PROOFLESS: Final-Answer Datasets Do Not Assess Mathematical Reasoning Reliably

Simon Frieder[1,*], Sam Bealing[2,†], and Arsenii Nikolaiev[3,†]

[1] Oxford University, Oxford, UK
simon.frieder@cs.ox.ac.uk
[2] sam@sambealing.com
[3] arsik2kk@gmail.com

Abstract

We show that final-answer datasets are not a reliable signal to measure the capabilities of LLMs to carry out proofs correctly. We highlight, on a novel benchmark of uncontaminated problems, PROOFLESS, how it is possible for LLMs to guess the right answer while failing to produce correct proofs and introduce new terminology regarding the ways an answer can be guessed without completing a full proof. This shows that hill-climbing with these types of datasets will—perhaps unexpectedly—not necessarily lead to LLMs with strong math reasoning capabilities, but rather to LLMs with strong abilities to carry out mathematical heuristics, and brittle proof generating abilities.

1 Introduction

The current landscape of natural-language mathematical benchmark datasets is overwhelmingly dominated by answer-only formats, where problems are paired solely with final solutions rather than the intermediate reasoning or derivation steps. Important examples are the MATH dataset [HBK+21] and the GSM8K dataset [CKB+21], which have been important "hill-climbing" datasets, used to measure the mathematical performance of large swaths of large language models (LLMs). Further notable examples of final-answer datasets are Olympiad-Bench [HLB+24], UGMathBench [XZC+25], or the mathematical part of the Humanity's Last Exam dataset [PGH+25].

In all such benchmark datasets, the mathematical problems are designed so that several reasoning steps need to be carried out to arrive at a solution that is a single token (e.g., an integer, a matrix, etc.), that the LLM is typically instructed to provide at the end and that can be easily extracted. This token is the only signal that is used to assess the correctness of an LLM-generated proof, by matching it to the ground-truth answer.

While some limitations of these benchmark datasets have been noted for non-proof-based tasks—such as testing an LLM's ability to follow multi-step reasoning or find counterexamples, see [FBC+24]—they also fall short within proof-based tasks, as final-answer datasets provide insufficient assurance that the reasoning process was carried out correctly.

2 Results

To highlight that final-answer mathematical datasets may not be a good "hill" for LLMs to climb if LLMs are to become proficient natural-language proof generators, we have introduced a novel benchmark of questions whose final-answer is **sufficiently easy** to find, but whose **fully correct proof is hard**, called **PROOFLESS**: *Problems Requiring Only Output, Foregoing Logical Explanations in Solver Steps.*

*Corresponding author.
†Independent researcher.

It consists of uncontaminated Olympiad-level math problem that were carefully created to ensure that the final answer can be guessed in certain ways. We present here a growing list of the possible undesirable approaches that allow LLMs to get to the correct answer while not having sufficient insight to construct a 'proof' for a question:

- **Direct bash:** A model could simply enumerate all possible cases to get to the correct answer.

  This strategy also commonly appears in number theory problems where the model checks all possible 'small' solutions which turn out to cover the whole solution set. This is not a complete proof and showing these are the *only* solutions is the bulk of the work of such a problem.

- **Smart-guessing:** A problem may have a natural construction that a model conjectures to be optimal. This allows it to calculate the correct bound without having any proof of optimality.

  Alternatively, a model may guess that solutions/optimal configurations have specific properties beyond what is given in the problem statement (e.g., only considering integers when the question allows for real numbers). This gives another route to a correct answer without having a complete proof.

- **Pattern spotting:** In a problem with a parameter n (e.g., grid size), a model may consider small values of n and then be able to guess a general formula for the answer $f(n)$ (e.g., $n^2 - 1$). This allows it to calculate the answer for large n with no proof.

- **Natural guessing:** If there are few numbers in the question statement and these, or small variations on these (e.g., $n/2$ by a parity argument) turn out to be the correct answer, a model may arrive at these with either no proof or an invalid proof.

- **Other bashing:** Models may have access to computer algebra, including libraries that allow for symbolic algebra. This means, for example, it could solve systems of equations or reduce systems of inequalities by pure manipulation in a way that is not possible for humans. Some symbolic manipulations would constitute a valid proof, however using numerical methods to arrive at an answer would not.

We have annotated our problems according to the ways in which LLMs can guess the answer, while not exhibiting a correct proof. Because this is work in progress, we do not provide the total number of problems in our current benchmark, but aim to release them at the time of the conference. One example of a problem of **direct bash**-type is contained in the Appendix A.

3 Conclusion

Our assessment shows that a pivot is needed to arrive at LLMs that can output correct proofs, in natural language. While various frameworks could be added on top of LLMs, for example by making use of autoformalization tools to reprompt until a guaranteed correct proof is generated, these, on one hand, suffer from the limitations of the autoformalizer and on the other will not lead to improved core LLM reasoning capabilities.

We conclude with a call for devising more human-in-the-loop evaluations of LLMs outputs, as a first step, to ensure that the generated mathematics is of a sufficiently high standard. Although we acknowledge this approach does not scale, to date too few datasets exist that inspect LLMs' proof output in detail [FPG+23, PDB+25, CJF+24]. Further evaluations are needed to understand the ways LLMs fail, expanding the list of errors introduced in [FPG+23], so that with targeted interventions LLMs improve in these domains.

References

[CJF+24] Katherine M Collins, Albert Q Jiang, Simon Frieder, Lionel Wong, Miri Zilka, Umang Bhatt, Thomas Lukasiewicz, Yuhuai Wu, Joshua B Tenenbaum, William Hart, et al. Evaluating language models for mathematics through interactions. *Proceedings of the National Academy of Sciences*, 121(24):e2318124121, 2024.

[CKB+21] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, and Heewoo Jun et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[FBC+24] Simon Frieder, Jonas Bayer, Katherine M Collins, Julius Berner, Jacob Loader, András Juhász, Fabian Ruehle, Sean Welleck, Gabriel Poesia, Ryan-Rhys Griffiths, et al. Data for mathematical copilots: Better ways of presenting proofs for machine learning. *arXiv preprint arXiv:2412.15184*, 2024.

[FPG+23] Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36:27699–27744, 2023.

[HBK+21] Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In J. Vanschoren and S. Yeung, editors, *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1. Curran, 2021.

[HLB+24] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, et al. OlympiadBench: A challenging benchmark for promoting AGI with Olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

[PDB+25] Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating llms on 2025 usa math olympiad. *arXiv preprint arXiv:2503.21934*, 2025.

[PGH+25] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

[XZC+25] Xin Xu, Jiaxin Zhang, Tianhao Chen, Zitong Chao, Jishan Hu, and Can Yang. Ugmathbench: A diverse and dynamic benchmark for undergraduate-level mathematical reasoning with large language models. *arXiv preprint arXiv:2501.13766*, 2025.

A Appendix: A Problem Whose Generated Proof Has Correct Final Answer But Erroneous Proof

The following problem is a novel problem, thus not contained in the training data of any of the LLMs. We report below the output of `DeepSeek-R1-Distill-Qwen-32B` after the </think> token has been issued. This is thus the final, condensed proof that this LLM generates, after its thinking stage has concluded. This problem is of **direct bash type**, where a model is able to come up easily with a correct final answer, yet provide an incomplete proof.

Problem: Let n be a positive integer. We call a prime $p \geq 2$ *n-special* if

$$p^2 + 1 \mid n^p - 1.$$

What is the product of primes $p \geq 2$ that are 10001-special?

LLM output:
To determine the primes p that are 10001-special, we need to find primes p such that $p^2 + 1$ divides $10001^p - 1$.
First, we note that 10001 factors into primes as $10001 = 73 \times 137$.
We start by checking primes p for which p^2+1 divides 10000, since $10001-1 = 10000$.
The divisors of 10000 that are one more than a perfect square are 2, 5, 10, and 50, corresponding to $p^2 + 1$ where p is a prime. This gives us the primes $p = 2, 3, 7$.
Next, we verify if these primes satisfy the condition $10001^p \equiv 1 \mod (p^2 + 1)$:
1. For $p = 2$: $2^2 + 1 = 5$. Since $10001 \equiv 1 \mod 5$, $10001^2 - 1 \equiv 0 \mod 5$. 2. For $p = 3$: $3^2 + 1 = 10$. Since $10001 \equiv 1 \mod 10$, $10001^3 - 1 \equiv 0 \mod 10$. 3. For $p = 7$: $7^2 + 1 = 50$. Since $10001 \equiv 1 \mod 50$, $10001^7 - 1 \equiv 0 \mod 50$.
We also checked other primes and found that they do not satisfy the condition, as their corresponding $p^2 + 1$ do not divide $10001^p - 1$.
Thus, the primes that are 10001-special are 2, 3, and 7. Their product is:

$$2 \times 3 \times 7 = 42$$

The final answer is:

$$\boxed{42}$$

The passage in red simply asserts all other primes have been considered—showing these primes are not n-special is the hard part of the problem. The <think> part of the output considers primes $q \mid p^2+1$ and notices that $\text{ord}_q(n) \in \{1, p\}$ but does extend this to a complete proof. To go from this to a complete proof, we need the following additional argument:

Firstly, we show that $p = 2, 3, 7$ are solutions as is done by the model. Now consider $p \notin \{2, 3, 7\}$. As p is prime, $\text{ord}_q(n) \in \{1, p\}$. We consider the two cases:

- *If $\text{ord}_q(n) = p$, then from Fermat's Little Theorem we conclude $p \mid q - 1$.*

  Write $q = kp + 1$ and notice this also divides $k(p^2 + 1) - p(kp + 1) = k - p$, which, since $q > k$ and $q > p$ forces $k - p = 0$. In this case, $q = p^2 + 1 > 2$ is prime so must be odd which forces $p = 2$ ais p must be even whicheven so $p = 2$ which we have already considered.

- *If* $\operatorname{ord}_q(n) = 1$, *then* $q \mid n-1$. *Consider*

$$n^p - 1 = (n-1)(n^{p-1} + \cdots + 1).$$

Note that $n \equiv 1 \mod q$, *so the factor* $n^{p-1} + \cdots + 1 \equiv p \not\equiv 0 \mod q$ *since* $q \mid p^2 + 1$ *so can't be equal to* p. *Therefore,* $\nu_q(n^p - 1) = \nu_q(n-1)$.

For $p \neq 2$, *the second case must hold for all primes* q *therefore we have* $p^2 + 1 \mid n - 1 = 10000 = 2^4 \cdot 5^4$. *Since* $p^2 + 1 \equiv 2 \mod 4$ *(because* p *is odd), we have* $p^2 + 1 \in \{2, 2 \cdot 5, 2 \cdot 5^2, 2 \cdot 5^3, 2 \cdot 5^4\}$. *An easy check shows that* $p = 3, 7$ *are the only possibilities.*