UDC: 004.89 https://doi.org/10.15407/jai2024.04.096

A. Nikolaiev[1]*, O. Derevianchenko[2] * Work done while at Cambridge
[1,2]Taras Shevchenko National University of Kyiv, Ukraine
4d, Glushkova str., Kyiv, 03680
[1]nikolaev@knu.ua
[2]olexandrder@knu.ua
[1]https://orcid.org/0000-0003-3359-4936
[2]https://orcid.org/0009-0006-9308-8567

COMPARISON OF PROBLEM-SOLVING PERFORMANCE ACROSS MATHEMATICAL DOMAINS WITH LARGE LANGUAGE MODELS

Abstract. This study investigates problem-solving performance across four mathematical domains, using statistical techniques to analyse domain-specific differences. By leveraging the NuminaMath-TIR dataset, we categorized problems into algebra, geometry, number theory, and combinatorics, selecting 8,000 problems for the analysis. Models including GPT-4o-mini, Mathstral-7B, Qwen2.5-Math-7B, and Llama-3.1-8B-Instruct were applied to assess answer correctness. Significant differences in solution accuracy were identified, with algebra showing the highest correctness rates and combinatorics the lowest. The results highlight the impact of domain on model performance and suggest the potential for tool-integrated reasoning (TIR) techniques to enhance consistency across domains. Future work can explore targeted model training improvements, aiming to optimize educational technologies and adaptive learning systems.

Keywords: Artificial Intelligence; Mathematical Problems; Natural Language Processing; Large Language Models; Automated Reasoning.

Introduction

Mathematics education often reveals varying levels of student success across different domains. Understanding these differences can inform educational practices and resource allocation. Identifying problem areas within mathematical domains allows educators to tailor strategies for improvement, enhancing overall learning outcomes. The same is true for computational models, including large language models (LLMs).

This study aims to assess problem-solving efficacy across multiple mathematical domains and to determine if significant differences exist in correctness rates. In this research, we address the following two questions:

1. Are there significant differences in the performances of LLMs across distinct mathematical domains?

2. Which domains exhibit the highest and lowest problem-solving success?

1 Analysis of recent research and publications

Automation in addressing mathematically complex problems has been a topic of interest for over 50 years. In 1974, Victor Glushkov pioneered efforts in this domain by exploring the use of formal languages for documenting mathematical texts and automating the search for theorem proofs [3].

Recent advancements in artificial intelligence (AI), particularly in natural language processing (NLP), have introduced promising tools to facilitate mathematical problem-solving processes. However, the diversity in problem domains and overlaps highlights the importance of investigating best practices and models for the future development of computational models.

LLMs such as GPT-4 (OpenAI, 2023 [8]) and PaLM-2 (Anil et al., 2023 [1]), have shown significant progress in a broad spectrum of language-related tasks, particularly in addressing the longstanding challenge of mathematical reasoning. These models represent a leap forward in processing and understanding natural language; however, they frequently encounter challenges when tasked with advanced mathematical reasoning.

In contrast, open-source models like LLaMA-2 (Touvron et al., 2023 [10]) and Falcon (Penedo et al., 2023 [9]) continue to face difficulties in complex mathematical tasks. The mathematical performance of language models has been improved in existing work either through step-by-step natural language reasoning [12], or by synthesizing and executing programs to arrive at correct answers [2]. These two

methodologies leverage distinct, yet complementary strengths.

With models like AlphaGeometry and AlphaProof, which include neuro-symbolic systems for solution search and an LLM to formalise processed text, there has been shown significant progress in high-level competitions like the International Mathematical Olympiad [11]. However, these models and data used for training are not available, though considering the approaches used to solve problems of such level, it's visible that a diversity of problems need to be addressed with different techniques.

Natural language reasoning using these models is well-suited for tasks involving semantic analysis, planning, and abstract reasoning, which often require a commonsense understanding. However, this approach often falls short when precise computations, symbolic manipulation, or algorithmic processing are required. Conversely, program synthesis and execution provide the necessary robustness for such tasks by accurately performing operations and delegating complex calculations to tools designed to handle specific problem types, such as equation solvers.

This study applies a systematic approach comparison between results in the domain-related mathematical problems utilising several language models. It's claimed that tool-integrated reasoning (TIR) is pivotal in enhancing computational problem-solving accuracy [4], yet collecting and annotating data for this purpose can be resource intensive.

Authors of the paper work with NuminaMathTIR a dataset of 70,000 problems from the NuminaMath-CoT dataset by focusing on those with numerical outputs, primarily integers. This was achieved by utilising a pipeline powered by GPT-4, to generate ToRA-like reasoning paths, which were used to execute code and derive complete solutions. Solutions were iteratively filtered to match reference answers to maintain accuracy and consistency, a process repeated three times. This method allowed the production of high-quality TIR data with reduced annotation costs and time [6].

2 Research methods

In the first stage, we selected and filtered 8,000 mathematical problems with numerical outputs from the NuminaMath-TIR dataset and categorized them into four distinct domains: algebra, geometry, number theory, and combinatorics, with 2,000 problems per domain. This categorization was achieved using keyword matching to ensure that each problem uniquely belonged to one of the specified domains, resulting in a dataset with a balanced representation across the domains.

Next, we use open-source and propriety models to generate solutions and assess their reasoning abilities across domains. To verify numerical answer validity, problem statements were presented to the model to generate solutions, which were later compared to the baseline answers.

To assess how well LLMs solve problems per domain, we integrated into the experiments four different models: the open-source models Mathstral7B, Qwen2.5-Math-7B, Llama-3.1-8B-Instruct – ran locally, and propriety model GPT-4o-mini – accessed via an API. This approach allowed us to evaluate and compare the performance and capabilities of both open-source and proprietary models in handling complex problem sets across various domains. Detailed specifications of the models tested can be found in the *Section 3.3 Detailed specifications of LLMs tested.*

For the input, we have a Parquet file, which comprises 4 columns with details of each problem, including problem topic, answer, problem, and solution, which represent the problem's domain, baseline answer, statement, and solution.

For each problem per model, we perform a series of queries during which we: (i) introduce a problem (without any additional prompt added) to get a solution from a model; (ii) extract the final answer with the help of a regular expression; (iii) and compare it with the baseline answer, indicating whether answers are correct ("1") or incorrect ("0").

We discovered from the dataset used for the experiments that many answers returned as mathematical LaTeX formulas, which was also true for the LLMs during the solutions generation process, e.g.

$\\boxed{\\frac{12 - 5\\sqrt3}{36}}$

For that reason, we applied regular expressions to extract the numerical answers from the generated texts. For the cases when there was no *boxed*{} found, we extracted the last numerical answer present in the solution as a final response provided by the model. Also, in some cases, the problems were presented in a multi-choice option format.

Results are saved in two additional columns: *messages* and *answer_model*, which correspond to dialogue with the model and whether the answer provided by the model was correct. Several problem examples are listed in the *Section 3.4 Problems examples.*

Finally, for the analysis part of the results obtained, we calculated average correctness rates, expressed as percentages to convey domain difficulty levels, and performed two statistical tests to assess whether the differences are statistically significant across domains:

(i) Chi-Square test of independence to assess the distribution of correct and incorrect answers per model and collectively, with the latter calculated as the median of values of the combined model outputs, which is used to determine performance by domain;

(ii) Permutation test, performed by shuffling domain labels and comparing the mean differences between pairs of topics across all models per problem, was conducted to validate the differences in domain relevance.

3 Experiment setup and data processing

We used the GPU unit Quadro RTX 8000 to experiment with open-source models with 48 GB of RAM. Although memory was not completely occupied during the experimentation process, it is visible that GPU-Util was 100% busy (Figure 1), indicating the bottleneck of the data processing with several instances of LLM running locally on the same GPU-unit. Interference with the GPT-4o-mini model has been done via OpenAI API calls.

To speed up the processing processes we ran 4 scripts to process each domain simultaneously, details on how much time was used for processing data are included in Table 1.

Table 1. Durations for processing data

Model	# of processes	Queries via	Time to process
GPT-4o-mini	4	API	9.5-11.5 hours
Mathstral-7B	4	Local	18.5-21.5 hours
Qwen2.5-Math-7B	4	Local	40-60 hours
Llama3.1-8B-Instruct	4	Local	53-54 hours

Code and data used for the analysis are available in the repository: https://github.com/andynik/math-domains-comp-24.

3.1 Open-source models

For the open-source models for the experiment, we selected lightweight open-source models: Mathstral-7B by Mistral company, Qwen-2.5-math7B by Alibaba Cloud company, and Llama3.1-8B-Instruct by Meta.

Mathstral-7B is a model specialising in mathematical and scientific tasks, based on Mistral-7B [5].

Qwen2.5-Math series is expanded to support using both CoT and Tool-integrated Reasoning (TIR) to solve math problems in both Chinese and English. The Qwen2.5-Math series models have achieved significant performance improvements compared to the previous generation of the models on the Chinese and English mathematics benchmarks with CoT [13].

The Llama 3.1 instruction-tuned text-only models are optimized for multilingual dialogue use cases [7].

3.2 GPT- 4o-mini

As closed-access, we utilised one of the latest OpenAI large language models, GPT-4o-mini. OpenAI would not disclose exactly how large GPT-4o-mini is, but it is mentioned on the website to be in the same tier as other small AI models, such as Claude Haiku and Gemini 1.5 Flash. The model is probably bigger than 7B as opposed to open source compared in this paper but is not specifically trained on the mathematical data only.

During the experiment around 6.7 million tokens were processed and the model generated 4.3 million output tokens.

3.3 Detailed specifications of LLMs tested

Table 2 lists models and their descriptions, which have been used during the experiments. The open-source models have been quantised using K-means quantisation with the help of **llama.cpp** library. The quantisation helps to speed up the inference, make data more private, and use less bandwidth.

In the context of **llama.cpp**, Q4_K_M refers to a specific type of quantization method. The naming convention is as follows:

- "Q" stands for quantisation.
- "4" indicates the number of bits used in the quantisation process.
- "K" refers to the use of k-means clustering in the quantisation.
- "M" represents the size of the model after quantisation (S = Small, M = Medium, L = Large).

Models with detailed descriptions of the open-source (quantised versions) and OpenAI are available at:

✓ GPT- 4o-mini: (accessible via API): https://platform.openai.com/docs/models/gpt-4o-mini

✓ Mathstral-7B: https://huggingface.co/QuantFactory/mathstral-7B-v0.1-GGUF

✓ Qwen2.5-Math-7B: https://huggingface.co/QuantFactory/Qwen2.5-Math-7B-GGUF

✓ Llama-3.1-8B-Instruct: https://huggingface.co/QuantFactory/Meta-Llama-3.1-8B-Instruct-GGUF

Table 2. Specifications of models tested

Model	Params	Is quant.	Q. method	Context length	Knowledge cutoff	Model creator
GPT-4o-mini-2024-07-18	n/a	✗	-	128k	Oct 2023	OpenAI
Mathstral-7B	7.25B	✓	Q4_K_M	32k	n/a	Mistral AI
Qwen2.5-Math-7B	7.62B	✓	Q4_K_M	128k	n/a	Alibaba Cloud
Llama-3.1-8BInstruct	8.03B	✓	Q4_K_M	128k	Dec 2023	Meta

3.4 Problems examples

Table 3 represents some examples of the problems used for the tests presented in the way they appear in the dataset. Some of the answers are expected to be returned in the form of a test ("Choose the correct answer from options: A, B, C, D"), some answers might be expected as a continuation of the question from the problem statement (e.g. "The answer is _"), some answers might use mathematical constants like π.

Table 3. Examples of problems categorized by domain with answers

Domain	Answer	Problem
Algebra	D	Let $f(x) = x^2 + bx + c$. If the equation $f(x) = x$ has no real roots, then the equation $f(f(x)) = x$: A. has 4 real roots B. has 2 real roots C. has 1 real root D. has no real roots.
Combinatorics	3136	In how many ways can two rooks be arranged on a chessboard such that one cannot capture the other? (A rook can capture another if it is on the same row or column of the chessboard).
Geometry	60.0	A triangle has side lengths of 8, 15 and 17 units. What is the area of the triangle, in square units?
Number Theory	37	Find the greatest common divisor (GCD) of 8251 and 6105.
Algebra	-2	Simplify $(3 - 2i) - (5 - 2i)$.
Combinatorics	0.016	Out of a randomly selected sample of 500 parts, 8 are found to be defective. Find the frequency of defective parts.
Geometry	$100 + 75\pi$	A square has sides of length 10, and a circle centred at one of its vertices has radius 10. What is the area of the union of the regions enclosed by the square and the circle? Express your answer in terms of π.
Number Theory	1	Find the remainder when 2^{100} is divided by 101.
Algebra	16	John has just turned 39. 3 years ago, he was twice as old as James will be in 6 years. If James' older brother is 4 years older than James, how old is James' older brother?
Combinatorics	No	Is it possible to divide a 12×12 checkerboard into 'L' shapes made from three adjacent cells such that each horizontal and each vertical row on the board intersects the same number of 'L' shapes? (A row intersects an 'L' shape if it contains at least one of its cells.)
Geometry	B. 45°	In a regular tetrahedron $ABCD$, M and N are the midpoints of edges AB and CD respectively. Determine the angle between segment MN and edge AD. A. 30° B. 45° C. 60° D. 90°
Number Theory	0	The sum of all integers whose absolute value is greater than 1 but less than 3.5 is ___ .

3.4.1 Wrong problems

Because our algorithm relies on filtering by keywords, some of the problems have been classified wrongly, mainly due to the specifics of the terminology used in problem statements (e.g. "Pascal's triangle" etc.) or due to the problem's set phrases usage (e.g. "sitting around a round table" with misleading indication to geometry domain), which are designed as a story. Some examples of such cases are shown in the Table 4.

4 Analysis of Results

This section presents a statistical analysis of problem-solving performance across different mathematical domains on the models tested. First, we calculated the average performances of model across domains. Next, we utilised chi-square and permutation statistical methods to evaluate whether the distribution of correct and incorrect answers differs significantly between domains and to ascertain the average correctness rates for each domain.

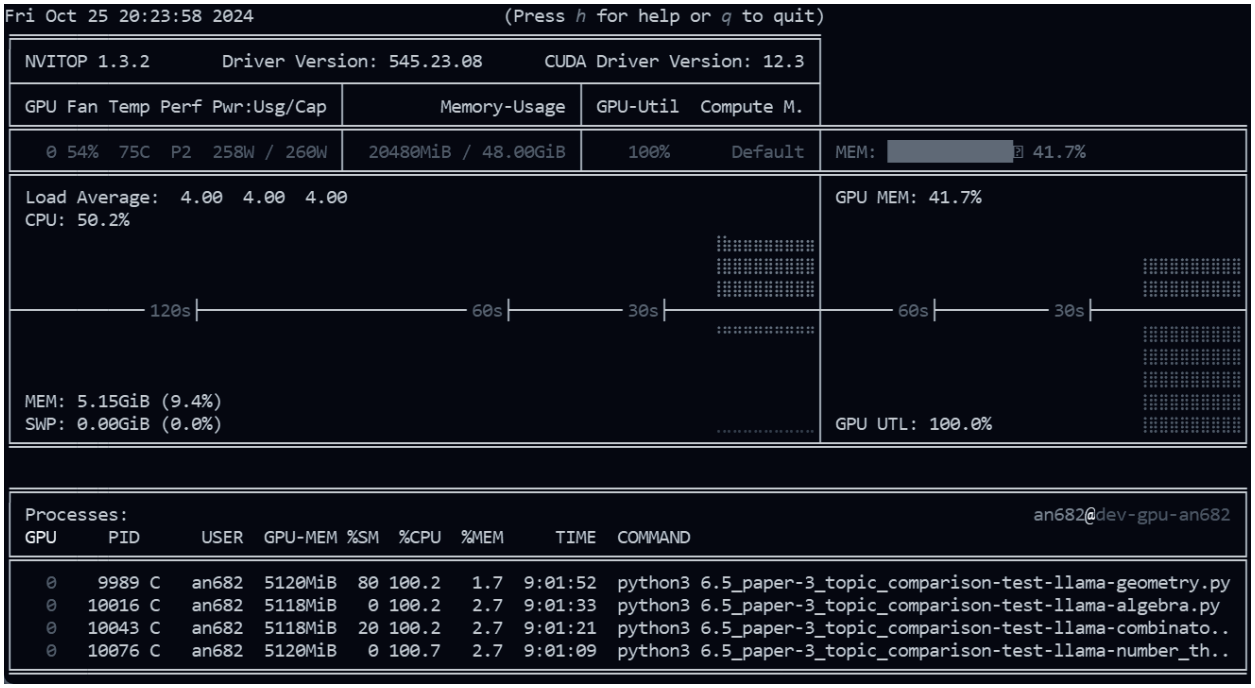

Fig. 1. Multiple processes running on the GPU unit utilising the CPU element to the 100%

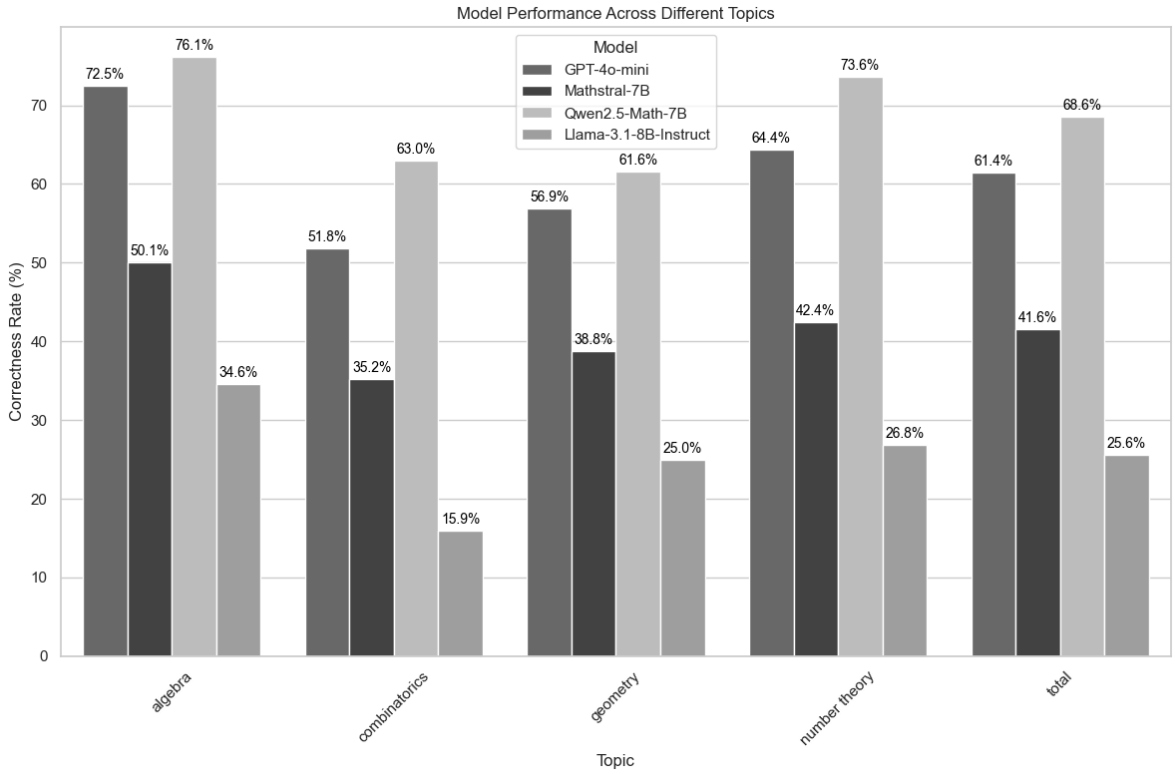

Fig. 2. Average solution rate between the domains

Table 4. Examples of wrongly identified problems with assigned and expected domains

Assigned domain	Expected domain	Problem
Geometry	Combinatorics	What is the 39th number in the row of Pascal's triangle that has 41 numbers?
Geometry	Combinatorics	99 gnomes are sitting around a round table. The hobbit Bilbo knows all the gnomes, but he cannot see how they are seated as his eyes are covered. Bilbo can name two any gnomes, and all the gnomes will chorus in response, telling him how many gnomes are sitting between these two gnomes (along the shortest arc). Can Bilbo find out at least one pair of adjacent sitting gnomes by asking no more than 50 questions?
Number Theory	Combinatorics	Wanda, Darren, Beatrice, and Chi are tutors in the school math lab. Their schedule is as follows: Darren works every third school day, Wanda works every fourth school day, Beatrice works every sixth school day, and Chi works every seventh school day. Today they are all working in the math lab. In how many school days from today will they next be together tutoring in the lab?

4.1 Models Performances across Domains

We computed the average correctness rate for each domain per model, representing the percentages of problems solved correctly. This measure provides insight into the overall effectiveness of the models with different mathematical domains.

From Figure 2, Qwen2.5-Math-7B emerged as the top performer with an average correctness rate of 68.6%, whereas Llama-3.1-8B-Instruct model showed the worst results of 25.6%.

Average. From Table 5 we observe that the highest averaged model performances across domains (AMPD) have been achieved on algebra topic with 58% on average, indicating that on average, models perform best in this domain. Number Theory follows closely with an APD of 52%, showing relatively high performance. Geometry and Combinatorics have lower APDs of 46% and 41%, respectively, suggesting these are more challenging domains for the models. The table also states that all the differences observed are statistically significant (details in section Permutation Test Analysis Results).

Algebra. Algebra consistently emerged as the strongest area across all models, with GPT-4o-mini achieving a correctness rate of 72.5% and Qwen2.5Math-7B reaching 76.1%. These figures indicate robust algebraic capabilities, as algebra tasks align well with the models' strengths.

Combinatorics. Combinatorial challenges were evident, as this domain consistently underperformed compared to others. Mathstral-7B struggled significantly with a correctness rate of only 35.2%, and GPT-4o-mini also found it challenging at 51.8%. The complexity of combinatorial reasoning remains a considerable hurdle.

Geometry. Performance varied in Geometry, with Mathstral-7B managing a correctness rate of 38.8%, indicating below-average proficiency. In comparison, GPT-4o-mini and Qwen2.5-Math-7B were more successful, achieving correctness rates of 56.9% and 61.6%, respectively, suggesting moderate success in this spatial domain for some models.

Number Theory. Number Theory showed mixed outcomes, with Mathstral-7B dipping to a correctness rate of 42.4%, indicating challenges. However, Qwen2.5-Math-7B demonstrated substantial competence in this domain, achieving a correctness rate of 73.6%, revealing varied proficiency and potential for targeted improvement.

Table 5. Absolute differences between Averaged Model Performances across Domains (AMPD). '*' represents that the domain difference was highly significant ($p < 0.001$)

Domain		Algebra	Combinatorics	Geometry	Number theory
	AMPD	0.58	0.41	0.46	0.52
Algebra	0.58	-	0.17*	0.13*	0.07*
Combinatorics	0.41		-	0.04*	0.10*
Geometry	0.46			-	0.06*
Number theory	0.52				-

4.2 Statistical significance

4.2.1 Chi-Square Test Analysis

Results

A Chi-Square test of independence was conducted to evaluate if the domain influences the distribution of correct and incorrect answers. The corresponding values are available in Table 6. Additionally, all investigated models have degrees of freedom $D_f = 3$, which is equal to the number of domains minus 1, and p_{value} way below the significance level of 0.05.

Table 6. Chi-Square test results for each model and a combined analysis of all models

Model	Chi-Square
GPT-4o-mini	205.64
Mathstral-7B	101.04
Qwen2.5-Math-7B	149.37
Llama-3.1-8B-Instruct	184.93
Combined analysis	520.16

From Table 6, we observe that the average chis-quare value $\chi^2 = 520.16$ underscores the overall influence of the domain across all models. The high average indicates that the domain significantly affects performance, consolidating that models exhibit varied performance outcomes across different domains.

We observe that the highest chi-square value was achieved by the model GPT-4o-mini of $\chi^2_{GPT} = 205.64$. This suggests a strong association between domains and the distribution of answers, indicating that the model performance significantly varies across different domains.

With the $\chi^2_{Mathstral} = 101.04$ Mathstral-7B model shows the least domain-specific variability in the correctness of responses. While there is still a significant influence of domain on performance, it is less pronounced compared to the other models.

Overall, the results highlight that all models experience some degree of domain influence, with GPT-4o-mini and Llama-3.1-8B-Instruct showing particularly high sensitivity to domain variations. One reason, why the indicated models specifically showed higher variability in performances across domains is due to the sensitivity of the regular expression formulas applied for the answer extraction, and this can be further investigated with better methods of data extraction.

4.2.2 Permutation Test Analysis

Results

To further investigate inter-domain relevance, we performed a permutation test. As shown in Figure 2, the permutation test consistently yielded $p_{value} < 0.001$ across mathematical domains, corroborating the chi-square test findings.

The results provided by statistical analysis imply that the answer to the first research question investigated ("Are there significant differences in the performances of LLMs across distinct mathematical domains?") is positive, as the likelihood of correctly solving a problem is significantly dependent on the specific domain.

As regards the second question ("Which domains exhibit the highest and lowest problem-solving success?") in our experiment we observed that the lowest the lowest performance rate was exhibited in

combinatorics, and the highest performance rate for algebra.

Conclusions and Further Research

The study effectively demonstrates the application of advanced computational models to assess problem-solving performance across distinct mathematical domains. By analysing a subset of the NuminaMath-TIR dataset, we identified substantial differences in correctness rates among the domains of algebra, geometry, number theory, and combinatorics. The results underscore a statistically significant association between the domain and problem-solving success, with combinatorics exhibiting the lowest average correctness rate.

Our analysis revealed consistent patterns across the models tested: algebra frequently emerged as the domain with the highest correctness rates for all models, indicating a robust understanding and capability in this area. Conversely, combinatorics consistently presented the greatest challenge. Understanding domain influences can guide efforts in model training, specifically targeting domains where models underperform. This could involve adapting training datasets to address gaps in model performance.

While all differences between domains in model performances were significant, we observed that some models, like GPT-4o-mini and Llama-3.1-8BInstruct, showed more variability in performance ($\chi^2 > 180$), whereas model Mathstral-7B and Qwen2.5-Math-7B had their performances are more closely aligned across domains ($\chi^2 < 150$), which is might be an indication of the importance of the techniques like tool-integrated reasoning (TIR) to produce models more stable results across distinct domains. Furthermore, LLM Qwen2.5-Math7B demonstrated superior overall performance with consistent strength across topics.

Our findings emphasize the importance of domain-aware evaluation within educational technology, offering a pathway for future studies to explore tailored curriculum enhancements and model improvements. This can foster the creation of more adaptive and intelligent learning systems that manage the intricacies of various mathematical domains effectively.

References

1. Rohan Anil et al. 2023. “PaLM 2 Technical Report”, available at https://arxiv.org/abs/2305.10403.
2. Gao, Luyu, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. “PAL: Program-Aided Language Models”, available at https://arxiv.org/abs/2211.10435.
3. Glushkov, V. M., K. P. Vershinin, Yu. V. Kapitonova, et al. 1974. “About a Formal Language for Recording Mathematical Texts: Automation of the Search for Proofs of Theorems in Mathematics.”
4. Gou, Zhibin, Zhihong Shao, Yeyun Gong, Yelong Shen, Yujiu Yang, Minlie Huang, Nan Duan, and Weizhu Chen. 2024. “ToRA: A Tool-Integrated Reasoning Agent for Mathematical Problem Solving”, available at https://arxiv.org/abs/2309.17452.
5. Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, et al. 2023, “Mistral 7b”, available at https://arxiv.org/abs/2310.06825.
6. LI, Jia, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, et al. 2024. “NuminaMath.” available at *GitHub Repository* Project Numina: https://github.com/project-numina/aimo-progress-prize.
7. Dubey, A., Jauhri, A., Pandey, A., Kadian et al. 2024. “The Llama3 Herd of Models”, available at https://arxiv.org/abs/2310.06825.
8. OpenAI. 2024. “GPT-4 Technical Report”, available at https://arxiv.org/abs/2303.08774.
9. Penedo, Guilherme, Quentin Malartic, Daniel Hesslow, Ruxandra Cojocaru, Alessandro Cappelli, Hamza Alobeidli, Baptiste Pannier, Ebtesam Almazrouei, and Julien Launay. 2023. “The RefinedWeb Dataset for Falcon LLM: Outperforming Curated Corpora with Web Data, and Web Data Only”, available at https://arxiv.org/abs/2306.01116.
10. Touvron, Hugo, and et al. 2023. “Llama 2: Open Foundation and Fine-Tuned Chat Models”, available at https://arxiv.org/abs/2307.09288.
11. Trinh, Trieu, Yuhuai Wu, Quoc Le, and Thang Luong. 2024. “Solving Olympiad Geometry Without Human Demonstrations”, available at https://doi.org/10.1038/s41586-023-06747-5.
12. Wei, Jason, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, available at https://arxiv.org/abs/2201.11903.
13. Yang, An, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, et al. 2024. “Qwen2.5-Math Technical Report: Toward Mathematical Expert Model via Self-Improvement”, available at https://arxiv.org/abs/2409.12122.

The article has been sent to the editors 01.11.24.
After processing 15.11.24.
Submitted for printing 30.12.24.