimization process, we have indeed gained several observations and lessons learned.

3 The Lessons

In this section, we present the critical lessons gained during the PRM training comprising two main aspects: (1) the limitations of commonly adopted MC estimation approaches in PRMs training, and (2) the bias in using BoN as the sole evaluation metric for optimizing PRMs.

3.1 Limitations of MC Estimation

Distinguishing PRMs from Value Models PRMs provide fine-grained supervision by evaluating the correctness of intermediate reasoning steps. In contrast, value models estimate the potential of reaching the correct final answer from the current step in the future. The key difference between PRM and value model lies in that PRMs function as deterministic evaluators of current step correctness, while value models operate as predictive estimators of future solution potential.

MC estimation attempts to estimate the potential of reaching the correct final answer in the future from the current step. When we follow this approach to construct data and train the PRMs, the value model principles are incorporated into PRMs training essentially. This methodology potentially introduces performance and generalization limitations which we discuss in subsequent sections.

MC Estimation vs. LLM-as-a-judge vs. Human Annotation Since the observation of MC estimation's limitations of identifying erroneous steps in Section 2, we conducted a comprehensive comparison of three distinct data construction approaches: MC estimation, LLM-as-a-judge, and human annotation. For the MC estimation approach, we respectively train the PRM on 445k open-source datasets Math-shepherd (Wang et al., 2024b) and our 860k similarly constructed dataset. For the LLM-as-a-judge approach, we use the same 860k dataset and employ Qwen2.5-72B-Instruct (Yang et al., 2024b) to verify the correctness of each step in the responses. For the human annotation approach, we use the open-source dataset PRM800K (Lightman et al., 2023) which consists of approximately 265k samples after deduplication against the test set.

The experimental results of Best-of-8 and PROCESSBENCH are shown in Table 2 and 3, respectively. In general, for Best-of-8, the PRM trained on our MC estimated data achieves the best accuracy and human annotation shows substantially inferior performance. For PROCESSBENCH, human annotation achieves the best performance with the least amount of data, followed by LLM-as-a-judge, while MC estimation performs the worst despite having the largest dataset overall. The contrasting trend in the two evaluation catches our attention and is thoroughly investigated in Section 3.2.

In terms of the ability of identifying the correctness of reasoning steps evaluated in PROCESS-

BENCH, it can be found that: (1) human annotation, despite being only performed on the MATH dataset, exhibited superior generalization capabilities on more complex tasks, such as OlympiadBench and Omni-MATH. (2) Given identical data with different annotation approaches, LLM-as-a-judge demonstrates better generalization performance on challenging problems than MC estimation. (3) For MC estimation, a comparison between our 860k dataset and Math-Shepherd 440k data indicates that performance improvements can still be achieved through data scaling.

Stringent Data Filtering Mechanisms Required in MC Estimation We attribute the inferior performance of MC estimation compared to LLM-as-a-judge and human annotation to its high noise in reasoning step correctness estimation and inaccurate error position identification due to its heavy dependence on the policy model. For instance, the policy model may generate correct final answers but incorrect reasoning steps, or incorrect answers based on correct steps.

Motivated by LLM-as-a-judge's encouraging results in Best-of-8 and PROCESSBENCH, we naturally propose a simple yet efficient consensus filtering mechanism that integrates LLM-as-a-judge with MC estimation. Based on the aforementioned 860K samples, the instances are only retained when both LLM-as-a-judge and MC estimation show consensus on the error reasoning step locations in the solution. As demonstrated in Figure 2, it can be found that only approximately 40% of the data are preserved after consensus filtering. For evaluation on PROCESSBENCH, the results reveal that the reduced dataset after consensus filtering significantly outperforms MC estimation, and notably, achieves comparable performance to LLM-as-a-judge while using only 40% of the data. Regarding the BoN evaluation, the performance variations among these three models are marginal.

Hard Label vs. Soft Label in MC Estimation Although we have previously demonstrated that MC estimation is not as effective as LLM-as-a-judge and human annotation, there remains a noteworthy point of MC estimation to be discussed, i.e., whether to train with soft label or hard label. We construct 3 million training data using MC estimation and apply the consensus filtering strategy subsequently, which reduces the dataset to 1.5 million samples. We respectively train PRMs using both soft labels and hard labels on 3 million and 1.5 million data.

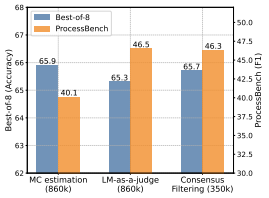

Figure 2: Performance comparison on Best-of-8 and PROCESSBENCH using PRMs trained with different data synthesis.

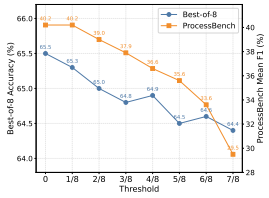

Figure 3: PRM Performance changes on Best-of-8 and PROCESSBENCH across different hard label thresholds.

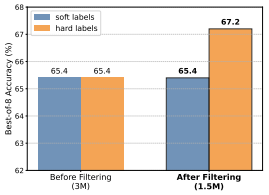

Figure 4: Performance comparison on Best-of-8 for the PRMs trained on soft and hard labels before and after filtering.

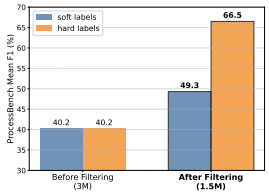

Figure 5: Performance comparison on PROCESSBENCH for the PRMs trained on soft and hard labels before and after consensus filtering.

The performance of trained PRMs on Best-of-8 and PROCESSBENCH are illustrated in Figure 4 and 5 separately. Before data filtering, the performance difference between soft and hard labels is not significant, which we attribute to the high noise level masking their distinctions. However, this difference becomes much more pronounced after data filtering, with hard labels substantially outperforming soft labels on both Best-of-8 and PROCESSBENCH. We consider the limitations of soft labels are: (1) As discussed in Section 3.1, the correctness of steps (i.e., rewards) should be deterministic. Training PRMs with soft labels that represent future possibilities introduces additional noise. For instance, when numerous completely correct steps are assigned with soft labels lower than 1, it actually reduces the model's ability to discriminate between positive and negative labels. (2) Only 8 completions for step correctness estimation exhibit high variance and are relatively crude. Although we can achieve better estimation accuracy by increasing the number of completions, the associated costs may outweigh the incremental benefits. Moreover, the experimental results indicate that the consensus filtering strategy yields performance benefits across both soft and hard label schemes.

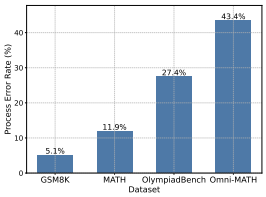

Figure 6: Proportion of incorrect reasoning steps in responses with correct final answers generated by policy model Qwen2.5-Math-7B-Instruct.

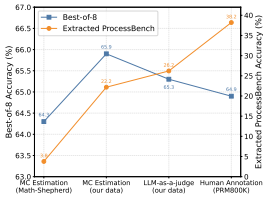

Figure 7: Performance trends on BoN and extracted PROCESSBENCH for models trained with different data sources.

Last but not least, we investigate the threshold selection for distinguishing between positive and negative labels based on the MC estimation result of 8 completions. Following our previous experimental setup, we conduct a series of experiments on the 3 million with threshold values from 1/8 to 7/8 at 1/8 intervals, with results shown in Figure 3. It can be easily observed that as the threshold increases, the performance deteriorates on both Best-of-8 and PROCESSBENCH, indicating that using an MC estimated value of 0 as the negative label and all others as positive labels yields the best results. In other words, we suggest a step is considered correct if any completion reaches the correct final answer in MC estimation. This threshold has also been employed throughout our all experimental studies.

3.2 Bias in BoN Evaluation

Although BoN evaluations are commonly used in previous PRM optimization, their effectiveness as a sole optimization criterion is worth careful consideration due to potential limitations in performance assessment.

Unreliable Policy Models Cause BoN-PRMs Misalignment In an ideal scenario, the responses generated by the policy model would exhibit both correct answers and accurate solution steps or conversely, flawed processes would correspond to incorrect answers. However, existing policy models are prone to generating responses with correct answers but flawed processes, while BoN inherently only focuses on answers, leading to a misalignment between the evaluation criteria of BoN and the PRM objectives of process verification. To provide empirical evidence for this phenomenon, we sample 8 responses from GSM8K, MATH, OlympiadBench, and Omni-MATH using the policy model Qwen2.5-Math-7B-Instruct. Then we randomly choose correct-answer responses from them and conduct thorough manual annotations. As detailed in Figure 6, a substantial percentage of responses contain process errors while maintaining correct answers. Notably, comparing easy task GSM8K and hard task Omni-MATH, this phenomenon becomes more pronounced as the problem's complexity increases. This implies that an effective PRM might assign low scores to responses with correct answers but flawed processes, resulting in overall lower performance on the BoN evaluation.

Limited Process Verification Capability in PRMs Lead to BoN Scores Inflation When the PRM cannot distinguish responses that have correct answers but flawed processes and assign them high scores, this leads to overestimated performance in the BoN evaluation, thereby creating an overly optimistic and potentially misleading assessment of PRM capabilities. To investigate the discriminative capability of PRMs for such cases, we extract instances from PROCESSBENCH where answers are correct but processes are erroneous and analysis the detection accuracy rates of PRMs for these cases. As shown in Figure 7, the PRMs trained on our MC estimated data, LLM-as-a-judge and PRM800K demonstrate opposite performance trends in BoN and extracted PROCESSBENCH evaluation. The model trained on our MC estimated data shows limited process verification capability but inflated results on the BoN. This limited discriminative capability indicates that PRMs struggle to differentiate between genuinely correct responses and those with merely superficial answer correctness in BoN evaluations. Consequently, this implies that beyond BoN evaluation, supplementary benchmarks are necessary to assess the actual capability of PRMs, especially in detecting process errors.

Process-to-Outcome Shift in BoN Optimized PRMs The majority of current PRMs are optimized towards BoN. However, the bias of BoN leads PRMs process-to-outcome shift. During the BoN selection process based on PRM-predicted scores following the scoring method for responses in (Lightman et al., 2023), it can be found that regardless of whether we employ the minimum score or the product of scores to evaluate the full solution, the lowest step score acts as the key limiting factor that affects the selection criteria of PRMs.

As shown in Figure 8, we analyze the distribution of minimum step scores assigned by multiple open-sourced PRMs, specifically focusing on

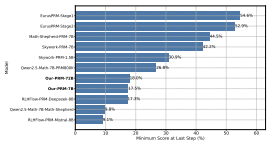

Figure 8: Percentage of responses where the minimum step score predict by PRMs appears in the final step (among all best of 8 responses).

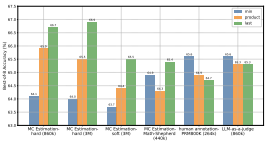

Figure 9: Performance on BoN across multiple PRMs with different scoring methods: minimum, product and last.

cases where the lowest score occurred at the final step, which typically contains the final answer. The results show that models EurusPRM-Stage1, EurusPRM-Stage2, Math-Shepherd-PRM-7B and Skywork-PRM-7B exhibit notably high proportions in this category, which exceed 40%.

This analysis reveals that some PRMs' performance in BoN evaluation is predominantly determined by final answer scores rather than intermediate reasoning steps. In other words, optimizing solely for the BoN evaluation has made current PRMs perform more like Outcome Reward Models (ORMs) in practice. Hence, it is essential to supplement response-level evaluation BoN with step-level assessment methods to avoid the process-to-outcome shift. In this paper, we employ process error localization tasks PROCESSBENCH.

Different PRMs, Different Optimal Scoring Strategies In BoN, the overall solution score is derived by combining individual step scores. When each step's score represents the probability of that specific step being correct, it's generally acceptable to combine these step-level scores (through methods like product or minimum) to calculate the overall solution score. However, in MC estimation, each step's score actually estimates the probability of reaching the correct final answer in the future from the current position. Given this forward-looking nature of MC estimation, we should neither multiply the estimated probabilities across steps (as these estimates are dependent on each other), nor simply take the minimum estimated value from a particular step as the overall score. Instead, the estimated value from the final step naturally integrates information from the entire solution process, making it more suitable as the final score for the complete solution.

To validate that, we evaluate BoN in different scoring strategies for the PRMs trained on MC estimation, LLM-as-a-judge, and human annotation data, as shown in Figure 9. We found that in MC estimation, using the last score shows significantly better performance than product and minimum approaches across multiple PRMs. And the product and minimum scores are better than the last for human annotation and LLM-as-a-judge.

4 Our Approach

This section presents our methodology for overcoming the previously discussed limitations and the details of our trained PRM achieving state-of-the-art performance.

4.1 Training Details

Based on the lessons learned, we implement a simple yet efficient consensus filtering mechanism by filtering out instances where there is a discrepancy between the LLM-annotated and MC-estimated process labels. This ensures the retained data maintains high quality and consistency in the reasoning process annotation. Specically, we use MC estimation to construct hard label, where a response is classified as negative only if none of the 8 completions achieves the correct final answer. Then, the LLM instantiated by Qwen2.5-Instruct-72B (Yang et al., 2024b) serves as a critic to verify the reasoning process for all responses step by step, i.e., LLM-as-a-judge. We employ cross-entropy loss on the tokens at the end of each step to train the binary classification task. We train the PRMs with 7B and 72B parameter, initialized with Qwen2.5-Math-7B-Instruct and Qwen2.5-Math-72B-Instruct respectively.

4.2 Experimental Setup

To validate the effectiveness of our trained PRMs, we respectively conduct the response-level BoN evaluation and the step-level process errors identification task PROCESSBENCH (Zheng et al., 2024).

Best-of-N We follow the experimental setting in Appendix A.2. In rm@8, we evaluate Outcome Reward Models (ORMs) and Process Reward Models (PRMs). For ORMs, we introduce Qwen2.5-Math-RM-72B (Yang et al., 2024c), which assigns a single score to each complete response. For PRMs, we compute the product of each step score as the final response score.

We compare with the following PRMs: (1) Math-Shepherd-PRM-7B (Wang et al., 2024b): determining process labels for each step by estimating the empirical probability of reaching the correct

Setting	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8 (Upper Bound)	98.1	92	49.3	80.5	59.6	52.6	90.5	74.7
maj@8	96.7	87.1	41.2	72.5	44.4	47.8	73.8	66.2
1.5B								
Skywork-PRM-1.5B	96.9	86.7	37.9	70.1	42.1	47.9	67.9	64.2
7B+								
Math-Shepherd-PRM-7B	**97.3**	85.4	37.9	70.6	40.4	47.2	70.5	64.2
RLHFlow-PRM-Mistral-8B	97.0	86.1	37.1	70.6	41.2	47.6	69.5	64.2
RLHFlow-PRM-Deepseek-8B	**97.3**	86.3	40.8	70.9	42.2	47.2	69.3	64.9
Skywork-PRM-7B	**97.3**	87.3	38.2	71.9	43.7	47.8	67.7	64.8
EurusPRM-Stage1	95.6	83.0	35.7	66.2	38.2	46.2	66.6	61.6
EurusPRM-Stage2	95.4	83.4	34.9	67.3	39.1	46.3	67.3	62.0
Qwen2.5-Math-7B-Math-Shepherd	96.9	86.5	36.8	71.4	41.6	47.7	69.3	64.3
Qwen2.5-Math-7B-PRM800K	96.9	86.9	37.1	71.2	44.0	47.6	70.9	64.9
★ Our PRM-7B	97.1	**88.0**	**42.6**	**74.5**	**47.6**	**48.7**	**74.5**	**67.6**
72B								
Qwen2.5-Math-RM-72B	**97.9**	88.5	42.6	**75.1**	**49.9**	**49.6**	78.7	68.9
★ Our PRM-72B	97.6	**88.7**	**46.0**	74.3	48.1	49.3	**81.1**	**69.3**

Table 4: Performance comparison on the Best-of-8 strategy of the policy model Qwen2.5-Math- 7B-Instruct. ★ represents the models we trained.

final answer. (2) RLHFlow-PRM-Mistral-8B & RLHFlow-PRM-Deepseek-8B (Xiong et al., 2024): two LLaMA-3.1-based PRMs that adopt Math-Shepherd's training methodology while implementing different solution generation models and optimization objectives. (3) Skywork-PRM-1.5B & Skywork-PRM-7B (Skywork, 2024): two recently released Qwen2.5-Math-based PRMs by Skywork. (4) EurusPRM-Stage1 & EurusPRM-Stage2 (Cui et al., 2025): two PRMs trained using Implicit PRM approach (Yuan et al., 2024) with 7B parameters, which obtains process rewards replying on the ORM trained on the response-level labels. (5) Qwen2.5-Math-7B-Math-Shepherd & Qwen2.5-Math-7B-PRM800K: two additional PRMs our developed by fine-tuning Qwen2.5-Math-7B-Instruct separately on the PRM800K (Lightman et al., 2023) and Math-Shepherd (Wang et al., 2024b) open-source datasets.

PROCESSBENCH The compared PRMs are consistent with the previously mentioned PRMs. For the LLM prompted as Critic Models, i.e., LLM-as-a-judge, we compare with proprietary language models GPT-4o-0806 (Hurst et al., 2024) and o1-mini (OpenAI, 2024), open-source language models Llama-3.3-70B-Instruct (Dubey et al., 2024), Qwen2.5-Math-72B-Instruct (Yang et al., 2024c), Qwen2.5-72B-Instruct (Yang et al., 2024b) and QwQ-32B-Preview (Qwen, 2024). We also decompose the N-step response trajectory into N separate instances to enable individual scoring by the ORM Qwen2.5-Math-RM-72B.

4.3 Experimental Results

Best-of-N The evaluation on policy model Qwen2.5-Math-7b-Instruct is shown in Table 4. Our PRM-7B demonstrates superior performance compared to other PRMs of equivalent model scale. Notably, it outperforms maj@8 across all 7 tasks, achieving an average improvement of 1.4%. Furthermore, our PRM-72B exhibits slightly better overall performance than Qwen2.5-Math-RM-72B, with particularly significant improvements observed in the Minerva Math and MMLU STEM tasks. Detailed experimental results, including BoN performance on Policy model Qwen2.5-Math-72b-Instruct, alternative scoring strategies, and evaluations on Chinese benchmarks, are comprehensively documented in the Appendix D.

PROCESSBENCH The evaluation results on PROCESSBENCH are presented in Table 5. When compared with LLM-as-judge, our PRM-7B in smaller model size demonstrates superior performance over all open-source models. For proprietary language models, our PRM-7B outperforms GPT-4o-0806, while there remains a performance gap compared to o1-mini. Furthermore, in comparison with existing PRMs, both our PRM-7B and 72B exhibit substan-

Model	GSM8K			MATH			OlympiadBench			Omni-MATH			Avg. F1
	error	correct	F1	error	correct	F1	error	correct	F1	error	correct	F1	
LLM-as-judge, Proprietary language models													
GPT-4-0806	70.0	91.2	79.2	54.4	76.6	63.6	45.8	58.4	51.4	45.2	65.6	53.5	61.9
o1-mini	88.9	97.9	**93.2**	83.5	95.1	**88.9**	80.2	95.6	**87.2**	74.8	91.7	**82.4**	**87.9**
LLM-as-judge, Open-source language models													
Llama-3.3-70B-Instruct	72.5	96.9	82.9	43.3	83.2	59.4	31.0	94.1	46.7	28.2	90.5	43.0	58.0
Qwen2.5-Math-72B-Instruct	49.8	96.9	65.8	36.0	94.3	52.1	19.5	97.3	32.5	19.0	96.3	31.7	45.5
Qwen2.5-72B-Instruct	62.8	96.9	76.2	46.3	93.1	61.8	38.7	92.6	54.6	36.6	90.9	52.2	61.2
QwQ-32B-Preview	81.6	95.3	**88.0**	78.1	79.3	**78.7**	61.4	54.6	**57.8**	55.7	68.0	**61.3**	**71.5**
PRMs													
1.5B													
Skywork-PRM-1.5B	50.2	71.5	59.0	37.9	65.2	48.0	15.4	26.0	19.3	13.6	32.8	19.2	36.4
7B+													
Math-Shepherd-PRM-7B	32.4	91.7	47.9	18.0	82.0	29.5	15.0	71.1	24.8	14.2	73.0	23.8	31.5
RLHFlow-PRM-Mistral-8B	33.8	99.0	50.4	21.7	72.2	33.4	8.2	43.1	13.8	9.6	45.2	15.8	28.4
RLHFlow-PRM-Deepseek-8B	24.2	98.4	38.8	21.4	80.0	33.8	10.1	51.0	16.9	10.9	51.9	16.9	26.6
Skywork-PRM-7B	61.8	82.9	70.8	43.8	62.2	53.6	17.9	31.9	22.9	14.0	41.9	21.0	42.1
EurusPRM-Stage1	46.9	42.0	44.3	33.3	38.2	35.6	23.9	19.8	21.7	21.9	24.5	23.1	31.2
EurusPRM-Stage2	51.2	44.0	47.3	36.4	35.0	35.7	25.7	18.0	21.2	23.1	19.1	20.9	31.3
Qwen2.5-Math-7B-Math-Shepherd	46.4	95.9	62.5	18.9	96.6	31.6	7.4	93.8	13.7	4.0	95.0	7.7	28.9
Qwen2.5-Math-7B-PRM800K	53.1	95.3	68.2	48.0	90.1	62.6	35.7	87.3	50.7	29.8	86.1	44.3	56.5
★ Our PRM-7B	72.0	96.4	**82.4**	68.0	90.4	**77.6**	55.7	85.5	**67.5**	55.2	83.0	**66.3**	**73.5**
72B													
Qwen2.5-Math-RM-72B	41.1	46.1	43.5	39.7	58.1	47.2	28.1	56.6	37.6	18.8	50.2	27.4	38.9
★ Our PRM-72B	78.7	97.9	**87.3**	74.2	88.2	**80.6**	67.9	82.0	**74.3**	64.8	78.8	**71.1**	**78.3**

Table 5: Performance comparison on PROCESSBENCH. ★ represents the models we trained. We report the results in the same calculation method with PROCESSBENCH.

tial advantages over their counterparts. An interesting observation worth noting is that the ORM Qwen2.5-Math-RM-72B exhibits considerable capability in identifying step errors, even surpassing some open-source PRMs.

5 Related Work

Reward Model in Mathematical Reasoning Mathematical reasoning reward models primarily fall into two categories: Outcome Reward Models (ORMs) that evaluate final answers, and Process Reward Models (PRMs) (Uesato et al., 2022; Lightman et al., 2023) that assess individual reasoning steps. Though PRMs show greater potential than ORMs (Lightman et al., 2023; Wang et al., 2024b), they rely on high-quality training data.

Mathematical Reasoning Step Verification Step verification methods usually include human annotation (Lightman et al., 2023) and automated approaches. Automated methods comprise: (1) backward-propagation based methods that infer step correctness from solution outcomes, including MC estimation (Wang et al., 2024b; Luo et al., 2024; Chen et al., 2024), progressive ORM labeling (Xi et al., 2024), credit assignment (Wang et al., 2024a; Cui et al., 2025; Yuan et al., 2024) techniques and so on; (2) prompting-based methods that leverage LLMs serve as critic, i.e., LLM-as-a-judge (Zhang et al., 2024; Gao et al., 2024; Xia et al., 2024) to assess step correctness directly. In this work, we integrate both MC estimation and LLM-as-a-judge methods.

6 Conclusion

In this paper, we present the critical lessons gained during developing PRMs and release a new state-of-the-art PRM. Firstly, we identify critical limitations in current data construction approaches for PRMs, demonstrating that MC estimation-based data construction yields inferior performance and generalization compared to LLM-as-a-judge and human annotation. Then we reveal the potential bias in using response-level BoN evaluation alone for PRMs and advocate for combining both response-level and step-level metrics. To address these issues, we propose an effective consensus filtering strategy combining MC estimation with LLM-as-a-judge. Our evaluation, incorporating both response-level BoN and identifying step-wise correctness task PROCESSBENCH, demonstrates significant improvements in data efficiency and model performance.

Limitation There are several limitations remained in our current work. Firstly, there exists a considerable performance gap between our PRMs and the BoN upper bound (pass@8), suggesting substantial optimization potential. Then the best practices for utilizing PRMs in reinforcement learning remain unexplored. Finally, although our approach combines LLM-as-a-judge with MC estimation for consensus filtering, the efficient utilization of existing high-quality human annotation data is still largely under-explored. For instance, gradually expanding high-quality datasets through weakly supervised methods can be investigated as a promising direction for future exploration.

References

Guoxin Chen, Minpeng Liao, Chengxi Li, and Kai Fan. 2024. Alphamath almost zero: Process supervision without process. *Preprint*, arXiv:2405.03553.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. 2025. Process reinforcement through implicit rewards.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Dayiheng Liu, Chang Zhou, Wen Xiao, Junjie Hu, Tianyu Liu, and Baobao Chang. 2024. Llm critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. *Preprint*, arXiv:2406.14024.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. Measuring massive multitask language understanding. In *ICLR*. OpenReview.net.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay V. Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with 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*.

Minpeng Liao, Chengxi Li, Wei Luo, Jing Wu, and Kai Fan. 2024. MARIO: math reasoning with code interpreter output - A reproducible pipeline. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 905–924. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, Yunxuan Li, Lei Shu, Yun Zhu, Lei Meng, Jiao Sun, and Abhinav Rastogi. 2024. Improve mathematical reasoning in language models by automated process supervision. *Preprint*, arXiv:2406.06592.

OpenAI. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

OpenAI. 2024. Openai o1-mini: Advancing cost-efficient reasoning.

Team Qwen. 2024. Qwq: Reflect deeply on the boundaries of the unknown.

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

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

Zhengyang Tang, Xingxing Zhang, Benyou Wang, and Furu Wei. 2024. Mathscale: Scaling instruction tuning for mathematical reasoning. In *Forty-first International Conference on Machine Learning, ICML*

2024, Vienna, Austria, July 21-27, 2024. OpenReview.net.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. *Preprint*, arXiv:2211.14275.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. 2024a. Q*: Improving multi-step reasoning for llms with deliberative planning. *Preprint*, arXiv:2406.14283.

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)*, pages 9426–9439.

Tianwen Wei, Jian Luan, Wei Liu, Shuang Dong, and Bin Wang. 2023. CMATH: can your language model pass chinese elementary school math test? *CoRR*, abs/2306.16636.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, Peng Sun, Tao Gui, Qi Zhang, and Xuanjing Huang. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. *Preprint*, arXiv:2402.05808.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating mathematical reasoning beyond accuracy. *Preprint*, arXiv:2404.05692.

Wei Xiong, Hanning Zhang, Nan Jiang, and Tong Zhang. 2024. An implementation of generative prm. `https://github.com/RLHFlow/RLHF-Reward-Modeling`.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. 2024b. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. 2024c. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. 2024. Free process rewards without process labels. *arXiv preprint arXiv:2412.01981*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2024. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

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. *arXiv preprint arXiv:2412.06559*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems*, volume 36, pages 46595–46623.

Wanjun Zhong, Ruixiang Cui, Yiduo Guo, Yaobo Liang, Shuai Lu, Yanlin Wang, Amin Saied, Weizhu Chen, and Nan Duan. 2024. Agieval: A human-centric benchmark for evaluating foundation models. In *NAACL-HLT (Findings)*, pages 2299–2314. Association for Computational Linguistics.

Qihao Zhu, Daya Guo, Zhihong Shao, Dejian Yang, Peiyi Wang, Runxin Xu, Y Wu, Yukun Li, Huazuo Gao, Shirong Ma, et al. 2024. Deepseek-coder-v2: Breaking the barrier of closed-source models in code intelligence. *arXiv preprint arXiv:2406.11931*.

A Details of Preliminary Trials

A.1 Training Setup

Training Data Synthesis We followed the commonly used MC estimation approach, Math-Shepherd (Wang et al., 2024b), to construct the PRM training data. Specifically, we collected a large-scale dataset of approximately 500,000 queries with golden answers. For each query, we generate 6-8 diverse responses by mixing outputs from the Qwen2-Math-Instruct and Qwen2.5-Math-Instruct series models (Yang et al., 2024c), spanning the model sizes of 7B and 72B parameters. These responses are systematically split into individual steps using the delimiter "\n\n". To assess the correctness of each step, we conduct 8 independent completions starting from this step using Qwen2.5-Math-Instruct series with the corresponding model size, estimating the step labels based on the empirical probabilities of each step yielding the correct final answer.

Training Details Our trained PRMs were initialized from the supervised fine-tuned Qwen2.5-Math-7B/72B-Instruct models (Yang et al., 2024c), where we replace the original language modeling head (used for next token prediction) with a scalar-value head, consisting of two linear layers. We trained PRMs with either hard labels or soft labels. For *hard* labels, we treat a step as correct if any one of the 8 completions yields the correct final answer, and negative otherwise. For *soft* labels, we determined the value (between 0 and 1) as the proportion of completions leading to the correct final answers. We calculated the cross-entropy (CE) loss and mean squared error (MSE) loss on the last tokens of each step for the binary classification task using hard labels and for the regression task using soft labels, respectively. Note that we eliminated all steps subsequent to those labeled as incorrect (label 0), as their validity becomes irrelevant after an error occurs. This removal was implemented to prevent potential model confusion during training.

A.2 Evaluation Setup

Best-of-N Consistent with previous work (Lightman et al., 2023; Wang et al., 2024b; Luo et al., 2024; Cobbe et al., 2021; Yang et al., 2024c), we employed the BoN evaluation, which selects the highest-scored response from N candidates according to a PRM. We denote the evaluation metric as "prm@N". Following Yang et al. (2024c), we sampled eight responses (i.e., $N = 8$) from Qwen2.5-Math-7B-Instruct across multiple mathematical benchmarks, including GSM8K (Cobbe et al., 2021), MATH (Hendrycks et al., 2021b), Minerva Math (Lewkowycz et al., 2022), GaoKao 2023 En (Liao et al., 2024), OlympiadBench (He et al., 2024), College Math (Tang et al., 2024), and MMLU STEM (Hendrycks et al., 2021a). Each candidate response is scored using the product of all the individual scores of each step within the response, as computed in Lightman et al. (2023). We also report the result of majority voting among eight samplings (maj@8) as the baseline, and pass@8 (i.e., the proportion of test samples where any of the 8 samplings lead to the correct final answers) as the upper bound.

PROCESSBENCH We also evaluated on PROCESSBENCH (Zheng et al., 2024) as a complement which measures the capability of models to identify erroneous steps in mathematical reasoning. Models are required to identify the first step that contains an error or conclude that all steps are correct. Following the evaluation methods for PRMs in PROCESSBENCH, we locate the first erroneous step from predict scores yielded by PRMs.

A.3 Evaluation Results

As shown in Table 6 and Table 7, we denote the models trained on our MC estimated dataset as Qwen2.5-Math-7B-PRM-MC-hard (trained with hard labels) and Qwen2.5-Math-7B-PRM-MC-soft (trained with soft labels), respectively, and compare them with a baseline model trained exclusively on the PRM800K (Lightman et al., 2023) dataset named Qwen2.5-Math-7B-PRM-PRM800K. The experimental results demonstrate that on the Best-of-8 evaluation, none of the PRMs achieved prm@8 scores superior to maj@8. Furthermore, on the PROCESSBENCH, Both Qwen2.5-Math-7B-PRM-MC-hard and Qwen2.5-Math-7B-PRM-MC-soft exhibit significantly inferior erroneous step localization capabilities compared to Qwen2.5-Math-7B-PRM-PRM800K.

Setting	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8 (Upper Bound)	98.1	92.0	49.3	80.5	59.6	52.6	90.5	74.7
maj@8	96.7	87.1	**41.2**	**72.5**	**44.4**	47.8	**73.8**	**66.2**
Qwen2.5-Math-7B-PRM800K	**96.9**	86.9	37.1	71.2	44.0	47.6	70.9	64.9
Qwen2.5-Math-7B-PRM-MC-hard	96.8	**87.3**	40.1	70.6	43.7	**48.1**	71.6	65.5
Qwen2.5-Math-7B-PRM-MC-soft	96.8	86.3	37.9	70.6	41.0	47.7	70.4	64.4

Table 6: Performance comparison on Best-of-8 using PRMs trained with MC estimated hard labels and soft labels, human-annotated PRM800K, denoted as Qwen2.5-Math-7B-PRM-MC-hard, Qwen2.5-Math-7B-PRM-MC-soft, and Qwen2.5-Math-7B-PRM800K, respectively.

Model	GSM8K			MATH			OlympiadBench			Omni-MATH			Avg. F1
	error	correct	F1	error	correct	F1	error	correct	F1	error	correct	F1	
Qwen2.5-Math-7B-PRM800K	53.1	95.3	68.2	48.0	90.1	**62.6**	35.7	87.3	**50.7**	29.8	86.1	**44.3**	**56.5**
Qwen2.5-Math-7B-PRM-MC-hard	67.1	90.2	77.0	35.2	65.8	45.8	13.2	28.0	17.9	13.3	41.9	20.2	40.2
Qwen2.5-Math-7B-PRM-MC-soft	65.7	93.3	**77.1**	35.7	64.5	46.0	13.2	29.2	18.1	12.9	40.2	19.6	40.2

Table 7: Performance comparison on PROCESSBENCH using PRMs trained with MC estimated hard labels and soft labels, human-annotated PRM800K, denoted as Qwen2.5-Math-7B-PRM-MC-hard, Qwen2.5-Math-7B-PRM-MC-soft, and Qwen2.5-Math-7B-PRM800K, respectively.

B Detailed Comparison of MC Estimation vs. LLM-as-a-judge vs. Human Annotation

The models trained on the different data construction methods including MC estimation, LLM-as-a-judge, and human annotation are evaluated on Best-of-8 and PROCESSBENCH. The detailed experimental results are shown in Table 8 and 9.

Setting	# samples	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
MC Estimation (Math-Shepherd)	440k	96.9	86.5	36.8	**71.4**	41.6	47.7	69.3	64.3
MC Estimation (our data)	860k	**97.0**	**87.6**	**41.9**	**71.4**	43.6	**48.2**	**71.9**	**65.9**
LLM-as-a-judge (our data)	860k	96.9	86.8	39.0	71.2	43.7	47.7	**71.9**	65.3
Human Annotation (PRM800K)	264k	96.9	86.9	37.1	71.2	**44.0**	47.6	70.9	64.9

Table 8: PRMs performance comparison on the Best-of-8 strategy of the policy model Qwen2.5-Math-7B-Instruct. The models are trained on the different data construction methods including MC estimation, LLM-as-a-judge, and human annotation.

Method	# samples	GSM8K			MATH			OlympiadBench			Omni-MATH			Avg.F1
		error	correct	F1	error	correct	F1	error	correct	F1	error	correct	F1	
MC Estimation (Math-Shepherd)	440k	46.4	95.9	62.5	18.9	96.6	31.6	7.4	93.8	13.7	4.0	95.0	7.7	28.9
MC Estimation (our data)	860k	62.3	91.2	**74.0**	35.2	71.9	47.3	12.7	41.3	19.4	12.1	54.4	19.8	40.1
LLM-as-a-judge (our data)	860k	44.0	99.0	60.9	33.5	94.8	49.5	24.7	97.1	39.4	22.3	95.4	36.1	46.5
Human Annotation (PRM800K)	264k	53.1	95.3	68.2	48.0	90.1	**62.6**	35.7	87.3	**50.7**	29.8	86.3	**44.3**	**56.5**

Table 9: PRMs performance comparison on PROCESSBENCH. The models are trained on the different data construction methods including MC estimation, LLM-as-a-judge, and human annotation.

C Process Verification Capability of Existing PRMs

The policy model may generate the responses that have correct answers but flawed processes. To investigate the discriminative capability of PRMs for such cases, we extract instances from PROCESSBENCH where answers are correct but processes are erroneous and analysis the detection accuracy rates of PRMs for these cases. As shown in Table 10, except our PRM-7B and 72B, all other open-sourced PRMs demonstrate detection accuracy rates below 50%.

	GSM8K	MATH	Olympiad Bench	Omni-MATH	Avg.
# samples	7	94	161	259	
1.5B					
Skywork-PRM-1.5B	42.9	36.2	12.4	13.9	26.4
7B+					
Math-Shepherd-PRM-7B	14.3	12.8	13.7	14.7	13.9
RLHFlow-PRM-Mistral-8B	14.3	13.8	7.5	10.0	11.4
RLHFlow-PRM-Deepseek-8B	0.0	18.1	9.9	10.8	9.7
Skywork-PRM-7B	**57.1**	26.6	14.3	13.1	27.8
EurusPRM-Stage1	28.6	25.5	19.9	20.1	23.5
EurusPRM-Stage2	42.9	27.7	18.0	20.8	27.4
Qwen2.5-Math-7B-Math-Shepherd	0.0	9.6	4.3	1.2	3.8
Qwen2.5-Math-7B-PRM800K	42.9	50.0	31.7	28.2	38.2
★ Our PRM-7B	42.9	**68.1**	**48.4**	**56.0**	**53.9**
72B					
★ Our PRM-72B	28.6	76.6	62.7	64.5	58.1

Table 10: The accuracy in identifying erroneous steps on the test cases of PROCESSBENCH containing correct answers but erroneous reasoning steps. "# samples" represents the number of test cases.

D Supplementary Experimental Results

D.1 The BoN Evaluation on Qwen2.5-Math-72b-Instruct

The BoN evaluation on policy model Qwen2.5-Math-72b-Instruct is shown in Table 11. Our PRM-7B outperforms other PRMs of equivalent model scale. However, its performance is inferior to maj@8, suggesting challenges in employing a 7B PRM for the supervision of 72B policy model-generated responses. Besides, Our PRM-72B surpasses maj@8 in prm@8 and is comparable with Qwen2.5-Math-RM-72B in orm@8.

D.2 The BoN Evaluation with Various Scoring Strategies

We demonstrate experimental results using the last step score, the minimum step score or the production of step scores as the solution-level score. The BoN results with policy model Qwen2.5-Math-7B-Instruct and Qwen2.5-Math-72B-Instruct are shown in Table 13 and Table 14 respectively.

D.3 The BoN Evaluation on Chinese Benchmarks

We evaluate across three Chinese benchmarks including Chinese math benchmarks CMATH (Wei et al., 2023), GaoKao Math Cloze (Zhong et al., 2024), and GaoKao Math QA (Zhong et al., 2024) following (Yang et al., 2024c), as shown in Table 15 and Table 16.

E PRM Guided Search

We further integrate PRM with greedy search by generating N candidate steps at each step, evaluating these candidates using PRM scoring, and selecting the highest-scoring step for subsequent expansion. For the policy model, we employed Qwen2.5-7B-Instruct which has greater diversity in generation to sample 8 candidates at each step, with sampling parameters set to $temperature = 1.0$ and $top\_p = 1.0$. We conduct comparative experiments with ORM in BoN approach. As shown in Table 12, Our PRM-72B with greedy search@8 is slightly superior performance compared to Qwen2.5-Math-RM-72B with orm@8. We argue the potentially smaller performance differential between PRM and ORM lies in the consistency of generated token counts between greedy search and BoN outputs. Furthermore, although greedy search always selects the highest-scoring candidate at each step, the highest-scoring step may not be the correct

Setting	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8	97.3	93.2	56.6	83.6	62.4	54.1	95.3	77.5
maj@8	96.0	88.6	47.8	73.8	50.1	50.2	84.9	70.2
1.5B								
Skywork-PRM-1.5B	96.5	88.1	45.2	74.3	48.4	49.7	79.7	68.8
7B+								
Math-Shepherd-PRM-7B	96.5	86.8	45.6	71.9	49.2	49.5	77.5	68.1
RLHFlow-PRM-Mistral-8B	96.6	87.5	46.3	73.5	48.9	49.4	**83.4**	69.4
RLHFlow-PRM-Deepseek-8B	96.5	87.7	44.5	73.5	48.7	49.4	84.6	69.3
Skywork-PRM-7B	**97.0**	89.0	47.1	75.3	49.8	49.9	76.3	69.2
EurusPRM-Stage1	95.4	85.6	44.1	72.5	46.5	49.2	80.3	67.7
EurusPRM-Stage2	95.3	85.1	44.9	72.5	47.1	49.0	80.2	67.7
Qwen2.5-Math-7B-Math-Shepherd	96.9	88.5	46.0	75.8	49.9	49.5	79.7	69.5
Qwen2.5-Math-7B-PRM800K	96.5	88.9	**47.4**	75.3	50.7	50.1	76.6	69.4
★ Our PRM-7B	96.8	**89.6**	46.7	**77.7**	**51.4**	**50.4**	76.4	**69.9**
72B								
Qwen2.5-Math-RM-72B	96.4	89.8	**47.4**	76.9	**54.5**	**50.6**	80.1	**70.8**
★ Our PRM-72B	**96.4**	**89.9**	46.0	**77.4**	52.9	50.1	**82.3**	70.7

Table 11: Performance comparison on the Best-of-8 strategy of the policy model Qwen2.5-Math-72B-Instruct. ★ represents the models we trained.

one. Therefore, implementing either Depth-First Search (DFS) with backtracking capabilities or search approaches incorporating score constraints could prove more suitable for this cases.

We choose the highest-scoring candidate at each step which the score predicted by PRM represents the correctness of this step. But such locally optimal choices may not lead to the correct final answer. In contrast, value models can predict the future probability of reaching the correct answer, rather than reflecting the correctness of the current step like rewards do, making them particularly well-suited for integration with search strategies. Based on these considerations, we believe there is still significant potential for exploration in the future regarding more appropriate search strategies or combining rewards and values to simultaneously consider both the correctness of the current step and the possibility of reaching the correct future outcomes.

Setting	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8 (Upper Bound)	96.9	89.6	48.2	79.7	58.4	55.0	81.6	72.8
pass@1	91.2	74.0	32.0	64.7	36.9	46.2	57.1	57.4
maj@8	93.7	80.3	37.1	69.9	45.8	48.5	61.9	62.5
orm@8								
Qwen2.5-Math-RM-72B	95.4	84.2	**38.6**	73.0	48.6	**50.1**	**75.6**	66.5
Greedy Search@8								
Skywork-PRM-7B	95.3	83.2	33.8	70.4	44.1	48.2	60.1	62.2
★ Our PRM-7B	95.5	82.6	32.0	71.4	44.9	48.8	69.6	63.5
★ Our PRM-72B	**95.9**	**84.7**	37.9	**73.2**	**48.9**	50.0	75.3	**66.6**

Table 12: The performance of PRM guided greedy search and ORM of Best-of-8 with policy model Qwen2.5-7B-Instruct. For greedy search, 8 candidates is proposed at each step.

F Prompt Template for LLM-as-a-judge

```
I will provide a math problem along with a solution. They will be formatted as
follows:

[Math Problem]

<math_problem>
...(math problem)...
</math_problem>

[Solution]

<paragraph_1>
...(paragraph 1 of solution)...
</paragraph_1>

...

<paragraph_n>
...(paragraph n of solution)...
</paragraph_n>

Your task is to review each paragraph of the solution in sequence, analyzing,
verifying, and critiquing the reasoning in detail. You need to provide the
analyses and the conclusion in the following format:

<analysis_1>
...(analysis of paragraph 1)...
</analysis_1>

...

<analysis_n>
...(analysis of paragraph n)...
</analysis_n>

<conclusion>
Correct/Incorrect
</conclusion>


* When you analyze each paragraph, you should use proper verification,
recalculation, or reflection to indicate whether it is logically and
mathematically valid. Please elaborate on the analysis process carefully.

* If an error is detected in any paragraph, you should describe the nature and
cause of the error in detail, and suggest how to correct the error or the correct
approach. Once a paragraph is found to contain any error, stop further analysis
of subsequent paragraphs (as they may depend on the identified error) and directly
provide the conclusion of "Incorrect."
```

```
For instance, given a solution of five paragraphs, if an error is found in the
third paragraph, you should reply in the following format:

<analysis_1>
...(analysis of paragraph 1)...
</analysis_1>

<analysis_2>
...(analysis of paragraph 2)...
</analysis_3>

<analysis_3>
...(analysis of paragraph 3; since an error is found here, also provide detailed
critique and correction guideline)...
</analysis_3>

<conclusion>
Incorrect
</conclusion>

Note that the analyses of paragraphs 4 and 5 should be skipped as the paragraph
3 has been found to contain an error.

* Respond with your analyses and conclusion directly.

---------------------------------------------------

The following is the math problem and the solution for you task:

[Math Problem]

{tagged_problem}

[Solution]

{tagged_response}
```

Setting	Scoring	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8 (Upper Bound)	-	98.1	92	49.3	80.5	59.6	52.6	90.5	74.7
maj@8	-	96.7	87.1	41.2	72.5	44.4	47.8	73.8	66.2
Math-Shepherd-PRM-7B	last	96.8	85.2	39.0	70.1	42.8	47.2	67.7	64.1
	product	97.3	85.4	37.9	70.6	40.4	47.2	70.5	64.2
	min	96.9	85.3	39.0	69.9	42.2	47.4	70.6	64.5
RLHFlow-PRM-Mistral-8B	last	97.0	85.3	39.0	71.2	44.0	47.1	64.0	63.9
	product	97.0	86.1	37.1	70.6	41.2	47.6	69.5	64.2
	min	97.0	84.3	37.1	69.4	40.4	46.9	68.7	63.4
RLHFlow-PRM-Deepseek-8B	last	97.0	84.7	35.7	70.4	43.0	46.8	63.8	63.1
	product	97.3	86.3	40.8	70.9	42.2	47.2	69.3	64.9
	min	97.3	84.5	38.2	69.6	40.7	46.5	67.6	63.5
Skywork-PRM-1.5B	last	96.8	86.4	39.0	71.7	45.0	47.9	68.2	65.0
	product	96.9	86.7	37.9	70.1	42.1	47.9	67.9	64.2
	min	96.6	86.6	37.9	71.9	43.1	48.2	66.9	64.5
Skywork-PRM-7B	last	97.2	87.3	41.2	73.8	45.8	48.3	65.3	65.6
	product	97.3	87.3	38.2	71.9	43.7	47.8	67.7	64.8
	min	96.7	87.0	39.7	71.2	42.5	48.2	66.6	64.6
EurusPRM-Stage1	last	94.7	79.7	32.7	61.6	33.8	45.7	63.4	58.8
	product	95.6	83.0	35.7	66.2	38.2	46.2	66.6	61.6
	min	95.8	83.3	39.0	67.8	37.9	46.6	67.4	62.5
EurusPRM-Stage2	last	94.7	79.7	33.1	61.3	34.2	45.7	63.5	58.9
	product	95.4	83.4	34.9	67.3	39.1	46.3	67.3	62.0
	min	96.1	83.6	39.3	68.8	38.8	46.7	67.5	63.0
Qwen2.5-Math-7B-Math-Shepherd	last	97.1	87.7	38.6	73.8	44.6	48.1	68.0	65.4
	product	96.9	86.5	36.8	71.4	41.6	47.7	69.3	64.3
	min	97.0	86.7	36.8	72.5	43.1	47.6	70.7	64.9
Qwen2.5-Math-7B-PRM800K	last	96.7	86.3	37.9	71.9	44.3	47.6	68.1	64.7
	product	96.9	86.9	37.1	71.2	44.0	47.6	70.9	64.9
	min	96.9	86.6	39.7	71.7	45.6	47.8	71.1	65.6
★ Our PRM-7B	last	96.9	87.2	39.0	73.5	45.5	48.5	72.0	66.1
	product	97.1	88.0	42.6	74.5	47.6	48.7	74.5	67.6
	min	97.0	87.8	42.3	74.3	46.2	48.3	74.1	67.1
★ Our PRM-72B	last	97.6	88.9	43.4	73.8	49.2	49.6	76.8	68.5
	product	97.6	88.7	46.0	74.3	48.1	49.3	81.1	69.3
	min	97.6	88.8	45.2	74.5	48.1	49.2	80.9	69.2

Table 13: Performance comparison on the Best-of-8 strategy of the policy model Qwen2.5-Math-7B-Instruct with 3 scoring strategies: last, product and minimum. ★ represents the models we trained.

Setting	Scoring	GSM8K	MATH	Minerva Math	GaoKao 2023 En	Olympiad Bench	College Math	MMLU STEM	Avg.
pass@8 (Upper Bound)	-	97.3	93.2	56.6	83.6	62.4	54.1	95.3	77.5
maj@8	-	96.0	88.6	47.8	73.8	50.1	50.2	84.9	70.2
Math-Shepherd-PRM-7B	last	96.2	87.0	46.7	73.0	47.3	49.8	76.3	68.0
	product	96.5	86.8	45.6	71.9	49.2	49.5	77.5	68.1
	min	96.1	86.8	45.6	73.2	48.6	49.9	76.0	68.0
RLHFlow-PRM-Mistral-8B	last	96.3	86.6	44.9	74.3	47.6	49.3	67.1	66.6
	product	96.6	87.5	46.3	73.5	48.9	49.4	83.4	69.4
	min	96.4	86.3	44.5	71.9	47.9	49.3	76.0	67.5
RLHFlow-PRM-Deepseek-8B	last	96.1	86.6	46.3	73.2	49.2	49.2	71.7	67.5
	product	96.5	87.7	44.5	73.5	48.7	49.4	84.6	69.3
	min	96.6	87.4	44.1	74.0	48.6	49.3	74.8	67.8
Skywork-PRM-1.5B	last	96.1	88.6	44.9	72.2	47.9	50.1	74.2	67.7
	product	96.5	88.1	45.2	74.3	48.4	49.7	79.7	68.8
	min	96.0	88.3	45.6	73.8	48.6	50.1	75.9	68.3
Skywork-PRM-7B	last	97.0	89.0	46.0	74.8	51.0	49.7	66.7	67.7
	product	97.0	89.0	47.1	75.3	49.8	49.9	76.3	69.2
	min	96.9	89.2	46.7	73.5	49.8	49.8	73.2	68.4
EurusPRM-Stage1	last	95.9	87.3	44.9	72.7	47.0	49.4	78.4	67.9
	product	95.4	85.6	44.1	72.5	46.5	49.2	80.3	67.7
	min	96.4	88.2	44.9	75.1	49.0	49.5	83.7	69.5
EurusPRM-Stage2	last	96.0	87.7	44.5	73.5	47.0	49.4	78.1	68.0
	product	95.3	85.1	44.9	72.5	47.1	49.0	80.2	67.7
	min	96.5	88.6	45.2	75.3	48.9	49.6	83.3	69.6
Qwen2.5-Math-7B-Math-Shepherd	last	97.0	89.6	44.9	77.4	50.8	50.5	74.9	69.3
	product	96.9	88.5	46.0	75.8	49.9	49.5	79.7	69.5
	min	97.0	88.6	46.0	74.8	50.2	49.6	79.6	69.4
Qwen2.5-Math-7B-PRM800K	last	96.7	88.8	47.1	76.1	50.1	49.5	71.8	68.6
	product	96.5	88.9	47.4	75.3	50.7	50.1	76.6	69.4
	min	96.5	89.1	47.1	76.1	50.7	49.9	75.3	69.2
★ Our PRM-7B	last	96.8	89.0	46.7	75.3	49.8	50.3	78.4	69.5
	product	96.8	89.6	46.7	77.7	51.4	50.4	76.4	69.9
	min	96.7	89.6	46.3	77.9	50.8	50.3	76.0	69.7
★ Our PRM-72B	last	96.3	89.8	47.8	76.6	53.3	50.9	80.5	70.7
	product	96.4	89.9	46.0	77.4	52.9	50.1	82.3	70.7
	min	96.4	89.7	46.3	77.7	52.4	50.4	81.2	70.6

Table 14: Performance comparison on the Best-of-8 strategy of the policy model Qwen2.5-Math-72B-Instruct with 3 scoring strategies: last, product and minimum. ★ represents the models we trained.

Setting	Scoring	CMATH	CN Middle School 24	GaoKao	Avg.
pass@8 (Upper Bound)	-	95.3	82.2	84.3	87.3
maj@8	-	92.7	78.2	68.1	79.7
Math-Shepherd-PRM-7B	last	91.8	80.2	63.0	78.3
	product	92.0	80.2	69.1	80.4
	min	91.5	80.2	69.8	80.5
RLHFlow-PRM-Mistral-8B	last	92.8	79.2	57.2	76.4
	product	92.7	77.2	65.8	78.6
	min	92.8	76.2	62.1	77.0
RLHFlow-PRM-Deepseek-8B	last	93.2	75.2	56.9	75.1
	product	92.7	76.2	63.6	77.5
	min	93.0	74.3	67.3	78.2
Skywork-PRM-1.5B	last	93.8	80.2	66.6	80.2
	product	92.8	79.2	66.3	79.4
	min	93.3	80.2	66.6	80.0
Skywork-PRM-7B	last	94.0	81.2	66.7	80.6
	product	93.3	79.2	68.1	80.2
	min	93.8	80.2	66.3	80.1
EurusPRM-Stage1	last	91.8	77.2	55.4	74.8
	product	91.7	77.2	52.6	73.8
	min	91.7	78.2	64.4	78.1
EurusPRM-Stage2	last	91.8	77.2	55.7	74.9
	product	92.0	77.2	52.4	73.9
	min	92.0	78.2	64.7	78.3
Qwen2.5-Math-7B-Math-Shepherd	last	93.0	81.2	65.4	79.9
	product	93.0	79.2	67.7	80.0
	min	92.5	80.2	69.8	80.8
Qwen2.5-Math-7B-PRM800K	last	92.8	78.2	67.1	79.4
	product	92.7	77.2	68.9	79.6
	min	93.0	77.2	69.4	79.9
★ Our PRM-7B	last	93.3	80.2	68.2	80.6
	product	93.7	80.2	70.1	81.3
	min	93.5	80.2	71.7	81.8
★ Our PRM-72B	last	94.3	80.2	72.1	82.2
	product	94.2	80.2	73.5	82.6
	min	94.2	80.2	73.1	82.5

Table 15: Best-of-8 performance comparison on the Chinese benchmarks with the policy model Qwen2.5-Math-7B-Instruct in 3 scoring strategies: last, product and minimum. ★ represents the PRMs we trained.

Setting	Scoring	CMATH	CN Middle School 24	GaoKao	Avg.
pass@8 (Upper Bound)	-	96.8	83.2	86.2	88.7
maj@8	-	95.3	79.2	75.0	83.2
Math-Shepherd-PRM-7B	last	93.7	78.2	73.2	81.7
	product	94.0	80.2	72.1	82.1
	min	93.5	80.2	73.9	82.5
RLHFlow-PRM-Mistral-8B	last	94.3	79.2	65.5	79.7
	product	93.8	79.2	72.0	81.7
	min	93.3	79.2	71.2	81.2
RLHFlow-PRM-Deepseek-8B	last	94.3	79.2	63.0	78.8
	product	94.3	79.2	72.5	82.0
	min	94.5	79.2	73.5	82.4
Skywork-PRM-1.5B	last	94.8	80.2	74.3	83.1
	product	93.8	79.2	69.7	80.9
	min	94.5	80.2	74.6	83.1
Skywork-PRM-7B	last	95.3	80.2	72.6	82.7
	product	94.7	80.2	71.5	82.1
	min	94.8	80.2	76.0	83.7
EurusPRM-Stage1	last	94.0	79.2	64.5	79.2
	product	93.8	80.2	64.5	79.5
	min	94.7	79.2	70.8	81.6
EurusPRM-Stage2	last	94.2	79.2	63.4	78.9
	product	93.7	80.2	65.4	79.8
	min	94.3	79.2	69.7	81.1
Qwen2.5-Math-7B-Math-Shepherd	last	95.0	81.2	74.6	83.6
	product	94.5	80.2	73.0	82.6
	min	94.3	80.2	71.5	82.0
Qwen2.5-Math-7B-PRM800K	last	94.2	79.2	76.5	83.3
	product	94.2	82.2	70.8	82.4
	min	93.8	80.2	72.9	82.3
★ Our PRM-7B	last	94.7	79.2	74.5	82.8
	product	94.3	81.2	77.6	84.4
	min	94.5	81.2	77.6	84.4
★ Our PRM-72B	last	96.0	79.2	76.1	83.8
	product	96.0	80.2	77.2	84.5
	min	95.8	80.2	77.5	84.5

Table 16: Best-of-8 performance comparison on the Chinese benchmarks with the policy model Qwen2.5-Math-72B-Instruct in 3 scoring strategies: last, product and minimum. ★ represents the PRMs we trained.