Can LLMs *understand* Math? Exploring the Pitfalls in Mathematical Reasoning

Aditeya Baral
Courant Institute of Mathematical Sciences
New York University
ab12057@nyu.edu

Ayush Rajesh Jhaveri
Courant Institute of Mathematical Sciences
New York University
aj4332@nyu.edu

Tiasa Singha Roy
Courant Institute of Mathematical Sciences
New York University
ts5478@nyu.edu

Yusuf Baig
Courant Institute of Mathematical Sciences
New York University
yb2510@nyu.edu

Abstract

Large language models (LLMs) demonstrate considerable potential in various natural language tasks but face significant challenges in mathematical reasoning, particularly in executing precise, multi-step logic. However, current evaluation frameworks judge their performance solely based on accuracy, which only accounts for the final answer. This study explores these pitfalls by employing a novel evaluation framework. We propose an evaluation metric called the MAPLE score, which holistically quantifies reasoning misalignment by integrating error rates, redundancy, and validity.

1 Introduction

Large Language Models (LLMs) have demonstrated impressive capabilities across tasks such as text generation, language translation, question answering, and sentiment analysis. However, their performance diminishes in complex reasoning tasks, particularly within the mathematical domain. While LLMs perform adequately on elementary math problems, they often need help with tasks requiring precise, step-by-step reasoning, leading to errors in solution validity and logical consistency. These limitations underscore the need for a holistic evaluation of their mathematical reasoning abilities to identify weaknesses and guide targeted improvements.

This study introduces a multi-stage evaluation methodology to systematically assess LLMs' mathematical reasoning capabilities. We prompt various LLMs to generate solutions using the MATH dataset. The process involves iterative self-reflection to evaluate reasoning steps, identify misalignments, and compile error labels such as calculation errors, misinterpretations, and incoherent outputs.

We propose the MAPLE (Mathematical Pitfalls and Logical Evaluation) score as a holistic metric to quantify reasoning misalignment. This score incorporates error rates, redundancy, and validity, offering a comprehensive evaluation. Our findings reveal patterns of errors and limitations across different mathematical topics and levels, providing insights into the challenges LLMs face in complex reasoning tasks.

2 Related Work

We build on the redundancy and validity scores introduced in ReasonEval[1], which advocates for evaluations beyond accuracy. Inspired by self-reflection[2], which enables LLMs to self-correct, we

adapt this approach to identify reasoning pitfalls. Unlike prior methods relying on external inputs, our approach uses oracle labels within the LLM to autonomously analyze and address reasoning failures.

3 Approach

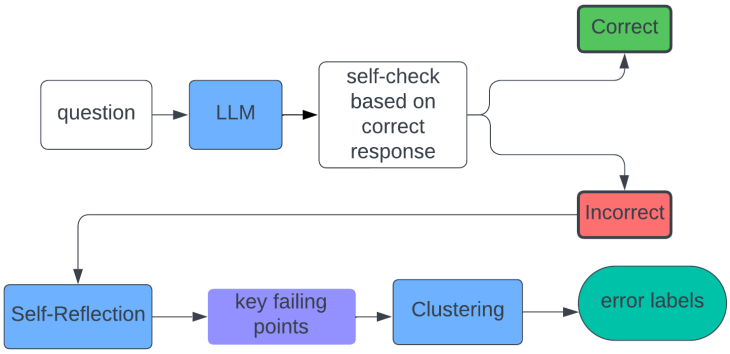

Figure 1: **Architecture of our LLM Agent evaluation and identification of errors**. The LLM's generated answer is evaluated in a multi-turn set-up to identify the failing points in the generated response using *self-reflection* and *clustering*.

3.1 Stage 1 - Evaluating the Final Answer and Approach

As shown in Figure 1, we initialize the LLM agent to generate an answer, a'_i based on the initial prompt p_i and question q_i. For each a'_i, we create a multi-turn setup, which includes providing the agent with the correct solution to induce self-checking. We provide the LLM with the correct final answer value a_{fi} to check. This approach verifies whether the final answers (a'_{fi}, a_{fi}) match to identify incorrect cases for the LLM agent.

For these incorrect responses, we invoke self-reflection for the incorrect samples. This step uses generated and correct response pair (a'_i, a_i) to return a generation analysis which highlights the points of misalignment of the reasoning steps with the actual solution. We use BERT[3] based embeddings to wrap these failing points and perform clustering to compile a consistent set of error labels L to encompass issues across the samples.

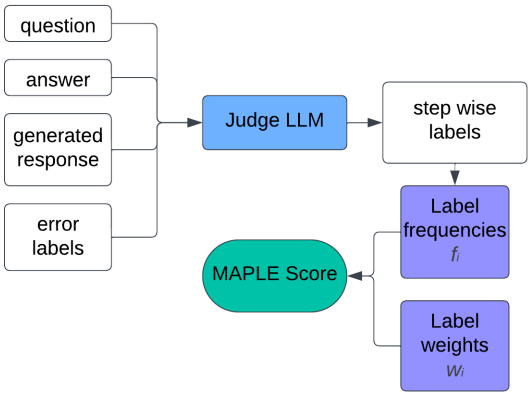

Figure 2: **Architecture of our Judge LLM Agent and MAPLE Score**. The Judge LLM provides step-wise analysis to compute MAPLE score using *label-frequencies* and *label-weights*.

3.2 Stage 2 - LLM as a Judge and Computing Incorrectness

While the compiled error labels L help provide insights into the broad classification of mathematical mistakes made while solving a problem, it is also crucial to identify and correlate each mathematical reasoning step with these labels.

As shown in Figure 2, we initialize a judge LLM and prompt it with the error labels L, the question q_i, the correct solution a_i, and the generated solution a_i' to create a set of error labels S corresponding to each erroneous reasoning step. These step-wise error labels are then used to compute the *degree of incorrectness* in the incorrect solution.

3.3 Stage 3 - Computing MAPLE Score

Given a step-wise collection of error labels $S = [[l_1, l_2, ..., l_n], [l_1, l_2, ..., l_n], ...]$, we first compute the frequency f_l of each label $l \in L$. This represents the relevance of a particular label to an incorrectly generated sample. We consider the logarithm of frequencies instead of raw frequencies to reduce sensitivity to large frequency values, with a $+1$ added term to ensure numerical stability when no errors are made.

We further compute the error rate e as the weighted average of the frequencies f_l weighted by the penalty score for each label w_l. The penalty score for each label, as shown in section A.1, was aggregated over the results of a human survey which ranked the error labels in increasing order of incorrectness. The penalty score allows us to individually weigh the contribution of each error label to compute the MAPLE score.

$$e = \frac{\sum_{l \in L} w_l \cdot log(1 + f_l)}{\sum_{l \in L} w_l} \tag{1}$$

The redundancy r and validity v of the overall solution are computed using ReasonEval[1] within the range of values $r, v \in [0, 1]$. The MAPLE score decreases with an increase in validity and increases with an increase in redundancy of the solution. Finally, we express the error metric e by applying the $tanh$ function such that MAPLE$_{score} \in [0, 1]$.

$$\text{MAPLE}_{score} = tanh\left(\frac{e \cdot v}{r}\right) \tag{2}$$

4 Experiments

4.1 Data

We use the MATH [4] dataset, which comprises 12,500 competition mathematics problems. The problems vary in complexity from levels 1 through 5 and span mathematics categories, consisting of Intermediate Algebra, Precalculus, Algebra, Prealgebra, Geometry, Counting & Probability, and Number Theory.

4.2 Experiment Setup

We choose the four models for our study from the four popular LLM families, namely Gemini, GPT-4, Llama, and Mixtral. More details on the models used can be found in section A.2.

4.3 Evaluation method

To evaluate the reliability of the MAPLE score, we perform a thorough validation process on the judge LLM predictions. We manually annotate a representative sample from the MATH dataset, consisting of incorrect mathematical answers generated by various LLM families. A multi-label approach is employed to comprehensively capture the types of errors present in the responses. The error labels predicted by the judge LLM are then compared against these human-annotated labels. This alignment accuracy can be found in section A.3 .

4.4 Results

4.4.1 Error Classification

Based on the clustering of the LLM self-reflections for incorrect answers, we obtained the following error labels:

1. **Complete misunderstanding**. The model completely fails to understand the question and its requirements.
2. **Partial misunderstanding**. The model partially fails to understand the question or its requirements.
3. **Incorrect Method**. The model applies a concept with the correct formula but it is unrelated to the given question.
4. **Incorrectly Applied Method**. The model chooses the right concept with the correct formula which is related to the given question and can be used to solve it, but applied it incorrectly.
5. **Calculation Error**. Errors in arithmetic calculations.
6. **Incoherent Output**. Junk text with repeated characters or phrases.
7. **No Solution**. Failure to reach a final answer.

The error labels are provided as prompts to the judge LLM, which identifies the errors present in the mathematical answers generated. These identified errors are subsequently used for computing the MAPLE score.

4.4.2 MAPLE Score Computation

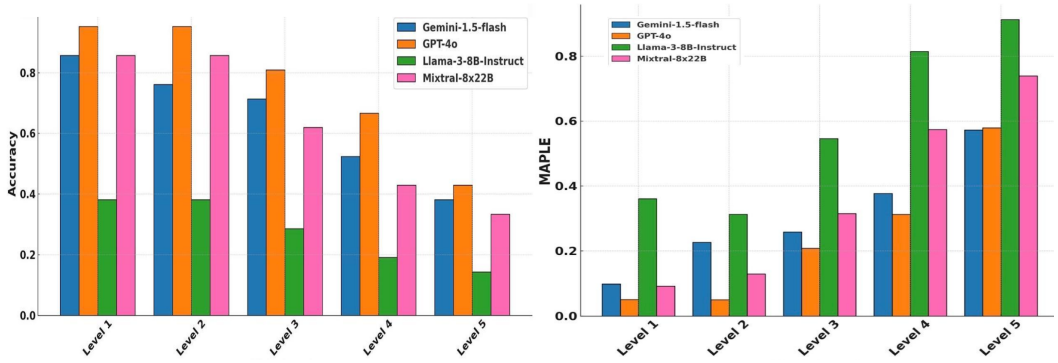

Figure 3: **Comparison of LLM performance across difficulty levels on the MATH Dataset**. Level 1 represents the easiest and Level 5 represents the toughest math problems. We observe a correlation between final answer accuracy and the degree of incorrectness represented by the MAPLE score.

We evaluated the mathematical answers generated by various LLMs for the MATH dataset using our proposed approach. The results, categorized by difficulty level, are presented in Figure 3. The left graph demonstrates that as the difficulty level increases, accuracy declines across all models. Conversely, the right graph shows that the MAPLE score rises with increased difficulty, with the highest MAPLE score observed for the Llama model. This suggests that the Llama model exhibits the most significant issues in mathematical reasoning.

Additionally, we performed a topic-wise analysis of the LLM-generated answers, the results of which are provided in section A.4.

5 Future Work

Future efforts will expand the evaluation framework to include a broader range of error types, such as topic-specific reasoning issues, and incorporate ranking of error labels for more nuanced scoring. Addressing hallucination in LLMs through fine-tuning for evaluation-specific tasks and exploring alternatives to LLMs as judges will enhance alignment with human judgment. Testing the framework on diverse models, datasets, and interdisciplinary reasoning tasks will validate its robustness. Additionally, refining methods to reduce redundancy and improve logical coherence in reasoning steps will be critical for advancing LLMs' mathematical problem-solving capabilities.

References

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

[2] Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36, 2024.

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[4] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

[5] Shuai Peng, Ke Yuan, Liangcai Gao, and Zhi Tang. MathBERT: A Pre-Trained Model for Mathematical Formula Understanding, 2021.

A Appendix

A.1 Error Label Penalty Weights

Table 1: Error Label Penalty Weights

Error Label	Penalty Weight
Complete Misunderstanding	0.95
Partial Misunderstanding	0.75
Incorrectly Applied Method	0.40
Calculation Error	0.10
Incoherent Output	1.00
No Solution	1.00

A.2 Experiment Setup

The four LLMs chosen for our study are Gemini-1.5-Flash from Gemini, GPT-4o from GPT-4, Llama-3-8B-Instruct[1] from LlaMa, and Mixtral-8x22B[2] from Mistral. All fine-tuned open-source models were taken from HuggingFace, while respective APIs were used for GPT-4o and Gemini-1.5-Flash. Outputs were generated with the temperature parameters set to `1` for Gemini-1.5-Flash and GPT-4o, `0.3` for Mixtral-8x22B, and `0.05` for Llama-3-8B-Instruct.

To cluster the failing points and compile a set of error labels L, we use MathBERT[5][3], a BERT model fine-tuned on math word problems. Similarly, we use Mathstral-7B[4], a fine-tuned Mistral model for mathematical problem solving from HuggingFace for our Judge LLM.

A.3 Evaluation of LLM as Judge

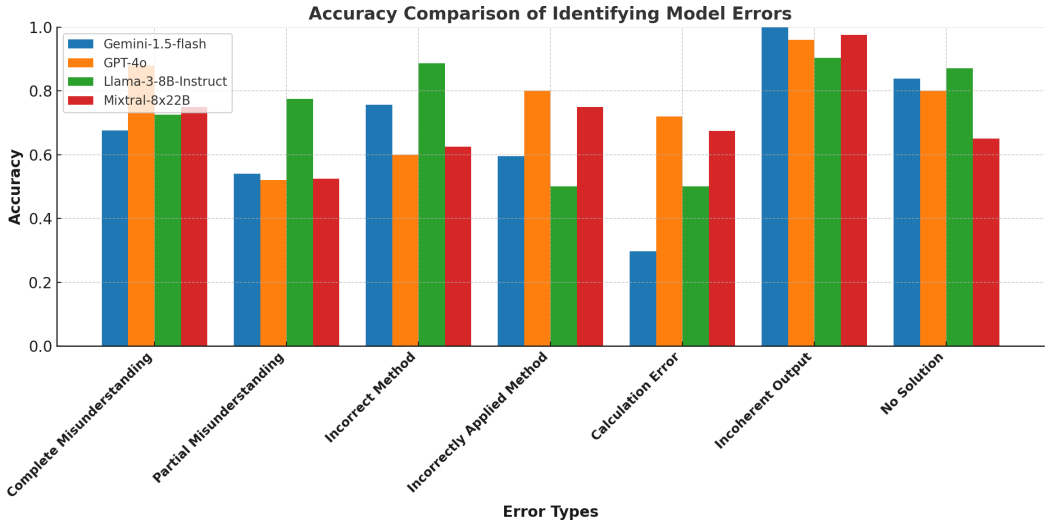

Figure 4: **Comparison of accuracy of the LLM as a Judge in predicting error labels for generated solutions**. We observe that most predictions match human annotations for a representative sample of 105 evenly-distributed examples across difficulty levels and topics.

[1] `https://huggingface.co/meta-llama/Meta-Llama-3-8B-Instruct`
[2] `https://huggingface.co/mistralai/Mixtral-8x22B-Instruct-v0.1`
[3] `https://huggingface.co/tbs17/MathBERT`
[4] `https://huggingface.co/mistralai/Mathstral-7B-v0.1`

A.4 Topic-wise Evaluation Scores

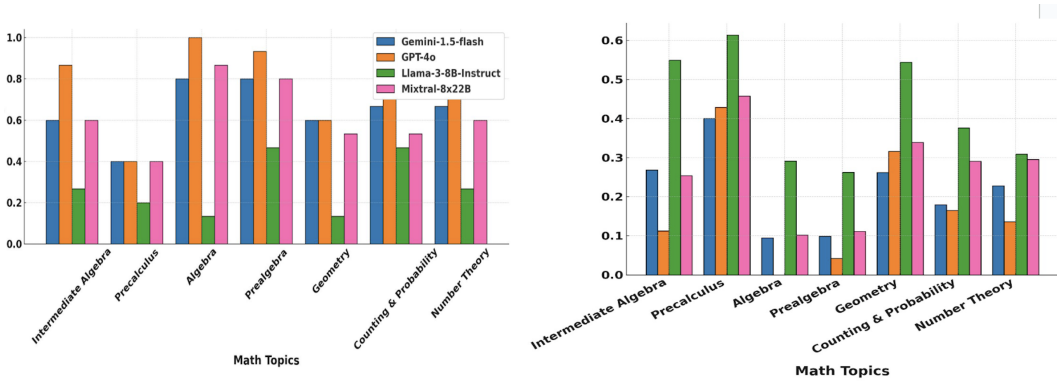

Figure 5: **Comparison of LLM performance across math topics on the MATH Dataset**. We observe that most models perform better at easier topics such as geometry while underperforming at tougher topics such as calculus.