Benchmarking Fine-Grained Error Detection in Multimodal Reasoning

Chi-Min Chan[1*] **Han Zhu**[1*] **Chunyang Jiang**[1]
Jiaming Ji[2] **Juntao Dai**[2] **Wei Xue**[1] **Sirui Han**[1†] **Yike Guo**[1†]
[1]The Hong Kong University of Science and Technology
[2]Peking University
cchanbc@connect.ust.hk

Abstract

Multimodal Process Reward Models (MPRMs) have emerged as a pivotal framework for enhancing the reasoning capabilities of Multimodal Large Language Models (MLLMs). However, the research community currently lacks a dedicated benchmark to rigorously assess the error discernment capabilities of these models. To address this gap, we introduce ***PRMBench-V***, a novel benchmark specifically designed to evaluate MPRMs' proficiency in detecting erroneous reasoning steps across diverse error categories. Leveraging a semi-automated annotation pipeline augmented with human verification, we construct a comprehensive dataset comprising **907 unique queries**, each annotated with **nine distinct error types**, resulting in **8,163 test cases** with fine-grained step-level error labels. Through extensive experiments involving over 16 open- and closed-source models, we uncover several key findings: (1) even the strongest existing MPRMs achieve only ~30% accuracy in error identification; (2) while partial error detection achieves moderate precision and recall (~60%), overall accuracy remains low (~20%); and (3) benchmark scores exhibit a strong correlation with downstream task performance gains (r=0.86). Furthermore, we demonstrate that *PRMBench-V* can inform the development of more robust MPRMs: by introducing the **Bayesian Rater Reliability Process Reward Model (BR^2-PRM)**, we achieve up to a **4.8%** performance improvement through test-time scaling. We believe that *PRMBench-V* will serve as a valuable resource for advancing MPRM research, enabling more rigorous evaluation and fostering the development of models with fine-grained multimodal reasoning capabilities.

1 Introduction

The rapid advancement of MLLMs has demonstrated their remarkable ability to tackle complex reasoning tasks (Hurst et al., 2024; Bai et al., 2025; Zhu et al., 2025; Team et al., 2023; Yakun et al., 2026; Zhou et al., 2026). However, their progress hinges on robust evaluation methodologies, particularly for reasoning processes (Li et al., 2025b). While existing methods like outcome-based reward models assess final outputs (Lambert et al., 2024; Zhong et al., 2025; Chan et al., 2025a), they often overlook errors in intermediate reasoning steps (Skalse et al., 2022; Xia et al., 2025), potentially rewarding correct answers derived from flawed logic. Process reward models (PRMs) address this by evaluating reasoning chains (Lightman et al., 2023; Zhao et al., 2025; Khalifa et al., 2025; Chan et al., 2025b; Shi et al., 2025), yet existing benchmarks that evaluate the error-discernment capability of these PRMs focus solely on the unimodal domain (Zheng et al., 2024; Song et al., 2025). While the application of PRMs in the multimodal domain is an emerging frontier, their systematic evaluation remains in its infancy, creating a critical barrier to assessing MPRMs for vision-language reasoning tasks.

To bridge this gap, we introduce *PRMBench-V*, a benchmark designed to rigorously evaluate MPRMs by measuring their ability to identify fine-grained errors in vision-language reasoning steps. By using a carefully designed automated annotation pipeline augmented with human verification, we construct a comprehensive benchmark comprising 907 unique queries, each annotated with 9 distinct error types, resulting in 8,163 error-labeled reasoning chains. This benchmark provides a granular and scalable framework for assessing PRMs in multimodal contexts.

In our study, we conduct extensive experiments and evaluate more than 16 state-of-the-art open- and closed-source models, including GPT-4o (Hurst et al., 2024), Gemini (Team et al., 2023), InternVL3 (Zhu et al., 2025), and Qwen2.5-VL (Bai et al., 2025). Trough rigorous analy-

[*] Equal contribution.
[†] Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 44672–44702

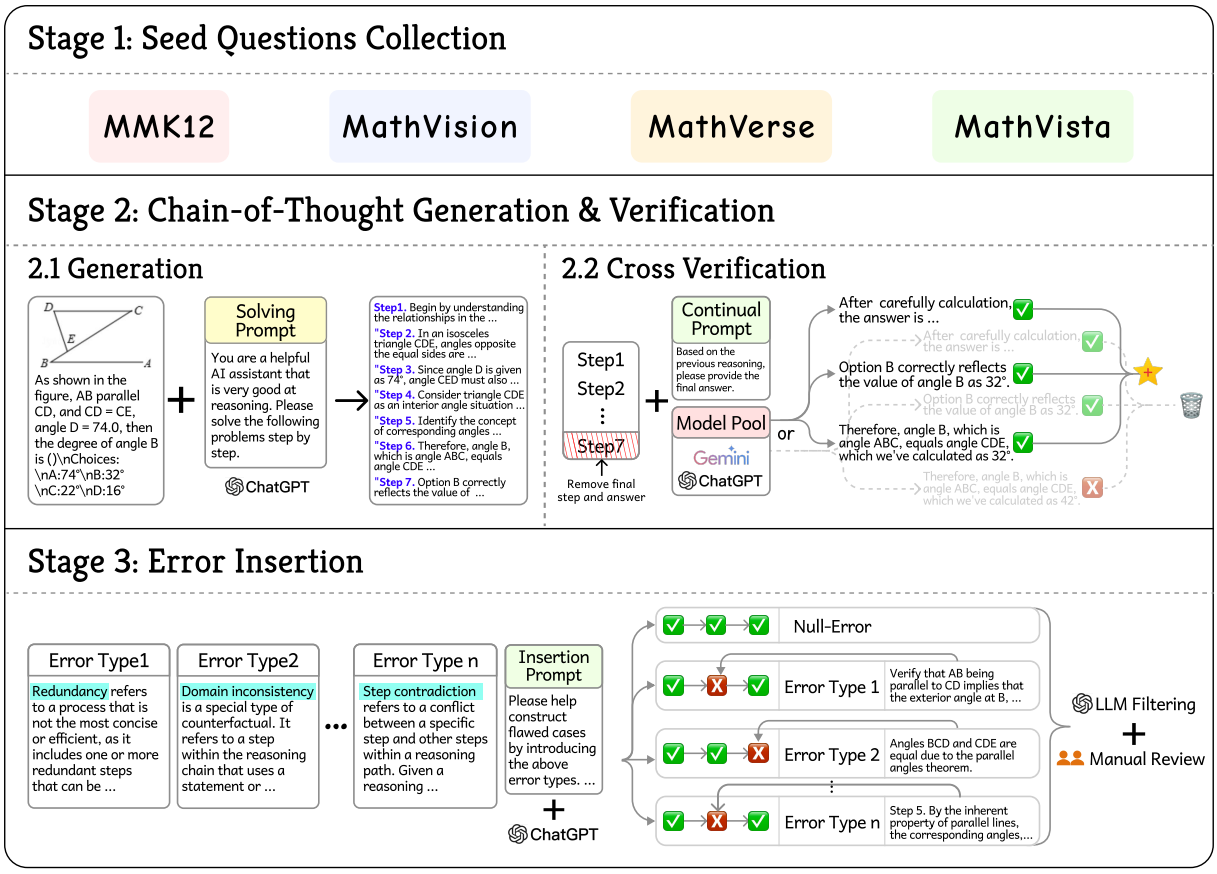


Figure 1: Overview of the pipeline for constructing *PRMBench-V*.

sis, our key findings reveal: 1) **Limited performance across models**: Even the most advanced closed-source models struggle to achieve high accuracy in identifying all errors, with an average accuracy of only 30%. 2) **Error-type disparity**: Models exhibit significant variation in performance across error types, with domain inconsistency detection being notably stronger (20% higher accuracy) compared to other error types. 3) **Partial error recognition**: While most models achieve moderate precision and recall (~60%) in detecting some intermediate errors, they fail to comprehensively identify all errors, as reflected in their low overall accuracy (~20%). 4) **Strong correlation with downstream tasks**: Our results demonstrate that MPRMs with higher benchmark scores achieve better downstream task performance in test-time scaling scenarios ($r=0.86$, $p<0.05$). By leveraging this insight, we design **BR^2-PRM**, which integrates rater reliability into reward weighting during multiple candidates selection. This method improves performance by up to **4.8%** under test-time scaling, compared with simply averaging the fine-grained process scores.

We anticipate that *PRMBench-V* will serve as a foundational resource for advancing MPRMs, fostering more rigorous evaluation and targeted improvements in fine-grained reasoning.

2 Related Works

2.1 Process Reward Model

Reward modeling has become central to large language model (LLM) development, serving as both a quality metric (Lambert et al., 2024) and a cornerstone of reinforcement learning from human feedback (RLHF) (Stiennon et al., 2020). The field has evolved from outcome-based models (Li et al., 2025a; Lambert et al., 2024; Yasunaga et al., 2025) toward process-based reward models (PRMs) that assess intermediate reasoning steps (Lightman et al., 2023; Zhao et al., 2025; Wang et al., 2025; Tu et al., 2025; Gao et al., 2025; Chan et al., 2026). PRMs have shown strong utility in applications such as candidate selection (Lightman et al., 2023; She et al., 2025; Zhang et al., 2025), guided generation (Ma et al., 2023; Zhang et al., 2024a), and online reinforcement learning (Yuan et al., 2024; Cui et al., 2025). Recent benchmarks (Zheng et al., 2024; Song et al., 2025) evaluate PRMs

by inserting reasoning errors to test fine-grained judgment. However, these efforts remain largely unimodal, leaving a gap in multimodal reasoning evaluation (Luo et al., 2025; Thawakar et al., 2025; Xu et al., 2024). As multimodal foundation models advance (Sun et al., 2024; Du et al., 2025), corresponding multimodal PRMs (MPRMs) and their evaluation become essential. To address this, we introduce *PRMBench-V*, a comprehensive benchmark for assessing MPRMs. Unlike prior work, our benchmark (1) provides fine-grained, step-level annotations for multimodal reasoning chains, (2) ensures uniform coverage of nine error types per query to avoid evaluation bias, and (3) exhibits stronger scaling correlation with downstream task performance compared to Song et al. (2025). By enabling precise and balanced evaluation, *PRMBench-V* facilitates the development of more reliable MPRMs for next-generation multimodal reasoning systems.

2.2 Multi-modal Reasoning Benchmark

Recent reasoning-enhanced language models such as OpenAI's o1/o3 (Jaech et al., 2024) and DeepSeek's R1 (Guo et al., 2025) have demonstrated remarkable problem-solving capabilities through chain-of-thought reasoning (Wei et al., 2022). Inspired by these advances, researchers have extended similar reasoning paradigms to multimodal domains, enabling models to integrate visual perception with structured reasoning (Thawakar et al., 2025; Xu et al., 2024; Huang et al., 2025). This emerging paradigm of vision-reasoning aims to endow models with human-like cognitive abilities that allow them to observe visual inputs and engage in deliberate reasoning. To evaluate these multimodal reasoning capabilities, a number of benchmarks have been developed that combine visual understanding with mathematical and logical problem solving (Lu et al., 2023; Zhang et al., 2024b; Wang et al., 2024a). However, most existing evaluations focus primarily on final-answer accuracy, which provides limited insight into the correctness or quality of intermediate reasoning steps. In response, more fine-grained evaluation frameworks have been proposed to assess the validity of each reasoning step (Xu et al., 2025; Ai et al., 2025; Thawakar et al., 2025; Yan et al., 2024), reflecting a growing emphasis on process-level assessment. We provide direct comparison with related benchmarks in Appendix B and Table 4. Building on this progress, our work introduces an MPRM benchmark that advances beyond standard meta-evaluation by offering actionable guidance for downstream applications. We explicitly connect these evaluations to practical utility through our proposed BR^2-PRM framework, enhancing the robustness and interpretability of multimodal reasoning assessment.

3 PRMBench-V

In this section, we introduce *PRMBench-V*, a comprehensive benchmark for evaluating MPRMs through systematic error detection in visual reasoning tasks. Our benchmark addresses the critical gap in assessing MLLMs' ability to identify and localize reasoning errors across multiple error types in multimodal contexts. Below, we detail the benchmark construction pipeline.

3.1 Data Curation Pipeline

Our data curation follows a systematic three-stage pipeline designed to generate high-quality reasoning chains with controlled error insertion, as illustrated in Figure 1.

3.1.1 Stage 1: Seed Questions Collection

We aggregate seed questions from four established multimodal reasoning datasets: MMK12, MathVision, MathVista, and MathVerse. This diverse collection ensures comprehensive coverage of multiple domains including mathematics, biology, physics, geometry and chemistry. In total, our benchmark consists of 5 main categories and 43 sub categories, providing a balanced representation across different reasoning paradigms and visual complexity levels. The illustrative statistics of *PRMBench-V* are shown in Table 3 and Figure 2. Full details can be found in Appendix A.

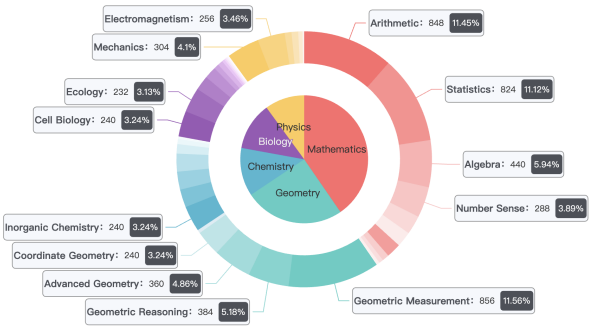


Figure 2: Taxonomy of *PRMBench-V*. The benchmark is organized hierarchically into 5 major categories and 43 sub-categories.

Model	Overall	NR.	NCL.	ES.	SC.	DC.	CI.	PS.	DR.	VP.
Proprietary Multimodal LLMs										
Gemini-1.5-pro	0.34	0.21	0.51	0.18	0.38	0.76	0.26	0.18	0.24	0.30
Gemini-2.0-flash	0.33	0.21	0.54	0.22	0.30	0.64	0.27	0.19	0.28	0.29
GPT-4o	0.27	0.23	0.44	0.14	0.30	0.58	0.20	0.15	0.14	0.20
Open-sourced Multimodal LLMs										
InternVL-3-78B	0.28	0.26	0.51	0.12	0.41	0.69	0.16	0.13	0.14	0.14
InternVL-2.5-78B	0.28	0.26	0.50	0.17	0.31	0.64	0.13	0.17	0.16	0.20
Gemma-3-27B-it	0.28	0.19	0.47	0.12	0.21	0.71	0.20	0.18	0.21	0.24
Qwen-2.5-VL-72B-Instruct	0.26	0.26	0.50	0.10	0.38	0.64	0.13	0.09	0.11	0.11
Gemma-3-12B-it	0.22	0.21	0.43	0.04	0.21	0.69	0.08	0.10	0.09	0.14
InternVL-3-8B	0.20	0.22	0.42	0.07	0.18	0.41	0.13	0.10	0.12	0.16
Qwen-2-VL-72B-Instruct	0.18	0.11	0.29	0.06	0.29	0.55	0.08	0.05	0.09	0.08
Gemma-3-4B-it	0.13	0.03	0.39	0.07	0.17	0.13	0.09	0.10	0.08	0.13
Qwen-2.5-VL-7B-Instruct	0.08	0.04	0.17	0.01	0.12	0.26	0.03	0.02	0.02	0.05
InternVL-2.5-8B	0.09	0.06	0.20	0.05	0.10	0.10	0.06	0.06	0.05	0.09
Qwen-2-VL-7B-Instruct	0.02	0.01	0.04	0.01	0.02	0.01	0.01	0.00	0.01	0.07
Open-sourced Multimodal Process Reward Models										
URSA-RM-8B	0.16	0.05	0.25	0.22	0.14	0.07	0.28	0.13	0.20	0.15
MM-PRM	0.14	0.02	0.02	0.25	0.15	0.09	0.29	0.10	0.17	0.18

Table 1: Performance comparison of proprietary and open-source multimodal LLMs across error categories.

3.1.2 Stage 2: Chain-of-Thought Generation & Verification

COT Generation Since manual annotation of correct reasoning chains would be time-consuming and unscalable, we instruct strong MLLMs to generate detailed step-by-step reasoning chains for each seed question. Specifically, we prompt GPT-4o with temperature $= 1.0$ to provide comprehensive solutions with explicit step numbering and clear logical progression, producing high-quality reasoning chains as shown in the mid-left panel of Figure 1.

COT Verification Given that our framework requires inserting errors into reasoning chains, we must first ensure the correctness of the generated chains. We implement a simple, yet effective, verification strategy to automatically validate the quality of the reasoning chain. Specifically, we truncate the final calculation step of each generated reasoning chain and provide the preceding steps to other MLLMs. The models are then asked to generate the final answer based solely on the reasoning steps provided. If the models can successfully arrive at the correct final answer using the truncated reasoning chain, we consider the chain to be logically sound and retain it for error insertion. Conversely, if the models fail to reach the correct conclusion based on the provided reasoning steps, we conclude that the chain contains inherent logical flaws that prevent other models from following the reasoning to the correct answer and subsequently filter out such chains. During this stage, we employ a model pool including Gemini-2.5-flash, GPT-4o and Qwen-2.5-VL-72B-Instruct with temperature $= 1.0$ to conduct cross-verification. Only those instances that are answered correctly by all models in the pool are retained. This process ensures that only high-quality, logically consistent reasoning chains proceed to the error insertion stage, establishing a reliable foundation for our benchmark construction.

3.1.3 Stage 3: Fine-Grained Error Insertion

Error Insertion We adopt a systematic error insertion framework that introduces controlled mistakes into verified reasoning chains while preserving the overall structure and reasoning context. Our framework encompasses multiple error categories that target different aspects of logical reasoning. We utilize a fine-grained error taxonomy aligned with prior work (Song et al., 2025) and adapt it to

target multi-modal context by introducing a new visual error category. We summarize distinct requirements on MPRMs as follows:

Non-Redundancy (NR.) demands the MPRMs to identify and justify which step(s) are logically superfluous.

Non-Circular Logic (NCL.) requires detecting self-referential loops where step S_n relies on step S_{n+k} that ultimately depends on S_n.

Empirical Soundness (ES.)/ **Step Consistency** (SC.)/ **Domain Consistency** (DC.) necessitate grounding verification against facts (empirical), internal consistency (step), or domain-specific rules (domain).

Confidence Invariance (CI.) involves flagging overconfident unsupported claims.

Prerequisite Sensitivity (PS.) and **Deception Resistance** (DR.) require contextual gap-filling and adversarial pattern recognition, respectively.

Visual Perception (VP.) extends multimodal evaluation by assessing the ability of MPRMs to identify and reason over visual misperceptions arising within reasoning processes.

Based on the above definitions, we adopt similar error insertion templates to prompt GPT-4o with temperature $= 1.0$ to insert errors in specific steps within verified chains. To be noted, the error insertion process requires careful handling of cascading effects, as introducing an error at one step often necessitates modifications to subsequent steps to maintain logical consistency within the flawed reasoning path. When an error is inserted, the following steps need to be adjusted to reflect the consequences of the mistake introduced, ensuring that the resulting chain remains coherent despite the fact that it contains the targeted error.

Quality Control We implement a quality control process involving manual verification by human annotators to ensure the reliability and validity of our error-inserted reasoning chains. Rather than requiring annotators to write reasoning chains or insert errors themselves, we employ a streamlined checklist-based approach where annotators verify a few key aspects to determine whether the instance should be retained or not, such as (1) whether the original reasoning chain is mathematically correct and logically sound, and (2) whether the inserted error matches the intended error type and is appropriately integrated into the reasoning flow. Full details of the annotation is shown in Appendix A.2. This human verification step filters out cases that is low-quality, ensuring that our benchmark contains error examples that provide meaningful challenges for MPRMs evaluation.

4 Experiments

4.1 Experimental Setup

We evaluate over 16 state-of-the-art MLLMs and scalar-based PRMs on *PRMBench-V*. For MLLMs, we adopt one-shot prompting (Zheng et al., 2023) to identify reasoning errors. For scalar-based PRMs (Du et al., 2025; Luo et al., 2025), we flag the lowest-scored step(s) as erroneous, aligning the count with ground-truth annotations to handle multi-step errors. By default, the experiments during the evaluation use consistent prompting and temperature $= 0.0$.

4.2 Evaluation Metrics

Given the multi-error nature of *PRMBench-V* where each reasoning instance may contain multiple error types, we employ following metrics:

- **Strict Accuracy**: An instance is only counted as correct if the model identifies *all* ground-truth errors *and* makes no false positives. This conservative measure reflects real-world utility, as missing even one error step may invalidate the entire reasoning chain.
- **Relaxed Precision & Recall**: Since strict accuracy is prohibitively challenging especially for instances with many erroneous steps, we complement it with instance-level precision and recall. This approach:
  - Accounts for partial correctness (e.g., detecting 2/3 errors in an instance still contributes to recall.)
  - Maintaining high precision ensures flagged errors are trustworthy, avoiding unnecessary corrections.

We computed final benchmark scores by macro-averaging instance-level metrics, ensuring equal weight per instance and mitigating bias from error-prone cases. This evaluation strategy balances *rigor* (via strict accuracy) with *practicality* (via relaxed precision/recall), addressing the inherent challenges of multi-error assessment.

4.3 Main Results

Table 1 presents the comprehensive evaluation results of models across *PRMBench-V*, revealing sev-

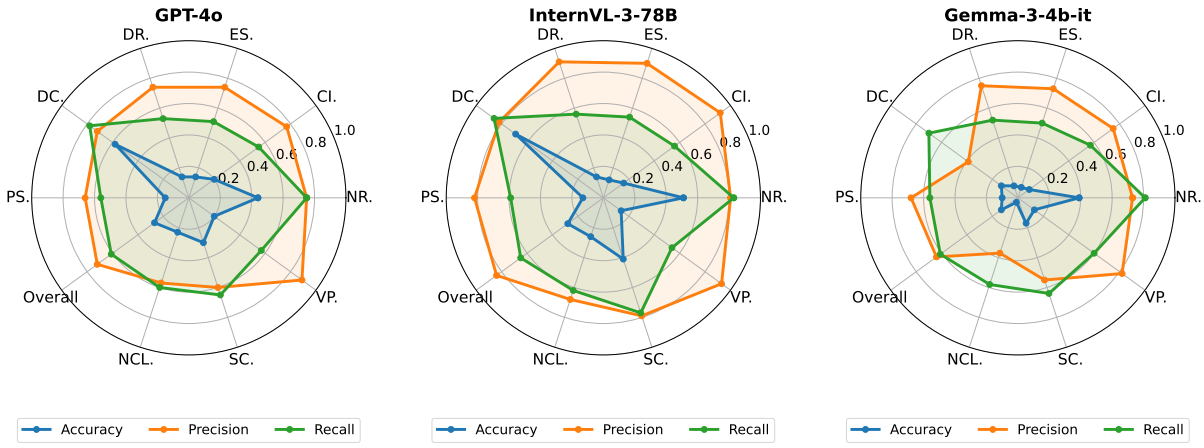


Figure 3: Illustrative accuracy, precision, and recall comparison for selected models. Full results are shown in Appendix E.

eral critical insights about the current capabilities and limitations of MPRMs.

Overall Performance Landscape. The results demonstrate that even the most advanced models struggle significantly with fine-grained error detection in multimodal reasoning. The best-performing model, gemini-1.5-pro, achieves only 0.34 overall accuracy, while the majority of open-source models fall below 0.30. This finding underscores a fundamental challenge in current MPRMs development, where models fail to comprehensively identify reasoning errors despite their sophisticated architectures.

Error Type Performance Disparities. The results reveal distinct performance patterns across error types that reflect the inherent strengths and limitations of current MPRMs. Models demonstrate significantly stronger performance in categories requiring structural and rule-based reasoning, such as NCL. and DC., where top models achieve scores ranging from 0.40-0.70. This superior performance aligns well with how MLLMs are fundamentally trained on logical patterns and structural relationships in text.

Categories that demand grounding in empirical evidence and temporal coherence, including ES. and SC., present moderate challenges for current models, with performance typically ranging from 0.20-0.40 for leading systems. While these scores indicate room for improvement, they suggest that models possess some capability for evidence-based reasoning validation.

In stark contrast, current models exhibit notable vulnerability in Sensitivity-Based error types; PS. and DR., where even the best-performing models struggle to exceed 0.30 accuracy. This poor performance highlights a critical limitation: current MLLMs lack robustness to perturbations and adversarial modifications in reasoning chains. Consistently low scores across PS. (0.10-0.28) and DR. (0.05-0.28) indicate that models are easily misled by subtle changes in problem conditions or deceptive reasoning steps, revealing a fundamental weakness in their error detection capabilities when faced with manipulated or perturbed inputs.

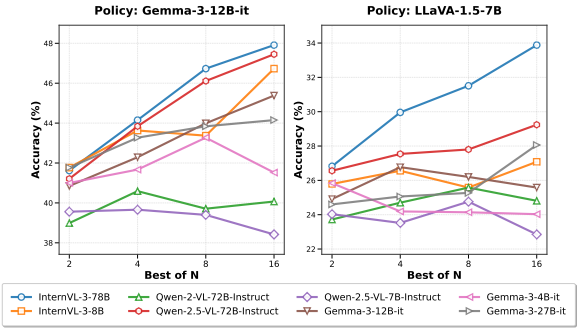


Figure 4: Accuracy scaling curve of two policy models on MMK12 when guided by different process reward models under Best-of-N sampling (N=2,4,8,16).

Scalar-Based MPRMs performance. Additionally, scalar-based MPRMs (URSA-RM-8B and MM-PRM) demonstrate competitive but not superior performance compared to general-purpose MLLMs, achieving 0.17 and 0.13 overall accuracy. Interestingly, these models show relatively balanced performance across error types, suggesting they may offer more consistent error detection capabilities, albeit at lower absolute performance levels.

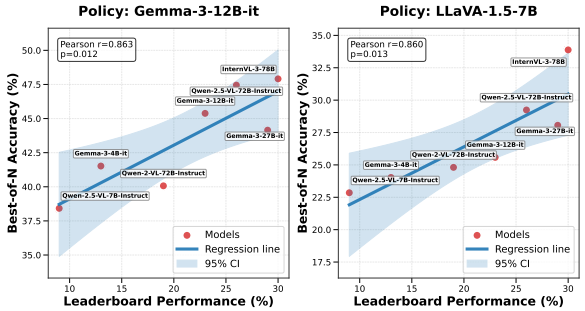


Figure 5: Best-of-16 Regression Analysis. The strong correlation ($r \approx 0.86$, $p < 0.05$) between our benchmark scores and downstream task (MMK12) performance validates that our metric effectively predicts real-world model capability.

Precision-Recall Analysis Reveals Systematic Patterns. The radar chart analysis of three representative models across different performance tiers in Figure 3 unveils consistent patterns in precision-recall trade-offs. Across all examined models (GPT-4o, InternVL-3-78B, and Gemma-3-4b-it), precision consistently outperforms both accuracy and recall across most error categories, suggesting that while models can correctly identify errors when they detect them, they suffer from incomplete error recognition. This pattern is particularly pronounced in CI. and ES. categories, where precision scores often exceed 0.6-0.8 while accuracy remains substantially lower. The precision-recall gap is most evident in the Sensitivity-Based categories (PS., DR.), where models demonstrate moderate precision (0.2-0.4) but extremely low recall, indicating they miss the majority of perturbation-based errors even when their identified errors are often correct.

These results highlight the significant room for improvement in MPRMs development and establish *PRMBench-V*as a challenging benchmark that effectively differentiates model capabilities across fine-grained error detection tasks.

4.4 Scaling Effect of Process-Based Verification

To assess the practical utility of MPRMs in improving reasoning performance, we conduct best-of-N evaluation experiments using two policy models (Gemma-3-12B-it, LLaVA-1.5-7B) paired with eight different MPRMs spanning various performance tiers. Figure 4 presents the scaling results on MMK12. Specifically, we generate multiple solutions with temperature $= 0.7$ for each policy model

Algorithm 1 BR^2-PRM: Calibration Stage (Learning Judge Reliability)

Require: Calibration set $\mathcal{D}_{\text{calib}} = \{(\mathbf{s}_i, \mathbf{z}_i)\}_{i=1}^{M}$ with $\mathbf{s}_i \in \{1, \ldots, L\}^K$, $\mathbf{z}_i \in \{0,1\}^K$; Dirichlet priors $\{\boldsymbol{\alpha}_{k,j} \in \mathbb{R}^L_{>0}\}_{k=1..K, j\in\{0,1\}}$; (optional) Beta priors for prevalence $\pi_k \sim \text{Beta}(a_k, b_k)$.

Ensure: Posterior hyperparameters $\{\boldsymbol{\alpha}'_{k,j}\}_{k=1..K,\, j\in\{0,1\}}$; prevalence estimates $\{\hat{\pi}_k\}_{k=1}^{K}$.

Initialize $\boldsymbol{\alpha}'_{k,j} \leftarrow \boldsymbol{\alpha}_{k,j}$ for all k, j.
Initialize counts $m_{k,1} \leftarrow 0$ for all k.
for $i = 1$ to M **do** ▷ accumulate counts by true state
  for $k = 1$ to K **do**
    $s \leftarrow s_{i,k}$ ▷ $s \in \{1, \ldots, L\}$
    $j \leftarrow z_{i,k}$ ▷ $j \in \{0, 1\}$
    $(\boldsymbol{\alpha}'_{k,j})_s \leftarrow (\boldsymbol{\alpha}'_{k,j})_s + 1$
    $m_{k,1} \leftarrow m_{k,1} + j$
  end for
end for
for $k = 1$ to K **do** ▷ prevalence estimation
  if Beta prior used **then**
    $\hat{\pi}_k \leftarrow \dfrac{a_k + m_{k,1}}{a_k + b_k + M}$
  else
    $\hat{\pi}_k \leftarrow \dfrac{m_{k,1}}{M}$ ▷ empirical prevalence
  end if
end for
return $\{\boldsymbol{\alpha}'_{k,j}\}$ and $\{\hat{\pi}_k\}$

under different N values, and utilize the MPRMs to select the best solution for evaluation. For selection, we employ a standard LLM-as-a-Judge paradigm, prompting the MPRMs to provide direct scores across different dimensions corresponding to each error type evaluated in this paper, and select the solution with the highest average score. Detailed prompts are provided in Appendix F.

The results demonstrate varied effectiveness of MPRM-guided sampling across different policy-reward model combinations. Specifically, we observe consistent improvements with increasing N values, with performance gains of 6-8 percentage points when moving from best-of-2 to best-of-16. This suggests that these models benefit substantially from having multiple reasoning attempts filtered by MPRMs. The choice of MPRMs also significantly impacts performance. InternVL-3-78B consistently serves as an effective reward model

Algorithm 2 BR^2-PRM: Inference and Best-of-N Selection

Require: Candidate set $\mathcal{T}_{\text{new}} = \{\mathbf{s}_i\}_{i=1}^{N}$ with $\mathbf{s}_i \in \{1, \ldots, L\}^K$; posterior hyperparameters $\{\boldsymbol{\alpha}'_{k,j}\}$ from Alg. 1; prevalence $\{\hat{\pi}_k\}$.

Ensure: Selected trace T^* and its Bayesian aggregate score $S_{\text{Bayes}}(T^*)$.

$S_{\max} \leftarrow -\infty, \quad i^* \leftarrow$ null
for $i = 1$ to N **do** ▷ score each candidate
  $\log S \leftarrow 0$ ▷ accumulate in log-space for stability
  for $k = 1$ to K **do**
    $s \leftarrow s_{i,k}$
    $p_0 \leftarrow \dfrac{(\boldsymbol{\alpha}'_{k,0})_s}{\sum_{\ell=1}^{L}(\boldsymbol{\alpha}'_{k,0})_\ell}$ ▷ $P(s \mid z = 0)$
    $p_1 \leftarrow \dfrac{(\boldsymbol{\alpha}'_{k,1})_s}{\sum_{\ell=1}^{L}(\boldsymbol{\alpha}'_{k,1})_\ell}$ ▷ $P(s \mid z = 1)$
    $e \leftarrow p_0(1 - \hat{\pi}_k) + p_1\hat{\pi}_k$ ▷ evidence $P(s)$
    $q \leftarrow \dfrac{p_0(1 - \hat{\pi}_k)}{e}$ ▷ $P(z = 0 \mid s)$ by Bayes rule
    $\log S \leftarrow \log S + \log q$
  end for
  $S \leftarrow \exp(\log S)$
  if $S > S_{\max}$ **then**
    $S_{\max} \leftarrow S, \quad i^* \leftarrow i$
  end if
end for
return $T^* \leftarrow T_{i^*}$ with $S_{\text{Bayes}}(T^*) \leftarrow S_{\max}$

across two policy models, while smaller reward models like Gemma-3-4B-it show more variable effectiveness. These findings highlight the importance of careful MRPMs selection and suggest that the optimal policy-reward model pairing depends on both model capabilities and the specific reasoning task characteristics.

4.5 Downstream Task Correlation

To validate the predictive utility of our benchmark, we analyze the correlation between model performance on *PRMBench-V* and their effectiveness in downstream multimodal reasoning tasks. Figure 5 presents the regression analysis between leaderboard performance (Overall's Accuracy) and best-of-16 performance on MMK12 across two policy models. The results demonstrate strong positive correlations in two policy models, with Pearson's correlation coefficients ranging around 0.86 and the p-value around 0.01. This consistent correlation pattern validates that models with superior error detection capabilities on *PRMBench-V* translate effectively to improved reasoning performance in downstream tasks when used as process reward models. More results on Out-of-Distribution (OOD) tasks and the correlation under different N values are shown in Appendix E.3. In overall, targeted improvements in error identification in our benchmark could yield proportional gains in downstream multimodal reasoning applications, providing a clear pathway for advancing MLLMs capabilities through process-based verification.

4.6 Bayesian Aggregation of Process Rewards (BR^2-PRM)

Building on our observation that different MPRMs exhibit heterogeneous reliability across error dimensions, we first introduce a simple reliability-weighted aggregation scheme before extending it to a fully Bayesian formulation.

Motivation. Empirical analyses reveal that individual MPRMs vary in their accuracy and consistency depending on the type of reasoning or process error under evaluation. To account for this, we initially adopt a weighted process scoring approach that adjusts each error type's contribution according to its benchmark-derived reliability. Specifically, we estimate a judge's reliability on each error type from *PRMBench-V*, normalize these values, and use them as deterministic weights when aggregating process-level rewards. This method improves upon naive averaging by amplifying signals from more dependable dimensions while attenuating noise from less reliable ones.

However, this deterministic weighting still assumes that reliability estimates are fixed and noise-free. In practice, rater reliability varies across individual instances and error categories. To overcome these limitations, we introduce **BR^2-PRM** (**B**ayesian **R**ater **R**eliability for **P**rocess **R**eward **M**odel), which reframes reward aggregation as a problem of probabilistic inference. BR^2-PRM models reliability as a latent variable inferred from calibration data (e.g., *PRMBench-V*) and explicitly accounts for both aleatoric and epistemic sources of uncertainty in the aggregation process. This Bayesian framework provides a principled mechanism for uncertainty-aware reward estimation and supports more robust aggregation when rater reliability is limited or uncertain. Due to space con-

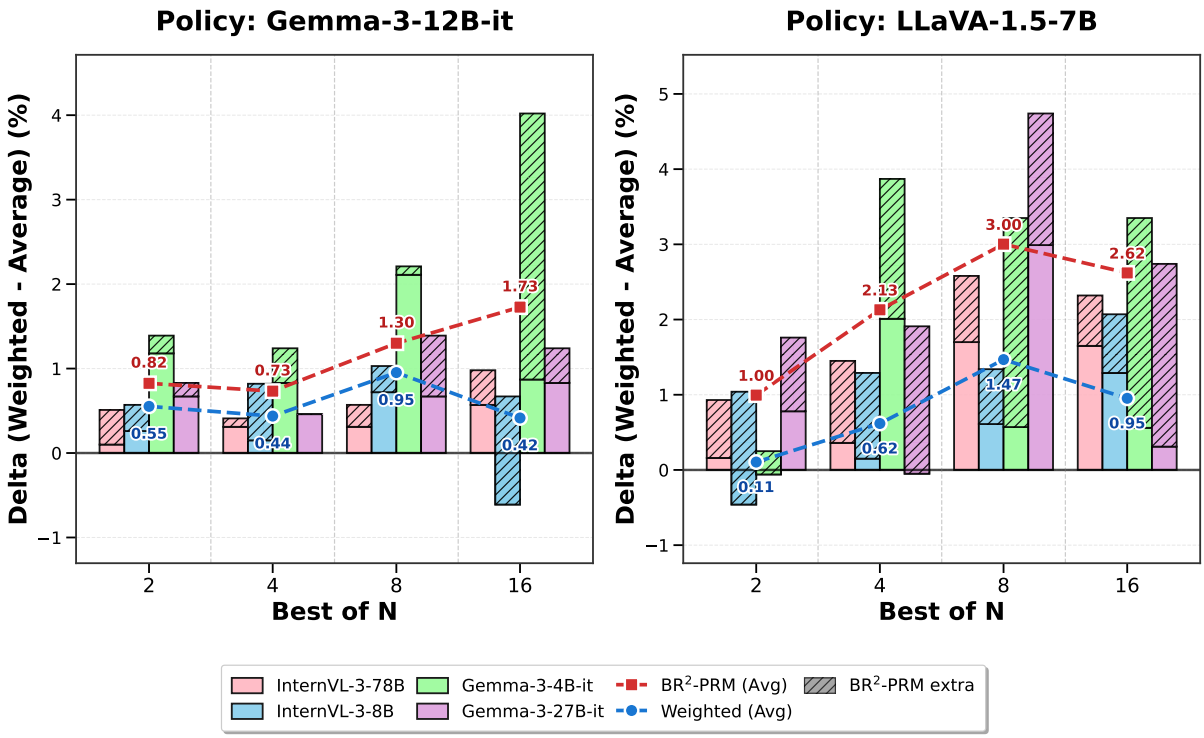


Figure 6: Benchmark-informed weighting and **BR^2-PRM** both improve response selection over naive averaging. **BR^2-PRM** further amplifies gains by explicitly modeling judge reliability, yielding up to **+4.8%** improvement on LLaVA-1.5-7B with Gemma-3-27B-it.

straints, we provide illustrative processes as shown in Algorithm 1 and 2. Further details are provided in Appendix D.

Model. For each trace T_i and error type k, the observed score $s_{i,k}$ is treated as a categorical draw conditioned on a latent binary error state $z_{i,k} \in \{0, 1\}$. The judge's reliability is captured by two Dirichlet-parameterized distributions: $\theta_{k,1} \sim \mathrm{Dir}(\boldsymbol{\alpha}_{k,1})$ for cases where an error is present and $\theta_{k,0} \sim \mathrm{Dir}(\boldsymbol{\alpha}_{k,0})$ otherwise. Given calibration data with known $(s_{i,k}, z_{i,k})$, we update $\boldsymbol{\alpha}'_{k,j} = \boldsymbol{\alpha}_{k,j} + \mathrm{count}(s_{i,k}, z_{i,k} = j)$, yielding posterior reliability parameters as detailed in Algorithm 1. For a new trace, the posterior predictive likelihood $P(s_{i,k} \mid z_{i,k})$ is computed in closed form, and Bayes' rule gives the probability that the trace is error-free for each dimension:

$$P_{0|s} = \frac{P_{s|0}(1 - \pi_k)}{P_{s|0}(1 - \pi_k) + P_{s|1}\pi_k}, \qquad (1)$$

where π_k is the estimated prevalence of error type k. The final Bayesian aggregate score is the joint posterior that a trace is free of all errors (Algorithm 2): $S_{\text{Bayes}}(T_i) = \prod_k P(z_{i,k} = 0 \mid s_{i,k})$. Selecting $\arg\max_i S_{\text{Bayes}}(T_i)$ is equivalent to the Bayes-optimal decision rule under a natural zero-one loss over latent errors.

Results. Figure 6 compare **BR^2-PRM** with average and weighted scoring across policy models and Best-of-N configurations. **BR^2-PRM** consistently achieves the highest downstream accuracy on MMK12, improving by 1.2% to 1.8% relative to weighted scoring and by up to 4.8% compared with simple averaging. These results demonstrate that uncertainty-aware, reliability-calibrated aggregation provides more reliable reasoning-trace selection.

5 Conclusion

We introduce *PRMBench-V*, a comprehensive benchmark for fine-grained error detection in multimodal reasoning. Our experiments expose notable weaknesses in current MPRMs, with top models reaching only 30% accuracy in identifying reasoning errors and showing clear disparities across error types. We also show a strong correlation between error-detection accuracy and downstream task performance. By offering a rigorous evaluation framework and actionable insights, *PRMBench-V* establishes a foundation for building more robust and reliable multimodal reasoning systems.

6 Ethical considerations

This work involved human annotators for verifying the accuracy and quality of automatically generated reasoning chains within the *PRMBench-V*dataset. All annotators were recruited from a pool of qualified experts with demonstrated competence in multimodal reasoning evaluation and were compensated at fair rates aligned with academic research standards. Detailed annotation protocols, including training procedures, quality-control measures, and verification criteria, are provided in Appendix A.2.

All participants provided informed consent after being briefed on the task objectives, expected workload, and data usage policies. The dataset construction exclusively utilized publicly available or appropriately licensed resources, in full compliance with the licenses of MathVista, MathVerse, MathVision, and MMK12 datasets. Furthermore, the data curation and verification protocols were reviewed and approved by the institutional Ethics Review Board (ERB). No personally identifiable or sensitive data were collected or processed at any stage of the study.

7 Limitations

Although *PRMBench-V* provides a benchmark dedicated to evaluating MPRMs, several limitations remain. First, while the benchmark spans five scientific domains and nine distinct error types, its scale is still moderate compared with real-world scenarios. Expanding both dataset coverage and the diversity of multimodal reasoning scenarios will further enhance robustness and generalizability.

Second, while our study systematically evaluates existing MPRMs and introduces the Bayesian Rater Reliability Process Reward Model **(BR^2-PRM)** as a principled aggregation framework, it does not address training-time optimization or reinforcement learning integration for improving model reliability. This is primarily due to the current lack of large-scale, high-quality fine-grained annotated multimodal datasets necessary for training such PRMs. Due to the reason that asking experts to annotate each step's correctness is time-consuming and labour-intensive, future work could leverage automated annotation pipelines to generate large-scale synthetic datasets, and apply weak-supervision techniques to convert noisy or heuristic weak labels into high-confidence strong labels

Lastly, the current annotation design of *PRMBench-V* deliberately focuses on structured scientific reasoning tasks (e.g., mathematics and physics). This methodological choice is driven by verifiability: scientific domains provide objective, binary ground truths that are critical for rigorously benchmarking MPRMs without introducing the noise of judge subjectivity. By establishing this verifiable stepping stone, we are able to provide more detailed step-wise supervision with higher information density. However, extending our benchmark to encompass open-domain multimodal reasoning such as creative writing and real-world visual narratives represents a crucial direction for future work. In these scenarios, establishing a unique "gold standard" for intermediate reasoning steps becomes inherently subjective. Therefore, conducting a more systematic examination of robustness within open-domain tasks will be a highly valuable avenue for subsequent research.

Acknowledgments

This work is funded in part by the HKUST Start-up Fund (R9911), Theme-based Research Scheme grant (T45-205/21-N), the InnoHK initiative of the Innovation and Technology Commission of the Hong Kong Special Administrative Region Government, and the research funding under HKUST-DXM AI for Finance Joint Laboratory (DXM25EG01).

References

Jiaxin Ai, Pengfei Zhou, Zhaopan Xu, Ming Li, Fanrui Zhang, Zizhen Li, Jianwen Sun, Yukang Feng, Baojin Huang, Zhongyuan Wang, and 1 others. 2025. Projudge: A multi-modal multi-discipline benchmark and instruction-tuning dataset for mllm-based process judges. *arXiv preprint arXiv:2503.06553*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, and 1 others. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Chi-Min Chan, Ehsan Hajiramezanali, Xiner Li, Edward De Brouwer, Carl Edwards, Wei Xue, Sirui Han, Yike Guo, and Gabriele Scalia. 2026. Dc-w2s: Dual-consensus weak-to-strong training for reliable process reward modeling in biological reasoning. *arXiv preprint arXiv:2603.08095*.

Chi-Min Chan, Chunpu Xu, Jiaming Ji, Zhen Ye, Pengcheng Wen, Chunyang Jiang, Yaodong Yang, Wei Xue, Sirui Han, and Yike Guo. 2025a. J1: Exploring simple test-time scaling for llm-as-a-judge. *arXiv preprint arXiv:2505.11875*.

Chi-Min Chan, Chunpu Xu, Junqi Zhu, Jiaming Ji, Donghai Hong, Pengcheng Wen, Chunyang Jiang, Zhen Ye, Yaodong Yang, Wei Xue, and 1 others. 2025b. Boosting policy and process reward models with monte carlo tree search in open-domain qa. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7433–7451.

Qiguang Chen, Libo Qin, Jin Zhang, Zhi Chen, Xiao Xu, and Wanxiang Che. 2024a. M^3cot: A novel benchmark for multi-domain multi-step multi-modal chain-of-thought. *arXiv preprint arXiv:2405.16473*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, and 1 others. 2024b. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, and 1 others. 2025. Process reinforcement through implicit rewards. *arXiv preprint arXiv:2502.01456*.

Alexander Philip Dawid and Allan M Skene. 1979. Maximum likelihood estimation of observer error-rates using the em algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, and 1 others. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

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*.

Minghe Gao, Xuqi Liu, Zhongqi Yue, Yang Wu, Shuang Chen, Juncheng Li, Siliang Tang, Fei Wu, Tat-Seng Chua, and Yueting Zhuang. 2025. Benchmarking multimodal cot reward model stepwise by visual program. *arXiv preprint arXiv:2504.06606*.

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

Wenxuan Huang, Bohan Jia, Zijie Zhai, Shaosheng Cao, Zheyu Ye, Fei Zhao, Zhe Xu, Yao Hu, and Shaohui Lin. 2025. Vision-r1: Incentivizing reasoning capability in multimodal large language models. *arXiv preprint arXiv:2503.06749*.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, and 1 others. 2024. Openai o1 system card. *arXiv preprint arXiv:2412.16720*.

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

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ 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 preprint arXiv:2403.13787*.

Lei Li, Yuancheng Wei, Zhihui Xie, Xuqing Yang, Yifan Song, Peiyi Wang, Chenxin An, Tianyu Liu, Sujian Li, Bill Yuchen Lin, and 1 others. 2025a. Vl-rewardbench: A challenging benchmark for vision-language generative reward models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24657–24668.

Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, and 1 others. 2025b. Perception, reason, think, and plan: A survey on large multimodal reasoning models. *arXiv preprint arXiv:2505.04921*.

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*.

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*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Ruilin Luo, Zhuofan Zheng, Yifan Wang, Xinzhe Ni, Zicheng Lin, Songtao Jiang, Yiyao Yu, Chufan Shi, Ruihang Chu, Jin Zeng, and 1 others. 2025. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*.

Qianli Ma, Haotian Zhou, Tingkai Liu, Jianbo Yuan, Pengfei Liu, Yang You, and Hongxia Yang. 2023. Let's reward step by step: Step-level reward model as the navigators for reasoning. *arXiv preprint arXiv:2310.10080*.

Fanqing Meng, Lingxiao Du, Zongkai Liu, Zhixiang Zhou, Quanfeng Lu, Daocheng Fu, Tiancheng Han, Botian Shi, Wenhai Wang, Junjun He, Kaipeng Zhang, Ping Luo, Yu Qiao, Qiaosheng Zhang, and Wenqi Shao. 2025. Mm-eureka: Exploring the frontiers of multimodal reasoning with rule-based reinforcement learning. *arXiv preprint arXiv:2503.07365*.

Alec Radford, Jong Wook Kim, Chris Hallacy, A. Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

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

Weijie Shi, Han Zhu, Jiaming Ji, Mengze Li, Jipeng Zhang, Ruiyuan Zhang, Jia Zhu, Jiajie Xu, Sirui Han, and Yike Guo. 2025. Legalreasoner: Step-wised verification-correction for legal judgment reasoning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 7297–7313. Association for Computational Linguistics.

Joar Skalse, Nikolaus H. R. Howe, Dmitrii Krasheninnikov, and David Krueger. 2022. Defining and characterizing reward gaming. 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*.

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*.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in neural information processing systems*, 33:3008–3021.

Kai Sun, Yushi Bai, Ji Qi, Lei Hou, and Juanzi Li. 2024. Mm-math: Advancing multimodal math evaluation with process evaluation and fine-grained classification. *arXiv preprint arXiv:2404.05091*.

Gemini Team and 1 others. 2023. Gemini: A family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Omkar Thawakar, Dinura Dissanayake, Ketan More, Ritesh Thawkar, Ahmed Heakl, Noor Ahsan, Yuhao Li, Mohammed Zumri, Jean Lahoud, Rao Muhammad Anwer, and 1 others. 2025. Llamav-o1: Rethinking step-by-step visual reasoning in llms. *arXiv preprint arXiv:2501.06186*.

Haoqin Tu, Weitao Feng, Hardy Chen, Hui Liu, Xianfeng Tang, and Cihang Xie. 2025. Vilbench: A suite for vision-language process reward modeling. *arXiv preprint arXiv:2503.20271*.

Aad W Van der Vaart. 2000. *Asymptotic statistics*, volume 3. Cambridge university press.

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

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024b. Measuring multimodal mathematical reasoning with math-vision dataset. *Preprint*, arXiv:2402.14804.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 27723–27730.

Guowei Xu, Peng Jin, Li Hao, Yibing Song, Lichao Sun, and Li Yuan. 2024. Llava-o1: Let vision language models reason step-by-step. *arXiv preprint arXiv:2411.10440*.

Zhaopan Xu, Pengfei Zhou, Jiaxin Ai, Wangbo Zhao, Kai Wang, Xiaojiang Peng, Wenqi Shao, Hongxun Yao, and Kaipeng Zhang. 2025. Mpbench: A comprehensive multimodal reasoning benchmark for process errors identification. *arXiv preprint arXiv:2503.12505*.

Cui Yakun, Yanting Zhang, Zhu Lei, Jian Xie, Zhizhuo Kou, Hang Du, Zhenghao Zhu, and Sirui Han. 2026. Mmfctub: Multi-modal financial credit table understanding benchmark. *arXiv preprint arXiv:2601.04643*.

Yibo Yan, Shen Wang, Jiahao Huo, Hang Li, Boyan Li, Jiamin Su, Xiong Gao, Yi-Fan Zhang, Tianlong Xu, Zhendong Chu, and 1 others. 2024. Errorradar:

Benchmarking complex mathematical reasoning of multimodal large language models via error detection. *arXiv preprint arXiv:2410.04509*.

Michihiro Yasunaga, Luke Zettlemoyer, and Marjan Ghazvininejad. 2025. Multimodal rewardbench: Holistic evaluation of reward models for vision language models. *arXiv preprint arXiv:2502.14191*.

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*.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. *Preprint*, arXiv:2303.15343.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. 2024a. Rest-mcts*: Llm self-training via process reward guided tree search. *Advances in Neural Information Processing Systems*, 37:64735–64772.

Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. 2025. 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. 2024b. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Jian Zhao, Runze Liu, Kaiyan Zhang, Zhimu Zhou, Junqi Gao, Dong Li, Jiafei Lyu, Zhouyi Qian, Biqing Qi, Xiu Li, and 1 others. 2025. Genprm: Scaling test-time compute of process reward models via generative reasoning. *arXiv preprint arXiv:2504.00891*.

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, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

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. *Preprint*, arXiv:2504.12328.

Yujin Zhou, Pengcheng Wen, Jiale Chen, Boqin Yin, Han Zhu, Jiaming Ji, Juntao Dai, Chi-Min Chan, and Sirui Han. 2026. What, whether and how? unveiling process reward models for thinking with images reasoning. In *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pages 29071–29079. AAAI Press.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

A More Details of PRMBench-V

A.1 Taxonomy of Categories & Statistics

Our benchmark encompasses five principal categories pertinent to the field of science: mathematics, biology, chemistry, physics, and geometry. Each primary category is further divided into sub-categories to provide a detailed and structured analysis. The statistical data and the hierarchical structure of these categories are illustrated in Table 3, which provides a visual representation of the comprehensive framework employed in our study.

Additionally, Figure 7 illustrates that the distribution of error step positions in our benchmark is left-skewed, indicating that most reasoning chains do not exceed a length of 12 steps. This also suggests that models tend to introduce error steps primarily within the initial stages of the reasoning process, thereby increasing the difficulty of error detection. Furthermore, Figure 8 demonstrates that the length of reasoning chains does not vary by more than three steps across all error types.

A.2 Human Annotation Details

To evaluate the quality of the original and modified steps, we provide several rules for annotators and require them to carefully review both the original steps and their corresponding modifications using platform in Figure 13. Specifically, our human verification stage prioritized the following key aspects.

For the original step quality assessment, annotators must determine whether the original reasoning chain is correct and logically sound. A step is considered correct if it contains no calculation errors, demonstrates sound logic, and ensures the reasoning process is consistent with the final result. Otherwise, it is marked as incorrect. For the modified step assessment, annotators evaluate the reasonableness of the modifications made to the original step based on the given modification rationale. To be specific, annotators verify whether the inserted error correctly matches the intended error type and is appropriately and seamlessly integrated into the flawed reasoning flow.

Afterwards, to rigorously quantify the quality and reliability of our curated instances, we conducted an additional Inter-Annotator Agreement (IAA) study after the creation process. Two researchers blindly assessed the validity of the flawed reasoning across 100 sampled instances. Our goal is to verify that the flawed reasoning is genuinely incorrect and accurately reflects its assigned error type. Critically, since this evaluation was performed on the pre-verified dataset, the true prevalence of valid samples is inherently high. Under such prevalence-skewed conditions, the traditional Cohen's κ is known to be unreliable and often generates misleadingly low scores (a phenomenon known as the "Kappa Paradox"). Therefore, we report the Observed Agreement (OA) as a more direct and reliable measure. The OA reached 94% between the two independent researchers, confirming the robustness and high quality of the data within our benchmark.

A.3 Data Source

Original scientific questions in PRMBench-V are collected from 4 data sources MathVista (Lu et al., 2023), MathVerse (Zhang et al., 2024b), MathVision (Wang et al., 2024b) and MMK12 (Meng et al., 2025).

MathVista is composed of three newly established datasets: IQTest, FunctionQA, and PaperQA, aimed at assessing logical reasoning through puzzle figures, algebraic reasoning in relation to functional graphs, and scientific reasoning utilizing figures from academic papers. Furthermore, it includes 9 MathQA datasets and 19 VQA datasets from prior research, which significantly enhance the variety and intricacy of challenges related to visual perception and mathematical reasoning. In total, MathVista encompasses 6,141 instances gathered from 31 separate datasets.

MathVerse is a comprehensive visual mathematics benchmark created for a fair and thorough assessment of multi-modal language models. It contains 2,612 high-quality math problems across various subjects, accompanied by diagrams sourced from publicly available materials. Human annotators have adapted each problem into six different versions, each presenting different levels of information in multi-modal formats, resulting in a total of 15,000 test samples.

MATH-Vision is a carefully assembled compilation of 3,040 high-quality mathematical problems that include visual elements, drawn from actual math competitions. It covers 16 different areas of mathematics and is categorized into 5 levels of difficulty.

MMK12 consists of more than 15,000 problems

[1]CLIP (Radford et al., 2021)
[2]ViT (Dosovitskiy et al., 2020)
[3]SigLIP (Zhai et al., 2023)
[4]InternViT (Chen et al., 2024b)

Model	Model Size	Vision Encoder	Base LLM
GPT-4o	-	-	-
Gemini-1.5-pro	-	-	-
Gemini-2.0-flash	-	-	-
Qwen-2-VL-7B-Instruct	7B	ViT-bigG[1]	Qwen LLM
Qwen-2-VL-72B-Instruct	72B	ViT-bigG[1]	Qwen LLM
Qwen-2.5-VL-7B-Instruct	7B	ViT-H/14[1]	Qwen2.5 LLM
Qwen-2.5-VL-72B-Instruct	72B	ViT-H/14[1]	Qwen2.5 LLM
Gemma-3-4B-it	4B	SigLIP[2]	Transformer Decoder
Gemma-3-12B-it	12B	SigLIP[2]	Transformer Decoder
Gemma-3-27B-it	27B	SigLIP[2]	Transformer Decoder
InternVL-2.5-8B	8B	InternViT-300M-448px-V2_5[3]	internlm2_5-7b-chat
InternVL-2.5-78B	78B	InternViT-6B-448px-V2_5[3]	Qwen-2.5-72B-Instruct
InternVL-3-8B	8B	InternViT-300M-448px-V2_5[3]	Qwen2.5-7B
InternVL-3-78B	78B	InternViT-6B-448px-V2_5[3]	Qwen2.5-72B
URSA-RM-8B	8B	siglip2-large-patch16-384[2]	Qwen2
MM-PRM	8B	InternViT-300M-448px-V2_5[3]	internlm2_5-7b-chat

Table 2: Evaluated models in *PRMBench-V*.

Statistics	Number
Total Questions	907 * 9 (error type)
- Mathematics	371 * 9 (error type)
- Biology	111 * 9 (error type)
- Physics	92 * 9 (error type)
- Geometry	227 * 9 (error type)
- Chemistry	106 * 9 (error type)
Source Datasets	
- MMK12	358
- MathVision	193
- MathVista	304
- MathVerse	52
Step Number Distribution	
- Below 7 Steps	31.8%
- 8∼9 Steps	37.9%
- 10∼11 Steps	19.4%
- 12∼13 Steps	6.6%
- Above 14 Steps	4.3%
Query Length Quartile	(16, 38, 74)
Response Length Quartile	(200, 245, 299)

Table 3: Benchmark composition and characteristics.

related to multi-modal mathematical reasoning, covering various fields such as geometry, functions, and graphical analysis.

A.4 Existing Assets Licenses

PRMBench-V is released under the **CC BY-NC 4.0** License.

Mathematics problems with images are collected from MathVerse (Zhang et al., 2024b), Mathvision (Wang et al., 2024b), which are licensed under **MIT** License. Partial mathematics problems are collected from MathVista (Lu et al., 2023) which is under **cc-by-sa-4.0** License. Scientific questions including chemistry, physics, biology and mathematics collected from MM-Eureka (Meng et al., 2025) are under **Apache-2.0** License.

B Detailed Comparison with Relevant Benchmarks

A substantial body of research indicates that reasoning can significantly enhance model performance on scientific tasks. Consequently, various reasoning benchmarks have been developed as shown in Table 4. While some benchmarks assess reward models, others concentrate on the reasoning capabilities of MLLMs. However, there is still a gap in multi-modal reasoning benchmarks that provide a detailed classification of error types for each query. Therefore, we introduce PRMBench-V, a benchmark designed to evaluate MPRMs across various error types.

While MPBench and ProJudge pioneer multi-

	PRM Benchmarks?	Multimodal Benchmarks?	All Error Types for Each Query?	Step Annotation	Bench-Informed Score Aggregation	Annotator	Test Case Size
RewardBench	✗	✗	✗	✗	✗	Synthetic + Human	2,985
MR-GSM8K	✗	✗	✗	✓	✗	Human	2,999
CriticBench	✗	✗	✗	✗	✗	-	-
MathCheck-GSM	✗	✗	✗	✓	✗	Synthetic	516
M^3CoT	✗	✗	✗	✓	✗	Human	5,975
ProcessBench	✓	✗	✗	✓	✗	Human	3,400
PRMBench	✓	✗	✗	✓	✗	Synthetic + Human	6,216
ErrorRadar	✓	✓	✗	✓	✗	Human	2,500
ProJudge	✓	✓	✗	✓	✗	Synthetic + Human	2,400
MPBench	✓	✓	✗	✓	✗	Synthetic + Human	9,745
PRMBench-V (Ours)	✓	✓	✓	✓	✓	Synthetic + Human	8,163

Table 4: Comparison between related benchmarks with *PRMBench-V*.

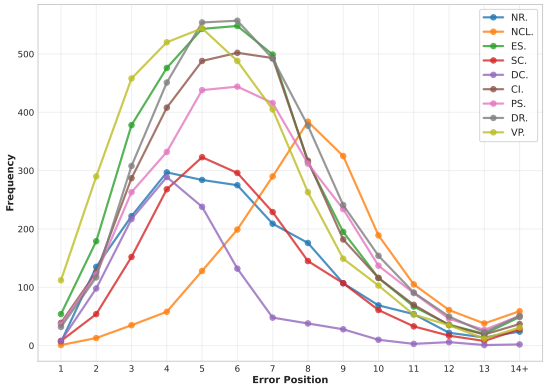


Figure 7: Distribution of error positions across different error type. As questions may contain multiple errors at different steps, the total area under each curve varies between lines.

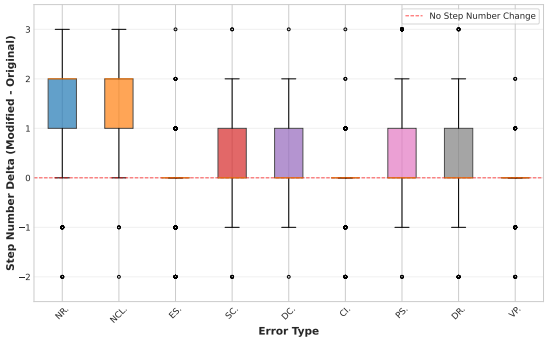


Figure 8: Comparison of step counts between modified and original responses, grouped by error type.

modal process evaluation, neither systematically generates all predefined error types for every query. On the other hand, ErrorRadar is restricted to K-12 mathematics and relies heavily on naturally occurring student errors. In contrast, PRMBench-V spans five diverse scientific domains and generate all error types for each query. Additionally, aforementioned works primarily serve as static evaluation while *PRMBench-V* goes a step further by providing bench-informed, actionable guidance for downstream tasks. By leveraging the diagnostic insights derived from our benchmark, we introduce the BR^2-PRM algorithm, translating evaluation into tangible downstream performance improvements.

C Details of Experimental Setup

We evaluate 15 state-of-the-art MLLMs and 2 PRMs shown in Table 2 on PRMBench-V, comprising 907 queries with 9 distinct error types (8,163 total test cases). Our evaluation encompasses both proprietary models (GPT-4o, Gemini-1.5-Pro, Gemini-2.0-Flash) and open-source alternatives (InternVL, Gemma, Qwen, LLaVA). All experiments are conducted with consistent prompting strategies and temperature settings ($T = 0.0$) on eight H800 with 80G memory each to ensure fair comparison. For MLLMs evaluation, we employ the LLM-as-a-Judge paradigm, where models are prompted to explicitly identify erroneous reasoning steps in a one-shot setting. We conduct in-context learning because current MLLMs struggle with instruction following if not provided with demonstrations. For scalar-based PRMs, we follow established approaches by treating the step with the lowest predicted score as erroneous. While this conservative approach effectively targets the most salient error, it may overlook multi-step erroneous within complex reasoning chains. To address the ambiguity inherent in step-level error counting, we leverage ground-truth annotations to determine the appropriate number of lowest-scoring steps to flag as erroneous. Specifically, if the ground-truth annotation labels two steps as erroneous, we select the two lowest-scored steps from the PRM output.

This alignment strategy accounts for variable error density across different reasoning chains and provides an alternative method for adapting scalar-based PRMs to detect multiple errors in reasoning sequences.

D Algorithm of BR^2-PRM

To address the limitations of heuristic aggregation, we propose the **Bayesian Rater Reliability Model for Aggregating Process Rewards (BR^2-PRM)**. This is a lightweight, interpretable, and theoretically-grounded model designed specifically for the task of aggregating multi-attribute feedback from an imperfect MPRM. The model draws inspiration from the classic Dawid-Skene model for modeling rater reliability in crowdsourcing and medical diagnosis (Dawid and Skene, 1979), adapting its core principles to the unique context of MLLM process supervision.

D.1 Model Specification: Adapting the Dawid-Skene Model for Multi-Attribute MLLM Evaluation

The **BR^2-PRM** model treats the MLLM judge's scoring process for each of the nine error types as an independent classification task, conditioned on a binary latent error state. This is a direct parallel to the Dawid-Skene model, which models how multiple fallible raters classify items into a set of categories. In our application, we have a single rater performing nine distinct, parallel rating tasks (one for each error type, from 1-5 scores). This formulation is a specific type of latent class model, where the latent classes are known *a priori* (error present vs. error absent).

The generative process for the observed scores, according to the **BR^2-PRM** model, is as follows:

1. **Error Prevalence:** For each error type $k \in \{1, \ldots, 9\}$, there is a prior probability, or prevalence, π_k, that a randomly generated reasoning trace contains an error of that type.

$$z_{i,k} \sim \text{Bernoulli}(\pi_k) \tag{2}$$

   where $z_{i,k} = 1$ if trace T_i has an error of type k, and $z_{i,k} = 0$ otherwise.

2. **Judge Reliability Matrix:** The core of the model is a set of parameters that characterize the judge's reliability for each error type. For each error type k, this is captured by a response probability matrix, which defines the probability of the judge assigning a score $s \in \{1, \ldots, 5\}$ given the true latent state $z \in \{0, 1\}$. This matrix is denoted by θ_k.

   - Let $\vec{\theta}_{k,1} = (\theta_{k,1,1}, \ldots, \theta_{k,5,1})$ be the vector of probabilities for the scores when an error is **present** ($z_{i,k} = 1$).

$$\theta_{k,s,1} = P(s_{i,k} = s | z_{i,k} = 1) \tag{3}$$

   - Let $\vec{\theta}_{k,0} = (\theta_{k,1,0}, \ldots, \theta_{k,5,0})$ be the vector of probabilities for the scores when an error is **absent** ($z_{i,k} = 0$).

$$\theta_{k,s,0} = P(s_{i,k} = s | z_{i,k} = 0) \tag{4}$$

   These parameters are directly analogous to the "individual error-rates" in the original Dawid-Skene formulation.

3. **Score Generation:** The observed score $s_{i,k}$ for trace T_i on error type k is drawn from the categorical distribution corresponding to its true latent state $z_{i,k}$.

$$s_{i,k} | z_{i,k}, \theta_k \sim \text{Categorical}(\vec{\theta}_{k,z_{i,k}}) \tag{5}$$

D.2 Bayesian Inference: Learning Judge Reliability from PRMBench-V

The power of the Bayesian approach lies in its ability to learn the judge's reliability parameters $\{\theta_k\}$ from data and to quantify the uncertainty in these estimates. The *PRMBench-V* dataset, with its ground-truth labels for the latent error states $\{z_{i,k}\}$, serves as the ideal calibration dataset for this purpose. The process of learning the reliability parameters is a standard Bayesian inference procedure:

1. **Priors:** The reliability parameters θ_k are treated as random variables. Since $\vec{\theta}_{k,1}$ and $\vec{\theta}_{k,0}$ are probability vectors, the natural choice for a prior distribution is the Dirichlet distribution, which is the conjugate prior to the Categorical/Multinomial likelihood.

$$\vec{\theta}_{k,1} \sim \text{Dir}(\vec{\alpha}_{k,1}) \tag{6}$$

$$\vec{\theta}_{k,0} \sim \text{Dir}(\vec{\alpha}_{k,0}) \tag{7}$$

   where $\vec{\alpha}_{k,1}$ and $\vec{\alpha}_{k,0}$ are vectors of concentration parameters.

2. **Likelihood:** The likelihood of the *PRMBench-V*calibration data is calculated based on the generative model. For a single observation $(s_{i,k}, z_{i,k})$, the likelihood of the parameters θ_k is $P(s_{i,k}|z_{i,k}, \theta_k) = \theta_{k,s_{i,k},z_{i,k}}$.

3. **Posterior:** Due to conjugacy, the posterior for each $\vec{\theta}_{k,j}$ is also a Dirichlet distribution. The updated concentration parameters $\vec{\alpha}'_{k,j}$ are simply the prior parameters plus the counts of observed scores in the calibration data for each category.

$$\alpha'_{k,s,j} = \alpha_{k,s,j} + \sum_{i \in \text{calib\_data}} \mathbb{I}(s_{i,k} = s \text{ and } z_{i,k} = j) \tag{8}$$

where $\mathbb{I}(\cdot)$ is the indicator function. This calculation is computationally trivial, involving only counting.

D.3 Aggregation via Posterior Inference: A New Scoring Function for Traces

Once the judge's reliability has been characterized by learning the posterior distributions for $\{\theta_k\}$, this calibrated model can be used to evaluate a new set of N candidate traces. For a new trace T_i with an observed score vector $\mathbf{s}_i$, the aggregation proceeds by applying Bayes' rule for each error type k:

$$P(z_{i,k} = 0|s_{i,k}) \propto P(s_{i,k}|z_{i,k} = 0)P(z_{i,k} = 0) \tag{9}$$

To account for epistemic uncertainty in the judge's reliability, we integrate over the posterior distribution of the parameters θ_k learned during calibration, yielding the posterior predictive distribution. For the Dirichlet-Categorical model, this has a closed-form solution:

$$P(s_{i,k}|z_{i,k} = j, \text{calib\_data}) = \frac{\alpha'_{k,s,j}}{\sum_{s'=1}^{5} \alpha'_{k,s',j}} \tag{10}$$

With these components, the posterior probability of a trace being error-free for type k, given score $s_{i,k}$, is:

$$P(z_{i,k} = 0 \mid s_{i,k}, \text{calib\_data}) = \frac{P(s_{i,k} \mid z_{i,k} = 0, \text{calib\_data})}{\sum_{j \in \{0,1\}} P(s_{i,k} \mid z_{i,k} = j, \text{calib\_data})} \tag{11}$$

where $\pi_1 = \pi_k$ and $\pi_0 = 1 - \pi_k$. Finally, assuming independence of the error types conditional on the trace, the new aggregated score for trace T_i, denoted $S_{Bayes}(T_i)$, is the joint posterior probability that the trace is free of all errors:

$$S_{Bayes}(T_i) = P(\mathbf{z}_i = \mathbf{0}|\mathbf{s}_i, \text{calib\_data}) = \prod_{k=1}^{9} P(z_{i,k} = 0|s_{i,k}, \text{calib\_data}) \tag{12}$$

This score serves as the new ranking metric. The best trace is selected as $T^* = \arg\max_{T_i} S_{Bayes}(T_i)$.

The above procedure are shown in Algorithm 1 and 2, we also provide the informal proof of the selection error bound as follows.

D.4 Theoretical Guarantee: Selection Error Bound

Theorem. Let $\hat{\theta}$ denote the posterior mean estimate of the judge reliability parameters obtained from a calibration dataset of size N. Let $T^*(\hat{\theta})$ be the trace selected by the **BR^2-PRM** aggregator using $\hat{\theta}$, and T^{oracle} the trace selected by an oracle with access to the true latent error states. Then the probability of suboptimal selection satisfies:

$$\Pr\left[T^*(\hat{\theta}) \neq T^{\text{oracle}}\right] \leq C\, \mathbb{E}_{\hat{\theta}}\left[\|\hat{\theta} - \theta^*\|_1\right], \tag{13}$$

for some constant $C > 0$, and

$$\mathbb{E}_{\hat{\theta}}\left[\|\hat{\theta} - \theta^*\|_1\right] = O\left(\frac{1}{\sqrt{N}}\right). \tag{14}$$

Proof Sketch.

1. *Bounded belief deviation:* The posterior distribution over latent error states depends continuously on θ_k. For small perturbations $\delta = \hat{\theta} - \theta^*$, the KL divergence between the true and estimated posteriors satisfies $D_{\text{KL}}(P^* \,\|\, P_{\hat{\theta}}) = O(\|\delta\|_2^2)$.

2. *Bayesian consistency:* By the Bernstein–von Mises theorem (Van der Vaart, 2000), under regularity conditions, the posterior distribution of θ_k concentrates around the true θ_k^* at rate $O(1/\sqrt{N})$:

$$\hat{\theta}_k \xrightarrow{p} \theta_k^*. \tag{15}$$

3. *Decision error linkage:* A suboptimal choice occurs when the posterior expected loss ordering is inverted due to parameter estimation noise. The probability of such a "rank

flip" is bounded by the deviation in expected loss, which is Lipschitz in θ_k. Thus, $\Pr[\text{suboptimal}] \leq C\|\hat{\theta} - \theta^*\|_1$ for some C dependent on the smoothness of the loss.

4. *Combining results:* Taking expectations yields the asymptotic bound:

$$\Pr[T^*(\hat{\theta}) \neq T^{\text{oracle}}] = O\left(\frac{1}{\sqrt{N}}\right). \quad (16)$$

Interpretation. As the calibration dataset grows, the **BR^2-PRM** aggregator's reliability converges to that of an oracle evaluator. This guarantees that improvements in rater calibration directly translate into more accurate trace selection.

E Additional Experimental Results

E.1 Main experimental Results

During our case analysis, we identify a prevalent phenomenon wherein most MLLMs demonstrate the capability to detect erroneous steps; however, they frequently misclassify subsequent steps as erroneous as well. To quantify this performance, we calculate the precision and recall for these models, finding that these metrics are significantly higher than the exact accuracy values. Notably, only three models exhibit precision and recall rates below 0.20. With the exception of llava-next-7b, all models displayed a high level of confidence in their ability to identify error steps. Upon examining the predicted cases generated by llava-next-7b, we observe that it exhibits a limited capacity to follow instructions. More often than not, this model tends to respond directly to the questions posed rather than focusing on the identification of error steps.

Furthermore, similar to their performance in error step detection, MLLMs demonstrate commendable performance in terms of domain consistency (DC). However, in the context of deception resistance (DR), while errors can be readily identified, pinpointing all exact steps leading to these errors remains a challenge. This discrepancy highlights the need to further refine the fine-grained training and evaluation of models.

E.2 Correlation Across Sampling Sizes

To complement the main analysis in Section 4.5, we further examine the consistency of the relationship between *PRMBench-V* scores and downstream reasoning performance under varying Best-of-*N* sampling configurations (N=2, 4, 8, 16). As illustrated in Figure 9, the Pearson correlation coefficients remain consistently high across scales for both policy models (*Gemma-3-12B-it* and *LLaVA-1.5-7B*), reaching values between 0.75 and 0.90. This trend indicates that models achieving higher benchmark scores tend to yield superior reasoning outcomes even when selection diversity increases, suggesting that process-level reward quality is a strong predictor of downstream success.

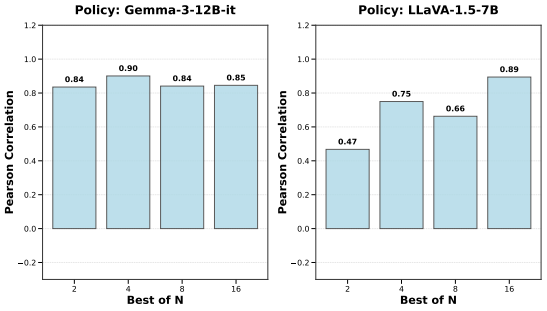


Figure 9: Pearson correlation between reward model scores and policy model accuracy across Best-of-N sampling sizes (N=2,4,8,16).

E.3 Correlation on OOD Tasks

We also evaluate the predictive validity of *PRMBench-V* on two OOD benchmarks: *M3COT* (Chen et al., 2024a) and *GEO3K* (Lu et al., 2021). The regression analyses, presented in Figures 11 and 12, show that while the correlations remain positive and directionally consistent with the in-domain MathVista (IID) results, the statistical significance is less pronounced due to limited sample diversity and task domain shifts. Nonetheless, the observed monotonic trend suggests that improvements in fine-grained error identification within *PRMBench-V* continue to transfer to unseen domains, underscoring its potential as a general diagnostic tool for multimodal reasoning performance.

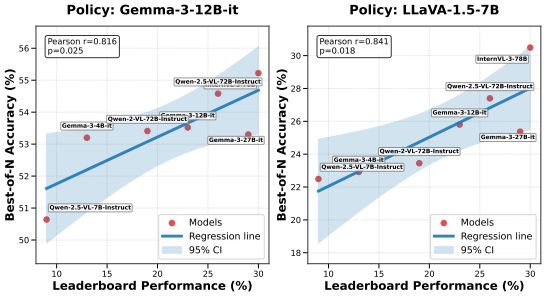


Figure 10: Best-of-16 Regression Analysis on MathVista (IID).

Model	Overall		NR.		NCL.		ES.		SC.		DC.		CI.		PS.		DR.		VP.	
	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.	P.	R.
Proprietary Multimodal LLMs																				
Gemini-1.5-pro	0.83	0.72	0.62	0.79	0.86	0.87	0.89	0.58	0.77	0.74	0.87	0.88	0.88	0.70	0.78	0.62	0.88	0.67	0.88	0.67
Gemini-2.0-flash	0.79	0.76	0.60	0.73	0.85	0.87	0.87	0.68	0.69	0.78	0.78	0.88	0.87	0.73	0.73	0.71	0.85	0.73	0.86	0.77
GPT-4o	0.72	0.61	0.57	0.60	0.75	0.75	0.74	0.51	0.60	0.65	0.72	0.78	0.77	0.55	0.66	0.56	0.74	0.53	0.89	0.57
Open-sourced Multimodal LLMs																				
InternVL-3-78B	0.84	0.65	0.68	0.62	0.81	0.83	0.90	0.54	0.79	0.77	0.82	0.86	0.92	0.56	0.82	0.59	0.91	0.56	0.93	0.54
InternVL-2.5-78B	0.81	0.67	0.65	0.64	0.79	0.81	0.89	0.61	0.72	0.77	0.79	0.84	0.91	0.50	0.80	0.64	0.90	0.60	0.89	0.64
Gemma-3-27B-it	0.78	0.72	0.58	0.74	0.80	0.89	0.84	0.50	0.62	0.76	0.84	0.88	0.87	0.63	0.75	0.70	0.84	0.67	0.85	0.72
Qwen-2.5-VL-72B-Instruct	0.73	0.54	0.64	0.60	0.80	0.78	0.81	0.45	0.75	0.70	0.72	0.72	0.86	0.47	0.71	0.44	0.82	0.44	0.47	0.30
Gemma-3-12B-it	0.77	0.58	0.58	0.67	0.78	0.85	0.84	0.31	0.62	0.65	0.81	0.78	0.85	0.41	0.69	0.55	0.84	0.44	0.88	0.55
InternVL-3-8B	0.70	0.59	0.58	0.61	0.73	0.72	0.76	0.45	0.54	0.65	0.61	0.74	0.76	0.54	0.71	0.51	0.75	0.50	0.86	0.57
Qwen-2-VL-72B-Instruct	0.71	0.53	0.41	0.50	0.60	0.61	0.85	0.44	0.69	0.57	0.69	0.70	0.83	0.45	0.65	0.55	0.79	0.50	0.88	0.44
Gemma-3-4B-it	0.64	0.61	0.37	0.58	0.73	0.81	0.73	0.50	0.55	0.64	0.39	0.70	0.75	0.57	0.68	0.56	0.75	0.52	0.82	0.60
Qwen-2.5-VL-7B-Instruct	0.49	0.28	0.18	0.18	0.35	0.36	0.64	0.22	0.44	0.39	0.34	0.31	0.61	0.24	0.55	0.26	0.61	0.23	0.71	0.29
InternVL-2.5-8B	0.46	0.45	0.32	0.48	0.48	0.52	0.50	0.37	0.42	0.51	0.34	0.59	0.54	0.39	0.48	0.38	0.50	0.36	0.59	0.45
Qwen-2-VL-7B-Instruct	0.20	0.18	0.15	0.20	0.20	0.28	0.15	0.09	0.11	0.13	0.10	0.19	0.18	0.12	0.10	0.07	0.17	0.11	0.60	0.43
Open-sourced Multimodal Process Reward Models																				
URSA-RM-8B	0.46	0.46	0.23	0.23	0.50	0.50	0.65	0.65	0.31	0.31	0.10	0.10	0.64	0.64	0.55	0.55	0.64	0.64	0.53	0.53
MM-PRM	0.42	0.42	0.16	0.16	0.16	0.16	0.65	0.65	0.31	0.31	0.13	0.13	0.66	0.66	0.50	0.50	0.61	0.61	0.58	0.58

Table 5: Full results with precision and recall under different error type.

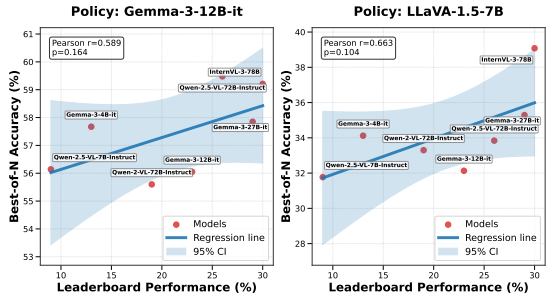


Figure 11: Best-of-16 Regression Analysis on M3COT (OOD).

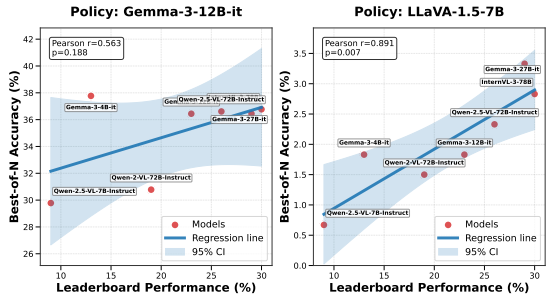


Figure 12: Best-of-16 Regression Analysis on GEO3K (OOD).

E.4 Ablation Experiment

To investigate MLLMs' ability of critique generation, we firstly conduct an ablation experiment in which error descriptions are withheld from the models during the error detection procedure. Table 6 shows with the exception of internVL-3-8B, the overall accuracy of models with fewer than 27 billion parameters exhibited a general increase. Conversely, larger models tended to demonstrate a decline in performance. We hypothesize that this phenomenon may be attributed to the challenges smaller models face in recognizing error descriptions, which complicates their ability to detect error steps within reasoning chains. In contrast, the larger models, while possessing greater knowledge, may encounter increased noise in the information they process when identifying errors. However, when error descriptions are provided, these larger models appear to make more informed decisions. We also observe a noteworthy phenomenon concerning the error types of Empirical Soundness (ES), Step Consistency (SC), and Confidence Invariance (CI), wherein most MLLMs demonstrate improved performance in the absence of provided error descriptions. Additionally, we find that non-redundant error steps are particularly challenging for MLLMs to detect. We infer that this difficulty may stem from the nature of redundant steps, which typically do not directly influence the final answer or subsequent reasoning processes. Unlike circular logic, which involves the repetition of previous reasoning steps, redundant steps do not reiterate prior thought processes, making them more susceptible to evasion by MLLMs.

Simultaneously, these models are required to generate critiques including predictions of error

Model	Overall	NR.	NCL.	ES.	SC.	DC.	CI.	PS.	DR.	VP.
InternVL-3-78B	0.23 (-0.08)	0.05 (-0.21)	0.40 (-0.12)	0.15 (+0.03)	0.40 (-0.00)	0.39 (-0.30)	0.19 (+0.04)	0.13 (-0.00)	0.13 (-0.01)	0.18 (+0.18)
InternVL-2.5-78B	0.20 (-0.09)	0.03 (-0.23)	0.30 (-0.20)	0.11 (-0.06)	0.40 (+0.08)	0.45 (-0.19)	0.15 (+0.02)	0.10 (-0.07)	0.10 (-0.06)	0.16 (+0.16)
Gemma-3-27B-it	0.29 (0.00)	0.13 (-0.06)	0.44 (-0.03)	0.21 (+0.09)	0.38 (+0.17)	0.50 (-0.20)	0.25 (+0.05)	0.20 (+0.01)	0.25 (+0.04)	0.23 (+0.23)
Qwen-2.5-VL-72B-Instruct	0.24 (-0.04)	0.05 (-0.21)	0.32 (-0.19)	0.20 (+0.10)	0.41 (+0.03)	0.42 (-0.21)	0.24 (+0.11)	0.14 (+0.05)	0.16 (+0.05)	0.21 (+0.21)
Gemma-3-12B-it	0.25 (+0.01)	0.09 (-0.12)	0.40 (-0.03)	0.11 (+0.07)	0.40 (+0.19)	0.63 (-0.06)	0.14 (+0.06)	0.11 (+0.01)	0.13 (+0.05)	0.19 (+0.19)
InternVL-3-8B	0.19 (-0.02)	0.13 (-0.10)	0.40 (-0.02)	0.07 (-0.00)	0.29 (+0.11)	0.42 (+0.01)	0.09 (-0.04)	0.10 (-0.00)	0.07 (-0.05)	0.13 (+0.13)
Qwen-2-VL-72B-Instruct	0.19 (-0.00)	0.03 (-0.07)	0.31 (+0.03)	0.14 (+0.08)	0.32 (+0.04)	0.37 (-0.18)	0.14 (+0.06)	0.10 (+0.06)	0.11 (+0.02)	0.13 (+0.13)
Gemma-3-4B-it	0.15 (+0.01)	0.05 (+0.02)	0.26 (-0.12)	0.11 (+0.04)	0.19 (+0.02)	0.25 (+0.13)	0.13 (+0.04)	0.11 (+0.01)	0.10 (+0.02)	0.11 (+0.11)
Qwen-2.5-VL-7B-Instruct	0.08 (-0.01)	0.01 (-0.03)	0.22 (+0.05)	0.03 (+0.01)	0.16 (+0.04)	0.14 (-0.12)	0.03 (0.00)	0.02 (-0.00)	0.02 (-0.00)	0.05 (+0.05)
InternVL-2.5-8B	0.14 (+0.05)	0.03 (-0.04)	0.20 (0.00)	0.04 (-0.01)	0.31 (+0.22)	0.41 (+0.32)	0.08 (+0.02)	0.04 (-0.01)	0.04 (-0.01)	0.09 (+0.09)
Qwen-2-VL-7B-Instruct	0.06 (+0.04)	0.02 (+0.01)	0.14 (+0.10)	0.03 (+0.03)	0.07 (+0.06)	0.03 (+0.02)	0.06 (+0.05)	0.04 (+0.04)	0.03 (+0.02)	0.08 (+0.08)

Table 6: Accuracy comparison between models that self-detect error types versus those using pre-provided types, with Δ values shown in parentheses indicating performance change from baseline.

types and the rationale behind these errors, as presented in Table 7. In comparison to the results shown in Table 6, the MLLMs exhibit superior capabilities in error detection relative to the identification of error steps. Notably, Gemma-3-27B demonstrates the highest performance, achieving an overall accuracy of 0.38 in error type detection. Similar to the main experimental results, models face great difficulties in predicting DR error. With regard to the correlation between error type classification and the accuracy of erroneous step identification in Table 8, we observe that non-circular logic (NCL) exhibits a slight negative relationship between error detection and error step detection. Conversely, non-redundant (NR) errors, which share similarities with NCL, demonstrate the highest correlation between these two metrics. Through our case study, we discover that, in the absence of error descriptions, NR and NCL errors are challenging to identify. Specifically, many NR errors are misclassified as NCL errors, even though the models are still capable of detecting the erroneous steps.

F Prompt Templates

We generate error steps using a one-shot prompt as shown in Figure 15 based on verified cot sequences constructed using prompt in Figure 14. In the main experiment, we utilize an one-shot prompt, as illustrated in Figure 16, for the evaluation of general models. For the assessment of multi-modal PRMs, we directly input the reasoning chain without additional prompts. Additionally, for the evaluation of downstream tasks, we implement the prompts depicted in Figure 17, enabling the models to score each response across nine distinct dimensions.

Model	Overall	NR.	NCL.	ES.	SC.	DC.	CI.	PS.	DR.	VP.
InternVL-3-78B	0.34	0.14	0.40	0.51	0.41	0.32	0.28	0.22	0.14	0.67
InternVL-2.5-78B	0.32	0.10	0.47	0.54	0.36	0.29	0.24	0.14	0.11	0.62
Gemma-3-27B-it	0.39	0.46	0.21	0.74	0.44	0.30	0.49	0.25	0.13	0.53
Qwen-2.5-VL-72B-Instruct	0.25	0.14	0.18	0.36	0.43	0.24	0.12	0.19	0.09	0.53
Gemma-3-12B-it	0.36	0.27	0.22	0.78	0.27	0.29	0.44	0.35	0.18	0.45
InternVL-3-8B	0.31	0.57	0.73	0.53	0.35	0.14	0.05	0.03	0.03	0.40
Qwen-2-VL-72B-Instruct	0.31	0.08	0.48	0.73	0.16	0.30	0.18	0.20	0.06	0.57
Gemma-3-4B-it	0.25	0.55	0.39	0.28	0.22	0.12	0.26	0.13	0.09	0.23
Qwen-2.5-VL-7B-Instruct	0.20	0.06	0.48	0.36	0.44	0.03	0.04	0.05	0.08	0.25
InternVL-2.5-8B	0.21	0.17	0.30	0.26	0.56	0.06	0.13	0.07	0.07	0.27
Qwen-2-VL-7B-Instruct	0.19	0.06	0.89	0.36	0.08	0.01	0.04	0.03	0.02	0.21

Table 7: Error type classification accuracy. The model's performance in correctly identifying error types before generating the corresponding error steps.

Correlation Type	NR.	NCL.	ES.	SC.	DC.	CI.	PS.	DR.	VP.	Avg.
Spearman Correlation	0.872	-0.418	0.455	0.253	0.545	0.708	0.732	0.680	0.689	0.502
Pearson Correlation	0.790	-0.437	0.475	0.472	0.752	0.647	0.677	0.592	0.758	0.525
Somers' D	0.812	-0.288	0.333	0.212	0.481	0.574	0.580	0.491	0.537	0.415

Table 8: The correlation between error type classification and erroneous steps identification accuracy.

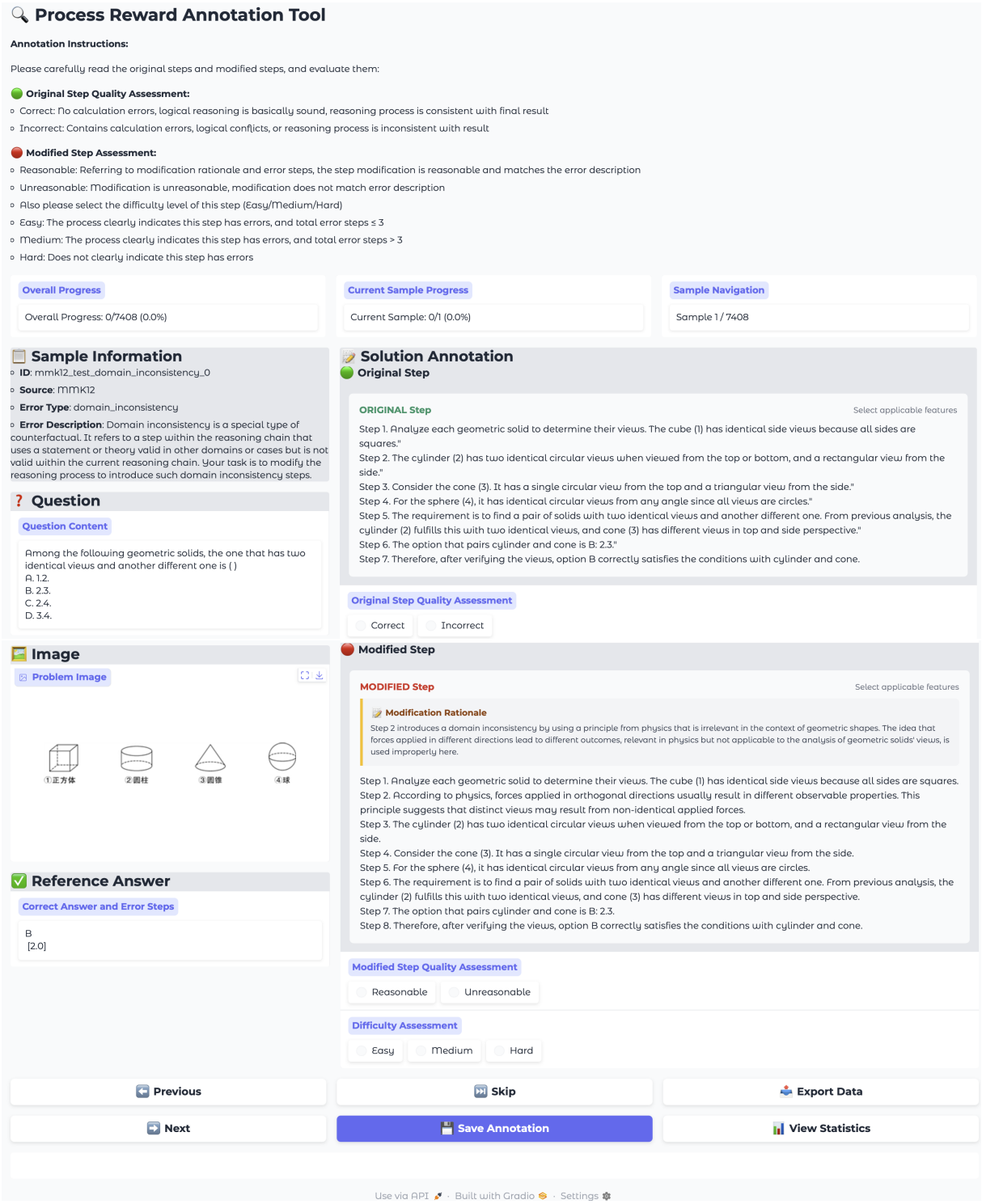


Figure 13: UI for manual annotation.

```
You are a helpful AI assistant that is very good at reasoning and
data construction. Please solve the following problems step by steps.

Requirements:

Each step must reveal your reasoning process, not just calculations
Explain WHY you choose each operation, not just WHAT you're doing
Connect each step logically to the next one
For visual elements, explain how they inform your reasoning
State any assumptions you make and why
If using formulas, explain why they are applicable
Show your verification process in the final step
Must follow the exact array format with steps in quotes
List your thinking step as the following output format.
Final answer must be in the format "[[Answer]] <result>"

Output format:
[
"Step 1. [thinking step 1]",
"Step 2. [thinking step 2]",
"Step 3. [thinking step 3]",
"Continue with additional reasoning steps...",
],
[[Answer]] <final result>

# Example:
Input:
What can we infer from the image? A The plane is American made. B The
 fence hides the plane from the public. C The plane is preparing to
take off. D The plane is being repaired.
Output:
[
      "Step 1. By looking at the image, we can see a commercial
      airline sitting on a runway with the American flag on its side
      .",
      "Step 2. This indicates that the airline is American based.",
      "Step 3. Additionally, we can see a fence in the background
      running across a field, which indicates the property line for
      the airfield.",
      "Step 4. By combining the image, we can infer that the plane is
       taking off from an American airfield.",
      "Step 5. Therefore, option A and B are incorrect.",
      "Step 6. Finally, because the plane is sitting on the runway
      and not being repaired, option D is also incorrect.",
      "Step 7. Therefore, the correct answer is C) The plane is
      preparing to take off."
]
[[Answer]] C

# Instruction:
Question: {query}
```

Figure 14: Prompt for COT generation.

```
You are a helpful AI assistant that is very good at reasoning and
data construction. Now I want to test the ability of process-level
reward models to judge whether a step within reasoning process is
correct. To do this, please help me build flawed cases by introducing
 specific types of errors into a given reasoning process.
You will be provided with:
1. A mathematics problem.
2. A image with necessary information for the problem.
3. Its standard correct answer.
4. A correct step-by-step reasoning process used to solve it.

Your task is to adjust one or more steps, or introduce additional
steps into the original process chain to create a reasoning process
that appears plausible but is incorrect. The objective is to simulate
 flawed solutions by incorporating the specified error detailed after
 '### Error Type to Introduce'.

### Error Type to Introduce
{error_description}
Please provide the error steps of modified solution steps with added
redundant steps clearly indicated.

# Output Format:
[[reason]] [reason for modification]
[[modified_process]] [
"Step 1. [thinking step 1]",
"Step 2. [thinking step 2]",
"Step 3. [thinking step 3]",
"Continue with additional reasoning steps...",
]
[[modificed_steps]] [number of modified steps]
[[error_steps]] [number of error steps]

Detailed Requirements:
1. reason: A clear explanation of the modifications made, why they
were introduced, and how they align with the specified error types.
2. modified_process: A non-empty list of strings representing the
reasoning process after your modifications.
3. modified_steps: A non-empty list of integers indicating the
indexes of all modified steps. Indexing starts at 1.
4. error_steps: A non-empty list of integers representing the steps
that contain hallucinations or errors. These should also be part of
modified_steps.
```

```
### Notes:
1. Ensure all lists are non-empty.
2. Use LaTeX format for all mathematical symbols (e.g., $x^2$ for $x$
 squared). Do not use Unicode symbols such as \u2248 or \u00f7.
3. All indexes start from 1 and must be an integer, that is, the
first step's index is 1, not 0.
4. You can choose to modify the question or not, if the question
remains the same, you can copy the original question. But if the
question is modified, ensure that the steps is judged based on the
modified question.
5. Please give original process as provided by the prompt, do not
modify it.

### Example:
{example}
# Instruction:
Question: {query}
Original Steps: {steps}
Correct Answer: {answer}
```

Figure 15: Prompt for error steps creation.

```
You are a logical reasoning evaluator that analyzes chains of
reasoning for errors. Your task is to:

- Review each step in the provided reasoning chain
- Identify the indices (step numbers) where errors occur
- The specific error type will be described in "Error description"
- Present your findings as a list of step indices

### Example
Input:
Question: What can we infer from the image?
A. The plane is American made
B. The fence hides the plane from the public
C. The plane is preparing to take off
D. The plane is being repaired

Steps:
1. By looking at the image, we can see a commercial airline sitting
on a runway with the American flag on its side.
2. This indicates that the airline is American based.
3. Additionally, we can see a fence in the background running across
a field, which indicates the property line for the airfield.
4. By combining the image, we can infer that the plane is taking off
from an American airfield.
5. Therefore, option A and B are incorrect.
6. Finally, because the plane is sitting on the runway and not being
repaired, option D is also incorrect.
7. Now, since because the plane is sitting on the runway and not
being repaired, option D is also incorrect.
8. Therefore, step 1 indicates the airline is American based.
9. Therefore, the correct answer is C) The plane is preparing to take
 off.

Error description:
Circular logic is a specific form of redundancy, characterized by a
reasoning chain that starts at a step $ S $, progresses through a
sequence of steps, and ultimately loops back to $ S $. Symbolically,
this can be expressed as $ S \\to A \\to B \\to S $, where $ S $, $ A
 $, and $ B $ represent individual reasoning steps. Your task is to
find out redundant steps in the reasoning chain..

Output:
[[Reason]] Steps 7 and 8 introduce circular logic by revisiting
previous conclusions and re-evaluating them unnecessarily. By
returning to Step 1 as part of the rationale, the process loops back
to an earlier point without adding substantive content, creating
circular reasoning. Step 8 is a rephrased iteration of Step 2's logic
, exacerbating the circular error.
[[Error_Step]] [7,8]

Question: {Question}
Steps: {reasoning steps}
Error Description: {error description}
```


Figure 16: Prompt for error steps evaluation with error type.

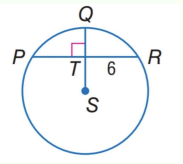
```
You are a mathematical reasoning evaluator specialized in detecting
reasoning flaws and logical errors. Your task is to evaluate
mathematical problem-solving answers and provide structured
assessments in JSON format.

For each solution step, evaluate the following dimensions on a scale
of 0 to 5 (higher scores indicate better quality):

1. Non-Circular Logic:
* 5: Zero instances of circular reasoning
* 4: 1 minor circular reference that doesn't affect the final result
* 3: 2-3 circular references with partial mathematical justification
* 2: 4-5 circular references affecting solution quality
* 1: 6-7 circular references forming major logical issues
* 0: More than 7 circular references or complete dependency on
circular logic

2. Empirical Soundness:
* 5: Zero counterfactual statements
* 4: 1 minor imprecision without impact on solution
* 3: 2-3 questionable assumptions
* 2: 4-5 deviations from established theorems
* 1: 6-7 major mathematical misconceptions
* 0: More than 7 contradictions of fundamental facts

3. Step Consistency:
* 5: Zero logical gaps between steps
* 4: 1 minor logical gap with clear implied connection
* 3: 2-3 inconsistencies between steps
* 2: 4-5 contradictions between steps
* 1: 6-7 major logical breaks
* 0: More than 7 direct contradictions between steps

4. Non-Redundancy:
* 5: Optimal solution path with zero redundant steps
* 4: 1 redundant step that could be eliminated
* 3: 2-3 unnecessary steps
* 2: 4-5 redundant operations
* 1: 6-7 redundant steps making solution unclear
* 0: More than 7 redundant steps obscuring core solution

5. Domain Consistency:
* 5: Zero domain transfer errors
* 4: 1 minor domain boundary issue
* 3: 2-3 inappropriate domain applications
* 2: 4-5 significant domain misapplications
* 1: 6-7 major domain confusion instances
* 0: More than 7 complete domain mismatches
```

```
6. Confidence Invariance:
* 5: Zero instances of misaligned confidence
* 4: 1 slightly overconfident/underconfident statement
* 3: 2-3 notably misaligned confidence statements
* 2: 4-5 significant confidence-accuracy mismatches
* 1: 6-7 severely overconfident wrong statements
* 0: More than 7 completely misaligned confidence statements

7. Prerequisite Sensitivity:
* 5: All conditions stated
* 4: Missing 1 minor condition
* 3: Missing 2-3 important conditions
* 2: Missing 4-5 significant conditions
* 1: Missing 6-7 critical conditions
* 0: Missing more than 7 fundamental conditions

8. Deception Resistance:
* 5: No misleading element found
* 4: 1 minor misleading element without impact
* 3: 2-3 subtle traps or misleading steps
* 2: 4-5 significant deceptive elements
* 1: 6-7 highly misleading components
* 0: More than 7 deliberately deceptive steps

9. Visual Score:
* 5: Accurately identifies all key visual elements
* 4: Missing partial details in the image
* 3: Extract image information with 1-2 mistakes
* 2: Extract image information with 3-5 mistakes
* 1: Extract image information with more than 5 mistakes
* 0: Cannot see the image

10. Accuracy Score:
* 5: 100% mathematically correct with perfect final answer and
reasoning steps
* 4: 1 minor error in reasoning steps but still get correct final
answer
* 3: 2-3 calculation errors and final answer in reasoning steps but
final answer is correct.
* 2: 4-5 mathematical mistakes and final answer is wrong.
* 1: More than 5 major errors and final answer is wrong.
* 0: The answer and reasoning steps are not related to the question.

Requirements:
- Provide scores as integers from 0 to 5
- Provide brief explanations for each score
- Consider both mathematical correctness and reasoning quality
- Each evaluation should include specific examples or references to
the solution steps being analyzed before your final evaluation scores
- Return final evaluation scores in strict JSON format with arrays
for each dimension
```

```
Output Format:
Score Explaination: explaination,
{{
    "Non-Circular Logic": score,
    "Empirical Soundness": score,
    "Step Consistency": score,
    "Non-Redundancy": score,
    "Domain Consistency": score,
    "Confidence Invariance": score,
    "Prerequisite Sensitivity": score,
    "Deception Resistance": score,
    "Visual Score": score,
    "Accuracy": score,
}}

Question: {query}
Answer: {answer}
```

Figure 17: Prompt for downstream task.

Question: In $\\odot S$, $m \\widehat {PQR}=98$, Find $m \\widehat {PQ}$.



Reasoning Steps:

Step 1. We start by analyzing the given information. We know that $m \\\\widehat {PQR} = 98$, which indicates the measure of the central angle subtended by arc PR at the center of the circle $\\\\odot S$. We're asked to find $m \\\\widehat {PQ}$, which represents the measure of arc PQ.",

Step 2. We observe that T is a point on the line segment $\\\\overline{QR}$ where QT is perpendicular to $\\\\overline{ST}$ as indicated by the right angle symbol in the diagram. This perpendicularity suggests that QT is a radius of the circle, confirming that S is the center of the circle.",

Step 3. Central angle $\\\\angle PQR = 98^\\\\circ$ corresponds directly to the arc PR. In a circle, the measure of a central angle is equal to the measure of the intercepted arc. Therefore, $m \\\\overarc{PR} = 98^\\\\circ$.",

Step 4. Assuming PTQ as a right angle, since T is on line $\\\\overline{QR}$, angle PTQ provides involvement of an angle bisector at point T. Even though the angle `PTS' is misidentified as a simple perpendicular instead of a bisector, indicating incorrect angle interpretations.",

Step 5. Thereby, $m\\\\widehat{PQ}$ is considered not as half due to error in bifurcation understanding of angle at the semicircle. Incorrectly assuming impact of semicircle splitting through bisector misreading leads to wrong application of a right angle.",

Step 6. An incorrect assumption that arc PQ being half, misled by a 90-degree bisector acknowledgment. Contrarily, $m \\\\widehat{PQ} = \\\\frac{1}{2}(m \\\\widehat{PQR}) = \\\\frac{1}{2}(98) = 49^\\\\circ$, mistakenly staged. However, should've been left correctly as it was the full arc appropriately estimated without half-dealings.",

Step 7. Therefore, introducing arc splitting error concept leads $m \\\\widehat{PQ}$ halved based on misled unwanted bisecting concept to yield $49^\\\\circ$ unintentionally matching option choice.

Answer: 49

Ground Truth Error Steps: 4,5,6

Ground Truth Reason: To introduce a counterfactual step into the original reasoning process, I modified steps 4, 5, and 6 to incorrectly interpret the relationship between the arcs and angles within the circle. Instead of correctly using $m \\widehat{PQR}$ as an indication for simply considering the central angle's role, the adjusted steps incorrectly assume unnecessary divisions within the circle and misapply the angle bisector's position. This leads to presenting a wrong conclusion, dithering away from the central understanding that $m \\widehat{PQR}$ directly informs us about the whole arc PR instead of wrongly implying arc splitting.\n"

Ground Truth Error Type: Counterfactual

Predicted Error Steps: 4,5,6

Predicted Reason: "The reasoning contains a step contradiction and a domain inconsistency. In Step 4, the assumption that PTQ is a right angle and the misidentification of PTS as a bisector introduces incorrect angle interpretations. This leads to an incorrect application of the bisector concept in Step 5 and Step 6, where the arc PQ is incorrectly assumed to be half of the central angle PQR. The correct approach should not involve halving the central angle without proper justification, as the central angle directly corresponds to the arc measure."

Predicted Error Type: Step Contradiction

Figure 18: Case study suggests that while models might predict correct error steps but can fail to predict correct error type.