PyraMathBench: A Comprehensive Framework for Evaluating and Improving Mathematical Capability in Large Language Models

Anonymous ACL submission

Abstract

Despite the critical role of mathematical capabilities in large language models (LLMs) across various applications, few frameworks comprehensively evaluate these abilities from foundational to advanced levels. This gap hinders the exploration of the weaknesses in the mathematical abilities of LLMs. In this paper, we introduce PyraMathBench[1], a framework designed to assess the mathematical capability of LLMs across four difficulty aspects, emphasizing the breakdown of complex tasks into simpler foundational components. PyraMathBench includes tailored single-modal and multimodal subtasks to rigorously evaluate model performance. We also propose the plug-and-play math model, a dynamic toolkit that enhances the mathematical processing abilities of LLMs, especially in Calculation tasks requiring intricate computation. Subsequent experiments with existing LLMs have led to the following findings: (i) LLMs' limited capacity for abstraction, task decomposition, and equation solving hinder their reasoning process. (ii) MLLMs predominantly rely on textual information when inferring Visual Reasoning Problems.

1 Introduction

Numbers play an integral role in text and are ubiquitous across a wide range of natural language processing (NLP) tasks (Yuan et al., 2023; Sundararaman et al., 2020). Mathematical reasoning is essential for NLP performance, especially in domains like scientific research (Spithourakis and Riedel, 2018) and financial documents (Chen et al., 2019; Jiang et al., 2020). Despite rapid advancements in big data and computational power, large language models (LLMs) like GPT-4 and Llama continue to struggle with mathematical tasks (Patel et al., 2021; Zhao et al., 2023), in part due to flaws in the tokenization of numbers (Liu and Low, 2023; Yuan et al., 2023) and hallucination (Ji et al., 2023; Chen et al., 2023). A model's ability to handle mathematical tasks serves as a critical indicator of its overall competence in solving real-world problems and performing abstract reasoning (Wei et al., 2022). However, current LLM-based mathematical problem-solving remains largely opaque, lacking mechanisms for analyzing errors or diagnosing failure modes, leading to an urgent need for a high-quality comprehensive mathematical evaluation benchmark.

Current benchmarks predominantly assess the mathematical reasoning abilities of language models through math word problems (MWPs). Datasets like GSM8K (Cobbe et al., 2021) and APE210K (Zhao et al., 2020), based on elementary-level problems, and benchmarks such as MATH (Hendrycks et al., 2021), ARB (Sawada et al.), and FrontierMath (Glazer et al., 2024), which involve competition-level problems like the IMO and AMC, are widely used. However, these benchmarks do not fully capture the limitations of LLMs' capabilities. For example, when models provide incorrect answers, it remains unclear whether the failure stems from computational errors or misinterpretation of the question. Some efforts, such as LILA (Mishra et al., 2022), attempt to address this by breaking down tasks into subtasks. Akhtar et al. (2023) introduced a framework to probe LLMs' numerical reasoning at various levels. But these frameworks lack cross-task correlations, testing LLMs' abilities in isolation. In the realm of Multi-Modal Large Language Models (MLLMs), benchmarks like MathVista (Lu et al., 2023) and MathVerse (Zhang et al., 2024) primarily emphasize image comprehension, neglecting a detailed exploration of the text modality's role in numerical reasoning.

Piaget's cognitive theory (Piaget, 1970) divides the progress of human cognition into four stages: the sensorimotor stage, pre-operational stage, con-

[1] https://osf.io/h4fwr/?view_only=392e23bd1b2443cd802b4c9ccef93dee

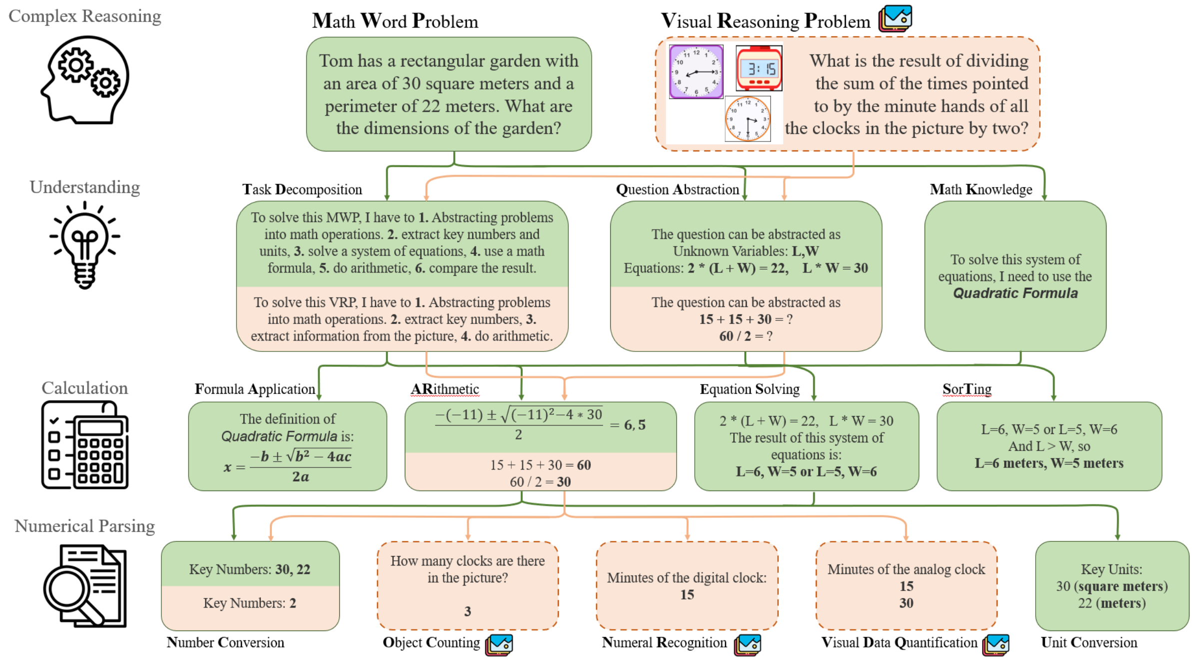

Figure 1: Two examples of decomposing complex reasoning problems into subtasks, with dashed lines representing multimodal tasks.

crete operational stage, and formal operational stage. The cognition shifts from simple and direct to complex and abstract. Under this framework, mathematical ability can be thought of as a hierarchy, akin to a pyramid structure, where complex tasks are broken down into simpler foundational components. By isolating and evaluating these core tasks, we can better understand the interplay between foundational skills and higher-level reasoning. This hierarchical approach not only assesses complex tasks but also identifies how weaknesses in basic skills can affect overall performance.

Based on this, we propose the PyraMathBench (PMB), a comprehensive hierarchical benchmark that includes 27,215 questions derived from 7,404 math word problems, covering 4 key cognitive aspects, 14 subcategories, and 2 modalities, ensuring a comprehensive evaluation. Additionally, the subtasks are decomposed from real math word problems rather than generated, making them more applicable to practical scenarios. PMB also incorporates the compositional relationships between tasks, enabling a deeper analysis of LLMs' strengths and weaknesses. Using PMB, we evaluated a variety of SOTA LLMs, identifying areas for improvement and offering valuable insights into the factors that influence performance in various aspects of mathematical reasoning. The key findings are summarized as follows:

- The models DeepSeek-R1, GPT-4o, and GPT-4o Mini are classified in the top tier of mathematical capabilities, demonstrating powerful reasoning and computational capabilities.

- A key weakness observed across the LLMs is their limited capacity for abstraction, task decomposition, and equation solving.

- MLLMs predominantly rely on textual information when inferring Visual Reasoning Problems.

2 The Taxonomy of PyraMathBench

The core motivation behind PMB's taxonomy is the recognition that mathematical tasks often require multiple layers of cognitive aspects and computational skills, ranging from simple numerical parsing to intricate logical reasoning. An LLM's ability to solve a high-level math word problem is contingent upon its proficiency in handling lower-level subcomponents. Traditional benchmarks conflate different cognitive aspects of mathematical competency without an explicit framework for isolating these subskills, making it difficult to diagnose specific failure points. By decomposing complex mathematical tasks into distinct hierarchical aspects, PMB provides a systematic method to evaluate capability at each stage of mathematical cognition, allowing for a more interpretable assessment of LLM performance.

Inspired by previous research (Xu et al., 2022; Akhtar et al., 2023) and Piaget's cognitive theory, our benchmark taxonomizes tasks into four hierarchical aspects (A1–A4), encompassing 14 distinct tasks. Figure 1 shows the composition of subtasks at each aspect and examples of subtask annotation. Here are concise definitions for each subtask.

- **Complex Reasoning.** This aspect represents the most advanced level of mathematical problem-solving, requiring the integration of multiple cognitive processes and mathematical principles. Complex reasoning tasks require sophisticated logical deductions, image comprehension ability, and multistep problem-solving. Models must demonstrate the ability to connect different types of information, identify abstract relationships, and apply higher-order reasoning strategies.

- **Understanding.** At this aspect, the focus is on the model's ability to comprehend and interpret mathematical content, transforming unstructured textual or visual information into actionable mathematical representations. Tasks in the Understanding category test the model's ability to make sense of mathematical descriptions, extract necessary information, and recognize patterns or structures.

- **Calculation.** The Calculation aspect involves performing arithmetic operations and applying standard mathematical formulas to compute solutions. Tasks for this aspect require the model to perform accurate numerical manipulations and apply mathematical formulas correctly. This aspect primarily tests the model's computational efficiency and correctness.

- **Numerical Parsing.** For the Numerical Parsing aspect, the tasks focus on the foundational abilities necessary to parse and process numerical information. This aspect tests the model's ability to recognize, interpret, and extract numerical data in various formats and contexts. It requires the model to handle the raw mathematical content and prepare it for further computation. Due to the limitations of tokenization, many LLMs perform poorly on such simple tasks.

We provide specific descriptions, prompts, and examples for each subtask in Appendix B.

3 Construction and Statistics

Data Sources. The PMB dataset integrates six existing evaluation datasets and practice questions. The data collection adheres to the following guidelines: 1) It includes common mathematical problems and visual reasoning tasks to represent the typical problem distribution. 2) Each problem is structured to allow clear decomposition into subtasks, facilitating unambiguous labeling. 3) The dataset is varied in difficulty, ensuring the inclusion of challenging tasks to effectively evaluate the performance of LLMs. We excluded non-mathematical content from the datasets. Based on this, we collected 6 datasets as data sources: ASDiv (Miao et al., 2021), alg514 (Kushman et al., 2014), Dolphin 18K (Shi et al., 2015), SVAMP (Patel et al., 2021), TAT-QA (Zhu et al., 2021), and MathVista (Lu et al., 2023), supplemented with some math practice.

Subtasks Annotation. The dataset annotation is conducted by three experts proficient in high school-level mathematics. The subtask questions are evenly distributed among the three experts for annotation, while the corresponding answers require validation by at least two experts. Once annotated, the answers are evaluated using the metrics outlined in Section 4. If the score falls below 90, the question is deemed ambiguous and subsequently discarded. We also utilized the table data from TAT-QA to create images to expand the variety of multi-modal tasks.

Certain datasets provide well-structured answer inference processes or automated question generation tools, facilitating the extraction of subtask questions. Additionally, we standardize the mathematical representations across different datasets, ensuring compatibility with both Python interpreters and LaTeX (the latter being used for more complex expressions). For floating-point answers, numerical values are rounded to six decimal places.

Statistics. Figure 2 presents the distribution of subtasks. PyraMathBench offers several advantages over existing evaluation methods: (1) **Comprehensive Coverage** – PMB includes a diverse array of tasks, spanning four primary areas of mathematical reasoning and 14 subcategories, derived from 13,735 questions across 4,536 Math Word Problems. This extensive dataset facilitates a thorough assessment of models across a wide range of topics and difficulty levels, ensuring broad coverage of mathematical challenges. (2) **Compositionality**

of Subtasks – PMB structures subtasks derived from the same Math Word Problem, allowing for detailed performance analysis. This compositional approach enables the isolation and evaluation of a model's ability to break down complex problems into simpler components, providing insights into foundational skill deficiencies and their impact on overall performance. (3) **Multimodal Tasks** – By incorporating both unimodal and multimodal tasks, PMB enables a more comprehensive evaluation of LLMs. This allows assessing models' ability to process different input types and engage in complex forms of reasoning.

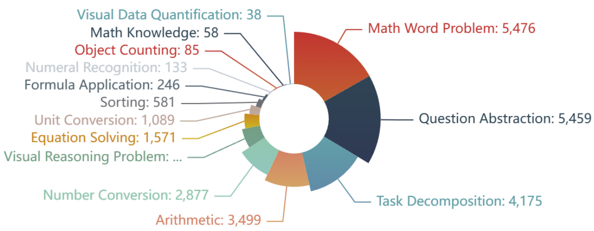

Figure 2: Subtask distribution within PyraMathBench

4 Models and Evaluation Metrics

Using PMB, we evaluated seven state-of-the-art LLMs, including GPT-4o (2024-11-20 version), GPT-4o-mini (2024-7-18 version)[2], LLaVA 13B (Liu et al., 2023), DeepSeek-R1 (DeepSeek-AI et al., 2025), Qwen2.5 14B (Yang et al., 2024), Llama3.1 8B (Grattafiori et al., 2024), Gemma2 9B (Team et al., 2024), and Mistral 7B (Jiang et al., 2023). The evaluation parameters were set as follows: `temperature` = 0.8, `top_k` = 40, and `top_p` = 0.9. The system prompt for each task contains the essential task setup and a detailed description of the question. Additionally, hints specific to each task are provided for LLMs that do not support structured output to guide the format of the answer. To simulate real-world mathematical question-answering scenarios, we employed zero-shot settings with Chain of Thought (CoT) prompting (Wei et al., 2022).

The evaluation of LLMs' mathematical capabilities incorporates their capacity to follow output format instructions, similar to grading practical exam questions. Specifically, LLMs are assessed on their ability to extract answers from designated fields in the prompt, with structured output being advantageous. The answer types include four formats: 1) a number or list of numbers, 2) expressions, 3) brief text, and 4) multiple-choice options.

Numerical answers are evaluated using Equation 1, where y represents the reference answer and $\hat{y}$ represents the model response. The equation considers two answers equivalent if their absolute difference is less than 10×10^{-4}, awarding a full score. For deviating answers, the score is determined by the absolute difference and the relative magnitude of the two numbers. If no corresponding answer is present in the model's output, the score is zero. To handle diverse mathematical expression formats, we employ a program based on SymPy to check equivalence and compute numerical results. This program uses heuristic methods to convert expression evaluations into numerical assessments. The text response format in PMB is relatively fixed and short, so we apply Jaccard similarity and semantic similarity as metrics. Multiple reference answers are provided in PMB, and the highest score derived from comparisons between the model output and reference answers is used as the final score. For multiple-choice questions, we calculate the perfect match rate. Finally, all scores are normalized to a range of 0 to 100.

$$\text{Score}(y, \hat{y}) = \begin{cases} 100, & \text{if } |\hat{y} - y| < 10 \times 10^{-4} \\ 0, & \text{if } \hat{y} = \textsc{Undefined} \\ \max(0, \frac{|\hat{y}-y|}{\max(1,y,\hat{y}) \times 50}), & \text{otherwise} \end{cases} \tag{1}$$

5 Main Results

The models DeepSeek-R1, GPT-4o, and GPT-4o Mini are classified in the top tier of mathematical capabilities, with DeepSeek-R1 demonstrating the highest overall performance. In Table 1, we can see that DeepSeek-R1 outperformed the other models in six out of ten text-only tasks. Interestingly, the open-source model Qwen2.5, despite having a smaller model size of only 14B parameters, achieved a performance comparable to that of the aforementioned models, showcasing competitive mathematical reasoning abilities.

In terms of performance on Math Word Problems, **a key weakness observed across the LLMs is their limited capacity for abstraction, task decomposition, and equation solving**. These deficiencies hinder their ability to effectively address complex mathematical tasks.

Furthermore, it can be observed that **MLLMs predominantly rely on textual information for**

[2]https://platform.openai.com/docs/models

Model	**Size**	MWP	*VRP**	QA	TD	MK	Arithmetic	ES
*GPT-4o**	-	92.1	**76.2**	65.9	81.1	75.6	96.1	50.1
*GPT-4o mini**	-	89.4	69.9	64.5	76.1	80.8	95.6	50.4
*LLaVA**	13B	34.4	25.3	11.4	36.1	61.0	60.9	8.0
`DeepSeek-R1`	671B	**93.9**	-	**92.3**	**84.2**	75.0	**96.5**	42.9
`Qwen2.5`	14B	91.0	-	64.1	72.9	**83.5**	90.3	**58.0**
`Llama3.1`	8B	56.9	-	54.1	69.3	78.3	81.4	34.9
`Gemma2`	9B	87.4	-	53.1	77.7	72.1	94.5	54.8
`Mistral`	7B	42.3	-	6.6	78.8	71.0	66.5	21.9

Model	**Size**	Sorting	FA	NC	UC	*NR**	*VDQ**	*OC**
*GPT-4o**	-	**96.4**	65.0	83.9	70.7	12.0	5.4	2.4
*GPT-4o mini**	-	94.5	76.7	73.6	72.9	17.6	**22.2**	1.1
*LLaVA**	13B	55.2	34.6	65.1	29.5	2.8	**8.3**	**7.1**
`DeepSeek-R1`	671B	94.8	73.7	**86.9**	**80.7**	-	-	-
`Qwen2.5`	14B	96.2	**79.4**	81.1	36.2	-	-	-
`Llama3.1`	8B	83.9	59.5	72.7	30.0	-	-	-
`Gemma2`	9B	93.9	71.7	69.0	14.9	-	-	-
`Mistral`	7B	90.4	51.7	64.3	53.9	-	-	-

Table 1: Main results of 8 LLMs on the 14 subtasks of PyraMathBench. *Italics** represents multimodal tasks.

Visual Reasoning Problems. Their ability to extract and process mathematical information from images remains relatively underdeveloped, even when the images involved are simple in nature. This suggests that LLMs require further advancements in their multimodal capabilities to enhance their performance in tasks that involve visual data.

Next, we will summarize the performance of LLMs with regard to particular difficult aspects. In complex reasoning tasks, DeepSeek leads with a score of 93.9 in the MWP task, followed by GPT-4o (93.9) and Qwen2.5 (91.0). The significant performance decline of Mistral (42.3) and LLaVA (34.4) is primarily due to limited instruction-following and mathematical reasoning abilities. Notably, LLaVA, which is not designed for complex mathematical tasks, shows a rather low performance in the VRP task at 25.3, in contrast to GPT-4o (76.2) and GPT-4o mini (69.9). However, even the latter two models do not achieve exceptional results.

LLMs demonstrate a marked decline in the Understanding aspect, which is closely linked to the accuracy of complex reasoning. This aspect focuses on assessing LLMs' ability to exhibit the reasoning process. Among the models, DeepSeek-R1 stands out with a score of 92.3 in the QA subtask, significantly surpassing GPT-4o (65.9). In contrast, Mistral and LLaVA, due to their limited support for structured output and weaker instruction-following abilities, struggle with providing valid expressions and consequently perform poorly in this task. Task decomposition ability, however, remains relatively consistent across LLMs, ranging from 72 to 85, indicating that, despite differences in reasoning skills, many mainstream LLMs share a similar reasoning process.

In the Calculation aspect, the leading LLMs achieve scores of around 90 in Arithmetic and Sorting subtasks. It should be noted that this does not necessarily reflect strong computational capabilities, but rather because the arithmetic and sorting questions decomposed from MWP and VRQ are relatively easy. A notable weakness across all LLMs is their weak ability to solve equations, with even the top performer, Qwen2.5, scoring only 58.0. Our analysis in Section 6 suggests that this shortage significantly hampers LLM performance in more complex problems. In the Formula Application subtask, Qwen2.5 leads with a score of 79.4, followed by GPT-4o mini (76.7) and DeepSeek-R1 (73.7). This task requires selecting the correct formula from variations; the unsatisfactory performance highlights the importance of eliminating hallucinations in mathematical reasoning.

The most notable data in the Numerical Parsing aspect is the poor performance of MLLMs. Indeed, the scores of three MLLMs on three tasks are even lower than 10. A case analysis shows that MLLMs are almost entirely unable to effectively extract mathematical information on these primary school-level problems, and they mainly rely on the information provided in the text to solve the VRQ. Although the highest score for digit recognition is only 17.6 points for GPT-4o mini, they can actually recognize a considerable number of digits in the image, but cannot determine which digits are useful for solving the problem. As a result, MLLMs may also exhibit serious hallucinations in the presence of redundant information in the image.

6 Quantitive Analysis

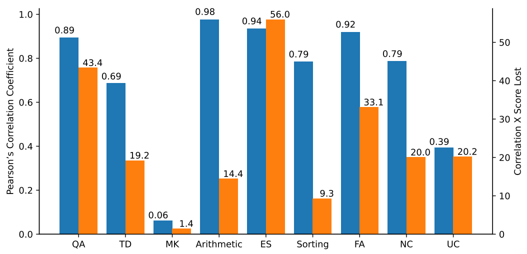

Figure 3: The Pearson's correlation with MWP and (Correlation X Score Lost) of each subtask.

Influence of Subtasks on High-level Task To quantify the influence of various abilities on LLM performance, we computed the Pearson Correlation Coefficient between the scores of MWPs and each subtask. A higher correlation value signifies greater relevance of the subtask to overall MWP performance. The results are shown in Figure 4. Additionally, to identify LLMs' potential weaknesses, we multiplied the average score loss on each subtask by its corresponding correlation coefficient with MWP. This approach highlights the subtasks that perform poorly and have a significant impact on MWP scores. From a correlation perspective, tasks such as Arithmetic (0.98), ES (0.94), FA (0.92), and QA (0.89) show strong ties to MWP performance. However, when considering score losses, it becomes evident that ES (56.0), QA (43.4), and FA (33.1) represent the key weaknesses of LLMs, as their scores for the Arithmetic task is already satisfying. This suggests that LLMs need to enhance their ability to solve equations and handle abstract reasoning problems as a top priority.

Multi-modal Through case analysis, we identified that the failure of MLLMs in NR tasks stems primarily from their inability to extract only the required numbers. Although these models recognize numbers with relatively high accuracy, they often randomly select numbers from the image without focusing on the relevant areas necessary for solving the problem. This leads to a significant accuracy drop when redundant data is present in the image (averaging 43.1 to 1.3). In VDQ tasks, MLLMs exhibit prominent hallucinations, resulting in inferences and analyses that deviate from the actual content of the image. In OC tasks, MLLMs fail not only due to their inability to select the correct objects based on instructions but also due to poor performance in counting large, patterned groups (e.g., $10times10$ arranged blocks). Hence, MLLMs struggle to extract meaningful information from images when addressing visual reasoning problems, relying primarily on text-based data. This suggests that some previous work (Liu et al., 2024) focused on enhancing feature extraction through key region-of-interest identification in images may fail to yield sufficiently satisfactory results in mathematical contexts.

Difficulty in Information Identification The Numerical Parsing tasks require LLMs to extract accurate and relevant information from data presented in various formats. However, analysis of LLM responses in this aspect revealed a consistent issue in the multimodal task NR, where LLMs tend to over-identify irrelevant information. Though this issue was somewhat mitigated in NR compared to other tasks. To assess the impact of this behavior on model performance and robustness, we introduced an unrelated, random problem before each task (e.g., inserting a word problem requiring solving an equation before an arithmetic question). This manipulation led to an average score reduction across the text-modality subtasks for four aspects: 11.3 points for Complex Reasoning, 8.7 points for Understanding, 21.5 points for Calculation, and 29.7 points for Numerical Parsing. These findings highlight that the inclusion of extraneous information significantly impairs LLM performance on mathematical tasks. Further case analysis revealed that while LLMs struggle to identify relevant data in lower-level tasks, they effectively discard incorrect answers through logical reasoning in higher-level tasks, leading to a lesser performance degradation in those cases.

7 The Plug-and-Play Math Model

Despite the impressive language modeling capabilities of large language models (LLMs), their performance in tasks involving simple arithmetic, number recognition, and factual retrieval remains

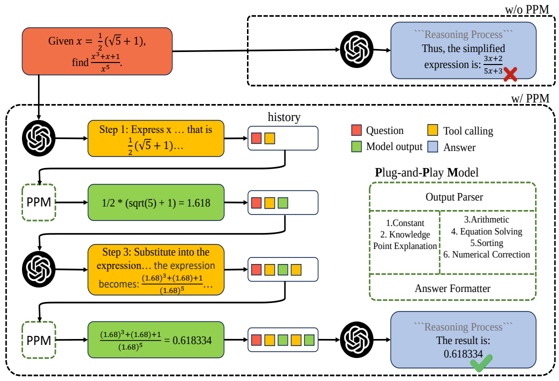

Figure 4: Accuracy comparison of four models on five subtasks w/ and w/o plug-and-play math model.

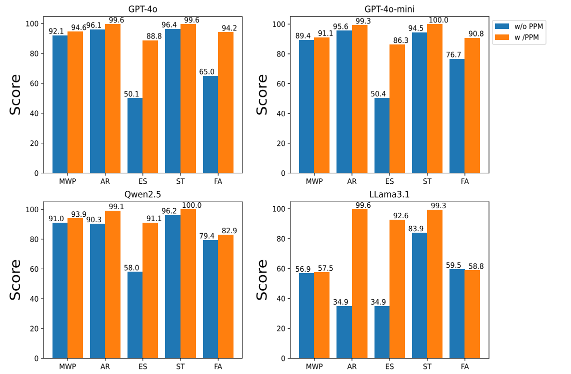

Figure 5: Accuracy comparison of four models on five subtasks w/ and w/o plug-and-play math model.

suboptimal. This limitation is primarily due to the tokenization and training approach inherent to LLMs, making substantial improvements in these areas difficult through simple model adjustments. However, our quantitative analysis indicates that LLMs' accuracy on low-level tasks significantly influences their performance on more complex tasks. To address this, we propose a **Plug-and-Play Math Model** (PMM), a dynamic toolkit that enhances the mathematical processing abilities of LLMs. Unlike prior approaches that rely on self-supervised fine-tuning, this model focuses on simplifying tool integration by streamlining API calls.

This model supports several functions, including 1) Arithmetic, 2) Equation Solving, 3) Sorting, 4) Knowledge Point Explanation, 5) Constant Storage, and 6) Numerical Correction. As depicted in Figure 4, the LLM can generate natural language requests for tool calls invarious formats (e.g. latex, unicode, Markdown), which the model then parses to identify task requirements and return appropriate results. This method reduces the communication overhead and increases the accuracy of tool utilization while preserving the core competencies of the LLM. Detailed descriptions of the PMM's functions are provided in Appendix B.

Result To evaluate the effectiveness of the PMM, we conducted comparative experiments on four LLMs that support tool calling, assessing performance on the MWP task and four subtasks within the Calculation aspect. The Calculation aspect subtasks were selected to test PMM's plug-and-play capability, as these tasks can be solved directly via tool calls. PMM utilizes an Output Parser to analyze LLM tool calls, extract specific requests, perform calculations, and return the results.

The results show a substantial improvement in performance after applying PMM. The average score for Arithmetic and Sorting tasks reached 99.4% and 99.7%, respectively, while the score for Equation Solving increased by 41.4%. Formula Application saw less significant improvement, primarily due to the varied expressions of formulas and mismatches between the stored definitions in PMM and those in the questions. Nevertheless, the significant gains in the Calculation aspect highlight PMM's effectiveness for simple, single-step problems.

For MWP tasks, which require multi-step reasoning, the average score improvement was 2.5%. Notably, the Question Abstract capability of GPT-4o, GPT-4o-mini, and Qwen2.5, which have significantly improved in the MWP task, is higher than Llama3.1. This ability is a key factor in PMM's success with more complex tasks. In conclusion, PMM can enhance the math capabilities of LLMs, particularly for single-step calculations and numerically intensive problems.

8 Related Work

The evaluation of LLMs in mathematical reasoning has seen significant advancements through the development of various benchmarks targeting distinct cognitive tasks and problem-solving abilities. MWPs have been a central focus, as they mirror real-world applications of mathematical reasoning and knowledge integration. Datasets like GSM8K (Cobbe et al., 2021), APE210K (Zhao et al., 2020), MATH401 (Yuan et al., 2023), and Math23K (Wang et al., 2017) provide diverse problem sets ranging from elementary to undergraduate levels, assessing foundational to advanced reasoning skills. In pursuit of more rigorous assessments, the Advanced Reasoning Benchmark (Sawada et al.) sourced from graduate-level exams and professional resources, covering topics from undergraduate to early graduate curricula. OlympiadBench (He et al., 2024), FrontierMath (Glazer et al., 2024), PutnamBench (Tsoukalas et al., 2024), and OmniMATH (Gao et al., 2024) focus on olympiad-level mathematics, curating problems from international competitions like IMO and AMC. However, these benchmarks do not fully capture the limitations of LLMs' capabilities. For example, when models provide incorrect answers, it remains unclear whether the failure stems from computational errors or misinterpretation of the question. Some efforts, such as LILA (Mishra et al., 2022), attempt to address this by breaking down tasks into subtasks. Akhtar et al. (2023) introduced a framework to probe LLMs' numerical reasoning at various levels. But these frameworks lack cross-task correlations, testing LLMs' abilities in isolation.

The mathematical ability of MLLM is also a focus in both academia and industry, MathVista (Lu et al., 2023) is a benchmark designed to combine challenges from diverse mathematical and visual tasks and systematically analyze the mathematical reasoning capabilities of SOTA MLLMs in visually complex scenarios. MathVerse (Zhang et al., 2024) meticulously collects 2,612 high-quality, multi-subject math problems with diagrams to assess whether and how much MLLMs can truly understand the visual diagrams for mathematical reasoning. However, these evaluation benchmarks primarily emphasize image comprehension, neglecting a detailed exploration of text modality's role in numerical reasoning.

9 Conclusion

This paper proposes PyraMathBench, a comprehensive hierarchical benchmark that includes 27,215 questions derived from 7,404 math word problems, covering 4 key cognitive aspects, 14 subcategories, and 2 modalities, ensuring a comprehensive evaluation. Our evaluation of multiple LLMs and MLLMs highlights their limitations in problem abstraction, equation solving, and image-based information extraction, which impede accurate inferences on complex mathematical tasks. These findings underscore the need for improved logical reasoning and the reduction of multimodal hallucinations. Through quantitative analysis, we assess the deficiencies of each LLM and the influence of individual subtasks on high-level task performance. We also propose the plug-and-play math model, a dynamic toolkit designed to enhance the mathematical capabilities of LLMs. Experimental results demonstrate that this model significantly improves LLMs' performance in computational and complex reasoning tasks.

10 Limitations

This study annotates subtasks by decomposing the MWP and VRQ problems, though it is important to note that this decomposition is not the only possible approach regarding task types and content. While various strategies have been employed to mitigate the impact of this issue during evaluation, it might still influences the results, particularly in the Understanding aspect. Furthermore, our task decomposition method does not independently evaluate the full range of LLM language capabilities, which means our classification system does not include all atomic tasks. This is a direction for our future work. Moreover, the current study focuses on English only. Additional research could be conducted on a diverse range of further languages.

While the plug-and-play math model is designed to enhance LLMs' performance on PyraMathBench's subtasks, it is primarily optimized for these specific tasks. Consequently, its effectiveness may not be as pronounced in other mathematical domains, such as formula proofs or algebraic calculations, which are not part of the current subtask set.

References

Mubashara Akhtar, Abhilash Shankarampeta, Vivek Gupta, Arpit Patil, Oana Cocarascu, and Elena Simperl. 2023. Exploring the numerical reasoning capabilities of language models: A comprehensive analysis on tabular data. *arXiv preprint arXiv:2311.02216*.

Anthony Chen, Panupong Pasupat, Sameer Singh, Hongrae Lee, and Kelvin Guu. 2023. Purr: Efficiently editing language model hallucinations by denoising language model corruptions. *arXiv preprint arXiv:2305.14908*.

Chung-Chi Chen, Hen-Hsen Huang, Hiroya Takamura, and Hsin-Hsi Chen. 2019. Numeracy-600k: Learning numeracy for detecting exaggerated information in market comments. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6307–6313.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro

Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

DeepSeek-AI, Daya Guo, and Dejian Yang et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, et al. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, et al. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Aaron Grattafiori, Abhimanyu Dubey, and Abhinav Jauhri et al. 2024. The llama 3 herd of models.

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

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

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *ACM Computing Surveys*, 55(12):1–38.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Chengyue Jiang, Zhonglin Nian, Kaihao Guo, Shanbo Chu, Yinggong Zhao, Libin Shen, and Kewei Tu. 2020. Learning numeral embedding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2586–2599, Online. Association for Computational Linguistics.

Nate Kushman, Yoav Artzi, Luke Zettlemoyer, and Regina Barzilay. 2014. Learning to automatically solve algebra word problems. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 271–281, Baltimore, Maryland. Association for Computational Linguistics.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning.

Tiedong Liu and Bryan Kian Hsiang Low. 2023. Goat: Fine-tuned llama outperforms gpt-4 on arithmetic tasks. *arXiv preprint arXiv:2305.14201*.

Zuyan Liu, Yuhao Dong, Yongming Rao, Jie Zhou, and Jiwen Lu. 2024. Chain-of-spot: Interactive reasoning improves large vision-language models.

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*.

Shen-Yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2021. A diverse corpus for evaluating and developing english math word problem solvers. *arXiv preprint arXiv:2106.15772*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and Ashwin Kalyan. 2022. LILA: A unified benchmark for mathematical reasoning. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5807–5832, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2080–2094, Online. Association for Computational Linguistics.

Jean Piaget. 1970. Piaget's theory. In Paul H. Mussen, editor, *Carmichael's Manual of Child Psychology*, volume 1, pages 703–732. John Wiley & Sons, New York.

Tomohiro Sawada, Daniel Paleka, Alexander Havrilla, Pranav Tadepalli, Paula Vidas, Alexander Kranias, John J Nay, Kshitij Gupta, and Aran Komatsuzaki. Arb: Advanced reasoning benchmark for large language models.

Shuming Shi, Yuehui Wang, Chin-Yew Lin, Xiaojiang Liu, and Yong Rui. 2015. Automatically solving number word problems by semantic parsing and reasoning. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1132–1142, Lisbon, Portugal. Association for Computational Linguistics.

Georgios Spithourakis and Sebastian Riedel. 2018. Numeracy for language models: Evaluating and improving their ability to predict numbers. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2104–2115, Melbourne, Australia. Association for Computational Linguistics.

Dhanasekar Sundararaman, Shijing Si, Vivek Subramanian, Guoyin Wang, Devamanyu Hazarika, and Lawrence Carin. 2020. Methods for numeracy-preserving word embeddings. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4742–4753.

Gemma Team, Morgane Riviere, Shreya Pathak, Pier Giuseppe Sessa, Cassidy Hardin, Surya Bhupatiraju, Léonard Hussenot, Thomas Mesnard, Bobak Shahriari, Alexandre Ramé, et al. 2024. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. 2024. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *arXiv preprint arXiv:2407.11214*.

Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep neural solver for math word problems. In *Proceedings of the 2017 conference on empirical methods in natural language processing*, pages 845–854.

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

Jialiang Xu, Mengyu Zhou, Xinyi He, Shi Han, and Dongmei Zhang. 2022. Towards robust numerical question answering: Diagnosing numerical capabilities of nlp systems. *arXiv preprint arXiv:2211.07455*.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Zheng Yuan, Hongyi Yuan, Chuanqi Tan, Wei Wang, and Songfang Huang. 2023. How well do large language models perform in arithmetic tasks? *arXiv preprint arXiv:2304.02015*.

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

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*.

Wei Zhao, Mingyue Shang, Yang Liu, Liang Wang, and Jingming Liu. 2020. Ape210k: A large-scale and template-rich dataset of math word problems. *arXiv preprint arXiv:2009.11506*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

A Details of the Plug-and-Play Math Model

The Plug-and-Play Math Model supports several functions, including 1) Arithmetic, 2) Equation Solving, 3) Sorting, 4) Knowledge Point Explanation, 5) Constant Storage, and 6) Numerical Correction. Here are the detailed descriptions of each function:

- **Arithmetic.** Detects arithmetic operations requested by LLMs, such as addition, subtraction, multiplication, division, roots, and exponents, and computes the corresponding results.
- **Equation Solving.** Identifies equation-solving tasks involving multiple unknowns or variable definitions and provides numerical solutions for each unknown after solving.
- **Sorting.** Sorts a set of numbers, which may be expressed in various formats, and returns the ordered result.
- **Knowledge Point Explanation.** Supplies mathematical knowledge (e.g., formulas, definitions, and theorem proofs) in response to LLM queries from a local database.
- **Constant Storage.** Stores frequently used mathematical constants (e.g., e, π), retains data from the questions and previous problem-solving steps, providing this information upon request.
- **Numerical Correction.** Automatically compares the LLM's reasoning process with stored constants and alerts the model of potential numerical inaccuracies.

B Detailed Description of each Subtask

In this section, we provide detailed information on each subtask, including 1) aspects, 2) whether it is a multimodal task, 3) size, 4) design rationale and description, and 5) all versions of the prompt we used.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Math Word Problem (MWP)</td><td>Complex Reasoning</td><td>No</td><td>5476</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">Assesses the model's ability to solve mathematical problems presented in natural language and reasoning through complex, real-world problems and translating them into mathematical solutions.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="2">System</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\nThink step by step and answer the following math word problem.</td><td colspan="2">**Question**: \n{question}</td></tr>
</table>

Table 2: Detailed description of subtask Math Word Problem.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Visual Reasoning Porblem (VRP)</td><td>Complex Reasoning</td><td>Yes</td><td>1928</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">This subtask evaluates the model's ability to combine textual and visual information for solving mathematical problems. MLLMs need to reason across multiple modalities and extract relevant insights from both text and images. The tasks include various types such as geometry problems, VQA, and statistic reasoning.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="2">System</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\nThink step by step and answer the following visual reasoning problem based on the following image.</td><td colspan="2">**Question**: \n{question}</td></tr>
</table>

Table 3: Detailed description of subtask Visual Reasoning Problem.

Subtask	Aspect	Multi-modal	Size
Question Abstraction (QA)	Understanding	No	5459

Description
This subtask requires LLMs to convert natural language problems into solvable structured mathematical representations, including arithmetic, equations, and sorting numbers.

Prompt	
System (arithmetic)	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe Math Word Problem(MWP), as a manifestation of questions, can be understood as the process of solving it by computing an operational expression. The following will provide a math word problem that you need to abstract into an arithmetic expression that can be directly interpreted by Python, such as 6 * (5+3).\nYou can use functions from the ṁaths̈tandard library.	**Math Word Problem**: \n{MWP}
System (equation)	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nYou are a helpful AI robot, you can solve mathematical problem accurately, The Math Word Problem(MWP), as a manifestation of questions, can be understood as the process of solving a equation or system of equations. The following will provide a math word problem that you need to abstract into a equation or system of equations. Specifically, you need to first list the unknown variable(s) that need to be used after abstraction. If there are multiple unknown variables, use commas to separate them. Then list the abstract equation, and if there are multiple equations, list them in multiple lines.\nYou can use functions from the ṁaths̈tandard library.	**Math Word Problem**: \n{MWP}
System (sorting)	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe following will provide a math word problem(MWP) that you have to compare or sort some numbers in the MWP to solve it.\nExtract the numbers that need to be compared or sorted from the questions.	**Math Word Problem**: \n{MWP}

Table 4: Detailed description of subtask Question Abstraction.

Subtask	Aspect	Multi-modal	Size
Task Decompisition (TD)	Understanding	No	4175

Description
The LLMs are required to analyze the MWP and determine the necessary steps for solving it. The LLMs must have a sufficient understanding of the text and mathematical logic to answer correctly. This subtask has a certain degree of openness..

Prompt	
System	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe Math Word Problem(MWP), as a type of comprehensive mathematical problem, it may require various mathematical operations such as calculations and solving equations during to solve it.\n"Based on the MWP provided below, choose what mathematical operations are needed to solve it.	**Math Word Problem**: \n{MWP}\n\n**Mathematical operation list**: \nA: Additional information such as mathematical formulas, constants, theorems, etc. that are not directly provided in the question.\nB: Solve an equation or system of equations.\nC: Perform mathematical arithmetic.\nD: Sort or compare the data in the question.\nE: Identify and only identify the numbers in various formats provided in the information that are needed to solve the problem.\nF: Identify the numerical unit(s) required to obtain the answer.\nG: Identify and only identify the numbers in various formats provided in the image that are needed to solve the problem.\nH: Quantify data in images that are not directly presented in numerical terms.\nI: Count the number of certain objects in the picture.

Table 5: Detailed description of subtask Task Decompisition.

Subtask	Aspect	Multi-modal	Size
Math Knowledge (MK)	Understanding	No	58

Description
This subtask evaluates the model's ability to leverage fundamental mathematical knowledge, such as approximations of constants and geometric formulas that are not explicitly provided in the problem. For example, the approximation of e or applying the quadratic formula for the root of an equation.

Prompt	
System	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe following will provide a math word problem(MWP). To solve this MWP, an additional knowledge point, such as a theorem or formula, is required. Please answer the name of this knowledge point.	**Math Word Problem**: \n{question}

Table 6: Detailed description of subtask Math Knowledge.

Subtask	Aspect	Multi-modal	Size
Arithmetic	Calculation	No	3499

Description
This subtask evaluates the model's proficiency in performing basic mathematical operations such as four operations, root operation, exponential operation, etc.

Prompt	
System	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nPlease calculate the provided arithmetic expression.	**Arithmetic Expression**: {expression}

Table 7: Detailed description of subtask Arithmetic.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Equation Solving (EQ)</td><td>Calculation</td><td>No</td><td>1571</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">Require LLMs to solve both single-variable and systems of equations. It evaluates the model's algebraic skills and its capacity for handling more advanced mathematical structures.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="2">System (equation)</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\n"Please solve the provided equation.</td><td colspan="2">**Unknown Variable**: {variable}\n**Equation**: {equation}"</td></tr>
<tr><td colspan="2">System (system of equations)</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\n"Please solve the provided system of equations.</td><td colspan="2">**Unknown Variable**: {variables}\n**System of Equations**: {equations}"</td></tr>
</table>

Table 8: Detailed description of subtask Equation Solving.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Sorting</td><td>Calculation</td><td>No</td><td>581</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">This subtask evaluates a model's ability to arrange numbers or objects in a specific order, assesses its understanding of order relationships and computational reasoning.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="2">System</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\nPlease sort the following numbers in ascending order.</td><td colspan="2">**Numbers**: {numbers}</td></tr>
</table>

Table 9: Detailed description of subtask Sorting.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Formula Application (FA)</td><td>Calculation</td><td>No</td><td>246</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">This subtask requires the LLMs to recognize and apply specific formulas to solve problems and tests the LLMs' familiarity with mathematical relationships.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="2">System</td><td colspan="2">Human</td></tr>
<tr><td colspan="2">You are a helpful AI robot, you can solve mathematical problem accurately.\nChoose the correct definition for the following theorem or formula.</td><td colspan="2">**Theorem or Formula**: formula\n**Options**:\noptions</td></tr>
</table>

Table 10: Detailed description of subtask Formula Application.

Subtask	Aspect	Multi-modal	Size
Number Conversion (NC)	Numerical Parsing	No	2877

Description
This subtask evaluates an LLM's ability to recognize and interpret important numbers in different formats, such as Arabic numerals, written words, and scientific notation. For example, "one hundred and three" or "1.13e+2" should be converted into "113". LLM also needs to avoid identifying invalid information.

Prompt	
System	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe Math Word Problem(MWP), as a type of comprehensive mathematical problem, it requires identify important information in the question to solve the problem.\nThe following will provide a math word problem, and you need to identify and **ONLY** identify "the numbers in various formats provided in the information that are needed to solve the problem.	**MWP**: {MWP}

Table 11: Detailed description of subtask Number Conversion.

Subtask	Aspect	Multi-modal	Size
Unit Conversion (UC)	Numerical Parsing	No	1089

Description
In MWP, especially in physics-related problems, unit conversion is extremely important. This subtask measures an LLM's understanding of various units of measurement and its ability to convert between them. For example, converting "5 kW·h" to J or "100°C" to Fahrenheit.

Prompt	
System	Human
You are a helpful AI robot, you can solve mathematical problem accurately.\nThe Math Word Problem(MWP), as a type of comprehensive mathematical problem, it requires identify important information in the question to solve the problem.\nThe following will provide a math word problem, and you need to identify the number with unit(s) required to solve the MWP. (Ignore numbers without units.)	**MWP**: {MWP}

Table 12: Detailed description of subtask Unit Conversion.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Numeral Recognition (NR)</td><td>Numerical Parsing</td><td>Yes</td><td>133</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">This task assesses an LLM’s ability to extract mathematical content like numbers, variables, and formulas from images. The model may need to extract and interpret a formula from an image of handwritten notes. LLM also needs to avoid identifying invalid information.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="3">System</td><td>Human</td></tr>
<tr><td colspan="3">You are a helpful AI robot, you can solve mathematical problem accurately.\nThe Visual Reasoning Problem(VRP), as a type of comprehensive mathematical problem, it requires identify important information in the image to solve the problem.\nThe following will provide a visual reasoning problem and a image, you need to identify and **ONLY** identify the numbers in the image that are needed to solve the problem.</td><td>**VRP**: {VRP}</td></tr>
</table>

Table 13: Detailed description of subtask Numeral Recognition.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Visual Data Quantification (VDQ)</td><td>Numerical Parsing</td><td>Yes</td><td>38</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">In the image, some data is not directly presented in numerical form, such as the time pointed by the clock or the length of an object. This subtask evaluates the model’s ability to understand instructions and quantify nonvalue data in images.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="3">System</td><td>Human</td></tr>
<tr><td colspan="3">You are a helpful AI robot, you can solve mathematical problem accurately.\nIdentify the specified data from the following image. If the data is not presented directly in numerical form, you need to quantify it.</td><td>**Question**: \n{question}</td></tr>
</table>

Table 14: Detailed description of subtask Visual Data Quantification.

<table>
<tr><th>Subtask</th><th>Aspect</th><th>Multi-modal</th><th>Size</th></tr>
<tr><td>Object Counting (OC)</td><td>Numerical Parsing</td><td>Yes</td><td>85</td></tr>
<tr><th colspan="4">Description</th></tr>
<tr><td colspan="4">This subtask requires models to count specified objects in an image based on a given description. It tests the models’ visual reasoning and object recognition skills.</td></tr>
<tr><th colspan="4">Prompt</th></tr>
<tr><td colspan="3">System</td><td>Human</td></tr>
<tr><td colspan="3">You are a helpful AI robot, you can solve mathematical problem accurately.\nIdentify the number of specified objects from the following image.</td><td>**Target Object**: \n{question}</td></tr>
</table>

Table 15: Detailed description of subtask Object Counting.