MathSE: Improving Multimodal Mathematical Reasoning via Self-Evolving Iterative Reflection and Reward-Guided Fine-Tuning

Jinhao Chen[1,3*], **Zhen Yang**[2,3*†], **Jianxin Shi**[1], **Tianyu Wo**[1], **Jie Tang**[2†]

[1]School of Software, Beihang University
[2]Department of Computer Science and Technology, Tsinghua University
[3]Zhipu AI
[†]Corresponding Authors: yang-zhen@mail.tsinghua.edu.cn, jietang@mail.tsinghua.edu.cn

Abstract

Multimodal large language models (MLLMs) have demonstrated remarkable capabilities in vision-language answering tasks. Despite their strengths, these models often encounter challenges in achieving complex reasoning tasks such as mathematical problem-solving. Previous works have focused on fine-tuning on specialized mathematical datasets. However, these datasets are typically distilled directly from teacher models, which capture only static reasoning patterns and leaving substantial gaps compared to student models. This reliance on fixed teacher-derived datasets not only restricts the model's ability to adapt to novel or more intricate questions that extend beyond the confines of the training data, but also lacks the iterative depth needed for robust generalization. To overcome these limitations, we propose **MathSE**, a **Math**ematical **S**elf-**E**volving framework for MLLMs. In contrast to traditional one-shot fine-tuning paradigms, MathSE iteratively refines the model through cycles of inference, reflection, and reward-based feedback. Specifically, we leverage iterative fine-tuning by incorporating correct reasoning paths derived from previous-stage inference and integrating reflections from a specialized Outcome Reward Model (ORM). To verify the effectiveness of MathSE, we evaluate it on a suite of challenging benchmarks, demonstrating significant performance gains over backbone models. Notably, our experimental results on MathVL-test surpass the leading open-source multimodal mathematical reasoning model QVQ. Our code and models are available at https://mathse.github.io.

Introduction

Multimodal large language models (MLLMs) (OpenAI 2024; Anthropic 2024; Bai et al. 2023; Wang et al. 2024b; Bai et al. 2025; Wang et al. 2023b; Hong et al. 2024; Chen et al. 2024b) have recently garnered significant attention for their impressive ability to integrate visual and textual information, enabling them to effectively address a variety of vision-language answering tasks (Antol et al. 2015; Kafle and Kanan 2017; Mishra et al. 2019). However, their performance tends to falter when confronted with complex reasoning challenges, such as mathematical problem-solving. In order to enhance the mathematical reasoning ability of MLLMs, existing methods (Gao et al. 2023; Yang et al. 2024b; Cai et al. 2024; Shi et al. 2024; Zhang et al. 2024b; Peng et al. 2024; Luo et al. 2025) have primarily focused on fine-tuning these models on specialized mathematical datasets. These approaches typically involve distilling detailed, step-by-step reasoning from teacher models to generate rich, annotated datasets that capture mathematical problem-solving processes.

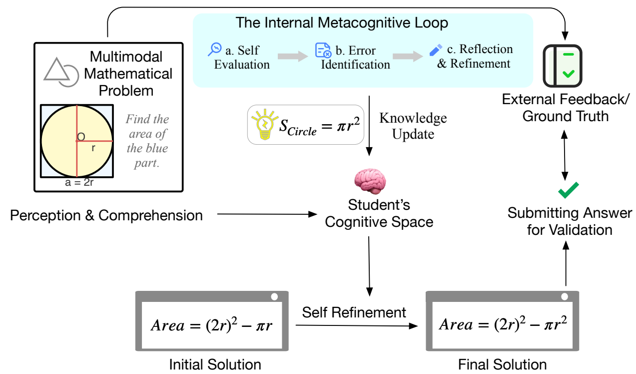


Figure 1: Illustration of the human learning process that inspires our approach.

Although previous methods have led to improvements in mathematical reasoning task, they still face notable limitations. The reliance on static, teacher-derived datasets often prevents models from adapting to novel or more intricate problems beyond the scope of the training data. Moreover, the step-by-step reasoning captured in these datasets frequently lacks the iterative depth necessary for robust generalization, leaving models ill-equipped to handle the evolving complexity inherent in mathematics. Such static and distilled datasets not only limit the dynamic, adaptive reasoning expected from student models but also widen the gap between the static patterns learned from teachers and the inherent data distribution from student models.

Inspired by human learning processes (Zimmerman 1990; Hattie and Timperley 2007), we recognize that effective learning process is inherently dynamic and iterative. As illustrated in Figure 1, human learning unfolds as a continuous cycle of instruction, practice, feedback, and improvement. In this paradigm, foundational knowledge is first acquired from teachers, distilling mathematical reasoning

*These authors contributed equally.

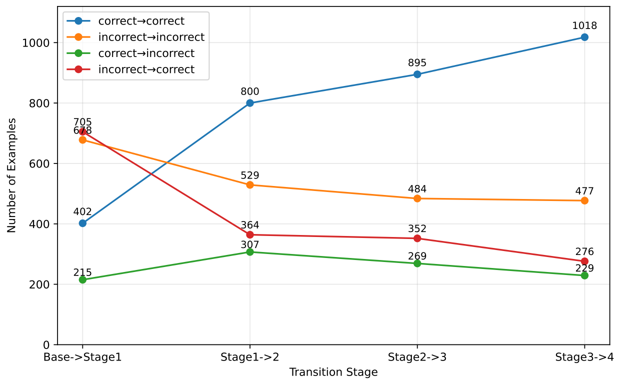


Figure 2: Accuracy changes during self-evolving process.

skills from teachers to students. This distilled knowledge forms the basis for independent practice, where students engage in self-guided problem solving to apply and reinforce what they've learned. As students tackle problems on their own, feedback plays a crucial role in identifying errors and improving their skills. Such continuous cycle of instruction, practice, feedback, and improvement enables students to progressively master mathematical problem-solving skills.

Drawing on these insights, we propose a novel framework that mirrors the dynamic, iterative nature of human learning. In this work, we propose a **Mathematical Self-Evolving** method, termed as MathSE. MathSE is built on the key idea of iterative fine-tuning and feedback-driven learning, designed to continuously enhance the mathematical reasoning capabilities of MLLMs. Specifically, our approach beings by fine-tuning a base model using a subset of GPT-4o distilled Chain-of-Thought (CoT) data, enabling it to grasp foundational mathematical reasoning skills. Once this initial training phase is complete, the model is employed to generate reasoning paths on the remaining dataset. Correct reasoning paths are then identified and leveraged for further fine-tuning. This creates a self-evolving learning cycle, where the model continuously refines its reasoning abilities by learning from its previous inferences. To distinguish between correct and flawed reasoning, we introduce a specialized Outcome Reward Model (ORM), which evaluates the entire reasoning process rather than the final answers. It identifies erroneous steps and delivers detailed error analyses. By leveraging the language understanding and reasoning capabilities of large language models (LLMs), the ORM not only signals correctness but also guides the model to reflect on and learn from its mistakes.

In our iterative framework, these incorrect reasoning paths, along with ORM-provided error steps and analyses, are fed back to the GPT-4o for reflection and refinement. The resulting corrected reasoning paths form a reflection dataset that is leveraged to train the final model. Such feedback from our ORM not only enables the model to recognize and correct its mistakes but also deepens its understanding of underlying reasoning flaws. As shown in Figure 2, as the self-evolving process progresses, more examples are consistently classified correctly, reflecting an improvement in overall accuracy. Such self-evolving training strategy enables the model to progressively enhance its problem-solving skills, effectively bridging the gap between static, teacher-derived datasets and the dynamic learning process characteristic of human students.

In order to verify the effectiveness and generalization of MathSE, we conduct experiments on three backbone models, including CogVLM2, Qwen2-VL-7B, and InternVL2.5-8B. Subsequently, we obtain a series of fine-tuned models namely MathSE-CogVLM2, MathSE-Qwen, and MathSE-InternVL. Experimental results demonstrate that MathSE achieves substantial improvements on multimodal math reasoning benchmarks, including MathVista, MathVL-test, MathVerse, and Math-Vision. With a parameter scale of approximately 10B, our models not only outperforms peer models of similar size but also attains performance levels comparable to state-of-the-art closed-source systems like Claude 3.5 Sonnet. Notably, the performance of our models on MathVL-test outperforms the leading open-source multimodal reasoning model QVQ (Team 2024).

Our contributions can be summarized as follows:

- **Method Perspective:** We propose a mathematical self-evolving framework (termed as MathSE) that iteratively improves multimodal math reasoning through reflection and reward-guided feedback.
- **Data Perspective:** We design a novel Outcome Reward Model (ORM) that provides step-wise error detection and analysis, guiding model refinement beyond mere answer evaluation.
- **Model Perspective:** Extensive experiments show significant performance gains on standard benchmarks, demonstrating the effectiveness of our approach. Code and model weights will be released soon.

Related Work

Multimodal Math Reasoning

Multimodal math reasoning requires models to process and integrate information from both textual and visual modalities to solve complex mathematical problems. Early approaches in this field primarily focused on text-only models with visual captions as input, such as PaLM (Chowdhery et al. 2022) and GPT-4 (Achiam et al. 2023), which demonstrated strong capabilities in language-based reasoning but struggled with visual content.

Recent developments in multimodal large language models (MLLMs) (Driess et al. 2023; Liu et al. 2024; Liu et al.; Wang et al. 2023b; Li et al. 2022; Dai et al. 2024; Bai et al. 2023; Chen et al. 2024b) have incorporated visual understanding to address these challenges. These models integrate vision encoders with language models to process images alongside text, achieving better performance in visual reasoning tasks.

However, despite their advancements, current multimodal large language models (MLLMs) are primarily limited to answering visual question answering (VQA) tasks and performing simple reasoning (Liu et al. 2024). They often struggle with more complex mathematical problems that require deeper logical reasoning, precise interpretation of visual elements, or multi-step problem-solving. This limitation highlights the gap between their current capabilities and the demands of advanced multimodal math reasoning.

Supervised Fine-Tuning and Knowledge Distillation

Supervised Fine-Tuning (SFT) has been widely used to adapt pre-trained models to specific tasks. By leveraging labeled datasets, SFT enables models to refine their understanding of task-specific patterns and improve performance on downstream applications. In the context of multimodal math reasoning, SFT has been employed to fine-tune models on datasets that combine textual and visual mathematical problems, such as ChartQA (Masry et al. 2022) and GeoQA (Chen et al. 2022b). Despite its effectiveness, SFT often requires large amounts of high-quality labeled data, which can be expensive and time-consuming to obtain.

Knowledge Distillation (KD) offers an alternative approach to enhance model performance by building high-quality math QA-pairs synthesized with frontier models like GPT-4o and Claude 3.5-Sonnet. However, most distillation-based methods (Taori et al. 2023; Zhang et al. 2023; Zhuang et al. 2024; Cai et al. 2024; Shi et al. 2024) focus on scaling the dataset size or leveraging outputs from larger teacher models, often ignoring the importance of in-distribution data. This oversight limits the student model's ability to generalize beyond the teacher's capabilities. Our approach emphasizes the role of in-distribution data and iterative refinement, moving beyond static fine-tuning to enable continuous self-evolution.

Reward Models for Reasoning Tasks

Reward Models (RMs) have been extensively explored in reinforcement learning from human feedback (RLHF), where they are used to align model outputs with human preferences. Traditional Outcome Reward Models (ORMs) (Cobbe et al. 2021; Stiennon et al. 2022; Yu, Gao, and Wang 2023) typically assign scalar rewards based on outcome quality without considering the reasoning process. In mathematical reasoning, such models fail to provide actionable feedback for improving intermediate steps.

Recent work attempts to address this through process-based supervision (Lightman et al. 2023; Wang et al. 2023a; Luo et al. 2024; Wu et al. 2024), where models are rewarded for correctness at each reasoning step. However, this paradigm requires exhaustive step-level correctness labels while potentially over-penalizing inconsequential mistakes.

In contrast, our **Outcome Reward Model (ORM)** introduces a paradigm shift by focusing on *diagnostic error analysis* rather than binary step-level evaluations. Unlike process-supervised RMs that require perfect intermediate verification, our ORM identifies *critical error step* in reasoning chains and provides *targeted error analysis*, enabling more effective revisions based on previous reasoning paths.

Self-Improvement and Reflection Mechanism

Large Language Models (LLMs) have increasingly demonstrated capabilities for self-improvement, where models leverage their own outputs to enhance performance without direct human supervision (Huang et al. 2022; Madaan et al. 2023). Qwen2.5-Math (Yang et al. 2024a) incorporates self-improvement mechanisms into its entire pipeline, enabling it to steadily improve problem-solving accuracy, minimize errors, and enhance generalization across diverse mathematical tasks.

To further augment this self-improvement process, recent work introduces reflection mechanisms that enable models to critically analyze their own reasoning traces (Lee et al. 2024; Renze and Guven 2024). ReAct (Yao et al. 2023) and Reflexion (Shinn et al. 2023) integrate reasoning and reflection to allow models to reconsider incorrect outputs. These methods demonstrate the potential of iterative self-improvement but are primarily applied to text-based tasks.

In the domain of multimodal reasoning, such reflective mechanisms remain underexplored. Our framework integrates reflection with ORM-guided feedback, enabling iterative self-improvement in multimodal math reasoning tasks. By combining reflection with error-specific feedback, our model can progressively enhance its reasoning capabilities.

Methodology

In this section, we present our mathematical self-evolving framework, designed to iteratively improve reasoning capabilities through a cycle of supervised fine-tuning, reward-guided feedback, and reflection. Our method consists of three key components: (1) initial supervised fine-tuning with GPT-4o distilled data, (2) specialized Outcome Reward Model (ORM) for error detection and analysis, and (3) iterative reflection and self-improvement.

Framework Overview

Figure 3 illustrates the overall workflow of MathSE framework. The process begins with fine-tuning a base model using a subset of GPT-4o distilled data. This fine-tuned model generates reasoning paths on the remaining dataset, which are then evaluated by a specialized Outcome Reward Model (ORM). The ORM identifies incorrect reasoning steps and provides a detailed error analysis. This process is repeated for multiple rounds to progressively enhance the model's reasoning ability. GPT-4o then reflects on the incorrect answer, corrects its reasoning. The improved reasoning paths are incorporated into the fine-tuning dataset to train the final model.

Supervised Fine-Tuning with GPT-4o Distilled Data

Our training pipeline begins with **Supervised Fine-Tuning (SFT)** performed on the base model using high-quality curated data distilled from GPT-4o.

The fine-tuning objective is formulated as:

$$\mathcal{L}_{\text{SFT}} = - \sum_{(x,y)\in\mathcal{D}_{\text{SFT}}} \log P_\theta(y|x) \qquad (1)$$

where $\mathcal{D}_{\text{SFT}}$ denotes the distilled dataset containing GPT-4o generated reasoning paths, x represents the input query,

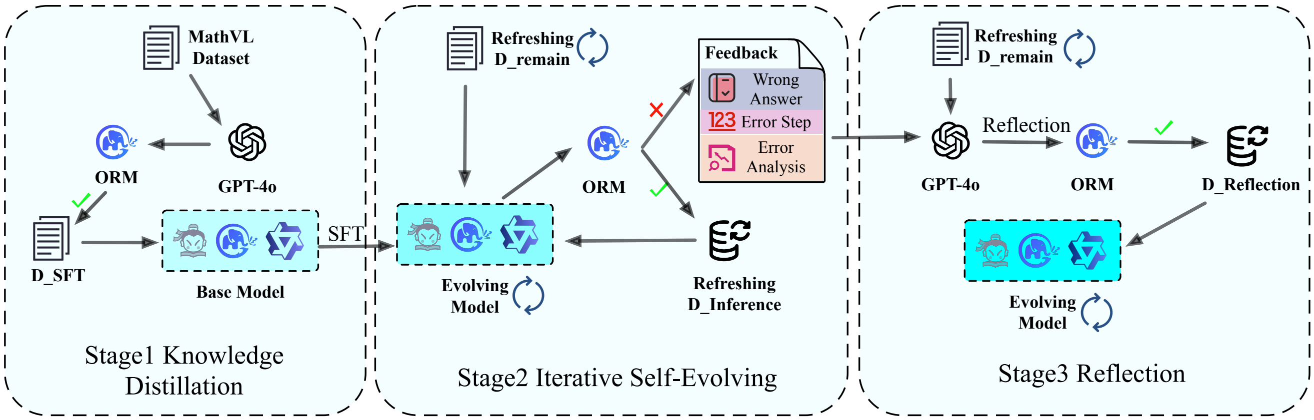


Figure 3: Overview of the MathSE Framework, which contains three stages to iteratively enhance mathematical reasoning abilities.

y corresponds to the target step-by-step reasoning response, and P_θ indicates the model's probability distribution.

Specialized Outcome Reward Model (ORM)

The specialized **Outcome Reward Model (ORM)** is a pivotal component of our framework, designed to provide both correctness evaluation and error analysis. Unlike traditional reward models that focus solely on output correctness, our ORM offers a comprehensive assessment by directly categorizing a reasoning path as either correct or incorrect. If an error is detected, the ORM pinpoints the faulty step and provides a detailed analysis of the error.

The ORM operates in the following two stages:

1. **Correctness Evaluation**: The ORM evaluates the entire reasoning path $r_i = \{s_1, s_2, \ldots, s_k\}$, assigning it as either correct or incorrect.
2. **Error Identification and Analysis**: If the reasoning path is identified as incorrect, the ORM directly specifies the step s_j where the error occurred and provides a detailed explanation E_i of the reasoning flaw that led to the mistake.

To train our ORM, we first constructed a comprehensive training dataset consisting of 60k reasoning samples. Specifically, we collected 30k incorrect reasoning paths along with their correct solutions, and leveraged GPT-4o to automatically generate detailed annotations, including the precise location of errors and corresponding error analysis. These annotated error cases were then combined with 30k correct Chain-of-Thought (CoT) reasoning examples to form a balanced dataset. We performed Supervised Fine-tuning (SFT) on CogVLM2 (Wang et al. 2023b) using this curated dataset, enabling the model to effectively evaluate reasoning correctness and provide precise error analysis when necessary.

Reflection and Self-Improvement

Incorrect reasoning paths, along with ORM-provided error steps and analyses, are fed back into the model for reflection. We leverage the language understanding and reasoning abilities of large models like GPT-4o to prompt the model to analyze its mistakes and generate improved solutions.

The reflection process involves the following prompt format:

```
Here is your previous solution:
[Incorrect Reasoning Path]
Error Step: [Faulty Step]
Error Analysis: [Explanation]
Please reflect and correct your
solution.
```

The refined reasoning paths $R_{\text{reflected}}$ are combined with previously correct paths R_{correct} to form the next round of training data:

$$\mathcal{D}_{\text{next}} = R_{\text{correct}} \cup R_{\text{reflected}} \tag{2}$$

Iterative Training Process

The MathSE framework operates through an iterative training process that refines the model's reasoning capabilities over multiple rounds. The process begins by generating an initial set of correct reasoning paths, D_{SFT}, using GPT-4o on the entire dataset D. These paths form the basis for fine-tuning the base model M_{base}, yielding the initial iteration model M_0. This first step ensures that the model starts with a set of well-formed reasoning patterns.

In each subsequent round, the model generates reasoning paths on the remaining dataset D_{remain}, which consists of data points not yet covered by the initial reasoning paths. These newly generated paths are then evaluated using the Output Reward Model (ORM). The ORM identifies correct paths, R_{correct}, and provides feedback on incorrect paths. The incorrect reasoning paths are reflected upon and corrected by generating new paths, $R_{\text{reflected}}$, based on the ORM feedback.

As the model progresses, the remaining dataset is updated to exclude the correct and reflected paths, and the training dataset $\mathcal{D}_{\text{SFT}}$ is expanded by incorporating both the correct and reflected paths. This updated training dataset is used to fine-tune the model for the next round, resulting in an improved model M_i. This cycle is repeated for a set number of rounds, T, leading to the model's gradual improvement.

The process can be formalized in Algorithm 1, which outlines the iterative training procedure. By integrating supervised fine-tuning, reward-guided feedback, and reflection, our self-evolving framework effectively enhances multimodal math reasoning. This iterative improvement allows the model to adapt and generalize, leading to state-of-the-art performance on benchmark datasets.

Algorithm 1: Framework of MathSE.

```
Generate correct reasoning paths D_SFT using GPT-4o on a subset of dataset D.
Initialize base model M_base.
Fine-tune M_base on D_SFT to obtain model M_0.
Set D_remain = D − D_SFT as the remaining dataset.
Initialize R_incorrect to collect incorrect reasoning paths.
for each round i = 1 to K do
    Use M_{i−1} to generate reasoning paths R_gen on D_remain.
    Evaluate R_gen with the ORM to obtain correct paths R_correct and incorrect paths with feedback.
    Update R_incorrect with newly identified incorrect paths and their ORM feedback.
    Update remaining dataset: D_remain = D_remain − R_correct.
    Update training data: D_SFT = D_SFT ∪ R_correct.
    Fine-tune to obtain the new model M_i using D_SFT.
end for
Use GPT-4o to reflect on R_incorrect with ORM feedback, generating reflected reasoning paths R_reflected.
Evaluate R_reflected with ORM to identify correct reflected paths R_reflect_correct.
Update training data: D_SFT = D_SFT ∪ R_reflect_correct.
Fine-tune to obtain the final model M_final using the updated D_SFT.
```

Experiments

In this section, we evaluate the effectiveness and generalization of our **MathSE** framework on multiple standard benchmarks.

Experimental Setup

Dataset Our study leverages the MathVL dataset (Yang et al. 2024b), a comprehensive educational dataset containing 341,346 multimodal mathematics exercises specifically designed for Chinese K12 education system and several open-source dataset including GeoQA+ (Cao and Xiao 2022), Geometry3K (Lu et al. 2021), ChartQA (Masry et al. 2022), and UniGEO-Calculation (Chen et al. 2022a). Furthermore, to enhance the dataset's diversity and coverage across mathematical domains, we strategically integrated carefully selected open-source datasets including MultiMath (Peng et al. 2024), MAVIS(Zhang et al. 2024b), Math-PUMA (Zhuang et al. 2024), and MathV360K (Shi et al. 2024). Our multimodal dataset spans elementary to senior high school curricula, encompassing diverse mathematical disciplines including arithmetic, algebra, geometry, probability, and applied word problems. While preserving the original question stems and multimodal contexts from MathVL, we employ MathSE to regenerate answer solutions. MathSE significantly extends the average answer length from 325 characters to 792 characters, providing more complete problem-solving procedures and more structured reasoning pathways.

Public Benchmarks We evaluate our model on three widely used multimodal math reasoning benchmarks and our specially constructed MathVL-test dataset:

- **MathVista** (Lu et al. 2024): A benchmark designed for visual math reasoning tasks, combining textual and diagrammatic information.
- **MathVerse** (Zhang et al. 2024a): Covers a wide range of math problems requiring both textual and visual comprehension.
- **MathVision** (Wang et al. 2024a): Focuses on complex multimodal math problems, challenging both vision and reasoning capabilities.
- **MathVL-test** (Yang et al. 2024b): A curated dataset designed to evaluate the integration of visual understanding and mathematical reasoning.

Baselines We compare our model with several Multimodal Large Language Models(MLLMs), including both closed-source MLLMs (OpenAI 2023, 2024; Anthropic 2024; Team et al. 2023) and open-source MLLMs (Team 2024; Bai et al. 2025; GLM et al. 2024; Dong et al. 2024; Gao et al. 2023; Chen et al. 2023; Zhang et al. 2024b; Peng et al. 2024; Luo et al. 2025). These models represent the current state-of-the-art in multimodal reasoning and serve as strong baselines for evaluating our approach.

Implementation Details In this study, we carefully selected three backbone models for our experiments, each chosen for their unique capabilities and performance in visual-language tasks. The models are as follows:

- **CogVLM2** (Hong et al. 2024) is an open-source multimodal large language model based on Meta-Llama-3-8B-Instruct. The model architecture supports context lengths up to 8K tokens and processes images at resolutions up to 1344 × 1344 pixels.
- **Qwen2-VL-7B** (Wang et al. 2024b) is built upon the Qwen2-7B language model backbone with a 675M ViT-based vision encoder. It features Naive Dynamic Resolution for handling arbitrary image sizes and Multimodal Rotary Position Embedding (M-ROPE) for processing textual, visual, and video positional information.
- **InternVL2.5-8B** (Chen et al. 2024a) is a multimodal large language model that combines InternViT-300M-448px-V2.5 as the vision encoder with internlm2.5-7b-chat as the language model. It employs dynamic high-resolution strategies with random JPEG compression for enhanced robustness.

For reproducibility and transparency, implementation details are documented in the supplementary material, which includes SFT configuration, prompts for data generation, training data examples, and error cases.

Main Results

Results on MathVL-test. The experimental results on MathVL-test depicted in Table 1 demonstrate the effectiveness of our iterative training framework across multiple stages. All three models show substantial improvements, with MathSE-InternVL exhibiting particularly promising results by achieving the highest accuracy of 65%.

Model	Training Data	Accuracy(%)
GPT-4o	-	51.05
Claude 3.5 Sonnet	-	46.84
Gemini-1.5-pro	-	52.03
QVQ-72B	-	52.25
Qwen2.5-VL-7B	-	50.50
CogVLM2	-	30.85
+ Distilled Data	100K	55.35
+ Self-Evolving	240K	62.35
+ Reflection	280K	64.70
Qwen2-VL-7B	-	40.60
+ Distilled Data	100K	48.80
+ Self-Evolving	240K	55.15
+ Reflection	280K	57.00
InternVL2.5-8B	-	33.20
+ Distilled Data	100K	58.45
+ Self-Evolving	240K	64.45
+ Reflection	280K	65.13

Table 1: Accuracy on MathVL-test across various backbones.

Results on Several Benchmarks. Table 2 presents the performance comparison between our model and baseline models across the three widely used multimodal math reasoning benchmarks: MathVista, MathVerse, and Math-Vision. Our approach significantly improves the performance of base models, with average score increases of 15.91% for CogVLM2, 12.28% for Qwen2-VL-7B, and 8.04% for InternVL2.5-8B. Notably, MathSE framework shows the most significant gains on MathVista (GPS), boosting CogVLM2 by 31.06%, Qwen2-VL-7B by 25.96%, and InternVL2.5-8B by 15.39%, demonstrating particularly strong capabilities on geometry-focused mathematical reasoning tasks.

Reflection Mechanism	Accuracy (%)
w/o Reflection	62.35
w/ GPT-4o Feedback	64.25
w/ ORM Feedback (Ours)	**64.70**

Table 3: Comparison of different reflection mechanisms on the MathVL-test set.

Ablation Studies

To assess the contribution of each component in our framework, we conduct ablation studies by selectively removing or modifying key modules. All experiments in this section are based on CogVLM2 as the backbone model, ensuring a consistent evaluation of the proposed methods.

Reflection Mechanism. We evaluate the impact of the reflection mechanism by comparing three configurations: (1) the proposed **ORM feedback**, where the Output Reward Model generates feedback for incorrect reasoning paths; (2) **GPT-4o feedback**, where GPT-4o replaces the ORM to provide feedback; and (3) **No reflection**, where the reflection mechanism is entirely removed. The results summarized in Table 3 show that the proposed ORM feedback achieves the highest accuracy (64.70%), outperforming both GPT-4o feedback (64.25%) and the no-reflection baseline (62.35%). This demonstrates the effectiveness of the ORM in providing precise feedback for improving reasoning paths, as well as the importance of the reflection mechanism in the self-evolving process.

Distilled data vs. self-evolving training data. This study examines how different data generation approaches and their resulting training distributions impact final model capabilities, comparing two paradigms:

- **Ours (Proposed)**: A combination of 90k GPT-4o-generated reasoning paths and 150k reasoning paths generated iteratively by our self-evolving model after filtering and refinement (90k + 150k = 240k total). This setup balances in-distribution and out-of-distribution data across different iterations.
- **Full GPT-4o**: A dataset of 240k reasoning paths fully generated by GPT-4o, all filtered by the Output Reward Model (ORM) to ensure correctness. This configuration lacks the iterative self-evolving mechanism, relying solely on GPT-4o-generated paths.

Method	Data Size	Accuracy (%)
Base	-	30.85
Full GPT-4o	$\approx$ 240K	58.00
Self-Evolving (Ours)	$\approx$ 240K	**62.35**

Table 4: Comparison of data generation methods on the MathVL-test set, with improvement ratio compared to base method.

The results in Table 4 show that our proposed method achieves the highest accuracy (62.35%), significantly outperforming the full GPT-4o baseline (58.00%). While GPT-4o provides high-quality initial data, it lacks the iterative adaptability necessary to address distributional shifts effectively. In contrast, our method leverages both pre-generated and model-refined data, leading to better generalization across diverse reasoning tasks.

Different ORM data curation. To evaluate the effectiveness of our ORM design, we conducted experiments comparing two different reward modeling approaches: our proposed ORM that incorporates both error step identification

Model	Method	MathVista (GPS)	MathVista	MathVerse	MathVision	**Average**
		Closed Source Models				
GPT-4V	-	50.50	49.90	50.80	22.76	43.49
GPT-4o	-	**64.71**	63.80	56.65	30.39	53.89
Claude 3 Opus	-	52.91	50.50	31.77	27.13	40.58
Claude 3.5 Sonnet	-	64.42	67.70	48.98	37.99	**54.77**
Gemini-1.5 Pro	-	53.85	63.90	51.08	19.24	47.02
Gemini-2.0 Flash	-	-	**73.10**	**47.80**	**41.30**	-
		Open Source Models				
GLM-4V-9B	-	46.12	46.70	35.66	15.31	35.95
InternLM-XC2	-	63.00	57.60	24.40	14.54	39.89
ShareGPT4V-G-7B	-	32.69	45.07	16.24	12.86	26.72
ShareGPT4V-G-13B	-	43.27	49.14	16.37	14.45	30.81
G-LLaVA-7B	-	53.40	28.46	12.70	12.07	26.66
G-LLaVA-13B	-	56.70	35.84	14.59	13.27	30.10
Qwen2.5-VL-7B	-	-	**68.20**	31.50	25.10	-
MAVIS-7B	-	64.10	-	28.40	-	-
URSA-8B	-	**79.30**	59.80	**45.70**	**32.60**	**54.35**
Math-LLaVA-13B	-	57.70	46.60	20.10	15.69	34.51
MultiMath-7B	-	66.80	50.00	26.90	-	-
CogVLM2	Base	39.61	40.85	25.76	13.20	29.86
	MathSE-CogVLM2	70.67	53.90	38.83	19.67	45.77
Qwen2-VL-7B	Base	43.75	59.40	35.53	16.92	37.94
	MathSE-Qwen	69.71	60.60	**45.36**	**25.22**	50.22
InternVL2.5-8B	Base	59.13	58.40	35.66	17.11	42.58
	MathSE-InternVL	**74.52**	**61.60**	43.65	22.70	**50.62**

Table 2: Performance on MathVista(GPS), MathVista, MathVerse, MathVision for different models.

and detailed error analysis, and a baseline method that only provides binary (correct/incorrect) feedback. The baseline was trained using the same training data as our ORM, but retained only the correct/incorrect judgments. We constructed ORM-2K, a balanced test set containing 1,000 correct and 1,000 incorrect reasoning paths, to assess the models' performance. As shown in Table 5, our proposed ORM significantly outperforms the binary-only baseline, achieving higher accuracy across both positive (correct) and negative (incorrect) cases. This demonstrates that incorporating fine-grained error analysis helps the model develop a more robust understanding of reasoning correctness, even when measuring only the binary judgment capability.

Error Correction Analysis

We analyze the effectiveness of MathSE in correcting its own previous errors. As shown in Figure 2, the number of consistently correct samples (correct→correct) significantly increases at each transition stage—from 402 initially to 1018 by the final stage—demonstrating strong knowledge retention. Furthermore, the count of persistently incorrect examples (incorrect→incorrect) and those changing from correct to incorrect (correct→incorrect) both show a decreasing trend, reinforcing the model's improved stability in predictions. Overall, these observations highlight MathSE's capability to progressively enhance its accuracy by effectively leveraging past mistakes and continuously refining its learned representations.

Method	Data Size	Positive Acc (%)	Negative Acc (%)	Overall Acc (%)
Binary	60K	85.40	99.90	92.65
Ours	**60K**	**94.20**	**100.00**	**97.10**

Table 5: Performance comparison of different reward modeling approaches on ORM-2K test set. "Binary" represents the baseline method with binary feedback, while "Ours" indicates our proposed ORM with error step identification and analysis.

Conclusion

In this paper, we introduced a novel Mathematical Self-Evolving framework which significantly enhances the reasoning capabilities of MLLMs through iterative fine-tuning, reward-guided feedback, and reflection. Our approach addresses the limitations of traditional one-shot fine-tuning and traditional reward models by introducing a specialized outcome reward model and a reflection mechanism that enables the model to progressively improve its reasoning ability. Experimental results demonstrate MathSE's substantial performance improvements across challenging multimodal mathematical reasoning benchmarks.

References

Achiam, J.; Adler, S.; Agarwal, S.; Ahmad, L.; Akkaya, I.; Aleman, F. L.; Almeida, D.; Altenschmidt, J.; Altman, S.; Anadkat, S.; et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Anthropic. 2024. The Claude 3 Model Family: Opus, Sonnet, Haiku.

Antol, S.; Agrawal, A.; Lu, J.; Mitchell, M.; Batra, D.; Zitnick, C. L.; and Parikh, D. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*, 2425–2433.

Bai, J.; Bai, S.; Yang, S.; Wang, S.; Tan, S.; Wang, P.; Lin, J.; Zhou, C.; and Zhou, J. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; Zhong, H.; Zhu, Y.; Yang, M.; Li, Z.; Wan, J.; Wang, P.; Ding, W.; Fu, Z.; Xu, Y.; Ye, J.; Zhang, X.; Xie, T.; Cheng, Z.; Zhang, H.; Yang, Z.; Xu, H.; and Lin, J. 2025. Qwen2.5-VL Technical Report. *arXiv preprint arXiv:2502.13923*.

Cai, S.; Bao, K.; Guo, H.; Zhang, J.; Song, J.; and Zheng, B. 2024. GeoGPT4V: Towards Geometric Multi-modal Large Language Models with Geometric Image Generation. arXiv:2406.11503.

Cao, J.; and Xiao, J. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, 1511–1520.

Chen, J.; Li, T.; Qin, J.; Lu, P.; Lin, L.; Chen, C.; and Liang, X. 2022a. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. *arXiv preprint arXiv:2212.02746*.

Chen, J.; Tang, J.; Qin, J.; Liang, X.; Liu, L.; Xing, E. P.; and Lin, L. 2022b. GeoQA: A Geometric Question Answering Benchmark Towards Multimodal Numerical Reasoning. arXiv:2105.14517.

Chen, L.; Li, J.; Dong, X.; Zhang, P.; He, C.; Wang, J.; Zhao, F.; and Lin, D. 2023. ShareGPT4V: Improving Large Multi-Modal Models with Better Captions. *arXiv preprint arXiv:2311.12793*.

Chen, Z.; Wang, W.; Cao, Y.; Liu, Y.; Gao, Z.; Cui, E.; Zhu, J.; Ye, S.; Tian, H.; Liu, Z.; et al. 2024a. Expanding Performance Boundaries of Open-Source Multimodal Models with Model, Data, and Test-Time Scaling. *arXiv preprint arXiv:2412.05271*.

Chen, Z.; Wu, J.; Wang, W.; Su, W.; Chen, G.; Xing, S.; Zhong, M.; Zhang, Q.; Zhu, X.; Lu, L.; et al. 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*, 24185–24198.

Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H. W.; Sutton, C.; Gehrmann, S.; Schuh, P.; Shi, K.; Tsvyashchenko, S.; Maynez, J.; Rao, A.; Barnes, P.; Tay, Y.; Shazeer, N. M.; Prabhakaran, V.; Reif, E.; Du, N.; Hutchinson, B. C.; Pope, R.; Bradbury, J.; Austin, J.; Isard, M.; Gur-Ari, G.; Yin, P.; Duke, T.; Levskaya, A.; Ghemawat, S.; Dev, S.; Michalewski, H.; García, X.; Misra, V.; Robinson, K.; Fedus, L.; Zhou, D.; Ippolito, D.; Luan, D.; Lim, H.; Zoph, B.; Spiridonov, A.; Sepassi, R.; Dohan, D.; Agrawal, S.; Omernick, M.; Dai, A. M.; Pillai, T. S.; Pellat, M.; Lewkowycz, A.; Moreira, E. O.; Child, R.; Polozov, O.; Lee, K.; Zhou, Z.; Wang, X.; Saeta, B.; Diaz, M.; Firat, O.; Catasta, M.; Wei, J.; Meier-Hellstern, K. S.; Eck, D.; Dean, J.; Petrov, S.; and Fiedel, N. 2022. PaLM: Scaling Language Modeling with Pathways.

Cobbe, K.; Kosaraju, V.; Bavarian, M.; Chen, M.; Jun, H.; Kaiser, L.; Plappert, M.; Tworek, J.; Hilton, J.; Nakano, R.; Hesse, C.; and Schulman, J. 2021. Training Verifiers to Solve Math Word Problems. arXiv:2110.14168.

Dai, W.; Li, J.; Li, D.; Tiong, A. M. H.; Zhao, J.; Wang, W.; Li, B.; Fung, P. N.; and Hoi, S. 2024. Instructblip: Towards general-purpose vision-language models with instruction tuning. *Advances in Neural Information Processing Systems*, 36.

Dong, X.; Zhang, P.; Zang, Y.; Cao, Y.; Wang, B.; Ouyang, L.; Wei, X.; Zhang, S.; Duan, H.; Cao, M.; Zhang, W.; Li, Y.; Yan, H.; Gao, Y.; Zhang, X.; Li, W.; Li, J.; Chen, K.; He, C.; Zhang, X.; Qiao, Y.; Lin, D.; and Wang, J. 2024. InternLM-XComposer2: Mastering Free-form Text-Image Composition and Comprehension in Vision-Language Large Model. *arXiv preprint arXiv:2401.16420*.

Driess, D.; Xia, F.; Sajjadi, M. S. M.; Lynch, C.; Chowdhery, A.; Ichter, B.; Wahid, A.; Tompson, J.; Vuong, Q.; Yu, T.; Huang, W.; Chebotar, Y.; Sermanet, P.; Duckworth, D.; Levine, S.; Vanhoucke, V.; Hausman, K.; Toussaint, M.; Greff, K.; Zeng, A.; Mordatch, I.; and Florence, P. 2023. PaLM-E: An Embodied Multimodal Language Model. In *arXiv preprint arXiv:2303.03378*.

Gao, J.; Pi, R.; Zhang, J.; Ye, J.; Zhong, W.; Wang, Y.; Hong, L.; Han, J.; Xu, H.; Li, Z.; et al. 2023. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*.

GLM, T.; Zeng, A.; Xu, B.; Wang, B.; Zhang, C.; Yin, D.; Rojas, D.; Feng, G.; Zhao, H.; Lai, H.; Yu, H.; Wang, H.; Sun, J.; Zhang, J.; Cheng, J.; Gui, J.; Tang, J.; Zhang, J.; Li, J.; Zhao, L.; Wu, L.; Zhong, L.; Liu, M.; Huang, M.; Zhang, P.; Zheng, Q.; Lu, R.; Duan, S.; Zhang, S.; Cao, S.; Yang, S.; Tam, W. L.; Zhao, W.; Liu, X.; Xia, X.; Zhang, X.; Gu, X.; Lv, X.; Liu, X.; Liu, X.; Yang, X.; Song, X.; Zhang, X.; An, Y.; Xu, Y.; Niu, Y.; Yang, Y.; Li, Y.; Bai, Y.; Dong, Y.; Qi, Z.; Wang, Z.; Yang, Z.; Du, Z.; Hou, Z.; and Wang, Z. 2024. ChatGLM: A Family of Large Language Models from GLM-130B to GLM-4 All Tools. arXiv:2406.12793.

Hattie, J.; and Timperley, H. 2007. The power of feedback. *Review of educational research*, 77(1): 81–112.

Hong, W.; Wang, W.; Ding, M.; Yu, W.; Lv, Q.; Wang, Y.; Cheng, Y.; Huang, S.; Ji, J.; Xue, Z.; et al. 2024. CogVLM2: Visual Language Models for Image and Video Understanding. *arXiv preprint arXiv:2408.16500*.

Huang, J.; Gu, S. S.; Hou, L.; Wu, Y.; Wang, X.; Yu, H.; and Han, J. 2022. Large Language Models Can Self-Improve. arXiv:2210.11610.

Kafle, K.; and Kanan, C. 2017. Visual question answering: Datasets, algorithms, and future challenges. *Computer Vision and Image Understanding*, 163: 3–20.

Lee, K.; Hwang, D.; Park, S.; Jang, Y.; and Lee, M. 2024. Reinforcement Learning from Reflective Feedback (RLRF): Aligning and Improving LLMs via Fine-Grained Self-Reflection. arXiv:2403.14238.

Li, J.; Li, D.; Xiong, C.; and Hoi, S. 2022. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, 12888–12900. PMLR.

Lightman, H.; Kosaraju, V.; Burda, Y.; Edwards, H.; Baker, B.; Lee, T.; Leike, J.; Schulman, J.; Sutskever, I.; and Cobbe, K. 2023. Let's Verify Step by Step. *arXiv preprint arXiv:2305.20050*.

Liu, H.; Li, C.; Li, Y.; Li, B.; Zhang, Y.; Shen, S.; and Lee, Y. J. ???? Llava-next: Improved reasoning, ocr, and world knowledge (January 2024). *URL https://llava-vl. github. io/blog/2024-01-30-llava-next*, 1(8).

Liu, H.; Li, C.; Wu, Q.; and Lee, Y. J. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2024. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *International Conference on Learning Representations (ICLR)*.

Lu, P.; Gong, R.; Jiang, S.; Qiu, L.; Huang, S.; Liang, X.; and Zhu, S.-C. 2021. Inter-GPS: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Luo, L.; Liu, Y.; Liu, R.; Phatale, S.; Guo, M.; Lara, H.; Li, Y.; Shu, L.; Zhu, Y.; Meng, L.; Sun, J.; and Rastogi, A. 2024. Improve Mathematical Reasoning in Language Models by Automated Process Supervision. arXiv:2406.06592.

Luo, R.; Zheng, Z.; Wang, Y.; Yu, Y.; Ni, X.; Lin, Z.; Zeng, J.; and Yang, Y. 2025. URSA: Understanding and Verifying Chain-of-thought Reasoning in Multimodal Mathematics. *arXiv preprint arXiv:2501.04686*.

Madaan, A.; Tandon, N.; Gupta, P.; Hallinan, S.; Gao, L.; Wiegreffe, S.; Alon, U.; Dziri, N.; Prabhumoye, S.; Yang, Y.; Gupta, S.; Majumder, B. P.; Hermann, K.; Welleck, S.; Yazdanbakhsh, A.; and Clark, P. 2023. Self-Refine: Iterative Refinement with Self-Feedback. arXiv:2303.17651.

Masry, A.; Long, D. X.; Tan, J. Q.; Joty, S.; and Hoque, E. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Mishra, A.; Shekhar, S.; Singh, A. K.; and Chakraborty, A. 2019. Ocr-vqa: Visual question answering by reading text in images. In *2019 international conference on document analysis and recognition (ICDAR)*, 947–952. IEEE.

OpenAI. 2023. GPT-4V(ision) System Card. In *technical report*.

OpenAI. 2024. GPT-4o System Card.

Peng, S.; Fu, D.; Gao, L.; Zhong, X.; Fu, H.; and Tang, Z. 2024. MultiMath: Bridging Visual and Mathematical Reasoning for Large Language Models. arXiv:2409.00147.

Renze, M.; and Guven, E. 2024. Self-Reflection in LLM Agents: Effects on Problem-Solving Performance. arXiv:2405.06682.

Shi, W.; Hu, Z.; Bin, Y.; Liu, J.; Yang, Y.; Ng, S.-K.; Bing, L.; and Lee, R. K.-W. 2024. Math-LLaVA: Bootstrapping Mathematical Reasoning for Multimodal Large Language Models. arXiv:2406.17294.

Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv:2303.11366.

Stiennon, N.; Ouyang, L.; Wu, J.; Ziegler, D. M.; Lowe, R.; Voss, C.; Radford, A.; Amodei, D.; and Christiano, P. 2022. Learning to summarize from human feedback. arXiv:2009.01325.

Taori, R.; Gulrajani, I.; Zhang, T.; Dubois, Y.; Li, X.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Stanford Alpaca: An Instruction-following LLaMA model. https://github.com/tatsu-lab/stanford_alpaca.

Team, G.; Anil, R.; Borgeaud, S.; Wu, Y.; Alayrac, J.-B.; Yu, J.; Soricut, R.; Schalkwyk, J.; Dai, A. M.; Hauth, A.; et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Team, Q. 2024. QVQ: To See the World with Wisdom.

Wang, K.; Pan, J.; Shi, W.; Lu, Z.; Zhan, M.; and Li, H. 2024a. Measuring Multimodal Mathematical Reasoning with MATH-Vision Dataset. arXiv:2402.14804.

Wang, P.; Bai, S.; Tan, S.; Wang, S.; Fan, Z.; Bai, J.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Fan, Y.; Dang, K.; Du, M.; Ren, X.; Men, R.; Liu, D.; Zhou, C.; Zhou, J.; and Lin, J. 2024b. Qwen2-VL: Enhancing Vision-Language Model's Perception of the World at Any Resolution. *arXiv preprint arXiv:2409.12191*.

Wang, P.; Li, L.; Shao, Z.; Xu, R.; Dai, D.; Li, Y.; Chen, D.; Wu, Y.; and Sui, Z. 2023a. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *CoRR, abs/2312.08935*.

Wang, W.; Lv, Q.; Yu, W.; Hong, W.; Qi, J.; Wang, Y.; Ji, J.; Yang, Z.; Zhao, L.; Song, X.; et al. 2023b. Cogvlm: Visual expert for pretrained language models. *arXiv preprint arXiv:2311.03079*.

Wu, J.; Feng, M.; Zhang, S.; Che, F.; Wen, Z.; and Tao, J. 2024. Beyond Examples: High-level Automated Reasoning Paradigm in In-Context Learning via MCTS. arXiv:2411.18478.

Yang, A.; Zhang, B.; Hui, B.; Gao, B.; Yu, B.; Li, C.; Liu, D.; Tu, J.; Zhou, J.; Lin, J.; Lu, K.; Xue, M.; Lin, R.; Liu, T.; Ren, X.; and Zhang, Z. 2024a. Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement. arXiv:2409.12122.

Yang, Z.; Chen, J.; Du, Z.; Yu, W.; Wang, W.; Hong, W.; Jiang, Z.; Xu, B.; and Tang, J. 2024b. MathGLM-Vision: Solving Mathematical Problems with Multi-Modal Large Language Model. arXiv:2409.13729.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; and Cao, Y. 2023. ReAct: Synergizing Reasoning and Acting in Language Models. In *International Conference on Learning Representations (ICLR)*.

Yu, F.; Gao, A.; and Wang, B. 2023. Outcome-supervised Verifiers for Planning in Mathematical Reasoning. arXiv:2311.09724.

Zhang, R.; Han, J.; Liu, C.; Gao, P.; Zhou, A.; Hu, X.; Yan, S.; Lu, P.; Li, H.; and Qiao, Y. 2023. LLaMA-Adapter: Efficient Fine-tuning of Language Models with Zero-init Attention. *arXiv preprint arXiv:2303.16199*.

Zhang, R.; Jiang, D.; Zhang, Y.; Lin, H.; Guo, Z.; Qiu, P.; Zhou, A.; Lu, P.; Chang, K.-W.; Gao, P.; et al. 2024a. MathVerse: Does Your Multi-modal LLM Truly See the Diagrams in Visual Math Problems? *arXiv preprint arXiv:2403.14624*.

Zhang, R.; Wei, X.; Jiang, D.; Zhang, Y.; Guo, Z.; Tong, C.; Liu, J.; Zhou, A.; Wei, B.; Zhang, S.; Gao, P.; and Li, H. 2024b. MAVIS: Mathematical Visual Instruction Tuning. arXiv:2407.08739.

Zhuang, W.; Huang, X.; Zhang, X.; and Zeng, J. 2024. Math-PUMA: Progressive Upward Multimodal Alignment to Enhance Mathematical Reasoning. *arXiv preprint arXiv:2408.08640*.

Zimmerman, B. J. 1990. Self-regulated learning and academic achievement: An overview. *Educational psychologist*, 25(1): 3–17.

SFT Configuration

Table 6 presents the detailed configurations of Supervised Fine-Tuning (SFT) process.

parameters	CogVLM2	Qwen2-VL-7B	InternVL2.5-8B
Total steps	5,0000	88,620	49,000
Global Batch Size	128	128	128
Learning Rate	$2e^{-5}$	$1e^{-5}$	$2e^{-5}$
Learning Rate Schedule	cosine decay	cosine decay	cosine decay
Warmup Ratio	0.01	0.1	0.03
Weight Decay	$5e^{-2}$	$5e^{-2}$	$5e^{-2}$
Optimizer	AdamW	AdamW	AdamW
Input Resolution	1344 * 1344	Dynamic	448 * 448

Table 6: The detailed setup of the SFT procedures.

Error Analysis

To better understand the limitations of our models, we conducted a detailed error analysis on MathVL-test. As shown in Figure 4, reasoning errors constitute the largest proportion of mistakes across all models (63.1% for MathSE-Qwen, 62.9% for MathSE-CogVLM2, and 65.3% for MathSE-InternVL), indicating that complex reasoning remains a significant challenge in multimodal understanding. Question misunderstanding errors form the second largest category (20.8% for MathSE-Qwen, 21.6% for MathSE-CogVLM2, and 27.0% for MathSE-InternVL), suggesting that these models sometimes struggle to properly interpret the user's intent. Knowledge errors (9.4% for MathSE-CogVLM2, 7.7% for MathSE-Qwen, and 6.3% for MathSE-InternVL) represent the third most common error type, highlighting gaps in the factual knowledge required for solving mathematical problems. Calculation errors remain relatively infrequent across all models (1.7% for MathSE-CogVLM2, 1.1% for MathSE-Qwen, and 1.0% for MathSE-InternVL), suggesting that basic arithmetic operations are not a major hurdle. Notably, vision recognition errors show significant variation between models, with MathSE-CogVLM2 exhibiting the highest rate (4.1%) compared to MathSE-Qwen (0.6%) and MathSE-InternVL (0.4%), indicating potential differences in the vision processing capabilities of these models. These findings suggest that improving reasoning abilities and question understanding should be prioritized in future mathematical MLLMs.

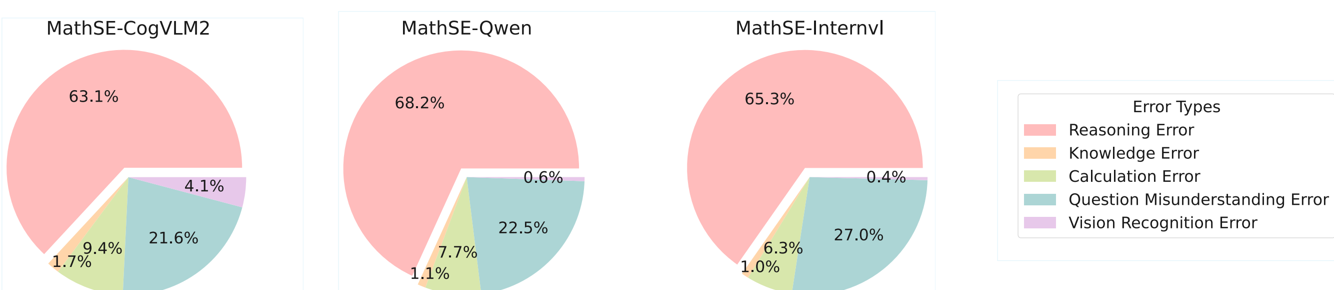


Figure 4: Distribution of different error types across three models on MathVL-test.

Prompts for Data Generation

To generate reasoning paths and reflection feedback, we designed specific prompts for each stage. Below are some representative examples.

Reasoning Path Generation Prompt

You are an excellent mathematics teacher. Please provide a step-by-step detailed solution and answer to the following question based on the question and image provided. Make sure to summarize at the end by stating `"The answer to`

`this problem is"` followed by the final result.
[question]
[images]

Reasoning Path Generation Prompt

You are an excellent mathematics teacher. Please provide a step-by-step detailed solution and answer to the following question based on the question and image provided. Make sure to summarize at the end by stating `"The answer to this problem is"` followed by the final result.
[question]
[images]

Feedback Generation Prompt

You are a professional solution evaluation system. You need to carefully compare the predicted answer with the standard answer, evaluate its correctness, and provide improvement suggestions.

Please evaluate the following solution:
Question: *[question]*
Predicted answer: *[predict]*
Standard answer: *[ground_answer]*

Please strictly follow these evaluation steps:

1. Carefully read the question requirements and evaluation criteria;
2. Compare the predicted answer with the standard answer step by step;
3. Check the correctness of key reasoning steps;
4. Analyze possible causes of errors or areas that can be improved.

Your output must strictly follow this JSON format:

```
{
  "status": "CORRECT" or "WRONG",
  "error_step": "Specific step where the error occurred",
  "error_analysis": "Analysis of the error cause",
  "improvement_suggestion": "How to improve the answer process"
}
```

Evaluation requirements:

- Status must be either `CORRECT` or `WRONG`;
- If wrong, clearly indicate the step where the problem occurs;
- Error analysis must be specific and clearly explain the cause;
- If the answer is correct, still provide improvement suggestions (e.g., remove redundant reasoning, improve clarity, strengthen logic);
- Analysis must be objective and evidence-based, avoiding subjective guesses.

Notes:

- If the predicted answer has a different format but identical essence, consider it correct;
- If multiple correct answers exist, matching any of them is valid;
- Do not reveal the correct final answer in the explanation.

Reflection Prompt

Please carefully review the following information:
Question: *[question]*
Original answer (wrong_answer): *[wrong_answer]*
Error step (wrong_step): *[wrong_step]*
Error analysis (wrong_analysis): *[wrong_analysis]*
And the image I uploaded

```
  Please note that the authenticity of wrong_step and wrong_analysis may be
questionable, but the original answer is confirmed to be wrong. Based on this
information, please re-examine and reflect on the original incorrect answer,
try to identify the errors, and regenerate a more accurate and reasonable
answer. In your response, please elaborate on your thinking process and
reasoning logic in detail to better verify the accuracy of the new answer.
Only provide your reconsidered answer, and make sure to summarize at the end
by stating "The answer to this problem is" followed by the final result.
```

Training Data Examples

Correct Reasoning Path Example

Figure 5 illustrates an example of correct reasoning path generated by Math-Vision. These validated reasoning paths, verified as accurate by our ORM, are subsequently incorporated into the training corpus to further enhance Math-Vision's reasoning capabilities.

Refined Reasoning Path with ORM Feedback

Figure 6 presents an example of incorrect reasoning with ORM feedback, demonstrating how reflection mechanisms help identify and correct errors in the reasoning process.

Error Cases

We provide examples of common reasoning errors identified by the ORM, including reasoning errors (Figure 7), knowledge errors (Figure 8), calculation errors (Figure 9), question misunderstanding errors (Figure 10), and vision recognition errors (Figure 11).

Example of Correct Reasoning Path

Question

As shown in the figure, in a cube ABCD-$A_1B_1C_1D_1$ with edge length 1, there are two moving points E and F on segment B_1D_1, where EF = 1/2. Consider the following four conclusions:

① AC ⊥ BE

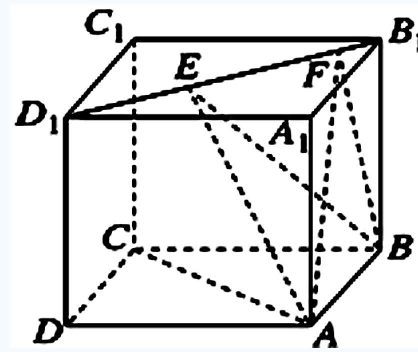
② EF ∥ plane ABCD

③ The volume of triangular pyramid A-BEF is constant

④ The area of triangle AEF equals the area of triangle BEF.

The number of incorrect conclusions is (   )

A. 0 B. 1 C. 2 D. 3



Model's response

To solve this problem, we need to analyze the correctness of each conclusion step by step.

Conclusion ①: AC⊥BE

1. In the cube ABCD-$A_1B_1C_1D_1$, AC is perpendicular to plane DD_1B_1B.
2. Since BE is in plane DD_1B_1B, therefore AC⊥BE.

Conclusion ②: EF∥plane ABCD

1. EF is in the plane containing BD_1, while BD_1 is parallel to BC and BC is in plane ABCD.
2. Therefore EF∥plane ABCD.

Conclusion ③: The volume of triangular pyramid A-BEF is constant

1. The area of triangle BEF is constant, because EF and the distance from B to E and F do not change.
2. The distance from point A to plane BEF is constant, as it is one height of the cube.
3. Therefore the volume of triangular pyramid A-BEF is constant.

Conclusion ④: The area of triangle AEF equals the area of triangle BEF

1. Since the distances from points A and B to segment EF are different, the heights of the two triangles are different.
2. Therefore the area of triangle AEF is actually not equal to the area of triangle BEF.

In summary, only conclusion ④ is incorrect.

This problem tests basic geometric relationships, with a rigorous analysis of each option.

The answer to this problem is B (1 incorrect conclusion).

ORM's Feedback

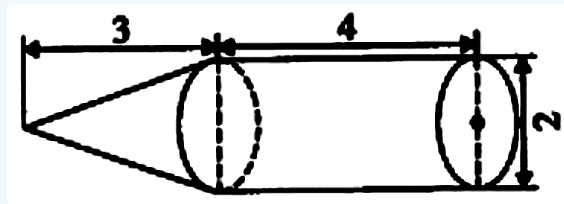
{
"**status**": "CORRECT",
"**error_step**": null,
"**error_analysis**": null
}

Figure 5: Example of correct reasoning path generated by Math-Vision.

Example of Reflection

Question

Calculate the volume as shown in the figure. (Unit: dm)



Model's response

we need to calculate the volume of a cylinder and a cone and add them together.
1. **Volume of the cylinder**: - Radius is 2 decimeters, height is 4 decimeters.\n - Volume formula: \\(V = \\pi r^2 h \\).\n - Substituting the data: \\(V_{\\text{cylinder}} = \\pi \\times 2^2 \\times 4 = 16\\pi \\, \\text{cubic decimeters} \\).
2. **Volume of the cone**: - Radius is 2 decimeters, height is 3 decimeters.\n - Volume formula: \\(V = \\frac{1}{3} \\pi r^2 h \\).\n - Substituting the data: \\(V_{\\text{cone}} = \\frac{1}{3} \\times \\pi \\times 2^2 \\times 3 = 4\\pi \\, \\text{cubic decimeters} \\).
3. **Total volume**:\n - Adding the volumes of the cylinder and cone: \\(V_{\\text{total}} = V_{\\text{cylinder}} + V_{\\text{cone}} = 16\\pi + 4\\pi = 20\\pi \\, \\text{cubic decimeters} \\). \n\nCalculating with \\(\\pi\\) approximated as 3.14:\n\\[V_{\\text{total}} \\approx 20 \\times 3.14 = 62.8 \\, \\text{cubic decimeters} \\]\n\nThe answer to this problem is 62.8 cubic decimeters.

ORM's Feedback

{
"**status**": "WRONG",
"**error_step**": "Incorrect use and calculation of the radius for the cylinder and cone",
"**error_analysis**": "The predicted answer incorrectly assumed the radius of both the cylinder and cone to be 2 decimeters, while according to the figure, the diameter of both the cylinder and cone is 2 decimeters, so the corresponding radius should be 1 decimeter."
}

After Reflection

Review and Correct Answer Calculation:\n\n#### Thinking Process:\nBased on the figure, we need to calculate the total volume of a geometric solid composed of a **cylinder and a cone**. The figure provides information about diameter, length, etc.
Below are the detailed calculation steps and reasoning: #### **1. Correctly Understanding the Figure Information:**\nFrom the figure, we can observe the following information:\n- The cylinder's diameter is 2 decimeters, so the **radius \\(r \\) is 1 decimeter**.\n- The cylinder's height (or length) is 4 decimeters.\n- The cone's diameter is the same as the cylinder's, 2 decimeters, so the **radius \\(r \\) is also 1 decimeter**.\n- The cone's height is 3 decimeters.\n\n---\n\n#### **2. Calculating the Volume of the Cylinder:**\nThe formula for the volume of a cylinder is:\n\\[\nV_{\\text{cylinder}} = \\pi r^2 h\n\\]\nSubstituting the data:\n\\[\nr = 1 \\, \\text{decimeter}, \\quad h = 4 \\, \\text{decimeters}\n\\]\n\\[\nV_{\\text{cylinder}} = \\pi \\times (1)^2 \\times 4 = 4\\pi \\, \\text{cubic decimeters}\n\\]\n\n---\n\n#### **3. Calculating the Volume of the Cone:**\nThe formula for the volume of a cone is: \n\\[\nV_{\\text{cone}} = \\frac{1}{3} \\pi r^2 h\n\\]\nSubstituting the data:\n\\[\nr = 1 \\, \\text{decimeter}, \\quad h = 3 \\, \\text{decimeters}\n\\]\n\\[\nV_{\\text{cone}} = \\frac{1}{3} \\times \\pi \\times (1)^2 \\times 3 = \\pi \\, \\text{cubic decimeters}\n\\]\n\n---\n\n#### **4. Calculating the Total Volume:**\nAdding the volumes of the cylinder and cone:\n\\[\nV_{\\text{total}} = V_{\\text{cylinder}} + V_{\\text{cone}} = 4\\pi + \\pi = 5\\pi \\, \\text{cubic decimeters}\n\\]\nTaking \\(\\pi\\) approximately as 3.14:\n\\[\nV_{\\text{total}} \\approx 5 \\times 3.14 = 15.7 \\, \\text{cubic decimeters}\n\\]\n\n---\n\n#### **Conclusion:**\nThe answer to this problem is:\n\\[\n\\boxed{15.7 \\, \\text{cubic decimeters}}\n\\]

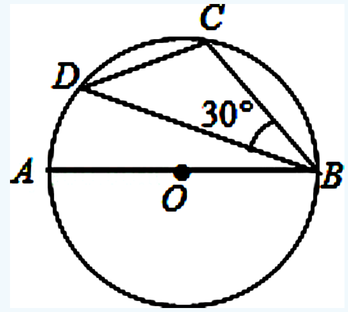
Figure 6: Example of reflection generated with incorrect reasoning path and ORM's Feedback.

Reasoning Error

Question

As shown in the figure, in circle O with diameter AB = 8, if angle CBD = 30°, then CD = ____.

Ground Truth 4



Model's answer

The problem requires us to find the length of CD, given that the diameter AB of circle O is 8, and angle CBD = 30°.
We can solve this problem through the following steps:
1. Determine the radius of the circle:
  - Since AB is a diameter, the radius OB = OA/2 = 8/2 = 4.
2. Understand the angle relationships:
  - Angle CBD = 30°. Since AB is a diameter, angle ACB is a right angle (90°), so triangle ACB is a right triangle, and angle CAB = 60°.
3. Calculate the length of CB:
  - In the right triangle ACB, we have angle CAB = 60°, angle ACB = 90°.
  - Using the sine law or special angle trigonometric functions, we can derive CB = AB * sin(angle CAB) = 8 * sin(60°) = 8 * √3/2 = 4√3.
4. Solve for the length of CD:
  - Angle CBD = 30°, angle DCB = 60° (since segment BCD is a chord on the same circle).
  - In triangle BCD, CD corresponds to the side opposite to angle DCB = 60°.
  - Again using the sine law or cosine law, we can find CD = CB * cos(angle CBD) = 4√3 * cos(30°) = 4√3 * (√3/2) = 6.

Therefore, the answer to this problem is CD = 6.

ORM's Feedback

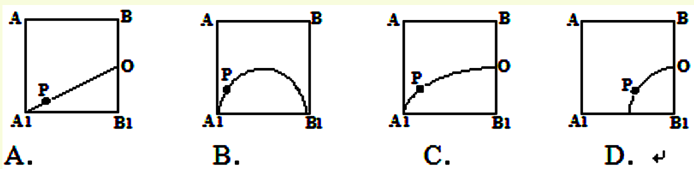
{
"**status**": "WRONG",
"**error_step**": "Steps 2 and 4: Misinterpretation of geometric relationships",
"**error_analysis**": "The solution contains multiple reasoning errors. In Step 2, it incorrectly assumes that ∠CAB = 60° without proper justification. In Step 4, there's a critical error in assuming that ∠DCB = 60° based on faulty reasoning about 'line segment BCD being on the same circle'. The solution fails to properly use the inscribed angle theorem, which states that an inscribed angle is half the central angle that subtends the same arc."
}

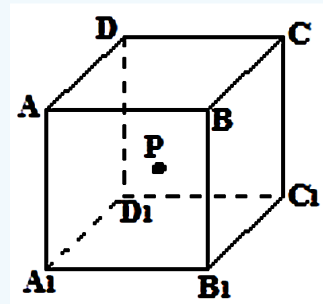
Figure 7: Example of reasoning error identified by our ORM.

Knowledge Error

Question

As shown in the figure, within the side face AB_1 of the cube $ABCD\text{-}A_1B_1C_1D_1$, there is a moving point P whose distance to line AB equals its distance to line B_1C_1. The shape of the curve on which the moving point P is located is ( )





Ground Truth

C

Model's answer

In the cube $ABCD\text{-}A_1B_1C_1D_1$, we analyze the distance from the moving point P to the two lines AB and B_1C_1.
Step 1: Understanding the problem: Point P is located on the side face AB_1 of the cube. The distances from P to AB and B_1C_1 are equal.
Step 2: Geometric relationship analysis: Line AB is parallel to line B_1C_1 and both are located in plane ABCD. Therefore, finding the perpendicular distance from P to these two lines is essentially solving for the intersection of a cylindrical surface with a plane.
Step 3: Parabola definition: In space, when a point's distance to a fixed point (focus) equals its distance to a fixed line (directrix), the trajectory is a parabola. Therefore, here we can view AB and B_1C_1 as fixed points, and P's trajectory satisfies this condition.
Step 4: Examining the graphical options: A. Parabola passing through the intersection of A_1B_1 B. Parabola passing through the extension of B_1C_1 C. The domain of the parabola doesn't match the cube structure D. Inverse proportional function form, doesn't correspond
Conclusion: Based on the above analysis, the curve on which the moving point P is located should be a parabola. Therefore, the correct answer is option A.

ORM's Feedback

{
"**status**": "WRONG",
"**error_step**": "Step 3",
"**error_analysis**": "The solution misapplied fundamental principles of 3D geometry when determining the locus of points equidistant from two lines."
}

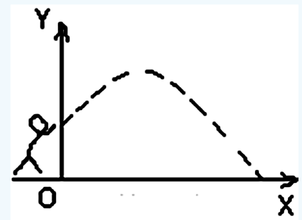
Figure 8: Example of knowledge error identified by our ORM.

Calculation Error

Question

As shown in the figure, if Xiao Li throws a shot put, and the height y (meters) of the shot put from the ground is expressed as $y=-\frac{1}{8}x^2+\frac{1}{2}x-\frac{3}{2}$, then the maximum height of the shot put from the ground during its motion is _____ meters.

Ground Truth 2



Model's answer

To solve this problem, we need to find the maximum value of the function \(y = -\frac{1}{8} x^2 + \frac{1}{2}x - \frac{3}{2} \) during the shot put's trajectory. This is a quadratic function with a downward-facing parabola, so we need to find its maximum value.
Solution Steps:
1. Determine the vertex form of the quadratic function:
For a quadratic function of the form \(ax^2 + bx + c \), where \(a = -\frac{1}{8}, b = \frac{1}{2}, c = -\frac{3}{2} \).
The x-coordinate of the vertex is: x = -\frac{b}{2a} = -\frac{\frac{1}{2}}{2 \times -\frac{1}{8}} = 4
2. Calculate the function value at the vertex: Substitute \(x = 4 \) into the original function to find the y-value: y = -\frac{1}{8}(4)^2 + \frac{1}{2}(4) - \frac{3}{2}
y = -\frac{16}{8} + 2 - \frac{3}{2}
y = -2 + 2 - \frac{3}{2} = -\frac{3}{2}.
Therefore, when the shot put reaches its highest point, the height from the ground is \(-\frac{3}{2}\) meters.
Summary:
The answer to this problem is \(-\frac{3}{2}\) meters.

ORM's Feedback

{
"**status**": "WRONG",
"**error_step**": "Step 1: Finding the vertex x-coordinate",
"**error_analysis**": "The error occurs in finding the vertex x-coordinate. Using the formula x = -b/(2a) with a = -1/8 and b = 1/2, we get x = -0.5/(2×(-1/8)) = -0.5/(-1/4) = 0.5/(1/4) = 2. The solution incorrectly calculated this as x = 4."
}

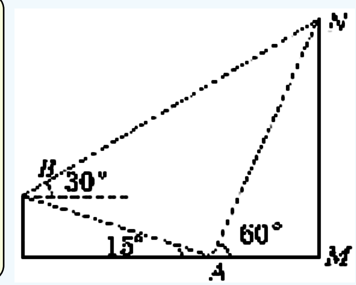
Figure 9: Example of calculation error identified by our ORM.

Question Understanding Error

Question

As shown in the figure, on a viewing stand with a slope of 15°, a row of seats along the straight line AB is coplanar with the flagpole along line MN. From the first seat A and the last seat B in this row, the elevation angles to the top of the flagpole N are measured as 60° and 30° respectively. The distance between seats A and B is $10\sqrt{6}$ meters. Find the height of the flagpole in meters.

Ground Truth 30



Model's answer

To solve this problem, we need to use trigonometric functions and the law of sines to calculate the height of the flagpole. Here are the specific steps:
Step 1: Identify the known values: The slope of the viewing stand \(\tan 15^\circ = \frac{\text{opposite}}{\text{adjacent}}\).The elevation angle to the top of the flagpole N is 60° from point A and 30° from point B. The distance between points A and B is \(AB = 10\sqrt{6}\) meters.
Step 2: Calculate the height from point A to point M. At point A, the angle between the flagpole MN and the horizontal plane is 15°, so: h_A = AB \cdot \sin 15^\circ
Step 3: Calculate the height from point B to point M. At point B, the angle between the flagpole MN and the horizontal plane is 30°, therefore: h_B = AB \cdot \sin 30^\circ
Step 4: Calculate the height of the flagpole: By finding the difference \(h_B - h_A\), we can get the height of the flagpole: h = h_B - h_A = (AB \cdot \sin 30^\circ) - (AB \cdot \sin 15^\circ)
Step 5: Substitute the values and calculate: First, calculate the approximate values of \(\sin 30^\circ\) and \(\sin 15^\circ\): \sin 30^\circ = \frac{1}{2}, \quad \sin 15^\circ = \frac{\sqrt{6} - \sqrt{2}}{4} Substitute into the formula: h = 10\sqrt{6} \times \frac{1}{2} - 10\sqrt{6} \times \frac{\sqrt{6} - \sqrt{2}}{4} Simplify: h = 5\sqrt{6} - 5(\sqrt{6} - \sqrt{2}) = 5\sqrt{2}
Summary
The answer to this problem is \(5\sqrt{2}\) meters.

ORM's Feedback

{
"**status**": "WRONG",
"**error_step**": "Step 2",
"**error_analysis**": "The solution misunderstood the geometry of the problem. It incorrectly interprets the 15° as the angle between the flagpole and horizontal plane, when it's actually the slope of the viewing platform."
}

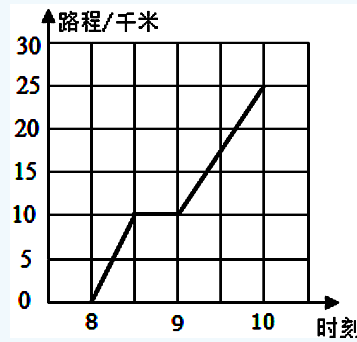
Figure 10: Example of question understanding error identified by our ORM.

Vision Recognition Error

Question

The figure shows statistics of a person riding a bicycle.
Look at the figure and fill in the blanks.
(1) This person traveled a total of ______ km.
(2) The person rested for ______ hours during the journey.
(3) From 9:00 to 9:30, their cycling speed was ______ kilometers/hour.

Ground Truth 25; 0.5; 15.



Model's answer

Based on the information in the figure, we can analyze and fill in the blanks:
(1) How many kilometers did this person travel in total?
- The graph shows the vertical axis reaching 25 kilometers.
Answer: This person traveled a total of 25 kilometers.
(2) How long did they rest during the journey?
- From the graph, we can see that between 9:00 and 9:30, the line is horizontal, indicating rest.
Answer: They rested for 0.5 hours during the journey.
(3) From 9:00 to 9:30, what was their cycling speed in kilometers/hour?
- Since the time from 9:00 to 9:30 was a rest period with no cycling, the speed should be 0.
Answer: From 9:00 to 9:30, their cycling speed was 0 kilometers/hour.
The answers to this question are: 25, 0.5, 0.

ORM's Feedback

{
"**status**": "WRONG",
"**error_step**": "(3) Calculation for the time period from 9:00 to 9:30",
"**error_analysis**": "The predicted answer incorrectly assumes the speed from 9:00 to 9:30 was 0 kilometers/hour. During this time period, the graph shows the cycling distance increasing from 10 kilometers to 17.5 kilometers, clearly indicating movement, and the actual speed can be calculated using the formula (distance ÷ time). Therefore, the predicted answer's judgment regarding the speed is incorrect. The remaining two parts of the answer (1) and (2) are correct."
}

Figure 11: Example of vision recognition error identified by our ORM.