GM-PRM: A Generative Multimodal Process Reward Model for Multimodal Mathematical Reasoning

Jianghangfan Zhang[1], Yibo Yan[1,2], Kening Zheng[3], Xin Zou[1], Song Dai[1], Xuming Hu[1,2]*

[1]Hong Kong University of Science and Technology (Guangzhou),
[2]Hong Kong University of Science and Technology
[3]University of Illinois Chicago

Correspondence: xuminghu@hkust-gz.edu.cn

Abstract

Multimodal Large Language Models (MLLMs) demonstrate remarkable capabilities but often *struggle with complex, multi-step mathematical reasoning*, where minor errors in visual perception or logical deduction can lead to complete failure. While Process Reward Models (PRMs) offer step-by-step supervision, existing multimodal PRMs are *limited to being binary verifiers that can identify but not correct errors*, offering little explanatory power. To address these deficiencies, we introduce the **Generative Multimodal Process Reward Model (GM-PRM)**, a novel paradigm that transforms the PRM from a passive judge into an active reasoning collaborator. Instead of a simple scalar score, GM-PRM provides a fine-grained, interpretable analysis of each reasoning step, evaluating its step intent, visual alignment, and logical soundness. More critically, GM-PRM is trained to generate a corrected version of the first erroneous step it identifies. This unique corrective capability enables our new test-time inference strategy, Refined Best-of-N (Refined-BoN). This framework actively enhances solution quality by using the PRM's generated correction to guide the policy model toward a more promising reasoning trajectory, thereby improving the diversity and correctness of the solution pool. We demonstrate that GM-PRM achieves state-of-the-art results on multiple multimodal math benchmarks, significantly boosting policy model performance with remarkable data efficiency, requiring only a 20K-sample training dataset. Our code will be released upon acceptance.

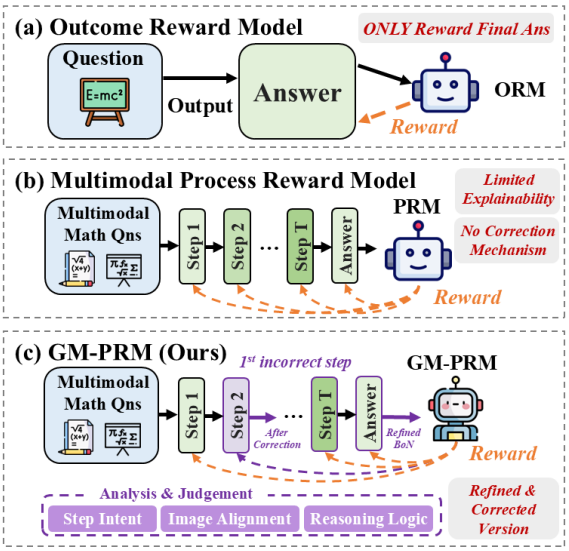


Figure 1: Comparison among ORM (a), PRM (b), and our proposed GM-PRM (c) for multimodal math reasoning.

1 Introduction

The advent of Multimodal Large Language Models (MLLMs) has marked a significant milestone in artificial intelligence, demonstrating remarkable capabilities in integrating and understanding visual and textual information (Caffagni et al., 2024; Yan et al., 2024c; Yan and Lee, 2024; Huo et al., 2024; Zheng et al., 2024b). While these models excel at general-purpose tasks such as image captioning and visual question answering, they often falter when confronted with complex, multi-step reasoning challenges, particularly within specialized domains like mathematics (Wang et al., 2024; Yan et al., 2024a, 2025a; Ahn et al., 2024). Solving multimodal mathematical problems requires not only accurate perception of visual elements (*e.g.,* geometric figures, function graphs) but also a rigorous, step-by-step logical deduction process (Shi et al., 2024; Zhuang et al., 2025; Yan et al., 2025b). *Minor errors in either image interpretation or logical inference can cascade, leading to entirely incorrect final answers.*

To mitigate these reasoning deficiencies, Process Reward Models (PRMs) have emerged as a promising paradigm (Gao et al., 2024; Zhong et al., 2025). Unlike outcome-based models that only re-

*Corresponding author

ward correct final answers (shown in Figure 1 (a)), PRMs provide fine-grained supervision by evaluating the correctness of each intermediate step in a reasoning chain (Zheng et al., 2024a; Lambert et al., 2024; Yan et al., 2024b), as shown in Figure 1 (b). This approach has proven effective in the language domain (Zeng et al., 2025; Yuan et al., 2024; Zhang et al., 2025a). However, extending PRMs to the multimodal context presents unique challenges (Miao et al., 2025; Du et al., 2025; Li et al., 2025b; Cao et al., 2025). Existing multimodal PRMs often function as binary classifiers, assigning a simple correct/incorrect label to each step, which offers *limited explanatory power*. Furthermore, they typically possess the ability to identify errors but *lack the mechanism to correct them*, leaving the reasoning process fundamentally broken. This limitation constrains their utility, especially within mechanisms like Best-of-N (BoN) sampling, which remain passive selection processes over a static set of potentially flawed solutions.

In this work, we introduce a novel **Generative Multimodal Process Reward Model (GM-PRM)** to address these limitations, as illustrated in Figure 1 (c). Our model transcends the role of a simple verifier and acts as an active reasoning collaborator. Instead of merely outputting a scalar score, our GM-PRM leverages its generative capabilities to produce a detailed, interpretable analysis for each reasoning step. This analysis deconstructs the step into three critical aspects: its fundamental **step intent**, the correctness of its **image alignment**, and the soundness of its **reasoning logic**. More importantly, our model is trained not only to identify errors but also to **generate a refined, corrected version** of the first incorrect step it encounters.

This unique corrective capability enables us to propose a new test-time inference strategy: the **Refined Best-of-N (Refined-BoN)** process. This dynamic framework moves beyond passive selection by actively enhancing the quality of candidate solutions. When our GM-PRM identifies a flawed step within a generated solution, it intervenes by providing a corrected step, which is then used to guide the policy model in generating a new, more promising reasoning trajectory. This iterative refinement process significantly improves the diversity and correctness of the solution pool, leading to a substantial boost in the policy model's problem-solving performance. Furthermore, we demonstrate that this powerful capability can be achieved with remarkable data efficiency, requiring a significantly smaller training dataset than previous approaches. Our primary contributions are as follows:

- We develop a generative multimodal PRM that **provides fine-grained, interpretable feedback for mathematical reasoning**. It analyzes each step's purpose, image alignment, and logical validity, moving beyond simple binary classification to offer deeper insight into the model's thought process.
- We introduce a novel Refined-BoN framework that **leverages the PRM's generative power to actively correct errors at test time**. It enhances the policy model's ability to find correct solutions by iteratively improving flawed reasoning paths.
- We demonstrate the **effectiveness and data efficiency** of GM-PRM, achieving state-of-the-art results on **multiple multimodal math benchmarks**. Our approach requires only a 20K sample dataset, highlighting the quality of data curation and the power of generative supervision strategy.

2 Related Work

2.1 Process Reward Models (PRMs)

PRMs have been proposed to evaluate the fine-grained step level for model reasoning. During the implementation process, annotating and obtaining a high-quality training dataset incurs a high cost. PRM800K (Lightman et al., 2023) is the first process supervision dataset completely annotated by humans. To mitigate annotation costs, Math-Shepherd (Wang et al., 2023) proposes Monte Carlo (MC) estimation, while OmegaPRM (Luo et al., 2024) leverages Monte Carlo Tree Search (MCTS) to automatically evaluate each reasoning step, both utilizing the generation capabilities of Large Language Models (LLMs). Subsequent research has enhanced the effectiveness of PRMs through various methods, including VersaPRM (Zeng et al., 2025), Implicit PRM (Yuan et al., 2024), OpenPRM (Zhang et al., 2025a), PQM (Li and Li, 2024), PAV (Setlur et al., 2024), and others. Furthermore, GenRM (Zhao et al., 2025), ThinkPRM (Khalifa et al., 2025) and R-PRM (She et al., 2025) extend the method of using model generation analysis to evaluate steps to PRMs. Recently, DeepSeek-GRM (Liu et al., 2025) analyzes responses from multiple perspectives and directly generates and aggregates the

scores to achieve the estimation of the entire reasoning process. There are also many studies on benchmarks of PRMs such as ProcessBench (Zheng et al., 2024a), PRMBench (Song et al., 2025), and Socratic-PRMBench (Li et al., 2025a).

2.2 Multimodal PRMs

After achieving certain results and progress in the research of language modality in PRMs, research on PRMs has also begun to shift towards multimodal tasks. M-STAR (Liu et al., 2024a) proposes and implements multimodal PRM on multimodal problems. URSA (Luo et al., 2025) constructs a dataset by inserting errors and utilizes it to train a multimodal PRM. VisualPRM (Wang et al., 2025b) not only uses MC estimation to construct a multimodal VisualPRM400K training dataset, but also proposes a benchmark for multimodal PRMs called VisualProcessBench, which is entirely annotated by humans. Moreover, Athena-PRM (Wang et al., 2025a), PRM-BAS (Hu et al., 2025), MM-PRM (Du et al., 2025) and DreamPRM (Cao et al., 2025) also improve the capability of multimodal PRMs. Although several studies have explored multimodal PRMs, applying them to multimodal tasks effectively still poses certain challenges, such as insufficient interpretability of the labels assigned to each reasoning step and the inability to correct identified erroneous steps. In our work, we introduce a generative multimodal PRM, GM-PRM to solve the above problems.

3 Methodology

In this section, we first describe how to utilize PRMs and generative PRMs combined with the BoN method to improve the performance of policy models for mathematical problems in Section 3.1. Then, we introduce our process to implement multimodal generative PRM, including data construction and model training in Section 3.2. Finally, we propose a novel Refined-BoN framework for PRMs to enhance its performance beyond traditional BoN method in Section 3.3.

3.1 PRMs for Mathematical Problem

In this section, we present the implementation methods of PRM and GM-PRM, and provide formal and detailed explanations of their usage through mathematical notation.

3.1.1 Problem and Reasoning Steps Generation

Let Q denote a mathematical problem. Firstly, an LLM π is involved in solving the mathematical problem Q. To facilitate reasoning, the problem is combined with a prompt P, which includes specific instructions guiding the generation of a step-by-step reasoning process and a final answer. This composite input is then fed into the LLM. When generating a response, π generates a sequence of reasoning steps, denoted as $R = \{r_1, r_2, \ldots, r_T\}$, where T represents the total number of reasoning steps to the given mathematical problem. The above process can be explained as follows:

$$R = \pi(Q \parallel P), \tag{1}$$

where $\parallel$ denotes the concatenation of the problem Q and the prompt P, and $\pi(\cdot)$ represents the inference process of LLM.

3.1.2 PRM

A single instance in a training dataset $\mathcal{D}$ to train a PRM comprises three components: (1) a problem statement, (2) a generated response consisting of multiple inference steps, and (3) a corresponding set of binary labels, each taking a value of either 0 or 1, indicating whether the associated reasoning step is incorrect or correct, respectively.

During training, the PRM is optimized using cross-entropy loss and supervised to align its predictions with the ground-truth labels. After being trained, the PRM model is capable of processing new reasoning steps generated by the LLM in response to a given mathematical problem, which means that the PRM is able to assign a scalar score to each individual reasoning step, reflecting the model's confidence in the correctness of each step:

$$f_{\text{PRM}} : (Q, R) \mapsto (s_1, s_2, \ldots, s_T), \tag{2}$$

where $f_{\text{PRM}} : (\cdot)$ represents the inference of PRM, $s_i \in [0, 1]$ denotes the confidence score assigned to the i-th reasoning step r_i, and T denotes the number of reasoning steps.

For generative PRM, the binary labels in the training dataset are replaced with textual analyses and judgments, each formulated as a textual choice such as "incorrect" or "correct". During inference, generative PRM also generates textual critiques and judgments for each step.

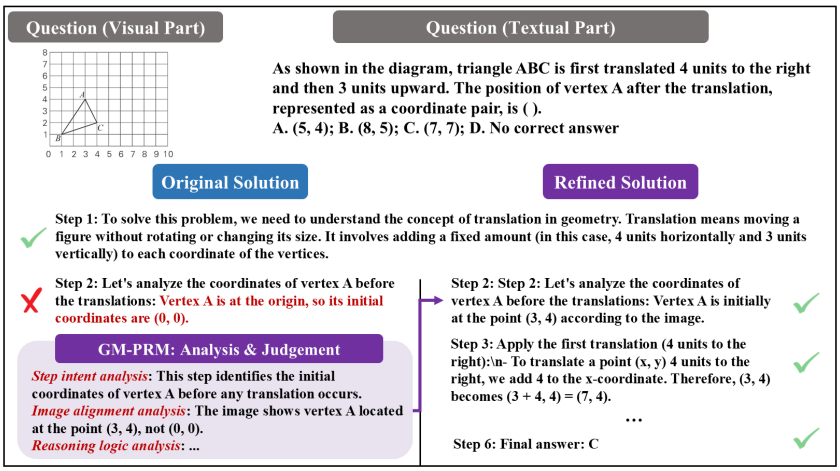


Figure 2: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the second step is incorrect and refines the second step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.

3.1.3 GM-PRM

By extending generative PRMs from the textual modality to a multimodal setting, we introduce GM-PRM. In this setting, mathematical problems are represented using both textual and visual information. The input to the policy model comprises the image of the problem, its textual description and task-specific instructions, which are processed jointly to generate reasoning steps. Similarly, during both training and inference, it is essential to provide GM-PRM with inputs from both visual and textual modalities, enabling it to perform cross-modal analysis when assigning correctness labels to each reasoning step:

$$f_{\text{GM-PRM}} : (Q, I, R) \mapsto (c_1, j_1, \ldots, c_T, j_T), \quad (3)$$

where $f_{\text{GM-PRM}} : (\cdot)$ represents the inference of GM-PRM, I denotes the image of the mathematical problem, c_i denotes the critique of the i-th reasoning step r_i, and j_i denotes the textual judgment assigned to the i-th reasoning step r_i.

3.2 Data Construction

In this section, we present our methodology employed to construct the training data for GM-PRM. The process consists of three key stages: (1) the selection of appropriate types and quantities of question data from the VisualPRM400K dataset (Wang et al., 2025b); (2) the generation of textual analysis and judgment data using GPT-4o; and (3) the filtering process of the generated data through MC estimation and LLM-as-a-judge techniques to ensure quality and reliability.

3.2.1 Data Selection

VisualPRM400K is a large-scale dataset containing approximately 400,000 multimodal process supervision samples. In our work, we select plane geometry- and function-related problems from VisualPRM400K to construct a specialized subset and supplement it with corresponding textual analysis for training GM-PRM. This targeted subset with textual critiques supports the effective training of GM-PRM, yielding strong performance on geometric and function-based mathematical reasoning tasks.

3.2.2 Generation of Analysis and Judgment

To obtain textual analyses and judgments, we employ GPT-4o to critique each reasoning step from 4 key aspects: **step intent**, **image alignment**, **reasoning logic**, and **step refinement**.

The aspect of **step intent** indicates identifying the purpose of each reasoning step. This initial analysis establishes a foundation that allows GM-PRM to interpret and evaluate each reasoning step in context more effectively. Furthermore, this level of understanding facilitates subsequent error detection and correction tasks, thereby enhancing the overall effectiveness of GM-PRM.

The second aspect is **image alignment**. When MLLMs are used for inference in solving multimodal problems, MLLMs often make errors in image alignment, such as misidentifying parallel relationships or incorrectly annotating angles, which leads to flawed solutions. To address this, we employ GPT-4o to produce textual analysis and judg-

ments in image alignment for inference steps, to form the dataset for training GM-PRM.

Reasoning logic is an indispensable presence in the step-by-step problem-solving process of MLLMs. However, the occurrences of logical inconsistencies and errors, such as miscalculations and incorrect inferences significantly impact the correctness of the reasoning steps and the final answers. Therefore, it is crucial for GM-PRM to be capable of identifying such logical flaws and making accurate judgments regarding the validity of the reasoning logic for each step. In our work, we employ GPT-4o to generate textual analysis and judgments of each step in reasoning logic to form the training dataset. The above process can be formulated as follows:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \{SI_i, IA_i, RL_i, FJ_i\}_{i=1}^{t}, \tag{4}$$

where $\mathcal{F} : (\cdot)$ represents the inference of GPT-4o, SI_i denotes the textual analysis of step intent for the i-th reasoning step, t denotes the number of the first incorrect step or the last step, $1 \leqslant t \leqslant T$, $IA_i = \{IAC_i, IAJ_i\}$ is the analysis which contains critique IAC_i and judgment IAJ_i in image alignment of the i-th reasoning step, $RL_i = \{RLC_i, RLJ_i\}$ denotes the analysis which contains critique RLC_i and judgment RLJ_i in reasoning logic of the i-th reasoning step, FJ_i denotes the final judgment of the i-th reasoning step.

Building on aforementioned three aspects, we further aim for GM-PRM to **correct the first identified erroneous step**. The above information enables GM-PRM to generate corrected reasoning steps that are logically coherent, visually accurate, and semantically aligned with the original step intent. The resulting corrected steps can then be used to construct more diverse and accurate inference solutions and ultimately produce more reliable final answers. In our work, we employ GPT-4o to generate a corrected version of the first identified error step in a reasoning process if the first error step is detected to exist by GPT-4o:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \begin{cases} RS, & \text{if incorrect step exists,} \\ \emptyset, & \text{otherwise.} \end{cases} \tag{5}$$

where RS denotes refined step of the first error step in a reasoning process.

In summary, we design a structured prompt for GPT-4o to generate comprehensive analysis data across four dimensions based on the provided problems, associated images, and step-by-step solutions:

$$\mathcal{F} : (Q, I, R \parallel P) \mapsto \mathcal{D}, \tag{6}$$

where $\mathcal{D}$ denotes the generated training dataset:

$$\mathcal{D} = \{(\{SI_i^k, IA_i^k, RL_i^k, FJ_i^k\}_{i=1}^{t}, RS^k)\}_{k=1}^{K}, \tag{7}$$

where $k \in \{1, 2, \ldots, K\}$ represents the k-th sample in the dataset, and K denotes the number of the training instances.

3.2.3 Data Filtering

The process of constructing training data using GPT-4o can be regarded as an implementation of LLM-as-a-judge methodology. Inspired by the combination of LLM-as-a-judge and MC estimation techniques (Zhang et al., 2025b), we employ the MC estimation technique proposed by Math-Shepherd (Wang et al., 2023) to effectively filter and curate the generated data.

Monte Carlo estimation is a strategy for automated annotation that leverages LLMs or MLLMs to generate multiple subsequent solutions for each step. When applying MC estimation to evaluate a step r_i, we use an LLM or an MLLM as a 'completer' to finalize multiple subsequent reasoning processes from this step:

$$f_{completer} \mapsto \{(r_{i+1}^j, \ldots, r_{L_j}^j, a^j)\}_{j=1}^{m}, \tag{8}$$

where a^j is the final answer of the j-th finalized solution and L_j is the total number of steps.

Within MC estimation, one type of evaluation method is commonly applied: hard estimation. In hard estimation, a step r_i is deemed correct if at least one subsequent solution reaches the correct final answer a^*; otherwise, it is incorrect:

$$l_i^{HE} = \begin{cases} 1, & \exists a_j, a_j = a^*, \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

In our data construction process, we employ hard estimation to label the correctness of individual reasoning steps. By integrating LLM-as-a-judge technique and MC estimation, we compare the labels acquired by MC estimation and judgments generated by GPT-4o. Data samples that receive consistent evaluations from both methods are selected as our final training dataset. By integrating these two methods, we aim to further enhance the reliability and quality of the training data, ensuring better performance of GM-PRM.

3.3 Refined-BoN Process

When applying Test-time Scaling (TTS) for LLMs and MLLMs, a widely adopted method is Best-of-N (BoN) approach. In the BoN process, a policy model is employed to generate N candidate solutions, which are then evaluated by reward models or self-consistency to select the optimal solution. However, during the BoN process, policy models are under identical prompting conditions when generating multiple solutions, which leads to the problem that the solutions often lack diversity and may exhibit limited correctness. In our work, we propose a novel Refined-BoN framework utilizing TTS techniques to enhance the diversity and accuracy of generated solutions, thereby improving the reasoning capabilities of policy models.

3.3.1 Refined-BoN Method

As shown in Figure 2, in Refined-BoN process, we first employ an MLLM as the policy model to generate $N/2$ initial solutions to a multimodal problem, and then these solutions are evaluated step-by-step by GM-PRM. For the subsequent $N/2$ solutions, the policy model generates them under varying conditions, informed by the evaluation of the preceding $N/2$ solutions: If GM-PRM identifies an incorrect reasoning step within a solution, it stops evaluating and refines the first erroneous step by generating a corrected version. This corrected step, along with all previously validated correct steps, is then input back into the policy model to continue the solution generation process. Conversely, if GM-PRM determines that all steps in a particular solution are correct, we employ the policy model to generate a new solution using the same prompt. Through this regeneration mechanism, we obtain the additional $N/2$ solutions. Then, we employ GM-PRM to evaluate the subsequent $N/2$ solutions.

3.3.2 Solution Selection

After applying the Refined-BoN process, we obtain N solutions for each problem, each accompanied by step-level correctness judgments. Moreover, we divide all the solutions into two categories: one where GM-PRM judges that it contains incorrect steps, and the other where GM-PRM judges that all its steps are correct. Furthermore, we take the corresponding probability of GM-PRM generating the associated "Correct" and "Incorrect" tokens as the score of each step.

Among the N generated solutions, if there exist solutions in which all reasoning steps are judged correct, we calculate the average of the scores of all steps in these solutions as the overall score of the solution, and select the solution with the highest average score as the optimal solution.

For N solutions to the problem, if GM-PRM determines that all N solutions contain incorrect steps, we calculate the average score of all steps in each solution as the overall score of the solution, and select the solution with the highest overall score as the final answer.

4 Experiments

In this section, we introduce our experimental setup to assess GM-PRM under the Refined-BoN process on five multimodal mathematical benchmarks in Section 4.1. In addition, we present the results of our experiments and three conclusions analyzed from the results in Section 4.2. Finally, we show the ablation studies in Section 4.3.

4.1 Experimental Setup

4.1.1 Benchmarks

We evaluate GM-PRM across five datasets, including MathVista (Lu et al., 2023), MathVision (Wang et al., 2024), MathVerse (Zhang et al., 2024), DynaMath (Zou et al., 2024) and WeMath (Qiao et al., 2024). The datasets contain diverse problem types, such as plane geometry, functions, puzzle tests, *etc*. We use Vision-Only subset of MathVerse dataset and Plane-Geometry subset of DynaMath.

4.1.2 Settings

We employ GM-PRM as the critic model for Refined-BoN evaluation and set N to 8 by default. We select six models as the policy models to generate step-by-step reasoning processes. When reasoning, we set the temperature of the policy models to 0.7 and top-p to 0.9. For comparison, we use the average accuracy of N sets of answers generated by policy models as baselines.

4.1.3 Training Details

To train GM-PRM, we use Qwen2.5-VL-7B-Instruct as our base model and perform supervised fine-tuning (SFT) with all parameters trainable but the frozen Vision Transformer (ViT) encoder. During the training process, we utilize bfloat16 mixed-precision and DeepSpeed with zero3 technology and set the training consists of 2 epochs. For batch size, the batch size on each training device is set to 2, and through gradient accumulation, the effective batch size is extended to 16. Moreover, we use two

MLLMs	MathVista	MathVision	MathVerse	DynaMath	WeMath	Average
MiniCPM-V2.6-8B	44.3	16.0	18.9	22.6	38.6	28.1
+ GM-PRM (Ours)	**51.0**	**18.1**	**24.4**	**25.7**	**51.0**	**34.0**
Improvements	+6.7	+2.1	+5.5	+3.1	+12.4	+5.9
Llama-3.2-11B-Vision	44.5	14.3	16.5	28.4	46.1	30.0
+ GM-PRM (Ours)	**49.5**	**18.2**	**18.8**	**32.7**	**53.4**	**34.5**
Improvements	+5.0	+3.9	+2.3	+4.3	+7.3	+4.5
Qwen2.5-VL-7B	63.2	25.1	32.8	35.0	60.6	43.3
+ GM-PRM (Ours)	**65.0**	**28.2**	**37.4**	**39.2**	**69.0**	**47.8**
Improvements	+1.8	+3.1	+4.6	+4.2	+8.4	+4.5
InternVL3-8B	50.6	20.3	25.0	27.0	50.9	34.8
+ GM-PRM (Ours)	**55.7**	**22.2**	**31.7**	**33.4**	**59.2**	**40.4**
Improvements	+5.1	+1.9	+6.7	+6.4	+8.3	+5.6
InternVL3-38B	68.3	34.9	37.8	40.1	66.4	49.5
+ GM-PRM (Ours)	**69.9**	**37.0**	**39.1**	**43.1**	**72.9**	**52.4**
Improvements	+1.6	+2.1	+1.3	+3.0	+6.5	+2.9
InternVL3-78B	68.0	34.6	36.0	38.1	65.7	48.5
+ GM-PRM (Ours)	**70.7**	**37.1**	**40.6**	**39.9**	**72.2**	**52.1**
Improvements	+2.7	+2.5	+4.6	+1.8	+6.5	+3.6

Table 1: Percentage accuracy scores (%) of multiple MLLMs across five datasets. For each MLLM, the first row shows the baseline, the second shows the final result with GM-PRM, and the third shows the improvement. Only positive improvements are underlined. The best results are highlighted in **bold**. All values are reported after rounding to one decimal place.

A800 GPUs to train GM-PRM, and the AdamW optimizer is used with an initial learning rate of 1×10^{-5}. The learning rate schedule involves a linear warm-up with the warm-up ratio equal to 0.05 followed by linear decay.

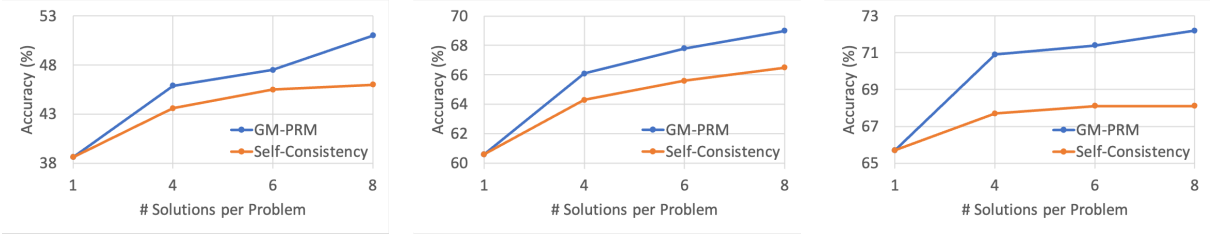
4.2 Main Results

As shown in Table 1, integrating GM-PRM with the Refined-BoN process consistently improves performance across five different benchmark datasets for six different MLLMs. On average, our method yields notable accuracy gains, with improvements of +5.9 for MiniCPM-V2.6-8B, +4.5 for Llama-3.2-11B-Vision, +4.5 for Qwen2.5-VL-7B, and +5.6 for InternVL3-8B.

A closer look at dataset-level results reveals that the improvements are not uniform. The WeMath benchmark shows the most significant enhancement, with MiniCPM-V2.6-8B improving by +12.4 points, highlighting the ability of our method to strengthen mathematical reasoning on challenging problems. Similarly, MathVerse and DynaMath exhibit consistent gains of +4.5–6.7 points across multiple models, suggesting that our approach particularly benefits datasets requiring complex symbolic manipulation and multi-step reasoning. In contrast, MathVision improvements are more modest (+1.9–3.9), indicating that the visual reasoning component may already be relatively strong in baseline models.

GM-PRM combined with the Refined-BoN process demonstrates strong generalization across diverse multimodal mathematical problems, with particularly remarkable gains in plane geometry tasks. As illustrated in Figure 4, even after excluding plane geometry and function problems, policy models still achieve notable improvements across the datasets. This indicates that although GM-PRM is primarily trained on a dataset composed of plane geometry and function problems, it generalizes effectively to other types of multimodal mathematical problems. Moreover, as shown by the averaged results in Figure 4, the improvements achieved by GM-PRM with Refined-BoN on plane geometry problems consistently exceed those on the overall dataset, function problems, and other categories, underscoring the exceptional effectiveness of our method in tackling plane geometry tasks.

The Refined-BoN process yields disproportionately larger gains for models with lower baseline performance. As shown in Table 1, InternVL3-38B starts with the highest initial average accuracy among all policy models (49.5%) and achieves a modest improvement of +2.9 points (+5.9%). In contrast, Qwen2.5-VL-7B, which has the highest baseline accuracy (43.3%) among models with fewer than 12 billion parameters, improves by +4.5 points (+10.4%), surpassing the relative gains of InternVL3-38B. Notably, MiniCPM-V-2.6-



(a) MiniCPM-V-2.6-8B (b) Qwen2.5-VL-7B (c) InternVL3-78B

Figure 3: The results of changing the value of N in the Refined-BoN process on the WeMath across different policies. As N increases, the effectiveness of GM-PRM in enhancing accuracy improves and surpasses that of Self-Consistency.

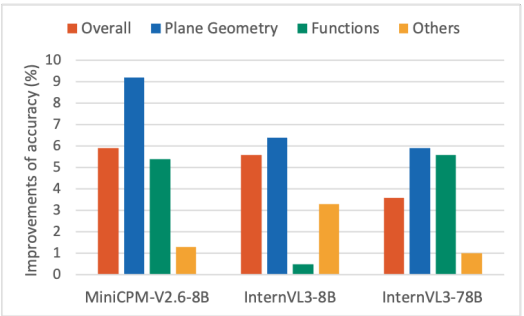


Figure 4: Improvements of the average percentage accuracy (%) of multiple MLLMs across different question types in MathVista, MathVision and MathVerse.

8B demonstrates the most significant relative improvement, achieving +5.9 points (+21.0%), despite its lower initial score. These results suggest that models with weaker baseline performance benefit more from the refinement mechanism of GM-PRM with Refined-BoN, likely because the process effectively corrects errors in reasoning steps, leaving greater room for improvement.

4.3 Number of solution samples N in Refined-BoN

Following Test-time Scaling technique, we vary the number of N in the Refined-BoN process to evaluate the performance of GM-PRM in comparison to the Self-Consistency baseline.

Figure 3 depicts WeMath accuracy as the number of sampled solutions per problem (N) increases from 1 to 8. Across all three backbones—MiniCPM-V-2.6-8B, Qwen2.5-VL-7B, and InternVL3-78B—both GM-PRM and the Self-Consistency (SC) baseline benefit from a larger sampling budget, yet GM-PRM exhibits a noticeably steeper growth curve.

Under the widely adopted Best-of-8 setting, GM-PRM delivers gains of 4.9 and 3.5 over SC on MiniCPM-V-2.6-8B and Qwen2.5-VL-7B, respectively. Even for the 78B-parameter InternVL3, GM-PRM maintains a substantial 4.1 margin. These results indicate that the proposed refinement strategy in this paper not only scales to larger models but also converts additional candidate solutions into accuracy more effectively than Self-Consistency, thereby underscoring the robustness and versatility of GM-PRM.

Furthermore, for MiniCPM-V-2.6-8B, GM-PRM surpasses the self-consistency baseline by 2.1, 2.2, and 4.9 points under the Best-of-4, Best-of-6, and Best-of-8 settings, respectively, indicating a steadily increasing performance gap between GM-PRM and self-consistency as N increases.

5 Conclusion

In this work, we introduced GM-PRM, a novel paradigm that transforms the reward model from a passive judge into an active reasoning collaborator for multimodal mathematics problem solving. By providing fine-grained, interpretable analysis and, more critically, generating step-level corrections for erroneous steps, GM-PRM moves beyond simple binary verification. This unique corrective capability powers our Refined Best-of-N (Refined-BoN) framework, which actively improves flawed reasoning trajectories during inference at test time. Our experiments demonstrate that this approach achieves state-of-the-art results on multiple benchmarks, significantly boosting policy model performance with remarkable data efficiency. The consistent gains across diverse models and problem types underscore the robustness and generalizability of our method. This shift from passive error detection to generative, collaborative correction represents a crucial advance in multimodal reasoning.

Limitations

Our approach has several limitations that are important to note. First, GM-PRM is trained on a curated subset of VisualPRM400K focusing primarily on plane geometry and function-related problems. Although we observe encouraging transfer to other categories (e.g., Figure 4), performance may degrade under larger domain shifts such as puzzle-style questions, diagram-heavy word problems, or open-ended proof-like reasoning that requires global restructuring rather than local step repair. Second, our training signals depend on GPT-4o-generated analyses/refinements and are further filtered using hard Monte Carlo (MC) estimation; both components can introduce systematic label noise and biases (e.g., overconfidence on visually ambiguous cues or preference for certain solution styles), which may be inherited by GM-PRM. Finally, our refinement mechanism corrects only the first detected erroneous step; when errors are entangled across multiple steps, or when fixing one step necessitates revising earlier assumptions or downstream derivations, the refinement may be insufficient and can occasionally yield plausible but incorrect corrections. In addition, we freeze the ViT encoder during SFT, which may limit gains on inputs where stronger low-level visual perception is the primary bottleneck.

Acknowledgement

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); CAAI-Ant Group Research Fund; Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 5 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 4 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: a survey. *arXiv preprint arXiv:2402.12451*.

Qi Cao, Ruiyi Wang, Ruiyi Zhang, Sai Ashish Somayajula, and Pengtao Xie. 2025. Dreamprm: Domain-reweighted process reward model for multimodal reasoning. *arXiv preprint arXiv:2505.20241*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. *Preprint*, arXiv:2312.14238.

Lingxiao Du, Fanqing Meng, Zongkai Liu, Zhixiang Zhou, Ping Luo, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-prm: Enhancing multimodal mathematical reasoning with scalable step-level supervision. *arXiv preprint arXiv:2505.13427*.

Jiaxuan Gao, Shusheng Xu, Wenjie Ye, Weilin Liu, Chuyi He, Wei Fu, Zhiyu Mei, Guangju Wang, and Yi Wu. 2024. On designing effective rl reward at training time for llm reasoning. *arXiv preprint arXiv:2410.15115*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Pengfei Hu, Zhenrong Zhang, Qikai Chang, Shuhang Liu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, Feng Ma, and 1 others. 2025. Prm-bas: Enhancing multimodal reasoning through prm-guided beam annealing search. *arXiv preprint arXiv:2504.10222*.

Jiahao Huo, Yibo Yan, Boren Hu, Yutao Yue, and Xuming Hu. 2024. Mmneuron: Discovering neuron-level domain-specific interpretation in multimodal large language model. *arXiv preprint arXiv:2406.11193*.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *Preprint*, arXiv:2504.16828.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, L Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, and 1 others. 2024. Rewardbench: Evaluating reward models for language modeling. arxiv 2024. *arXiv preprint arXiv:2403.13787*.

Qingyun Li, Zhe Chen, Weiyun Wang, Wenhai Wang, Shenglong Ye, Zhenjiang Jin, Guanzhou Chen, Yinan He, Zhangwei Gao, Erfei Cui, Jiashuo Yu, Hao Tian, Jiasheng Zhou, Chao Xu, Bin Wang, Xingjian Wei, Wei Li, Wenjian Zhang, Bo Zhang, and 3 others. 2024. Omnicorpus: A unified multimodal corpus of 10 billion-level images interleaved with text. *Preprint*, arXiv:2406.08418.

Wendi Li and Yixuan Li. 2024. Process reward model with q-value rankings. *arXiv preprint arXiv:2410.11287*.

Xiang Li, Haiyang Yu, Xinghua Zhang, Ziyang Huang, Shizhu He, Kang Liu, Jun Zhao, Fei Huang, and Yongbin Li. 2025a. Socratic-prmbench: Benchmarking process reward models with systematic reasoning patterns. *arXiv preprint arXiv:2505.23474*.

Zichao Li, Xueru Wen, Jie Lou, Yuqiu Ji, Yaojie Lu, Xianpei Han, Debing Zhang, and Le Sun. 2025b. The devil is in the details: Tackling unimodal spurious correlations for generalizable multimodal reward models. *arXiv preprint arXiv:2503.03122*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Wei Liu, Junlong Li, Xiwen Zhang, Fan Zhou, Yu Cheng, and Junxian He. 2024a. Diving into self-evolving training for multimodal reasoning. *Preprint*, arXiv:2412.17451.

Yangzhou Liu, Yue Cao, Zhangwei Gao, Weiyun Wang, Zhe Chen, Wenhai Wang, Hao Tian, Lewei Lu, Xizhou Zhu, Tong Lu, Yu Qiao, and Jifeng Dai. 2024b. Mminstruct: a high-quality multi-modal instruction tuning dataset with extensive diversity. *Science China Information Sciences*, 67(12).

Zhaoyang Liu, Yinan He, Wenhai Wang, Weiyun Wang, Yi Wang, Shoufa Chen, Qinglong Zhang, Zeqiang Lai, Yang Yang, Qingyun Li, Jiashuo Yu, Kunchang Li, Zhe Chen, Xue Yang, Xizhou Zhu, Yali Wang, Limin Wang, Ping Luo, Jifeng Dai, and Yu Qiao. 2023. Interngpt: Solving vision-centric tasks by interacting with chatgpt beyond language. *Preprint*, arXiv:2305.05662.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. *Preprint*, arXiv:2504.02495.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

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.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Ruihang Chu, Jin Zeng, and Yujiu Yang. 2025. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *Preprint*, arXiv:2501.04686.

Bingchen Miao, Yang Wu, Minghe Gao, Qifan Yu, Wendong Bu, Wenqiao Zhang, Yunfei Li, Siliang Tang, Tat-Seng Chua, and Juncheng Li. 2025. Boosting virtual agent learning and reasoning: A step-wise, multi-dimensional, and generalist reward model with benchmark. *arXiv preprint arXiv:2503.18665*.

Runqi Qiao, Qiuna Tan, Guanting Dong, Minhui Wu, Chong Sun, Xiaoshuai Song, Zhuoma GongQue, Shanglin Lei, Zhe Wei, Miaoxuan Zhang, and 1 others. 2024. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.

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. *arXiv preprint arXiv:2410.08146*.

Shuaijie She, Junxiao Liu, Yifeng Liu, Jiajun Chen, Xin Huang, and Shujian Huang. 2025. R-prm: Reasoning-driven process reward modeling. *Preprint*, arXiv:2503.21295.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-llava: Bootstrapping mathematical reasoning for multimodal large language models. *arXiv preprint arXiv:2406.17294*.

Mingyang Song, Zhaochen Su, Xiaoye Qu, Jiawei Zhou, and Yu Cheng. 2025. Prmbench: A fine-grained and challenging benchmark for process-level reward models. *arXiv preprint arXiv:2501.03124*.

Changyao Tian, Xizhou Zhu, Yuwen Xiong, Weiyun Wang, Zhe Chen, Wenhai Wang, Yuntao Chen, Lewei Lu, Tong Lu, Jie Zhou, Hongsheng Li, Yu Qiao, and Jifeng Dai. 2024. Mm-interleaved: Interleaved image-text generative modeling via multi-modal feature synchronizer. *Preprint*, arXiv:2401.10208.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.

Peiyi Wang, Lei Li, Zhihong Shao, RX Xu, Damai Dai, Yifei Li, Deli Chen, Y Wu, and Zhifang Sui. 2023. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR, abs/2312.08935*.

Shuai Wang, Zhenhua Liu, Jiaheng Wei, Xuanwu Yin, Dong Li, and Emad Barsoum. 2025a. Athena: Enhancing multimodal reasoning with data-efficient process reward models. *Preprint*, arXiv:2506.09532.

Weiyun Wang, Zhangwei Gao, Lianjie Chen, Zhe Chen, Jinguo Zhu, Xiangyu Zhao, Yangzhou Liu, Yue Cao, Shenglong Ye, Xizhou Zhu, and 1 others. 2025b. Visualprm: An effective process reward model for multimodal reasoning. *arXiv preprint arXiv:2503.10291*.

Yibo Yan and Joey Lee. 2024. Georeasoner: Reasoning on geospatially grounded context for natural language understanding. In *Proceedings of the 33rd ACM international conference on information and knowledge management*, pages 4163–4167.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2024a. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *arXiv preprint arXiv:2412.11936*.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, and 1 others. 2024b. Errorradar: Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.

Yibo Yan, Shen Wang, Jiahao Huo, Jingheng Ye, Zhendong Chu, Xuming Hu, Philip S Yu, Carla Gomes, Bart Selman, and Qingsong Wen. 2025a. Position: Multimodal large language models can significantly advance scientific reasoning. *arXiv preprint arXiv:2502.02871*.

Yibo Yan, Shen Wang, Jiahao Huo, Philip S Yu, Xuming Hu, and Qingsong Wen. 2025b. Mathagent: Leveraging a mixture-of-math-agent framework for real-world multimodal mathematical error detection. *arXiv preprint arXiv:2503.18132*.

Yibo Yan, Haomin Wen, Siru Zhong, Wei Chen, Haodong Chen, Qingsong Wen, Roger Zimmermann, and Yuxuan Liang. 2024c. Urbanclip: Learning text-enhanced urban region profiling with contrastive language-image pretraining from the web. In *Proceedings of the ACM Web Conference 2024*, pages 4006–4017.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Huarong Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, Xu Han, and 4 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *Preprint*, arXiv:2408.01800.

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*.

Thomas Zeng, Shuibai Zhang, Shutong Wu, Christian Classen, Daewon Chae, Ethan Ewer, Minjae Lee, Heeju Kim, Wonjun Kang, Jackson Kunde, and 1 others. 2025. Versaprm: Multi-domain process reward model via synthetic reasoning data. *arXiv preprint arXiv:2502.06737*.

Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. 2025a. Openprm: Building open-domain process-based reward models with preference trees. In *The Thirteenth International Conference on Learning Representations*.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025b. The lessons of developing process reward models in mathematical reasoning. *Preprint*, arXiv:2501.07301.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and Bowen Zhou. 2025. Genprm: Scaling test-time compute of process reward models via generative reasoning. *Preprint*, arXiv:2504.00891.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024a. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.

Kening Zheng, Junkai Chen, Yibo Yan, Xin Zou, and Xuming Hu. 2024b. Reefknot: A comprehensive benchmark for relation hallucination evaluation, analysis and mitigation in multimodal large language models. *arXiv preprint arXiv:2408.09429*.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. 2025. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *arXiv preprint arXiv:2504.12328*.

Wanrong Zhu, Jack Hessel, Anas Awadalla, Samir Yitzhak Gadre, Jesse Dodge, Alex Fang, Youngjae Yu, Ludwig Schmidt, William Yang Wang, and Yejin Choi. 2023. Multimodal c4: An open, billion-scale corpus of images interleaved with text. *Preprint*, arXiv:2304.06939.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2025. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26183–26191.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*.

A Appendix

A.1 More Related Work

A.1.1 Multimodal Large Language Models (MLLMs)

The advancement of artificial intelligence has advanced the development of Multimodal Large Language Models (MLLMs). MLLMs extend the capabilities of language-centric models by integrating multiple sensory inputs, primarily visual and auditory, with text. Unlike traditional Large Language Models (LLMs) which process solely textual data, MLLMs are designed to perceive and reason across modalities such as vision and language, thereby achieving the fusion and interaction of multimodal information. The development of MLLMs has been driven by extensive efforts, including enhancements across model structure and data curation. In terms of model structure, multiple studies (Bai et al., 2025; Liu et al., 2023; Yao et al., 2024) achieve notable performance through a method that utilizing connectors to align the embeddings of vision from Vision Foundation Models (VFMs) (Chen et al., 2024) with the latent space of LLMs (Bai et al., 2023; Touvron et al., 2023a,b). Alternatively, another line of research (Grattafiori et al., 2024; Tian et al., 2024) enhances pre-trained LLMs by adding supplementary layers to integrate visual features, which reduces the number of visual tokens but incurs additional training costs. Regarding dataset curation, recent research has achieved substantial advancements. Specifically, MultimodalC4 (Zhu et al., 2023) extends C4 corpus containing only text with images and constructs a corpus that supports pre-training for MLLMs. Furthermore, OmniCorpus (Li et al., 2024) delivers a large-scale yet noisy multimodal dataset suitable for pre-training, and MMInstruct (Liu et al., 2024b) presents an open-source collection of high-quality data designed for instruction tuning. The majority of research efforts have been concentrated on the training processes of MLLMs, leaving significant room for exploration in Test-Time Scaling (TTS) technique. In our work, we investigate the potential of enhancing the performance of MLLMs by incorporating Process Reward Model (PRM) into the TTS framework.

A.2 More Ablation Study

A.2.1 Methods for aggregating step scores

For PRMs, the method used to aggregate step scores into an overall solution score plays a critical role. In this part, we compare several different aggregation strategies, including averaging step scores, selecting the maximum step score, and selecting the minimum step score. Since step-by-step solutions that contain steps judged incorrect are often not evaluated or scored for all steps, this experiment focuses exclusively on solutions where all steps are judged correct.

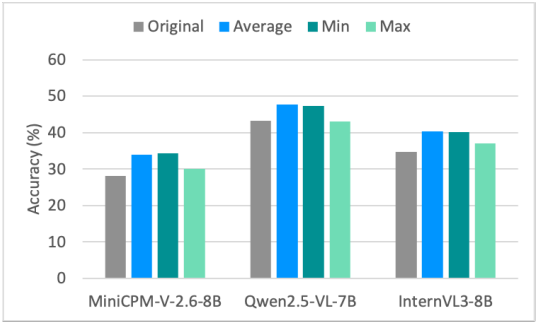


Figure 5: Average percentage accuracy (%) of MLLMs via different aggregation methods across five datasets.

The results are illustrated in the Figure 5. Across all policy models and datasets, we find that both averaging the step scores and selecting the minimum score significantly outperform the strategy of selecting the maximum score. This suggests that either the average or the minimum score provides

a more accurate reflection of the overall quality of a solution than the maximum score. Between the minimum and average aggregation methods, we observe that averaging performs slightly better. This improvement may stem from the fact that the average score takes into account all problem-solving steps, providing a more comprehensive evaluation, whereas the minimum score reflects only the step with the lowest score and thus offers a less holistic assessment.

A.2.2 Refined-BoN vs. BoN

The Refined-BoN process aims to enhance the diversity of N candidate solutions by refining the steps judged incorrect and integrating the refined steps with the steps judged correct into the prompt for the policy models. In this part, we use the Pass@k metric to evaluate the diversity and accuracy of policy models in generating multiple solutions to the given problems.

The results are summarized in the Table 2. Overall, the Refined-BoN process improves Pass@8 scores compared to the standard BoN process across multiple policy models and five benchmark datasets. Specifically, it increases the average Pass@8 values of MiniCPM-V-2.6-8B, Llama-3.2-11B-Vision, and InternVL3-8B by 0.9, 1.3, and 0.9 points, respectively, across the five datasets, demonstrating the effectiveness of the Refined-BoN approach in enhancing the diversity and correctness of the generated solutions.

MLLMs	BoN	Refined-BoN	Diff.
MiniCPM-V-2.6-8B	62.5	**63.4**	+0.9
Llama-3.2-11B-Vision	62.7	**64.0**	+1.3
InternVL3-8B	65.3	**66.2**	+0.9

Table 2: Average percentage Pass@8 scores of BoN and Refined-BoN across five datasets for different models.

A.3 Benchmark

We provide more details about the Refined-BoN test benchmarks in Table 3:

Benchmarks	Split	# Sample
DynaMath	Plane Geometry	770
MathVerse	Vision-Only	788
MathVista	Testmini	1000
WeMath	Testmini	1740
MathVision	Full	3040

Table 3: More details about the Refined-BoN test benchmarks.

A.4 Dataset

To ensure a balanced distribution of process labels, we carefully construct the training dataset. The final dataset used to train GM-PRM contains 19,614 samples in total, comprising 9,061 solutions that contain incorrect steps—as jointly identified by GPT-4o and Monte Carlo (MC) estimation—and 10,553 solutions in which all steps are judged to be correct.

A.5 Prompt

In this section, we introduce the prompts used to construct the training dataset and generate the reasoning processes and final answers. The prompt we guide the policy models to generate reasoning processes and final answers of multi-choice problems is represented in Figure 6.

Prompt for generating reasoning of multi-choice problems:

You are an expert in solving multimodal mathematical problems. I will provide a mathematical problem along with its corresponding image. According to the problem and the image, please first conduct step-by-step reasoning, and after your reasoning, please provide the correct option letter (e.g., A, B, C, D, E) using the format: "Final answer: ..."

Problem:
<Question>

Figure 6: Prompt for policy models to generate reasoning and final answers of multi-choice problem.

The prompt we guide the policy models to generate reasoning processes and final answers of free-form problems is represented in Figure 7.

Prompt for generating reasoning of free-form problems:

You are an expert in solving multimodal mathematical problems. I will provide a mathematical problem along with its corresponding image. According to the problem and the image, please first conduct step-by-step reasoning, and after your reasoning, please provide your final answer using the format: "Final answer: ..."

Problem:
<Question>

Figure 7: Prompt for policy models to generate reasoning and final answers of free-form problem.

The prompt we use to employ GPT-4o to generate the training dataset is shown in Figure 8.

A.6 More Cases

We collect 2 more cases of GM-PRM and the Refined-BoN process illustrated in Figure 9 and 10.

Prompt:

You are an expert in solving multimodal mathematical problems. You will be given:
1. A multimodal mathematical problem and its corresponding image.
2. A multiple-step solution (each step on a new line).

Task:
The tasks you need to do are:
1. Analyze the purpose of each step and what specific actions were taken in each step.
2. Analyze each step's correctness in terms of image alignment and reasoning logic.
- Image alignment: Whether the information and reasoning used in the step are consistent with the content of the provided image.
- Reasoning logic: Whether the reasoning is logically sound, calculations are correct, and information used matches that from previous steps and question.
When outputting judgements, you must choose one output from "Correct" or "Incorrect".
3. For the first incorrect step, correct it based on your analysis of its error, and output the corrected step at the end of your output.

Question:
The multimodal mathematical problem is as follows:
<Question>

Solution Steps:
The multiple-step solution is as follows:
<Solution Steps>

Figure 8: Prompt for GPT-4o to construct training dataset.

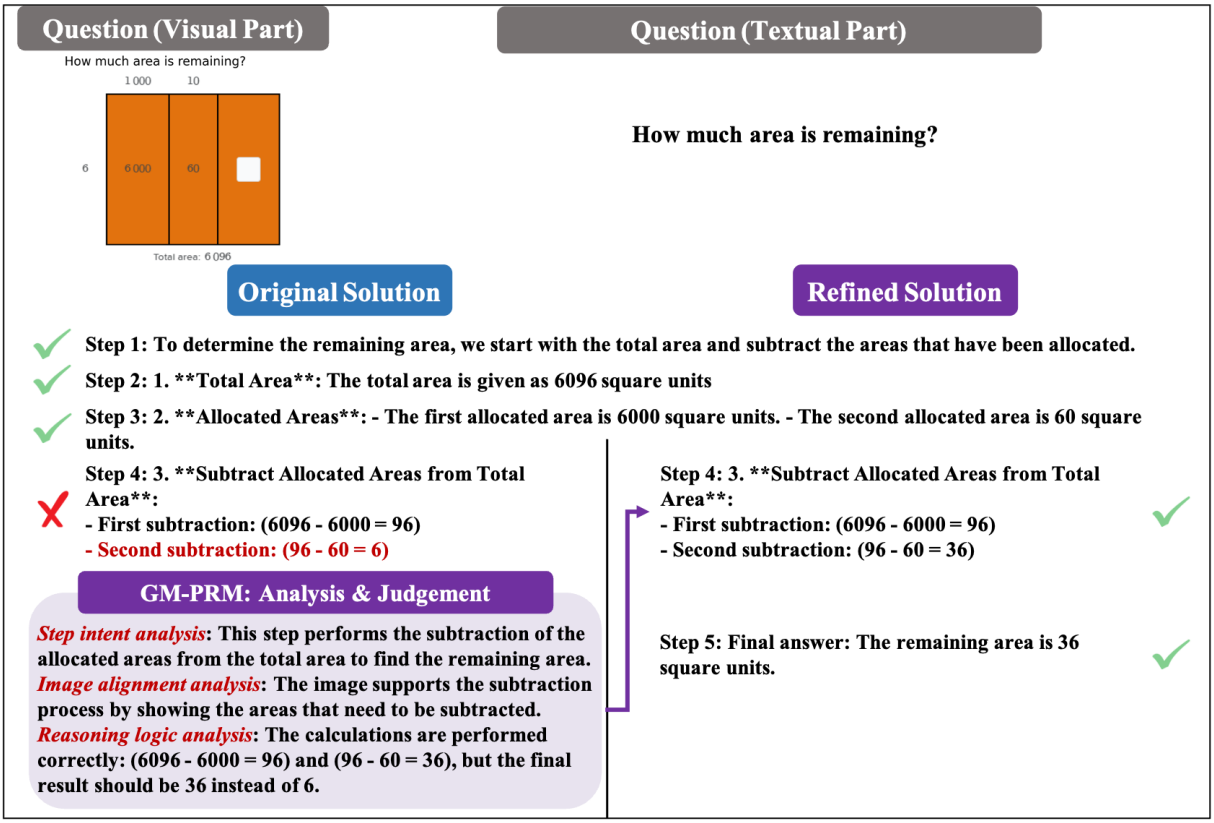
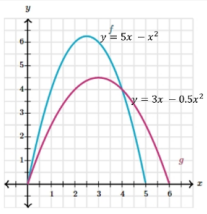


Figure 9: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the fourth step is incorrect regarding reasoning logic and refines the fourth step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.

Question (Visual Part)



Question (Textual Part)

In both equations x represents rainfall (in centimeters). When there is 0cm of rainfall, the number of mosquitos is the same as the number of bats. What is another rainfall amount where the number of mosquitos is the same as the number of bats?
Round your answer to the nearest half centimeter.

Original Solution

Step 1: To solve the problem, we need to find the point where the number of bats equals the number of mosquitos, other than the known 0 cm of rainfall.

Step 2: 1. **Identify the equations for mosquitoes and bats: - Mosquitoes: ($y = 5x - x^2$) - Bats: ($y = 3x - 0.5x^2$)**

Step 3: 2. **Set the equations equal to each other to find the points of intersection: [$5x - x^2 = 3x - 0.5x^2$]**

Step 4: 3 **Simplify the equation: [$5x - x^2 = 3x - 0.5x^2$] Rearrange the equation to collect like terms: [5x - =x - $x^2 + 0.5x^2 = 0$] [5x - =x + $0.5x^2 - x^2 = 0$] [5x - =x - $0.5x^2 = 0$] [5x - =x - $0.5x^2 = 0$] [5x - =x - $0.5x^2 = 0$]**

GM-PRM: Analysis & Judgement

***Step intent analysis*: This step simplifies the equation set in Step 3 to prepare it for solving.**
***Image alignment analysis*: The simplification process should reflect the intersection points seen in the image.**
***Reasoning logic analysis*: There is a mistake in the simplification process. The rearrangement and simplification steps contain errors in algebraic manipulation.**

Refined Solution

Step 4: 3. **Simplify the equation:** [$5x - x^2 = 3x - 0.5x^2$] Rearrange the equation to collect like terms: [5x - x^2 - = x + $0.5x^2 = 0$] [2x - $0.5x^2 = 0$] [$0.5x^2$ - 2x = 0] [x(0.5x - 2) = 0] This corrected simplification should lead to the correct roots for further analysis.**

Step 5: **Solve for (x): [x(0.5x - 2) = 0] This gives us two possible solutions for (x): 1. (x = 0) 2. (0.5x - 2 = 0) [0.5x = 2] [x = 4]**

...

Step 8: Final answer: (4) cm.

Figure 10: The illustration of a representative example *before* and *after* applying GM-PRM. In particular, GM-PRM first judges the steps of the original solution generated by the policy model. Subsequently, GM-PRM finds that the fourth step is incorrect regarding reasoning logic and refines the fourth step to generate the correct version. The correct steps are input to the policy model to generate the refined solution, and finally the correct answer is obtained.