Replacing the Human in the Loop: Towards Automated Evaluation of LLM Mathematical Proof Generalization

Carolina Dias-Alexiou, Edison Marrese-Taylor, Alfredo Solano, Yutaka Matsuo
Graduate School of Engineering, The University of Tokyo
{carolina.dias,emarrese,asolano,matsuo}@weblab.t.u-tokyo.ac.jp

Abstract

As we continue to extend their range of applications, the study of the generalization capabilities of large language models (LLM) has become an important topic. Previous works have recently proposed to study these abilities in the context of mathematical reasoning. This was done by asking models to reproduce proofs that they have most likely seen during training, but when key symbols are replaced. Since this evaluation relies heavily on human feedback, we study to what degree this evaluation can be automatically performed. To that end, we simulate the human evaluation proposed by previous work by interactively presenting the same questions to the LLMs as prompts. We analyze the degree of alignment between human and model answers via an accuracy and correlation analysis, as well as via inter-annotator agreement metrics. We test a selection of models, including ones with reasoning capabilities, and show LLMs currently have important limitations in this regard, which cannot be addressed by ICL.

1 Introduction

Mathematical reasoning is a key aspect of human intelligence that encompasses pattern recognition and logical entailment. The development of artificial intelligence systems capable of performing tasks such as solving applied and theoretical mathematical problems has been a long-standing focus of research in the fields of machine learning and natural language processing, dating back to the 1960s (Feigenbaum and Feldman, 1963; Bobrow, 1964).

With the advent of large language models (LLMs), there has been a lot of interest in equipping these models with the ability to reason. This ability could be of use to mathematicians, for example, to employ such LLMs for *interactive theorem provers* (ITPs), where the machine and the users work together to produce a formal proof. In order to track progress on this research topic, several tasks intended to measure the mathematical abilities of LLMs have been proposed over the last years .

Early work focused on testing the ability of models to understand mathematical expressions focused on the manipulation of simple expressions using standard notation (Allamanis et al., 2017; Evans et al., 2018). More recently, interest has shifted from these approaches to a question-answer format using natural language, which is more suitable for LLMs. The release of the MathQA (Amini et al., 2019), GSM8K (Cobbe et al., 2021) and GPQA (Rein et al., 2023) datasets extended the class of questions tested to many areas in mathematics.

A different angle to test models is to have them generate actual proofs. The first such benchmark is NaturalProofs (Welleck et al., 2021, 2022) which contains theorems paired with a gold proof, all in "informal" language. Similar datasets, such as GHOSTS (Frieder et al., 2023a) and MATH (Hendrycks et al., 2021) have also been developed recently. Furthermore, datasets like Omni-math (Gao et al.) and FrontierMath (Glazer et al., 2024) have introduced sets of challenging mathematics problems that can be verified computationally. Some of these datasets utilize the popular LLM-as-a-Judge approach, while others directly employ automated verification utilizing ad-hoc languages. We also find benchmarks that directly utilize ITPs in proof generation/validation (e.g. LEAN), including MiniF2F (Zheng et al., 2022), the LeanDojoBenchmark (Yang et al., 2023) and FormalMATH (Yu et al., 2025).

While the ideas behind these works are promising, our work follows a different approach: to evaluate the performance of machine learning models via model input perturbation. Prior works have used this approach as a means to test the robustness of the mathematical reasoning capabilities of models. Generally, this has meant generating perturbed, hard variations of previously-existing benchmarks or datasets to use as for such evaluations (Shi et al.,

Findings of the Association for Computational Linguistics: IJCNLP-AACL 2026, pages 996–1004
November 6-10, 2026 ©2026 Association for Computational Linguistics

2023; Li et al., 2024; Mirzadeh et al., 2024; Zhang et al., 2024; Gulati et al., 2024; Zou et al., 2024; Huang et al., 2025; Dias-Alexiou et al., 2025).

Finally, our work is also related to the usage of language models in automated grading and feedback systems for proof generation and other tasks related to mathematics, often with a focus on reasoning (Stephan et al., 2024; Mondorf and Plank, 2024; Ahn et al., 2024). For example, (Gao et al., 2024) proposed MATH-Minos, a system trained with stepwise natural language feedback which can identify various types of errors, including logical flaws. Similarly, (Miao et al., 2024) proposed SelfCheck, a mechanism to check if LLMs recognize errors in their own step-by-step reasoning, without resorting to external resources, testing on math- and logic-based datasets such as the ones discussed above. Yue and Klein (2025) directly evaluate LLM proof-writing capabilities using the LLM-as-a-judge framework to score the generated solutions. More recently, Zhang et al. (2025) used an ensemble of LLM Judges to evaluate reasoning in formal mathematics. Finally, approaches more tailored to educational settings have appeared. For example, Zhao et al. (2025) proposed a system to automatically grade induction proofs.

2 Background

As discussed in the previous Section, current approaches to evaluate the mathematical abilities of LLMs often focus on question answering, text completion, or similar tasks (Cobbe et al., 2021; Frieder et al., 2023b; Rein et al., 2023) where the content of the input is written in the style of informal mathematics. Alternatively, when it comes to formal mathematics, approaches focus on test passing via feeding the generated proof to the corresponding ITP language (Polu et al.; Han et al.; Lample et al., 2022; Azerbayev et al.; Zheng et al., 2022; Yang et al., 2023), so that it can be automatically checked. For approaches based on retrieval, or those that can be used to generate multiple candidate solutions, evaluation is performed in terms of the rate of top-k generations that pass the ITP test. Naturally, this setting requires that the model/system tested can either directly generate ITP code, or access an ITP environment via tool calling. An alternative to all of this is to use LLM-as-a-Judge. However, this does come at a cost. For example, it is known that LLMs are sensitive to the position of the examples presented, tend to prefer longer outputs and can exhibit a bias towards outputs generated by themselves or by models from the same family.

Here, we look at the approach by Dias-Alexiou et al. (2025) which utilizes the input-perturbation approach to study the mathematical reasoning abilities of LLMs. Concretely, they ask if LLMs models can recognize that two structures are the same even if they do not share the same nomenclature, and operationalize this by designing a mechanism to measure to what degree models can reproduce proofs that they have seen during training, but when the symbols do not match the ones seen.

In their approach, it is assumed that proofs are written in a standard system usually referred to as a Hilbert system. This is a deductive system that generates theorems from axioms (tautologies taken as starting point for further reasoning) and *modus ponens*. In this context, a proof is ultimately written as an ordered collection of statements with references (Welleck et al., 2021).

The evaluation consists of comparing a given proof against a gold standard proof by checking the following points: (i) Steps from the gold standard proof that appear in the generated proof, i.e., to what extent the model used the needed axioms (or given facts) and rules of inference; (ii) Steps that appear in the generated proof and were correctly applied. In other words, the steps that are correct; (iii) Steps that appear in the output providing an expression which is a step towards finalizing the proof, even if it was not correctly deduced, and (iv) Overall clarity of the proof, in terms of explanations.

In agreement with previous works such as Welleck et al. (2021), they indicate that in an ideal scenario, their evaluation should be performed by a qualified human annotator. In fact, we see that previous work has stressed the need to rely on experts for evaluation of theorem proving systems, including the work of Welleck et al. (2022), who carried out an in-depth annotation where an expert annotator is presented with the theorem, proof-so-far, and a generated next-step. Frieder et al. (2023b) also highlight that human evaluation of advanced mathematics that approaches research level is expensive and requires experts. There have been significant advancements in the capabilities of LLMs to act as judges for a variety of tasks, perhaps partly explained by their ability to perform reasoning. Thus, we think it is possible to partially automate the proposed evaluation, and suggest an empirical method to assess to what de-

gree it can be done. Comparing our proposal, we find that prior work by Ma et al. (2026) is perhaps the closest our work. The paper also evaluates the ability of LLMs to detect incorrect, classifying them into failure mode classes. Similarly, on "The Open Proof Corpus" (Dekoninck et al., 2026), a dataset of LLM-generated proofs that was labeled by humans as correct/incorrect, evaluation via accuracy on binary classification. In both cases, our scheme differs in two ways. First, we focus on zero-order logics only, where we have guarantees the proofs should be solvable. Second, our framework is much finer than binary classification, showing specifically where LLMs fail on judgment. Finally, we also mention the work of Mahdavi et al. (2025), who propose a methodology that moves away from binary classification into a score or scale. Although our evaluation moves in the same direction, their method is still not able to point at specific the types of mistakes or areas where models struggle.

3 Experiments and Results

3.1 Proposed Study

As mentioned earlier, a key point of the evaluation scheme we are adopting was to ensure that the generated proofs were checked by mathematicians. Here, we propose to automate this evaluation utilizing the LLM-as-a-Judge approach. We mimic their evaluation process by showing each judge-model the exact input fed to a previous model, the output generated by it, as well as the "gold standard" proof. We then proceed to ask each judge-model the same interactive questions asked to the human annotator. First, if the output of the model contains a proof. If so, then if each step in the "gold standard" appears in the output of the model. Additionally, if the step invokes an axiom or rule of inference, we ask the models to do the following: (a) If the step invokes an axiom, we ask to check if variables were substituted correctly; (b) If the step uses a rule of inference, we ask to check if the rule is properly invoked; (c) We ask the model to judge whether the step contributes to the proof in the sense that it stirs the overall flow of the proof in the right direction towards conclusion. Finally, we ask it to rate the output on a scale of 0 to 4 via "Very Incoherent, Incoherent, Neither Coherent nor Incoherent, Coherent, Very Coherent".

This evaluation protocol operates on proofs in propositional calculus, which we think (following the original paper's argument) represents a best-case scenario for LLM judges for several reasons. (1) *Foundational status*: All the machinery of propositional logic is included in higher-order logic and, thus, most of mathematics. Crucially, this setting does not involve quantifiers, which would introduce additional complexities in both proof generation and evaluation. (2) *Semantic completeness*: Propositional calculus is semantically complete. Any tautology can be proved using the formal axioms and rules of inference of the system. This means that failure in providing a proof is an issue of the prover and not of the logical system. (3)*Training exposure*: Being the subject of common undergraduate courses, demonstrations in propositional logic have been widely discussed online (e.g. Math Stack Exchange), so we can reasonably assume LLMs have been extensively exposed to these types of proofs during training. (4)*Self-contained*: The entire logical structure (axioms, rules of inference) can be included in the prompt, reducing assumptions and making evaluation fully grounded. Based on these principles, we think our findings represent a lower bound on difficulty. If LLM judges struggle to reliably assess correctness in this controlled, minimal, complete logical system, they are likely to perform worse reasoning in more complex mathematical domains.

For this task, we used the same proofs and human annotated data by Dias-Alexiou et al. (2025), and tested the following models as judges: GPT 4.1, GPT-OSS 20B and 120B (Int, 2026), Qwen 2.5 32B (Bai et al., 2025), and an equivalent DeepSeek distilled model DeepSeek-R1-Distill-Qwen-32B (Guo et al., 2025) – the first reasoning models to be released as open-weight models.

Based on the questions each of these models answered, we compute the key metrics used by the original paper to evaluate formal proofs.

- *Compliance*: Percentage of times the model generated output that contains a proof [1];
- *Extensiveness*: Percentage of steps from the gold standard proof that appear in the answer;
- *Correctness*: Percentage of steps that appear and were correctly applied;
- *Flow*: Percentage of steps that appear in the output providing an expression which is a step towards finalizing the proof, even if it was not correctly deduced;
- *Clarity*: Average coherence score reported.

[1] Here, we use a softer version of Compliance than the one used by Dias-Alexiou et al. (2025)

		GPT-4.1 -	GPT-oss (120B)	GPT-oss (20B)	Qwen 2.5 (DS, 32B)	Qwen 2.5 (32B)
Acc.	Compl.	**0.857**	**0.824**	**0.840**	0.277	0.403
	Extens.	**0.692**	**0.653**	**0.673**	**0.818**	**0.678**
	Correct.	**0.764**	**0.800**	**0.688**	0.000	0.486
	Flow	**0.918**	**1.000**	**0.983**	-	0.500
	Clarity	0.267	0.344	0.188	-	0.143
IAA	Compl. (κ)	**0.479*****	0.058	0.148	0.033	0.007
	Exten. (κ)	0.361***	0.313***	0.347***	0.290***	0.311***
	Correct. (κ)	0.185	0.181	0.151	0.000	0.097
	Flow (κ)	0.000	-	0.000	-	-0.069
	Clarity (τ)	**0.606*****	**0.586*****	**0.435*****	-	0.292
Corr.	Compl.	0.481***	0.111	0.282***	0.101	0.011
	Extens.	**0.558*****	**0.643*****	**0.712*****	0.294**	0.160*
	Correct.	**0.738*****	**0.815*****	**0.832*****	0.021*	0.101
	Flow	**0.689*****	**0.767*****	**0.842*****	0.076***	0.003
	Clarity	**0.583*****	**0.695*****	**0.530*****	-	0.051

Table 1: Results of our experiments using the LLM-as-a-Judge approach. Correlation is short for Pearson correlation. **Bold** indicates values higher than 0.500 for Accuracy and Correlation and values higher than 0.400 for IAA. Underline indicate negative corr. or IAA, while * denotes significance at 90, 95 and 99 confidence.

Our evaluation of the generated replies aims to study to what degree the behavior of the LLMs agrees with humans. For this, we utilize the human labels collected by the authors. On the one hand, taking the human labels as the gold standard, we report the accuracy of each LLM in answering each item on the questionnaire in a zero-shot fashion. On the other hand, we report inter-annotation agreement (IAA) between the LLM and the human answers via Cohen Kappa (κ) for Compliance, Extensiveness, Correctness and Flow and Kendal's Tau (τ) for Clarity, and Pearson correlation always.

Finally, we assess to what degree In-Context Learning (ICL) can help bridge the expected performance gap between humans and zero-shot LLMs. We tested 3 ICL strategies for providing human annotation examples to the LLM judge. (1) Random (rand): N annotated examples selected uniformly at random, re-sampled for each test item. Tests whether generic annotation examples improve task understanding. (2) Random Fixed (fixed): N examples selected once and reused across all test items. Controls for the variance introduced by per-item re-sampling. (3) Same Category (same): N examples drawn from items matching the test item's proof category (same context level and modifier). Tests whether domain-matched demonstrations provide additional benefit. Each strategy was tested with $N \in 1, 2, 5$ shots. To the best of our knowledge, no other study of this kind has been proposed.

3.2 Results

Table 1 summarizes the results of our zero-shot evaluation. It is important to note that Compliance is the only metric for which all questions are used to compute the scores, since all subsequent questions are only answered if the annotator and the judge agree that there is a proof to be evaluated. For all other metrics, the number of examples decreases and, in some cases, it can be as low as 10% of the original number or even zero. While we acknowledge that this makes some of the metrics hard to interpret, we note that this conditional evaluation is inherent to the task structure. In other words, one cannot judge, for example, step correctness without first establishing that a proof attempt exists. We also note that because top-performing LLMs achieve high gate agreement ($> 80\%$ overlap with humans), the downstream sample sizes are substantial (typically 90+ items out of 120).

We note that, even with reasoning, the distilled Qwen 2.5 model did not perform significantly better than its base model. In fact, the Compliance measure was worse for the modified model (27.7%) than for the base model (40.3%). Overall, we note that all these models generally struggle with instruction following. For example, for the first question (whether the tested model generated an answer), we often observed that models would start generating a proof rather than answering our question [2]. This suggests a significant gap between Qwen and GPTs in terms of their instruction following capabilities, making the former models essentially unsuitable for our needs.

Although GPT models have high values for accuracy and correlation, IAA values tend to be low. Even for GPT 4.1, which is the model with the highest overall scores, the IAA score for Correctness and Flow indicates a pure chance agreement. This becomes even more evident for the open-weight models where only Clarity has a significant IAA value. In general, these models showed good agreement with the annotator on the clarity of the overall text, but were not particularly good at judging the contents of the proof per se. Finally, all models show significant Extensiveness κ, suggesting all LLMs can detect step presence reasonably well. Conversely, Correctness κ is low for all models and none achieve significance here, suggesting step-correctness evaluation is unreliable.

To test if the input-output format interfered with the results, we also ran experiments using JSON-mode responses (both JSON-output-only

[2] Malformed responses, which occurred rarely ($< 2\%$ of outputs for most models) were logged and treated as missing values.

		GPT oss 20B			Qwen2.5-32B		
		0-shot	1-fixed	5-rand	0-shot	1-fixed	5-rand
Acc.	Compl.	**0.840**	**0.857**	**0.840**	0.403	**0.504**	0.395
	Extens.	**0.673**	**0.680**	**0.680**	**0.678**	**0.672**	**0.632**
	Correc.	**0.688**	**0.722**	**0.788**	0.486	**0.605**	**0.714**
	Flow	**0.983**	**0.968**	**0.951**	0.500	**0.750**	**0.643**
	Clarity	0.188	0.144	0.146	0.143	0.231	0.353
IAA	Compl. (κ)	0.148	0.279	0.195	0.007	0.159	0.036
	Extens. (κ)	0.347	0.350	0.355	0.311	0.279	0.169
	Correc. (κ)	0.151	0.298	**0.493**	0.097	0.193	**0.469**
	Flow (κ)	0.000	0.000	0.000	-0.069	**0.486**	0.125
	Clarity (τ)	**0.435**	**0.435**	**0.503**	0.292	0.198	0.216
Corr.	Compl.	0.282	0.403	0.281	0.011	0.256	0.065
	Extens.	**0.712**	**0.705**	**0.738**	0.160	0.141	-0.029
	Correct.	**0.832**	**0.854**	**0.867**	0.101	0.238	0.059
	Flow	**0.842**	**0.876**	**0.866**	0.003	0.118	0.060
	Clarity	**0.530**	**0.522**	**0.511**	0.051	-0.079	0.047

Table 2: Effect of selected ICL strategies on agreement with human annotations. For each model we show zero-shot and the best-performing few-shot condition. "rand" = random example selection (re-sampled per item); "fixed" = random examples fixed across all items. **Bold** indicates values > 0.500 (Acc./Corr.) or $>$ 0.400 (IAA). Underline indicates negative values.

	GPT-4.1			GPT-oss (120B)		
Metric	base	out	in+out	base	out	in+out
Completeness	0.479	0.076	0.000	0.058	0.010	0.257
Extensiveness	0.334	0.423	0.419	0.311	0.354	0.323
Correctness	0.185	0.114	0.049	0.181	0.242	0.207
Flow	0.000	0.000	0.000	-	0.000	0.488
Clarity	0.128	0.219	0.258	0.212	0.108	0.060

Table 3: Results of our experiments with constrained decoding via enforcing JSON inputs (in), outputs (out) and both (in+out), versus free text decoding (base).

and JSON-input+output) with GPT-4.1 and GPT-oss (120B) via vLLM. The results summarized on Table 3 show that the JSON mode produces mixed results. For GPT-4.1, we see it degrades the critical question, as Completeness drops from 0.479 to near zero because the model defaults to "yes" for nearly every item, while modestly improving some downstream metrics. For GPT-oss (120B), results are similarly mixed. We see JSON helps on binary step-level questions, but hurts on more nuanced judgments like Clarity. Overall, this evidence suggest that structured output does not materially change our conclusions. LLM judges remain unreliable on Correctness regardless of format.

We also implemented and tested a panel-of-judges approach. We experimenting aggregating evaluations from 5 models (GPT-oss (120B), GPT-oss (20B), GPT-4.1, Qwen 2.5 (32B), and Qwen 2.5 Distill (32B) into a single virtual panel verdict using configurable aggregation strategies. Concretely,

Metric	Best Single	κ	Best Panel	κ	Winner
Complet.	GPT-4.1	0.479	Majority	0.138	single
Extensiv.	GPT-oss (20B)	0.342	Any	0.347	panel
Correct.	GPT-4.1	0.185	Any	0.113	single
Flow	GPT-4.1	0.000	All	0.154	panel
Clarity	GPT-oss (120B)	0.212	All	0.165	single

Table 4: Results of our experiments with panels of LLM-as-a-Judge approaches, using several voting techniques, compared against best single models. No single voting or panel can consistently outperform single models

we tested: majority vote, mean, median, unanimous, and "any". For binary fields, each strategy applies its voting rule across panelists; for ordinal coherence scores, it uses mode/mean/median/max.

We note that the LLM panel approach does not consistently outperform the best individual model (see table 4). Panels win on 2 out of 5 of the core metrics (marginally), while individual models dominate on Completeness and Clarity. We also found that including weaker models dilutes panel quality on harder judgments. We think this supports our finding that the difficulty lies in the nature of mathematical correctness evaluation.

When it comes to the effects of ICL, shown in Table 2), we see this helps with the Correctness evaluation (judging whether axiom substitutions are done right) across both models, but this improvement does not generalize and other dimensions either stay flat or behave erratically, especially for the weaker model. In general, we observe that ICL does not have a positive impact on the ability of models to better mimic humans here, in some cases actually degrading performance with more shots.

4 Conclusions

We studied to what degree LLMs can be used to replace human verification when evaluating formal proofs in a controlled scenario where the original proofs are perturbed. We observed that open-weight models had difficulties following instructions, which limited their suitability to replace humans in this type of evaluation. It was also noted that, for these models, reasoning did not help them perform significantly better. We also noted that even for API type models, which had higher rates of accuracy, their inter-annotation agreement with the human annotator was not much better than chance in most cases, showing that current models offer limited ability in this regard, even with the help of ICL or the use of a panel-of judges.

Limitations

While our study may provide valuable insights into the mathematical reasoning abilities of large language models, it is subject to several limitations. First, our analysis is constrained to a finite and small set of tasks, which do not capture the full breadth of mathematical reasoning scenarios. Additionally, human evaluation, while essential for assessing nuanced reasoning steps, is inherently subjective and may introduce variability in judgments. The models examined and the techniques tested represent only a snapshot of available architectures and methods. Finally, our study focuses on English-language prompts.

References

2026. Introducing gpt-oss. https://openai.com/index/introducing-gpt-oss/.

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, St. Julian's, Malta. Association for Computational Linguistics.

Miltiadis Allamanis, Pankajan Chanthirasegaran, Pushmeet Kohli, and Charles Sutton. 2017. Learning Continuous Semantic Representations of Symbolic Expressions. In *Proceedings of the 34th International Conference on Machine Learning*, pages 80–88. PMLR.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen Marcus McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. Llemma: An open language model for mathematics. In *The Twelfth International Conference on Learning Representations*.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Daniel G Bobrow. 1964. Natural language input for a computer problem solving system. *AI Technical Reports*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, Kaloyan Tsvetkov, Nikolay Drenchev, Lazar Todorov, Kalina Nikolova, Nikolay Georgiev, Vanesa Kalinkova, Margulan Ismoldayev, Mislav Balunovic, and Martin Vechev. 2026. The open proof corpus: A large-scale study of llm-generated mathematical proofs. In *International Conference on Learning Representations*, volume 2026, pages 22214–22244.

Carolina Dias-Alexiou, Edison Marrese-Taylor, and Yutaka Matsuo. 2025. An in-depth human study of the mathematical reasoning abilities in Large Language Models.

Richard Evans, David Saxton, David Amos, Pushmeet Kohli, and Edward Grefenstette. 2018. Can Neural Networks Understand Logical Entailment? In *International Conference on Learning Representations*.

Edward A Feigenbaum and Julian Feldman. 1963. Computers and thought.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023a. Mathematical Capabilities of ChatGPT. *Advances in Neural Information Processing Systems*, 36:27699–27744.

Simon Frieder, Martin Trimmel, Rashid Alawadhi, and Klaus Gy. 2023b. LLM vs ITP. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

Bofei Gao, Zefan Cai, Runxin Xu, Peiyi Wang, Ce Zheng, Runji Lin, Keming Lu, Dayiheng Liu, Chang Zhou, Wen Xiao, and 1 others. 2024. Llm critics help catch bugs in mathematics: Towards a better mathematical verifier with natural language feedback. *arXiv preprint arXiv:2406.14024*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. Omni-math: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations*.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, and 1 others. 2024. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. *arXiv preprint arXiv:2411.04872*.

Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. 2024. Putnam-AXIOM: A functional and static benchmark for measuring higher level mathematical reasoning. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, and 175 others. 2025. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Jesse Michael Han, Jason Rute, Yuhuai Wu, Edward Ayers, and Stanislas Polu. Proof artifact co-training for theorem proving with language models. In *International Conference on Learning Representations*.

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. In *35th Conference on Neural Information Processing Systems (NeurIPS) Track on Datasets and Benchmarks*.

Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. 2025. MATH-perturb: Benchmarking LLMs' math reasoning abilities against hard perturbations. In *Forty-second International Conference on Machine Learning*.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. 2022. Hypertree proof search for neural theorem proving. *Advances in Neural Information Processing Systems (NeurIPS)*, 35:26337–26349.

Qintong Li, Leyang Cui, Xueliang Zhao, Lingpeng Kong, and Wei Bi. 2024. Gsm-plus: A comprehensive benchmark for evaluating the robustness of llms as mathematical problem solvers. *arXiv preprint arXiv:2402.19255*.

Wenjie Ma, Andrei Cojocaru, Neel Kolhe, Haihan Zhang, Vincent Zhuang, Matei Zaharia, and Sewon Min. 2026. Reliable fine-grained evaluation of natural language math proofs. In *International Conference on Learning Representations*, volume 2026, pages 72740–72778.

Hamed Mahdavi, Alireza Hashemi, Majid Daliri, Pegah Mohammadipour, Alireza Farhadi, Samira Malek, Yekta Yazdanifard, Amir Khasahmadi, and Vasant G. Honavar. 2025. Brains vs. Bytes: Evaluating LLM Proficiency in Olympiad Mathematics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2024. Selfcheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *The Twelfth International Conference on Learning Representations*.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2024. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. *arXiv preprint arXiv:2410.05229*.

Philipp Mondorf and Barbara Plank. 2024. Beyond accuracy: Evaluating the reasoning behavior of large language models - a survey. In *First Conference on Language Modeling*.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. Formal mathematics statement curriculum learning. In *The Eleventh International Conference on Learning Representations*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2023. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *Preprint*, arxiv:2311.12022.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Andreas Stephan, Dawei Zhu, Matthias Aßenmacher, Xiaoyu Shen, and Benjamin Roth. 2024. From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks. *arXiv preprint arXiv:2409.04168*.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hannaneh Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. In *Thirty-fifth Conference on Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*.

Sean Welleck, Jiacheng Liu, Ximing Lu, Hannaneh Hajishirzi, and Yejin Choi. 2022. Naturalprover: Grounded mathematical proof generation with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J. Prenger, and Animashree Anandkumar. 2023. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. *Advances in Neural Information Processing Systems*, 36:21573–21612.

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, Yandong Wen, Ge Zhang, and Weiyang Liu. 2025. Formalmath: Benchmarking formal mathematical reasoning of large language models. *Preprint*, arXiv:2505.02735.

Jonathan Yue and Daniel Klein. 2025. Benchmarking llms on advanced mathematical reasoning.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, and 1 others. 2024. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*.

Lan Zhang, Marco Valentino, and Andre Freitas. 2025. Beyond gold standards: Epistemic ensemble of llm judges for formal mathematical reasoning. *Preprint*, arXiv:2506.10903.

Chenyan Zhao, Mariana Silva, and Seth Poulsen. 2025. Autograding Mathematical Induction Proofs with Natural Language Processing. *International Journal of Artificial Intelligence in Education*.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2022. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *International Conference on Learning Representations (ICLR)*.

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2024. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. *arXiv preprint arXiv:2411.00836*.

A Costs and Computational Resources

Hardware and Costs Our experiments were performed on a large cluster, where we usually rely on a node with 8 NVIDIA H200 GPUs for LVLM inference. Inference on all examples, for all cases usually amounts to a few hours for the largest models. For our evaluation with API models, our experiments cost less than 100 USD.

Packages We utilize vLLM as an inference engine for the LLMs, via docker on version 0.11.0. Performance and IAA metrics are used directly from scipy (kendalltau) and sklearn metrics (cohen_kappa_score, accuracy_score).

Use of AI Assistants AI models were utilized to facilitate table generation from raw data.

B Proofs and Prompts

Example requested proof See figure 1

Prompt - Proof request

Given the following axioms:
(Ax 1)$((\phi \wedge \psi) \to \phi)$
(Ax 2)$((\phi \wedge \psi) \to \psi)$
(Ax 3)$(\phi \to (\psi \to (\phi \wedge \psi)))$
(Ax 4)$(\phi \to (\phi \vee \psi))$
(Ax 5)$(\phi \to (\psi \vee \phi))$
(Ax 6)$((\phi \to \chi) \to ((\psi \to \chi) \to ((\phi \vee \psi) \to \chi)))$
(Ax 7)$(\phi \to (\psi \to \phi))$
(Ax 8)$((\phi \to (\psi \to \chi)) \to ((\phi \to \psi) \to (\phi \to \chi)))$
(Ax 9)$((\phi \to \psi) \to ((\phi \to \neg\psi) \to \neg\phi))$
(Ax 10)$(\neg\phi \to (\phi \to \psi))$
(Ax 11)$(\phi \vee \neg\phi)$
(Ax 12)$((\phi \wedge \neg\phi) \to \psi)$
(Ax 13)$((\phi \to (\psi \wedge \neg\psi)) \to \neg\phi)$
(Ax 14)$(\neg\neg\phi \to \phi)$
(Modus Ponens)$\{P \to Q, P\} \vdash Q$
Using only the axioms provided, prove that
$(P \to Q), (Q \to R)\} \vdash (P \to R)$.

Figure 1: Example of prompt provided to the LLMs showing the content of the full context provided, namely, the axioms and rules of inference.

Example gold standard proof See figure 2

Question

Question: Prove that $\{(P \to Q), (Q \to R)\} \vdash (P \to R)$.
Answer:

$(P \to Q)$ (hyp)
$(Q \to R)$ (hyp)
$((Q \to R) \to (P \to (Q \to R)))$ (Ax 7)
$(P \to (Q \to R))$ 2, 3 MP
$((P \to (Q \to R)) \to ((P \to Q) \to (P \to R)))$ (Ax 8)
$((P \to Q) \to (P \to R))$ 4, 5 MP
$(P \to R)$ 1, 6 MP

Figure 2: Gold standard proof utilized for our study, which we allowed LLM judges to see.

Prompt for the Judge LLM (Completeness) See figure 3

Prompt - Evaluation request

Given the following question and answer.
Question: (markdown output)
Answer: (markdown output)
Is there a provided answer for the question?
Reply only YES or NO.

Figure 3: Example prompt fed to the judge LLM for the Completeness evaluation.