TMATH: A Dataset for Evaluating Large Language Models in Generating Educational Hints for Math Word Problems

Changyong Qi[1,2]*, Yuang Wei[1,2]*, Haoxin Xu[1,2], Longwei Zheng[3,4]†, Peiji Chen[5], Xiaoqing Gu[6]
[1]Lab of Artificial Intelligence for Education, East China Normal University, Shanghai, China
[2]Shanghai Institute of Artificial Intelligence for Education, East China Normal University, Shanghai, China
[3]School of Education, City University of Macau, Macau, China
[4]State Key Laboratory of Cognitive Intelligence, Hefei, China
[5]The University of Electro-Communications, Tokyo, Japan
[6]Department of Education Information Technology, East China Normal University, Shanghai, China

Abstract

Large Language Models (LLMs) are increasingly being applied in education, showing significant potential in personalized instruction, student feedback, and intelligent tutoring. Generating hints for Math Word Problems (MWPs) has become a critical application, particularly in helping students understand problem-solving steps and logic. However, existing models struggle to provide pedagogically sound guidance that fosters learning without offering direct answers. To address this issue, we introduce TMATH, a dataset specifically designed to evaluate LLMs' ability to generate high-quality hints for MWPs. TMATH contains diverse mathematical problems paired with carefully crafted, human-generated hints. To assess its impact, we fine-tuned a series of 7B-scale language models using TMATH. Our results, based on quantitative evaluations and expert assessments, show that while LLMs still face challenges in complex reasoning, the TMATH dataset significantly enhances their ability to generate more accurate and contextually appropriate educational hints. The dataset is available at https://github.com/qi-github-ui/TMATH.

1 Introduction

In recent years, Large Language Models (LLMs) have revolutionized natural language processing (NLP), with applications in personalized instruction, real-time student feedback, and intelligent tutoring systems (ITSs) (Hadi et al., 2024; Stamper et al., 2024). These applications are particularly transformative in education, where LLMs enhance both the quality and accessibility of learning. A critical area of focus is the generation of educational hints, especially for Math Word Problems (MWPs) (Zhang et al., 2024; Srivatsa and Kochmar, 2024; Ahn et al., 2024). By guiding students through problem-solving steps without revealing the answers, LLMs can foster deeper understanding and independent learning. However, current models often struggle to generate pedagogically sound hints, especially in tasks involving complex logic and problem-solving (Cohn et al., 2024).

While LLMs excel at generating fluent and coherent text, their ability to create high-quality educational hints remains underexplored (Gattupalli et al., 2023). Existing models frequently fail to provide guidance that encourages critical thinking, often producing hallucinations or irrelevant content when dealing with more complex logical problems (Azamfirei et al., 2023; Imani et al., 2023; Lin et al., 2024). This limitation not only diminishes their educational value but also raises questions about their suitability for tasks requiring step-by-step reasoning. To address these challenges, we introduce TMATH, a comprehensive dataset designed to rigorously evaluate the performance of LLMs in generating educational hints for MWPs. TMATH encompasses a wide variety of mathematical problems, ranging from elementary to advanced levels, each accompanied by human-generated, pedagogically sound hints. By focusing on the generation of educational hints, TMATH provides a much-needed resource for evaluating the ability of LLMs to not only solve mathematical problems but also offer meaningful guidance that fosters student learning. Our study makes several key contributions:

- We present TMATH, the first dataset specifically designed for evaluating LLMs' ability to generate educational hints in MWPs.
- We fine-tune several 7B-scale LLMs using TMATH and demonstrate that TMATH significantly enhances their performance in generating accurate, contextually appropriate hints.
- Through quantitative metrics and expert assessments, we identify critical areas for im-

*These authors contributed equally to this work.
†Corresponding author. E-mail: lwzheng@cityu.edu.mo

Proceedings of the 31st International Conference on Computational Linguistics, pages 5082–5093
January 19–24, 2025.

provement in LLMs' reasoning and hint generation abilities, particularly in complex problem-solving contexts.

2 Related Work

2.1 The Evolution and Applications of LLMs

In recent years, the evolution of LLMs, such as OpenAI's GPT series, has led to significant advancements in NLP and other complex tasks. The introduction of the GPT model (Radford et al., 2018) with its transformer architecture marked a breakthrough in unsupervised training, enabling the generation of coherent and contextually relevant text. Successive iterations, including GPT-2 (Radford et al., 2019) and GPT-3 (Brown et al., 2020), further enhanced text generation and comprehension by increasing model size and training data. GPT-3.5-turbo [1], for example, improved on these foundations by combining chatbot interactions with traditional text generation, allowing for multi-turn conversations and dynamic context understanding. GPT-4 (Achiam et al., 2023) not only improved the quality of multi-turn conversations but also strengthened performance in more complex tasks, such as mathematical reasoning and programming problem-solving. GPT-4o [2] further optimized processing efficiency, reducing cost and latency while maintaining high-generation capabilities. These models have been successfully applied in areas such as text generation (Mo et al., 2024), question-answering (Zhuang et al., 2024), automatic programming (Nam et al., 2024), and mathematical problem-solving (Yang et al., 2024b).

2.2 Proficiency of LLMs in Resolving Mathematical Problems

The burgeoning recognition of LLMs' potential in education extends to complex query comprehension and meaningful response generation, notably in MWPs solving. LLMs like Bloom were transformed into adept math tutors through LoRA fine-tuning strategies on an elementary school math dataset (Mangrulkar et al., 2022). Notably, the Goat model's fine-tuning on synthetic arithmetic data led to near-perfect large-number operation accuracy (Liu and Low, 2023), outstripping preceding models such as Bloom (Scao et al., 2022), OPT (Zhang et al., 2022), and GPT-NeoX (Black et al., 2022). This included Goat-7B's zero-shot learning surpassing PaLM-540's few-shot learning. To overcome ITSs' limitations with pre-registered problems and student-generated issues, a novel approach using LLMs to convert MWPs into Python code was proposed, integrated into internal system representations, demonstrating high accuracy and significant enhancement potential (Arnau-González et al., 2023). Additionally, a 2024 study shows LLMs have advanced in mathematical reasoning but still struggle with complex quantitative tasks (Ahn et al., 2024).

2.3 Applications of LLMs in Generating Hints

While LLMs have made significant strides in content generation, their primary focus has been on generating direct answers (Imani et al., 2023; Hasan et al., 2024) or programming prompts (Leinonen et al., 2023; Jury et al., 2024). In educational contexts, however, there are unique challenges. One major concern is that LLMs' ability to rapidly provide answers may hinder the development of students' critical thinking and problem-solving skills, both essential for academic and lifelong success (Tang et al., 2023; Grande et al., 2024). This highlights the need for research on how LLMs can offer Socratic, indirect teaching hints. Particularly in the context of mathematical problems, such hints can guide students to grasp the problem-solving process, promoting independent thinking rather than simply providing answers. Despite this potential, LLMs still exhibit inaccuracies and hallucinations when tasked with generating these pedagogical hints, raising concerns among educators.

3 Method

This study builds upon the MATH (Hendrycks et al., 2021) dataset to construct the TMATH dataset, which is designed to support the generation of problem-solving hints for MWPs using LLMs, as shown in Fig.1. The construction of TMATH follows a multi-stage analysis and design process to ensure the scientific validity and effectiveness of the generated hints. First, we conducted an in-depth analysis of the problem-solving steps in the original MATH dataset, extracting key information and identifying potential difficulties. Based on this analysis, we designed a hint generation mechanism following the Socratic teaching method, encouraging students to progressively solve problems

[1] https://chat.openai.com/

[2] https://www.techtarget.com/whatis/feature/GPT-4o-explained-Everything-you-need-to-know

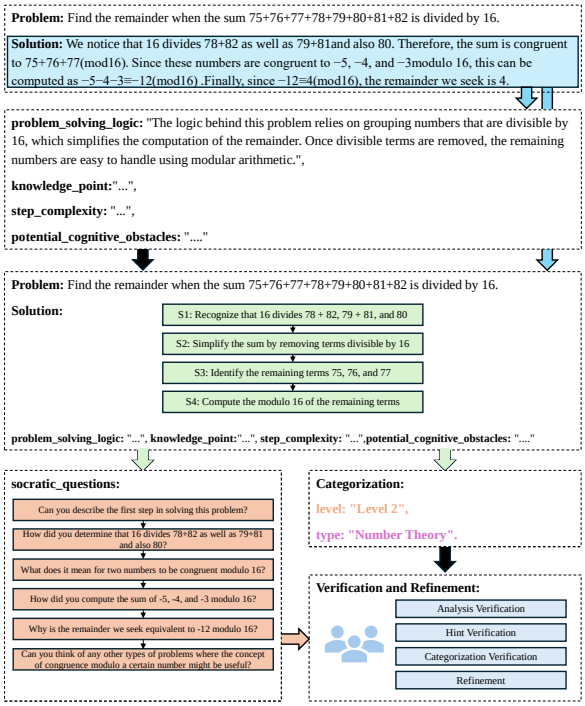


Figure 1: The framework for constructing the TMATH dataset involves several steps. First, the problem-solving structure of the original dataset is analyzed. Then, hint generation rules are designed based on the problem-solving steps. Next, the problems are categorized by difficulty and domain. Experts manually validate and refine the generated hints. Finally, all verified data is integrated to form a complete hint-based mathematical problem dataset.

through guiding questions. The dataset is categorized by difficulty and domain to comprehensively evaluate the model's performance across different areas of mathematics. Finally, the hints were rigorously validated and optimized by mathematics education experts to ensure their quality and guidance effectiveness.

3.1 Solution Analysis

We first conducted a systematic cognitive analysis of the problem-solving paths in the original MATH dataset, aiming to analyze the problem-solving logic, step complexity, and distribution of knowledge points. By evaluating key steps and common errors, we identified potential cognitive obstacles students may face and quantified the complexity of the steps to predict potential problem-solving difficulties, providing a theoretical foundation for hint generation.

3.2 Hint Generation Design

The hint generation design is based on the principles of Socratic teaching, aiming to guide students step by step in understanding and solving problems. Through analyzing each problem-solving step, we identified key steps and potential cognitive barriers. For these steps, hints are divided into three levels: basic understanding hints, thought guidance hints, and error correction hints. The generation of hints follows strict rule-based design to ensure alignment with educational guidance objectives.

3.3 Difficulty and Domain Categorization

We adopted a hierarchical difficulty model, dividing the problems into five levels based on complexity, corresponding to basic, intermediate, and advanced problems. The classification criteria are based on the number of required knowledge points, the complexity of solution steps, and the likelihood of common errors. Additionally, the problems are categorized into seven domains, such as algebra and geometry, to facilitate the evaluation of the model's generalization performance across different areas.

3.4 Human Verification and Refinement

To ensure the effectiveness of the hints, we invited experts in the field of mathematics education to validate the strategy and outcomes of the hint generation process. Each hint was evaluated by the experts to ensure that it guides students along the correct thinking path, followed by further adjustments and optimizations based on feedback. Ultimately, all hints underwent rigorous calibration and quality checks.

4 Dataset and Evaluation Framework

4.1 Dataset

TMATH, which is constructed based on the publicly available MATH dataset's problems and solution steps. Comprising approximately 4000 mathematical problems and Socratic-style hints, TMATH spans various domains from elementary to advanced mathematics. Each problem is paired with detailed step-by-step hints, providing a reliable foundation for training models in answer derivation and explanation. The dataset is organized into five levels of difficulty and categorized into seven subjects including geometry, algebra, intermediate algebra, prealgebra, precalculus, counting & probability, and number theory. Figure 2 presents

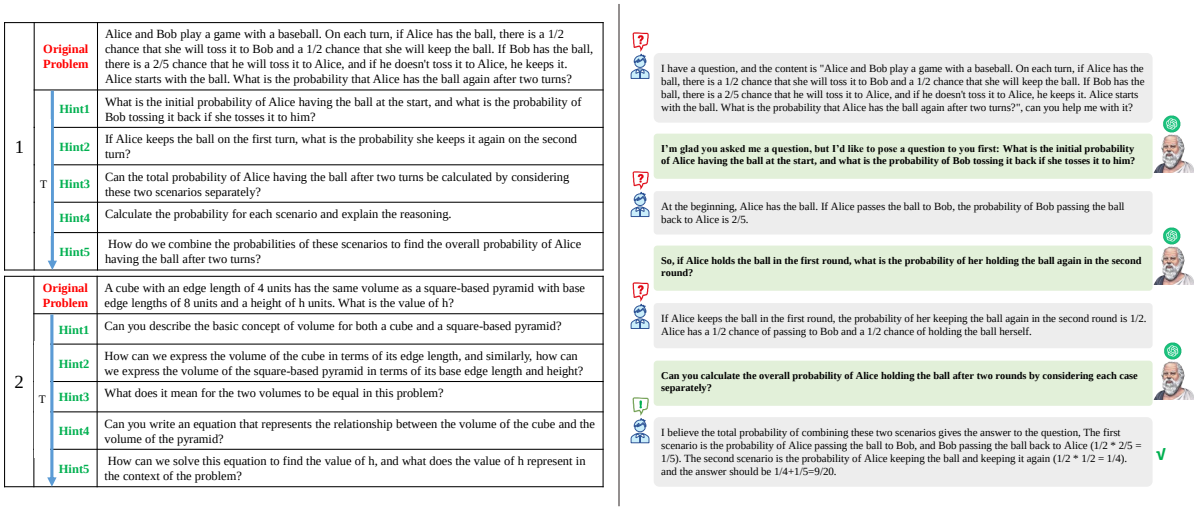


Figure 2: Examples from the TMATH Dataset with Socratic-Style Hints. The left column illustrates the structured, incremental hints designed to guide learners through the problem-solving journey, while the right column demonstrates the interactive prompts crafted by the fine-tuned LLMs.

selected examples from the TMATH dataset as well as instances of Socratic-style hints applied by LLMs fine-tuned on TMATH.

To ensure a comprehensive understanding of the mathematical problems and consider pedagogical elements in our hint design, we collaborated with a team of experts from relevant fields. Each problem was thoroughly discussed to determine the optimal hint design strategy, guided by the following core principles:

- Fostering Thought Orientation: The design of hints targets the stimulation of critical and independent reasoning, rather than mere problem solution. Through guided exploration, we aim to lead students to discern the core of problems, thereby igniting curiosity and drive for resolution.

- Clarity of Steps: The hints systematically elucidate strategies to guide students through the problem-solving process, facilitating the understanding of underlying concepts, principles, or algorithms, and steering the application of appropriate mathematical skills or tools.

- Problem Adaptability: The design of hints is calibrated to the complexity of problems, providing detailed insight and explicit solution paths for more intricate issues, while centering on fundamental concepts and basic problem-solving techniques for simpler tasks.

Principle	FTO	COS	PA
Annotator A & B	0.83	0.87	0.82
Annotator A & C	0.86	0.89	0.84
Annotator B & C	0.82	0.85	0.80
Fleiss' Kappa	0.84	0.88	0.81

Table 1: Inter-Annotator Agreement Results. FTO: Fostering Thought Orientation; COS: Clarity of Steps; PA: Problem Adaptability.

4.1.1 Inter-Annotator Agreement Analysis

To assess the consistency and reliability of the TMATH dataset annotations, an Inter-Annotator Agreement (IAA) analysis was conducted. Three annotators with expertise in mathematics education evaluated 100 pre-labeled problems annotated according to core principles. After a two-hour training session, annotators rated each hint's adherence to three core principles: Fostering Thought Orientation, Clarity of Steps, and Problem Adaptability. Ratings were on a 5-point scale, where 1 indicated poor adherence and 5 indicated excellent adherence. The scores were analyzed to estimate inter-annotator agreement, ensuring consistency and reliability. IAA was assessed using Fleiss' Kappa (Fleiss, 1971) for overall consistency and Cohen's Kappa (Fleiss and Cohen, 1973) for pairwise consistency between annotators on the three principles.

Analysis results, as shown in Table 1, reveal relatively strong consistency for Fostering Thought Orientation across annotator pairs (A & B, A & C,

B & C), with Cohen's Kappa values ranging from 0.82 to 0.86. This suggests that the annotators had a common understanding of how hints should promote independent reasoning. The slight difference between Annotator B & C (0.82) may reflect subtle differences in interpreting the application of this principle to specific hints. The Fleiss' Kappa values for Fostering Thought Orientation (0.84), Clarity of Steps (0.88), and Problem Adaptability (0.81) indicate substantial inter-annotator agreement, with all values exceeding 0.8. This strong agreement demonstrates the annotators' high level of consistency and reliability in evaluating the alignment of hints with the principles. The minor variation in Problem Adaptability (0.81) may stem from nuanced differences in interpreting the appropriate level of detail for more complex problems. Overall, these results affirm the robustness and consistency of the annotation process, providing a solid foundation for the dataset construction.

4.2 Evaluation Framework

With the introduction of the TMATH dataset, we designed a hybrid evaluation framework to assess the ability of LLMs to generate hints for MWPs, incorporating both quantitative evaluation metrics and expert assessments.

4.2.1 Quantitative Evaluation

In the quantitative assessment, we utilized the ROUGE metrics, a prevalent evaluation method in Natural Language Generation (NLG) tasks like summary and translation, to gauge the similarity between the model-generated and human-created reference hints. Specifically, we applied ROUGE-N and ROUGE-L to evaluate the coherence at various levels, with higher scores reflecting greater consistency and indicating superior model performance: ROUGE-N measures the overlap of n-grams, reflecting the consistency between generated hints and reference hints at the level of n-word sequences. ROUGE-L measures the longest common subsequence, the consistency between the generated hints and reference hints at the subsequence level. Additionally, we assessed the answer accuracy (ACC) for each problem to ascertain the efficacy of the generated hints in aiding correct responses. A balanced sample of 100 questions was extracted for each discipline, encompassing a uniform distribution of difficulty levels.

4.2.2 Expert Evaluation

Quantitative evaluation metrics provide a convenient way to generate tasks, but their relevance to human evaluation in open-ended generation tasks is relatively low. To mitigate this limitation, we engaged experts in education and mathematics to manually evaluate hints. Each expert evaluated 70 questions, with an average of 10 questions per domain, selected from varying disciplines and levels of difficulty. The evaluations were based on three criteria: (1) Correctness, assessing the alignment with problem-solving strategies; (2) Clarity, whether the hint is easy to understand; and (3) Guidance, determining the hint's ability to lead students toward solutions without directly providing answers. These aspects were analyzed using a five-point rating system, with the findings detailed in Table 2. The consistency of the evaluations was verified through an Inter-Rater Reliability Analysis, detailed in Appendix A.

5 Experimental Results and Analysis

We evaluated five LLMs: GPT-3.5-turbo, GPT-4o, ERNIE Bot [3], SparkDesk [4], and Qwen2 [5] (Yang et al., 2024a). We selected these models for comparative experiments as they represent the most widely used LLMs, each with a parameter size exceeding one hundred billion. We adopted the Zero-shot prompting (Alayrac et al., 2022) test method.

5.1 Quantitative Evaluation

Table 3 presents the performance of each model on key metrics such as R-1, R-2, R-L, R-AVG, and ACC. In terms of overall performance, we found the GPT-4o model to excel. It demonstrated remarkable performance in word-level matches, long sequence matches, and answer accuracy. Moreover, it achieved the highest overall average match, highlighting its superior capabilities in handling various mathematical problems. However, despite GPT-4o outperforming other models on most evaluation metrics, it lags behind the GPT-3.5-turbo model in R-2. This suggests that while GPT-4o has an advantage in generating text with better overall coherence, GPT-3.5-turbo is adept at capturing and generating certain specific phrases or word groups. Furthermore, we noticed that while ERNIE Bot, SparkDesk, and Qwen2 lag behind GPT-4o in

[3]https://cloud.baidu.com/product/wenxinworkshop
[4]https: //xinghuo.xfyun.cn/
[5]https://tongyi.aliyun.com/

Level	Correctness
1	The hint does not accurately reflect the steps or strategies to solve the problem and the deviation is significant
2	The hint has some correlation with the steps or strategies to solve the problem, but there are obvious inaccuracies
3	The hint generally reflects the steps or strategies to solve the problem, but there is room for improvement
4	The hint accurately reflects the steps or strategies to solve the problem, with only a few inaccuracies
5	The hint very accurately reflects the steps or strategies to solve the problem, with no inaccuracies at all
Level	**Clarity**
1	The hint is difficult to understand, too complex or semantically vague
2	The hint is somewhat difficult to understand and may require additional explanation to comprehend
3	The clarity of the hint is good, most students can understand it
4	The hint is clear and easy to understand, with only very few instances that may require additional explanation
5	The hint is very clear and all students can intuitively understand it
Level	**Guidance**
1	The hint directly provides the answer, without guiding the students to think
2	The hint guides the students to think to a certain extent, but the guidance is not obvious
3	The hint appropriately guides the students to think, but may still require further guidance
4	The hint effectively guides the students to think, with only a few instances that may require further guidance
5	The hint very effectively guides the students to think, completely without the need for additional guidance

Table 2: Expert Evaluation Scale.

Model	R-1	R-2	R-L	R-AVG	ACC
GPT-3.5-turbo	43.19	**27.24**	31.19	33.87	22.45
GPT-4o	**49.37**	21.13	**38.54**	**36.34**	**38.15**
ERNIE Bot	35.48	15.67	28.72	26.62	16.77
SparkDesk	26.17	9.73	20.59	18.83	20.33
Qwen2	31.37	14.24	23.54	23.05	17.16

Table 3: Overall Performance of Models.

overall performance, their performance on R-2 and ACC metrics is noteworthy. This implies that these models may exhibit superior performance under certain conditions or when dealing with specific types of problems, such as achieving higher accuracy and precision when dealing with simpler or discipline-specific problems.

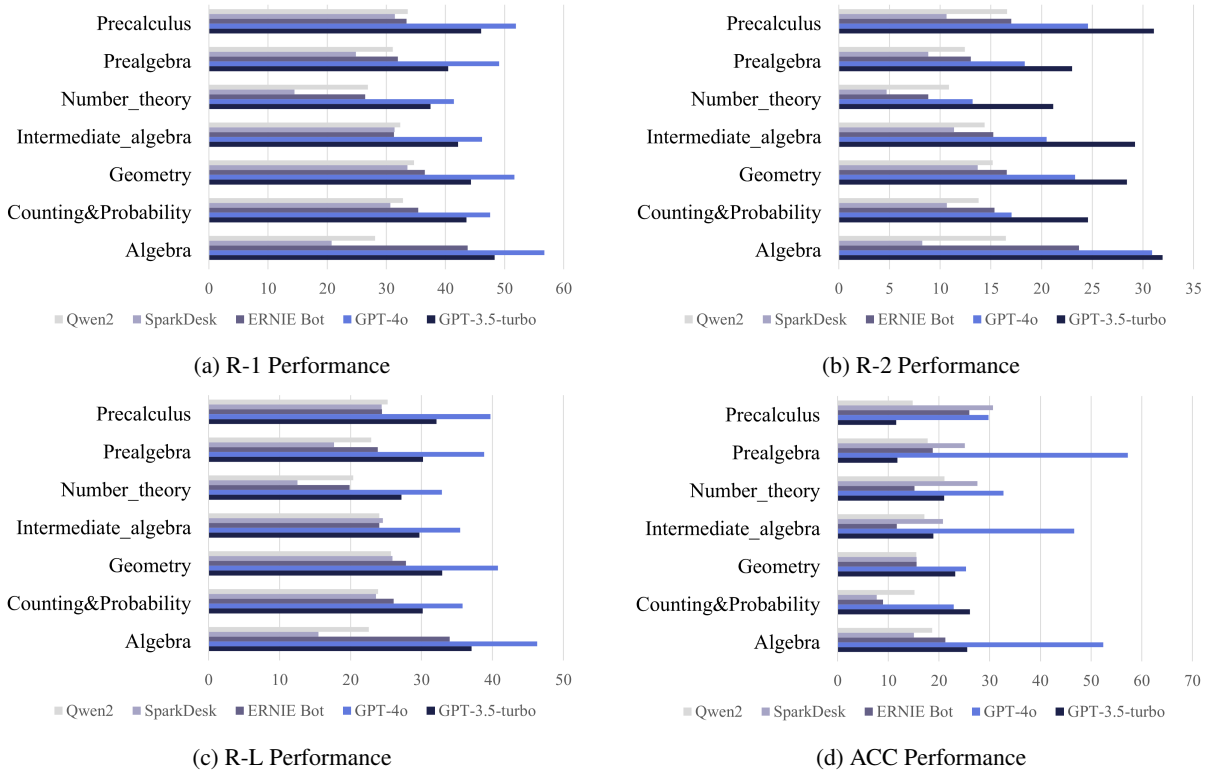
To further investigate the performance variation of LLMs across different subjects, we carried out an in-depth analysis of each model's performance in individual subjects. In Fig 3, the ROUGE-1 metric reveals GPT-4o's lead across all subjects, with ERNIE Bot, SparkDesk, and Qwen2 demonstrating stability in particular areas. The ROUGE-2 metric amplifies this trend, with SparkDesk and Qwen2 outperforming GPT-4o in Geometry and Prealgebra, suggesting their unique advantages for handling high logical complexity and precise symbolic tasks. Furthermore, the ACC metric shows SparkDesk surpassing GPT-4o in Geometry, Prealgebra, and Intermediate Algebra, likely reflecting its expertise in tackling problems with rigid structures and stringent logic.

5.2 Expert Evaluation

We invited two doctoral students with extensive backgrounds in mathematics and education to deeply assess the quality of the tutorial hints of each LLM. Table 4 presents the expert evaluation of various LLMs across three key dimensions: Correctness, Clarity, and Guidance. Overall, GPT-4o outperformed other models with an average score of 3.8, showing the highest scores in all three dimensions. GPT-3.5-turbo follows closely behind with an average score of 3.5, while ERNIE Bot, SparkDesk, and Qwen2 displayed lower performance across all metrics. Among the models, SparkDesk had the lowest average score of 2.9, indicating that it struggled to generate high-quality educational hints compared to the other models.

In terms of Correctness, which evaluates the accuracy of the hints generated by the models, GPT-4o achieved the highest score of 3.9, indicating that its hints were the most accurate and aligned with the correct problem-solving steps. GPT-3.5-turbo also performed reasonably well with a score of 3.6. In contrast, SparkDesk had the lowest score of 2.9, suggesting that its generated hints were less reliable and often deviated from the correct solution paths. This trend highlights GPT-4o's superior ability to understand complex problems and provide correct guidance, while other models like SparkDesk may require further improvement in this area.

For Clarity, which measures how understandable and clear the hints are, GPT-4o once again leads with a score of 3.9, demonstrating its ability to provide easy-to-understand explanations that stu-



(a) R-1 Performance

(b) R-2 Performance

(c) R-L Performance

(d) ACC Performance

Figure 3: Comparison of Performance Metrics Across Different Subjects Among Models.

dents can follow. GPT-3.5-turbo, with a score of 3.4, showed decent clarity but slightly lagged behind GPT-4o. Models like ERNIE Bot and Qwen2 performed moderately, while SparkDesk scored the lowest at 2.8. The lower clarity scores for SparkDesk and Qwen2 suggest that these models may generate more complex or vague hints, which could confuse students instead of aiding their understanding.

The Guidance dimension assesses how well the hints guide students through the problem-solving process without giving away the answers directly. GPT-4o excelled in this area as well, scoring 3.6, indicating that it effectively balanced providing guidance while promoting independent problem-solving. GPT-3.5-turbo also performed well with a score of 3.5. However, Qwen2 scored relatively lower at 2.9, implying that its hints might either be too direct or not sufficiently guiding students toward finding solutions on their own. SparkDesk, with a score of 3.0, showed a slight improvement in Guidance compared to its performance in other dimensions, but still lags behind the top models.

Overall, GPT-4o excels in generating educational hints, but all models still face challenges in handling complex problems, especially in providing deeper guidance.

Model	Correctness	Clarity	Guidance	AVG
GPT-3.5-turbo	3.6	3.4	3.5	3.5
GPT-4o	**3.9**	**3.9**	**3.6**	**3.8**
ERNIE Bot	3.2	3.1	3.3	3.2
SparkDesk	2.9	2.8	3.0	2.9
Qwen2	3.1	3.0	2.9	3.0

Table 4: Expert Evaluation of Model Performance.

6 Fine-Tuning Experiments and Analysis

To validate the quality of the TMATH dataset, we conducted a series of fine-tuning experiments using LLMs with 7B parameters, including Qwen2-7B (Yang et al., 2024a), GLM-7B (Du et al., 2022), Vicuna-7B (Chiang et al., 2023), and Gemma-7B (Team et al., 2023). These experiments were specifically designed to evaluate whether fine-tuning based on the TMATH dataset could enhance the LLMs' ability to not only generate effective hints for MWPs but also improve their performance in solving MWPs problems.

6.1 Performance of generating hints

To verify the impact of the TMATH dataset on enhancing the ability of models to generate effective educational hints, we conducted expert evaluations on four 7B models before and after fine-tuning. The

Model	Baseline [Fine-tuned]		
	Correctness	Clarity	Guidance
Qwen2-7B	2.5[+0.2]	2.3[+0.3]	2.3[+0.2]
GLM-7B	2.4[+0.1]	2.2[+0.4]	2.1[+0.3]
Vicuna-7B	2.6[+0.1]	2.4[+0.2]	2.3[+0.4]
Gemma-7B	2.5[+0.2]	2.3[+0.2]	2.2[+0.3]

Table 5: Expert Evaluation of 7B Models Pre- and Post-Fine-Tuning.

evaluation metrics included Correctness, Clarity, and Guidance, with the results presented in Table 5. The fine-tuned models showed improvements across all metrics. Fine-tuning led to significant improvements, especially for Qwen2-7B, which saw an increase of +0.2 in Correctness, +0.3 in Clarity, and +0.2 in Guidance, indicating that its generated hints became more accurate and clearer after fine-tuning. GLM-7B, Vicuna-7B, and Gemma-7B showed similar trends, with improvements ranging from +0.1 to +0.4 across the three metrics. These results suggest that fine-tuning with the TMATH dataset effectively enhances the models' ability to generate hints that are not only more accurate but also clearer and better aligned with educational goals. Although the improvements were modest in some cases, the overall trend indicates that the models perform more consistently after fine-tuning, particularly in terms of clarity and guidance.

6.2 Performance of solving MWPs

In this experiment, we fine-tuned several 7B models. Although the TMATH dataset is primarily designed for hint generation, effective hints guide the model to make correct choices at key steps, helping it identify and overcome difficulties, reducing reasoning errors, and thereby improving the model's understanding and problem-solving ability. By comparing the accuracy of problem-solving before and after fine-tuning, we can indirectly evaluate the quality of hint generation. Table 6 summarizes the performance of each model across different mathematical domains. The results show that all models exhibited significant improvements in problem-solving accuracy after fine-tuning, particularly in foundational areas such as Prealgebra and Algebra. Although the improvements in more complex areas like Number Theory and Precalculus were smaller, there was still noticeable progress. The performance gains in intermediate domains, such as Intermediate Algebra and Counting & Probability, were particularly notable, especially for GLM-7B and Vicuna-7B. These findings further validate the high quality and effectiveness of the TMATH dataset in generating useful hints.

7 Discussion

Drawing from our findings, LLMs' ability to generate hints for solving MWPs requires further refinement. We highlight two principal optimization avenues, targeting the intrinsic challenges identified in our study, to bolster LLMs' efficacy in educational contexts. Relevant work, such as the research on the limitations of LLMs in complex reasoning abilities (Choudhary and Reddy, 2023), and on how to enhance their generation accuracy through self-validation mechanisms (Weng et al., 2022), underpins our optimization pursuits, providing a theoretical and empirical foundation.

Optimize the learning mechanism of LLMs with chain of thought (CoT) intermediate supervision. An analysis of LLMs in generating hints for MWPs solving unveiled substantial inconsistencies in the granularity and depth of intermediate solution steps. These differences stem from three main capabilities: (1) the ability to align numbers in intermediate steps; (2) the ability to discern knowledge points in intermediate steps; (3) the ability to transition between steps appropriately. This discovery emphasizes an essential aspect: LLMs' hint generation can substantially profit from the CoT intermediate supervision stage. Consequently, fine-tuning the learning mechanism of LLMs with CoT intermediate supervision may elevate the model's acuity in handling quantitative problem facets, precisely identifying and employing mathematical concepts, and fortifying step-transition skills, thereby facilitating more fluid complex reasoning and detailed hint creation.

Enhance LLM's interpretable self-verification capabilities. In the context of generating hints for complex reasoning problems, the lack of robustness in LLMs can lead to the alteration of overall hint meaning, and consequently incorrect answer generation. The absence of an effective error-correction mechanism within LLMs results in their inability to self-correct erroneous hints, giving rise to illusions. Enhancing LLM's interpretable self-verification capabilities is thus imperative. This enhancement would not only enable the model to self-check and recalibrate its generated hints, managing uncertainty more effectively, but also allow for strategic adjustment, concentrating on error-prone steps

Model	Baseline [Fine-tuned]							
	Prealgebra	**Algebra**	**Number Theory**	**Counting & Probability**	**Geometry**	**Intermediate Algebra**	**Precalculus**	**Average(%)**
Qwen2-7B	6.5 [+0.9]	5.2 [+0.7]	5.3 [+0.7]	3.9 [+0.8]	6.8 [+0.7]	4.6 [+0.7]	5.7 [+0.7]	5.4 [+0.9]
GLM-7B	5.0 [+1.1]	4.8 [+0.9]	4.9 [+0.6]	2.5 [+1.1]	5.5 [+0.9]	6.1 [+0.8]	6.9 [+0.7]	5.1 [+0.8]
Vicuna-7B	6.2 [+1.0]	6.0 [+1.1]	5.2 [+0.8]	3.6 [+0.8]	6.6 [+0.8]	5.8 [+0.7]	6.8 [+0.7]	5.9 [+0.8]
Gemma-7B	3.8 [+1.4]	3.1 [+0.9]	4.4 [+0.9]	4.2 [+0.8]	3.8 [+1.7]	2.9 [+1.4]	4.2 [+1.7]	3.8 [+1.3]

Table 6: Accuracies across Subjects for 7B Models Pre- and Post-Fine-Tuning.

and investigating more effective problem-solving pathways.

8 Conclusion

This study introduces the TMATH dataset, a novel resource designed to evaluate and enhance the ability of LLMs to generate hints for MWPs. Through fine-tuning experiments on several 7B-parameter models, we demonstrated that the TMATH dataset significantly improves the models' hint generation capabilities. Expert evaluations revealed that fine-tuning with TMATH led to notable improvements in the accuracy, clarity, and guidance of the generated hints. However, our in-depth evaluation also uncovered several limitations in LLMs' ability to handle problems requiring deep understanding and critical thinking. These models still face challenges in context comprehension and explainable self-verification, particularly when solving complex problems.

9 Limitations

We must admit that despite the valuable insights our findings provide about the application of LLMs in the education field, our study still has certain limitations. Firstly, although TMATH is the first dataset covering a wide range of math problems and high-quality human-generated hints, its coverage of certain specific domains of problems might be limited; Secondly, our study employed a multi-angle evaluation approach, including quantitative indicators and expert evaluation, but these methods each have inherent limitations and biases. Quantitative metrics might overlook the genuine educational value, and expert assessments may be limited by individual perspectives and experiences. These will be important directions for our future research.

Acknowledgements

This work was supported by the Opening Foundation of the State Key Laboratory of Cognitive Intelligence (iED2023-008) and the National Natural Science Foundation of China (NSFC) under Grant No. 62477013. The authors would also like to acknowledge the support from the NSFC for the project "Multidimensional dynamic assessment and individualized intervention of scientific inquiry skill driven by complex task" under Grant No. 6906035.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 225–237.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. 2022. Flamingo: a visual language model for few-shot learning. *Advances in Neural Information Processing Systems*, 35:23716–23736.

Pablo Arnau-González, Ana Serrano-Mamolar, Stamos Katsigiannis, and Miguel Arevalillo-Herráez. 2023. Towards automatic tutoring of custom student-stated math word problems. In *International Conference on Artificial Intelligence in Education*, pages 639–644. Springer.

Razvan Azamfirei, Sapna R Kudchadkar, and James Fackler. 2023. Large language models and the perils of their hallucinations. *Critical Care*, 27(1):120.

Sid Black, Stella Biderman, Eric Hallahan, Quentin Anthony, Leo Gao, Laurence Golding, Horace He, Connor Leahy, Kyle McDonell, Jason Phang, et al. 2022. Gpt-neox-20b: An open-source autoregressive language model. *arXiv preprint arXiv:2204.06745*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind

Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Nurendra Choudhary and Chandan K Reddy. 2023. Complex logical reasoning over knowledge graphs using large language models. *arXiv preprint arXiv:2305.01157*.

Clayton Cohn, Nicole Hutchins, Tuan Le, and Gautam Biswas. 2024. A chain-of-thought prompting approach with llms for evaluating students' formative assessment responses in science. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23182–23190.

Zhengxiao Du, Yujie Qian, Xiao Liu, Ming Ding, Jiezhong Qiu, Zhilin Yang, and Jie Tang. 2022. GLM: General language model pretraining with autoregressive blank infilling. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 320–335, Dublin, Ireland. Association for Computational Linguistics.

Joseph L Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological bulletin*, 76(5):378.

Joseph L Fleiss and Jacob Cohen. 1973. The equivalence of weighted kappa and the intraclass correlation coefficient as measures of reliability. *Educational and psychological measurement*, 33(3):613–619.

Sai Satish Gattupalli, Will Lee, Danielle Allessio, Danielle Crabtree, Ivon Arroyo, Beverly P Woolf, and Beverly Woolf. 2023. Exploring pre-service teachers' perceptions of large language models-generated hints in online mathematics learning. In *LLM@ AIED*, pages 151–162.

Virginia Grande, Natalie Kiesler, and María Andreína Francisco R. 2024. Student perspectives on using a large language model (llm) for an assignment on professional ethics. In *Proceedings of the 2024 on Innovation and Technology in Computer Science Education V. 1*, pages 478–484.

Muhammad Usman Hadi, Qasem Al Tashi, Abbas Shah, Rizwan Qureshi, Amgad Muneer, Muhammad Irfan, Anas Zafar, Muhammad Bilal Shaikh, Naveed Akhtar, Jia Wu, et al. 2024. Large language models: a comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Preprints*.

ASM Hasan, Md Alvee Ehsan, Kefaya Benta Shahnoor, and Syeda Sumaiya Tasneem. 2024. *Automatic question & answer generation using generative Large Language Model (LLM)*. Ph.D. thesis, Brac University.

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. *NeurIPS*.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models. *arXiv preprint arXiv:2303.05398*.

Breanna Jury, Angela Lorusso, Juho Leinonen, Paul Denny, and Andrew Luxton-Reilly. 2024. Evaluating llm-generated worked examples in an introductory programming course. In *Proceedings of the 26th Australasian Computing Education Conference*, pages 77–86.

Juho Leinonen, Paul Denny, Stephen MacNeil, Sami Sarsa, Seth Bernstein, Joanne Kim, Andrew Tran, and Arto Hellas. 2023. Comparing code explanations created by students and large language models. *Preprint*, arXiv:2304.03938.

Zichao Lin, Shuyan Guan, Wending Zhang, Huiyan Zhang, Yugang Li, and Huaping Zhang. 2024. Towards trustworthy llms: a review on debiasing and dehallucinating in large language models. *Artificial Intelligence Review*, 57(9):1–50.

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

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, and Sayak Paul. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Yuhong Mo, Hao Qin, Yushan Dong, Ziyi Zhu, and Zhenglin Li. 2024. Large language model (llm) ai text generation detection based on transformer deep learning algorithm. *arXiv preprint arXiv:2405.06652*.

Daye Nam, Andrew Macvean, Vincent Hellendoorn, Bogdan Vasilescu, and Brad Myers. 2024. Using an llm to help with code understanding. In *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, pages 1–13.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. 2018. Improving language understanding by generative pre-training.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2022. Bloom: A 176b-parameter open-access multilingual language model. *arXiv preprint arXiv:2211.05100*.

Kv Aditya Srivatsa and Ekaterina Kochmar. 2024. What makes math word problems challenging for LLMs? In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 1138–1148, Mexico City, Mexico. Association for Computational Linguistics.

John Stamper, Ruiwei Xiao, and Xinying Hou. 2024. Enhancing llm-based feedback: Insights from intelligent tutoring systems and the learning sciences. In *International Conference on Artificial Intelligence in Education*, pages 32–43. Springer.

Ruixiang Tang, Yu-Neng Chuang, and Xia Hu. 2023. The science of detecting llm-generated texts. *arXiv preprint arXiv:2303.07205*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Yixuan Weng, Minjun Zhu, Shizhu He, Kang Liu, and Jun Zhao. 2022. Large language models are reasoners with self-verification. *arXiv preprint arXiv:2212.09561*.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, et al. 2024a. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Xiaocheng Yang, Bingsen Chen, and Yik-Cheung Tam. 2024b. Arithmetic reasoning with LLM: Prolog generation & permutation. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 699–710, Mexico City, Mexico. Association for Computational Linguistics.

Hanyu Zhang, Xiting Wang, Xiang Ao, and Qing He. 2024. Distillation with explanations from large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pages 5018–5028, Torino, Italia. ELRA and ICCL.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022. Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Yuchen Zhuang, Yue Yu, Kuan Wang, Haotian Sun, and Chao Zhang. 2024. Toolqa: A dataset for llm question answering with external tools. *Advances in Neural Information Processing Systems*, 36.

A Inter-Rater Reliability Analysis

Evaluation Metric	Correctness	Clarity	Guidance
Cohen's Kappa	0.85	0.88	0.82

Table 7: Cohen's Kappa Values for Inter-Rater Reliability Across Three Evaluation Metrics.

To ensure the consistency and reliability of expert evaluations, this study conducted an Inter-Rater Reliability (IRR) analysis. Two experts with specialized backgrounds in mathematics and education independently assessed 100 selected problems based on three core evaluation criteria: Correctness, Clarity, and Guidance. The evaluations were conducted using a standardized 5-point Likert scale with detailed rubrics, aiming to reduce subjectivity and ensure a systematic evaluation process.

Cohen's Kappa coefficient, a widely used statistical measure for inter-rater reliability, was employed to quantify the level of agreement between the two experts while accounting for chance agreement. The results, presented in Table 7, indicate substantial agreement across all three criteria. Clarity exhibited the highest level of consistency (0.88), reflecting strong alignment between evaluators in assessing the comprehensibility of the generated hints. Correctness also demonstrated excellent agreement (0.85), confirming consistency in evaluating the alignment of hints with problem-solving strategies. Although Guidance showed a slightly lower Kappa value (0.82), it remains within the range of substantial agreement.

B Additional Explanations and Examples

B.1 Clarification of Terms

To provide further clarity on the terms used in Fig.1, we offer detailed explanations below:

- **Knowledge Point**: Refers to the core mathematical concepts or principles required to solve a problem. Examples include algebraic operations, geometric theorems, or probability principles. Identifying these points is crucial for designing hints that target students' understanding gaps.

- **Step Complexity**: Describes the level of difficulty associated with each problem-solving

step. Factors influencing complexity include the number of operations required, the abstraction level, and the logical reasoning involved.

- **Potential Cognitive Obstacles**: Denotes the possible challenges students may face during problem-solving, such as misunderstanding formulas, skipping steps, or struggling with abstract reasoning. These obstacles guide the design of hints aimed at overcoming specific learning barriers.

B.2 Justification for the Term "Socratic-Style Hints"

The term "Socratic-style hints" is used to describe the pedagogical approach adopted in this study, inspired by the Socratic method. This approach involves guiding students through a sequence of thought-provoking questions or prompts that encourage them to arrive at solutions independently. For example, instead of directly providing the next step in solving an equation, the hint may prompt students to consider the role of a variable or the properties of an operation. This method aligns with Socratic teaching principles, which prioritize critical thinking and active learning over passive reception of answers.

B.3 Availability of Example Hints

Fig.1 demonstrates the framework for constructing hints, which is based on specific examples from the TMATH dataset. These examples were systematically designed to align with the methodology outlined in our study. For readers interested in additional examples, we have provided a comprehensive set of hints on our GitHub repository. The repository can be accessed at: https://github.com/qi-github-ui/TMATH. This resource includes a variety of problem types and corresponding hints, offering deeper insights into the hint-generation process.