LemmaBench: A Live, Research-Level Benchmark to Evaluate LLM Capabilities in Mathematics

Antoine Peyronnet [*1 2] **Fabian Gloeckle** [*2] **Amaury Hayat** [*2 3]

Abstract

We present a new approach for benchmarking Large Language Model (LLM) capabilities on research-level mathematics. Existing benchmarks largely rely on static, hand-curated sets of contest or textbook-style problems as proxies for mathematical research. Instead, we establish an updatable benchmark evaluating models directly on the latest research results in mathematics. This consists of an automatic pipeline that extracts lemmas from the arXiv and rewrites them into self-contained statements by making all assumptions and required definitions explicit. It results in a benchmark that can be updated regularly with new problems taken directly from human mathematical research, while previous instances can be used for training without compromising future evaluations. We benchmark current state-of-the-art LLMs, which went from 10-15% accuracy to 40.8% accuracy in theorem proving (pass@1) depending on the model, showing that there is both a very fast evolution of LLMs capabilities with respect to human research-level mathematics and still a large margin of progression for general public LLMs to reach human-level proving capabilities in a research context.

1. Introduction

Deep learning and machine learning methods have enjoyed tremendous success over the past decade, culminating in Nobel Prizes in physics and chemistry in 2024. It has become increasingly clear that AI methods are changing the practice of science, sometimes to the surprise of scientists and researchers in these disciplines. While machine learning methods are already useful in mathematics (Davies et al., 2021; Charton et al., 2024; Alfarano et al., 2024), theorem proving remains a major open challenge.

The exceptional progress made by Large Language Models (LLMs) in recent months and years offers a glimpse into a future where models could be capable of proving theorems autonomously at a human level. The performances of LLMs in this area have exploded over the past two years, mainly thanks to two elements: first, the use of end-to-end reinforcement learning from verifiable rewards (RLVR), which has led to the advent of so-called *reasoning models* (Guo et al., 2025; Rastogi et al., 2025); second, a major effort to gather and generate large mathematical data corpora, particularly for Olympiad-style exercises whether in natural language (Gao et al., 2024; LI et al., 2024; Fang et al., 2025) or in formal languages (Aram H. Markosyan, 2024; Li et al., 2024; Ying et al., 2024). All of this has led to several achievements: announcements of mathematical models capable of winning a gold medal at the International Mathematical Olympiads 2025 (Huang & Yang, 2025; Team, 2025; Castelvecchi, 2025) the arrival of reasoning models that prove to be highly efficient in mathematical symbolic computations, capable of performing in a few minutes the equivalent of several hours of human work (Diez et al., 2025); and the resolution of several open problems with the help of LLM-based automated theorem provers (Chen et al., 2026; Sothanaphan, 2026; Liu et al., 2026), specialized agent tools (Dixit et al., 2026) or even generic LLMs (Schmitt, 2025; Feng et al., 2026; Jang & Ryu, 2025). While such strong performance on research-level mathematics has been claimed (Glazer et al., 2025), the true capabilities are still debated.

While current models perform very well in calculations and in undergrad curriculum mathematics, two questions arise:

- **Data contamination**: how much of the performances of such models are tainted by possible contamination of the training data?
- **Generalization and performance in research-level mathematics**: how can these models, often post-trained on curated or competition-style math datasets, perform on specific areas of research-level mathemat-

[1]Ecole Normale Superieure de Rennes [2]Ecole des Ponts Paris [3]Korean Institute for Advanced Study. Correspondence to: Antoine Peyronnet <antoine.peyronnet@ens-rennes.fr>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026. Copyright 2026 by the author(s).

ics where, by definition, data is scarce?

This paper proposes another approach: relying on mathematicians' production to establish a research-level benchmark that can be updated as often as needed. Our pipeline, inspired by LiveCodeBench (Jain et al., 2025) in the domain of competitive coding, works as follows: it extracts lemmas from the latest research preprints on the arXiv and automatically gathers the required assumptions and definitions from elsewhere in the paper, thus making them self-contained statements that can be used as stand-alone benchmark elements. The advantages of such an automatically created benchmark are as follows:

(i) It can be updated regularly at low cost (up to weekly if needed), allowing contamination-free evaluation at all times; Human evaluation is only needed to assess the confidence score of the LLM as a judge and can be updated much less frequently (essentially when the judge is changed).

(ii) The same pipeline can be run on historical data, allowing the creation of large high-quality training datasets;

(iii) Domain composition can be steered freely, and domain-specific benchmarks can be created based on the categorization of research preprints on the arXiv;

(iv) The data is perfectly in-distribution for actual research-level lemma proving by definition of the data source.

Our **contributions** can be summarized as follows:

- We describe this **pipeline** for generating a live high-quality benchmark on lemma-proving in research-level mathematics from preprints on the arXiv, taking care to ensure *self-containedness* of the resulting statements which has been identified as a key limitation of data from the arXiv (Patel et al., 2023; Alexander et al., 2026) (Section 4.1).
- We **evaluate the quality** of the resulting self-contained lemma benchmark with manual inspection, indicating high precision in producing exact and correct mathematical statements (Section 4.2).
- Using the benchmark, we evaluate the **lemma-proving accuracy** of current state-of-the-art LLMs. Two evaluations are considered: the proofs provided are first evaluated with an LLM as a judge. A subset of the proofs are then judged by human expert mathematicians so as to provide a confidence score to the LLM as a judge evaluation. (Section 5). The main results are provided in Tables 3 and 5.

Instead of evaluating the theoretical upper bound of LLM performance in theorem proving, this paper asks a more practical question: how much of today's research-level mathematical theorem proving can already be performed using widely available, general-purpose LLMs, the tools most mathematicians are realistically expected to use in the near future—rather than specialized or advanced systems available only to a small number of researchers. We find that the capabilities of these general-purpose models have progressed rapidly over the past nine months, while a substantial gap to research-level proficiency remains.

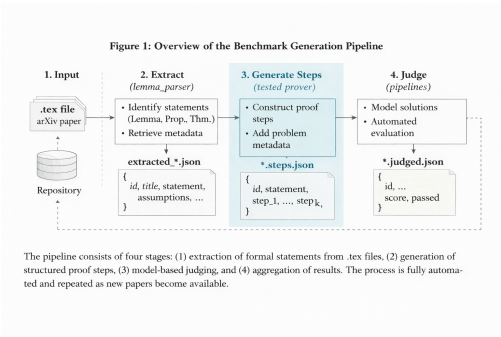

Figure 1. **Pipeline at a glance**: the pipeline retrieves papers from the arXiv, extract statements and make them self-contained. An external prover can then try to prove them step by step and each proof is inspected by a judge. The final score is the proportion of valid proofs. We used GPT-5, Gemini 2.5 Pro, Gemini 3 Pro, Claude Opus 4.5, and DeepSeek-R1 as a baseline for provers.

2. Related Work

Early mathematical reasoning benchmarks such as GSM8K (Cobbe et al., 2021), MathQA (Amini et al., 2019) and MATH (Hendrycks et al., 2021) target school- to undergraduate-level problems and are now largely saturated and exposed to contamination (Xu et al., 2024). A second line of work relies on expert-curated, often closed problem sets to push difficulty and limit leakage: FrontierMath (Glazer et al., 2025) keeps its problems closed-source, Soohak (Son et al., 2026) compiles over 400 problems from 64 mathematicians, First Proof (Abouzaid et al., 2026) releases ten expert questions for a single one-week evaluation, and the Leipzig Tier-4 set (Calvo Cortes et al., 2026) offers 102 open research-level problems. These approaches are valuable but costly to scale and update.

Closest to ours is the emerging family of *live*, arXiv-derived benchmarks. RealMath (Zhang et al., 2025) and MathArena/ArxivMath (MathArena Team, 2026) extract problems from recent papers but restrict themselves to statements with automatically verifiable answers; EternalMath (Ma et al., 2026) similarly targets code-verifiable questions. LemmaBench builds on this direction but differs in a key respect: rather than limiting the benchmark to verifiable final

answers, we evaluate *natural-language proofs* of research lemmas. This is harder to score, but closer to how mathematics is actually written, and it yields substantially more problems per unit of arXiv input (hundreds of lemmas per arXiv-week versus tens of verifiable problems per month). Our contribution is therefore not arXiv-based benchmarking in general, but self-contained arXiv *lemma proof* generation and its natural-language evaluation. Formal-language benchmarks such as miniF2F (Zheng et al., 2021), FIMO (Liu et al., 2023), Lean-Workbook (Ying et al., 2024) and PutnamBench (Tsoukalas et al., 2024) address a complementary, mostly pre-research setting. An extended discussion is given in Appendix A.

3. Problem Definition: Context Dependence in Extracted Theorems

Theorem statements in research preprints do not exist within a vacuum but tightly depend upon the definitions, objects and choices introduced previously in the paper. This raises the following question for a research-level mathematics benchmark: make long-context retrieval and contextual understanding part of an *end-to-end task*, or restrict the measured capability clearly to research-level *theorem proving*?

In this work, we opt for the second choice of removing context dependence by retrieving definitions and making statements self-contained. This allows us to provide a more targeted measure for theorem proving for which few metrics exist (for long-context understanding tasks in natural language processing, on the other hand, see for instance (Bai et al., 2024)), facilitates adoption and, at the same time, makes the pipeline useful for reinforcement learning prompt dataset construction.

What constitutes a *self-contained* theorem? In this work, we treat common undergraduate notions (e.g. the fields of rational, real and complex numbers, rings, modules and vector spaces, continuity, differentiability) as a known background corpus, and require notions and objects specific to the research conducted in the preprint to be retrieved. To give examples, the following lemmas are deemed *self-contained*, *not self-contained* and *self-contained given the retrieved context*.

Lemma 3.1 ((De Santis et al., 2026), Lemma 13)**.** *Suppose that*

$$au \leq b + c\log(u),$$

for some $a, c > 0$*, a scalar* $b \in \mathbb{R}$ *and* $u \geq 1$*. Then*

$$u \leq \max\left\{e^{b/c}, \frac{4c^2}{a^2}\right\}.$$

Example 1 (self-contained)

Lemma 3.2 ((Washburn & Zlatanović, 2026), Lemma 2.1)**.** *If* F *is reciprocal, then* G *and* H *are even.*

Example 2 (not self-contained)

Definition 3.3. Consider the following retrieved context definitions.

- Reciprocal cost: $F : \mathbb{R}_{>0} \to \mathbb{R}$ with $F(x) = F(x^{-1})$ for all $x > 0$; it is normalized if $F(1) = 0$.
- $G : \mathbb{R} \to \mathbb{R}$ defined by $G(t) := F(e^t)$.
- $H : \mathbb{R} \to \mathbb{R}$ defined by $H(t) := G(t) + 1 = F(e^t) + 1$.

Lemma 3.4 ((Washburn & Zlatanović, 2026), Lemma 2.1)**.** *If* F *is reciprocal, then* G *and* H *are even.*

Example 3 (self-contained given retrieved context)

Another example of a self-contained lemma retrieved by GPT-5 is given in F.1 in Appendix F

4. Method

4.1. Data extraction

While the research articles produced by mathematicians and stored in the arXiv represent a large source of research-level theorems, these theorems are often not self-contained: some assumptions needed to prove the statements may be scattered in the rest of the paper and only implicitly assumed in the actual statement. This is a notable reason why autoformalization of arXiv papers is challenging (Becker et al., 2025), just as their use in a theorem proving benchmark. To tackle this, we design a language-model based pipeline to identify the assumptions missing or implicitly assumed for each statement and extract them from the articles to make the statements self-contained.

Because our goal is to retrieve lemmas, the pipeline applies first a regex-based extractor to the LaTeX source and captures various lemma environments[1] We focus on lemmas because they are typically more granular and, on average, more tractable for current LLM provers; extending to other LaTeX theorem environments is, of course, straightforward.

[1]From the second iteration of the benchmark in Feb. 2026, we also discard all lemmas that bear a reference, indicating that they are already known in the literature and only recalled in the paper.

To obtain the definitions and assumptions needed to make a given statement self-contained, we investigate two complementary modes:

Full-context retrieval. Given a targeted statement S, we provide the full content of the article preceding this statement to an LLM which is then asked to retrieve all definitions D and assumptions H that would be relevant so that S is self-contained given D and H. Given a sufficiently capable model for long-context retrieval and reasoning, this approach should be able to fully resolve all dependencies that can be resolved from the preprint alone.

Vector Retrieval. For each targeted statement, a first LLM is asked to enumerate the non-trivial *objects* appearing in the statement – i.e., symbols, structures, or named results that are neither standard nor defined in the statement. We split the content of the article preceding the statement in paragraphs and, for each object, we use a regex to list all paragraphs containing it.

Separately, we compute dense embeddings of all paragraphs preceding the lemma and the literal query `"Definition of <object>"` using the `text-embedding-3-small` model in the OpenAI API and compute cosine similarity scores with ChromaDB. Concretely, we keep the top-k hits ($3 \leq k \leq 12$) whose similarity exceeds a threshold of $\tau = 0.75$.

Concatenating all paragraphs identified by the regex and selected by the retriever, for each non-trivial object in the statement, results in a context that can be passed to a second LLM call to extract the exact definitions and assumptions required to make the lemma fully self-contained.

The **full-context** method requires by design a larger number of tokens than vector retrieval in that it provides a much larger context. In our setup, full-context extraction needed about 5-10 times more tokens than vector retrieval depending on the model. However, it has a higher accuracy in retrieving the assumptions of the different lemmas as presented in the following subsection 4.2.

4.2. Self-Containedness Judge

After lemmas have been complemented with extracted assumptions, we filter for true self-containedness[2] and we then use an LLM-as-a-judge as an additional guardrail. For this, the lemma statement and its extracted assumptions are passed to the LLM with the task of assessing whether with the additional context, the lemma is self-contained, i.e. dependency-free except for notions deemed standard or classical. The model outputs a reasoning trace alongside a binary final judgment.

[2]From the second iteration of this benchmark in Feb. 2026, we first apply a deterministic filter to ensure that no external citations or undesirable environments are referenced in the extracted assumptions.

Evaluation We evaluate this extraction part of the pipeline up-front, settling on optimal choices for the final benchmark used for model evaluations in Section 5. Specifically, we conduct human annotations as well as model cross-validations to assess the LLM-based evaluation of self-containedness.

Human evaluation A significant proportion of the lemmas obtained were reviewed by human experts to check whether they were indeed self-contained and to evaluate the capabilities of the LLM-as-a-judge. Depending on the mode, between 75.5% and 96.5% of the lemmas vetted by the LLM-as-a-judge were indeed self-contained according to human mathematicians. The full results can be found in Table 1.

Extraction model	Mode	Judge model	SC (judge) (%)	PPV (%)
Gemini 2.5 Pro	Full context	Gemini 2.5 Pro	67.4	96.5
Gemini 2.5 Pro	Vector retrieval	Gemini 2.5 Pro	36.9	88.4
GPT-5	Full context	GPT-5	78.5	90.0
GPT-5	Vector retrieval	GPT-5	49.4	85.0
GPT-4	Full context	o3	47.6	78.1
GPT-4	Vector retrieval	o3	19.4	75.5
GPT-4	Full context	GPT-4	64.4	88.2
GPT-4	Vector retrieval	GPT-4	25.8	80.2

Table 1. **Evaluation of extraction modes and extraction models.** We report self-containedness (SC) proportions for 376 lemmas extracted with both modes and three different models, judged by LLM-as-a-judge using the same model. For GPT-4, we additionally use o3 as a stronger judge to measure the discrepancy. PPV (positive predictive value) is the proportion of lemmas judged self-contained that are confirmed by human evaluators.

Cross-model evaluation To further evaluate the robustness of self-containedness judgments, we performed a symmetric cross-validation: judging lemmas extracted by GPT-5 and Gemini 2.5 Pro by both models to assess the presence of systematic failure modes that each individual model cannot self-correct. The experiment uses full-context extraction and yields a cross-model confusion matrix for each of the extracted datasets.

Extracted by	Judged by	Total	Self-contained	Not self-contained
Gemini 2.5 Pro	GPT-5	254	183 (72%)	71 (28%)
confirmed by human judge		–	183 (100%)	17 (24%)
GPT-5	Gemini 2.5 Pro	296	293 (99%)	3 (1%)

Table 2. Cross-validation between Gemini 2.5 Pro and GPT-5 for detecting self-contained lemmas. Percentages on the second row refer to confirmation rates within each cell above.

We performed a cross-model validation of mathematical self-containedness between Gemini 2.5 Pro and GPT-5. Among the 254 lemmas that were extracted by Gemini and labeled as self-contained by Gemini, GPT-5 confirmed 72% and rejected 28%. Manual inspection showed that 100 % of those lemmas judged self contained by GPT-5 are truly self contained and 24% of those judged by GPT5 as not self-contained lemmas were in fact truly self-contained, indicat-

ing that GPT-5 acts as a conservative validator, producing almost no false positives but a non-negligible number of false negatives. These results show that the two models implement substantially different implicit criteria for self-containedness. Given that GPT-5 seemed to behave as a higher-precision classifier we chose GPT-5 as main extractor for the benchmark.

Results Full-context retrieval succeeds in complementing a larger number of lemmas by all required definitions and hypotheses to make them self-contained according to LLM-as-judge evaluation, and human evaluation. Accuracy and volume of self contained lemma rightfully retrieved are presented in Figure 3. Overall more than 60% of the lemmas identified by the regex in the initial step are successfully self-contained by the pipeline. Therefore we used full-context retrieval by default when establishing the benchmark.

5. Model Evaluation

We use the resulting benchmark to evaluate frontier LLMs on research-level lemma proving tasks. Specifically, we use a structured, step-wise format for proof presentation and independent LLM evaluators for judging correctness.

Generator. The candidate LLM *prover* is queried to produce a structured proof in numbered steps, with subgoals and the proof explicitly stated. We enforce minimal constraints: no invocation of undeclared external references; hypotheses must be cited at the moment of use; known theorems must be clearly announced when applied.

Independent LLM-as-a-judge. The lemma, its assumptions and the generated proof are presented to a separate LLM judge, which evaluates the overall proof.

Domain Stratification We stratify by domain, following the `math.*` classification on the arXiv (AG, AP, PR, NT, GR, MG, OC, FA, etc.). Our dataset was built from articles uploaded within a single week. We present here three iterations of the benchmark: one performed in September 2025 corresponding to articles submitted the last week of August 2025, another one in February 2026 corresponding to articles submitted the first week of February 2026, followed by one batch collecting articles submitted on the last week of April 2026. [3] . For each domain, we sampled 1 to 5 articles from the target week, with the exact selection guided by a mix of random choice and manual curation to ensure topical diversity. Table 6 reports the distribution for the lemma extracted from the last week of August 2025.

Evaluation: Metrics, Protocols, and Tables To evaluate the pipeline and compare models, we define several metrics covering extraction quality, proof validity, and human assessment.

Main metrics.

- **Self-Containment Pass** (SC)(%): percentage of lemmas validated as self-contained (`self_contained=1`). This reflects the robustness of the extraction pipeline and coherence judge. In our experiments, values above 60% are considered a satisfactory quality threshold.
- **Precision (PPV) — true self-contained pass (%):** Proportion of lemmas flagged as self-contained that are confirmed as such by human validation. This is the *precision* (positive predictive value) of the self-contained detector, i.e., how much we can trust the set of "pass" lemmas produced by a given method.
- **Proof Acceptance**: proportion of self-contained lemmas that were correctly proven by the LLM, tested at pass@1.
- **Human Confidence Score**: percentage of proof judged correct by the LLM-as-a-judge that are also judged correct by human experts.

For the week of August 2025, we extracted 376 lemmas. On average across models, about 240 (64%) of these are judged self-contained after the pipeline. On this benchmark of 240 elements, 7-12.3% are correctly proven depending on the model tested. The evaluation of the human score was performed on 10-20 such proofs per model by expert mathematicians (for a total of 44 proofs reviewed), chosen from reviewers of journals from different fields of mathematics. The proof judged correct by the LLM-as-a-judge were vetted correct by human standards in $67-83\%$ of the cases

Source	Gen→Judge	Proof Accept. (%)	Human confidence score (%)
Full-ctx (GPT-5)	GPT-5→GPT-5	12.3	83
Full-ctx (Gem-2.5)	Gem-2.5→Gem-2.5	7	67
Full-ctx (GPT-5)	DeepSeek-R1→DeepSeek-R1	11.9	80

Table 3. **Proof quality - September 2025** by source and generator → judge for the first iteration (week of August 2025, experiment performed in September 2025).

For the first week of February 2026, we select 100 preprints from which we extract a total of 677 lemmas appearing in 81 preprints using GPT-5. Out of these, 358 lemmas (52.8%) were judged as self-contained by GPT-5 after full-context definition and hypothesis extraction by GPT-5 and deterministic filtering. On the resulting benchmark 106 lemmas (29.6%) are correctly proven according to the most critical judge (GPT-5) with the best model (also GPT-5).

[3] We use the most recent article version regardless of the original submission

Generator	Judge	# Acc.	Rate (%)
GPT-5.5	GPT-5.5	**49**	**40.8**
	Gemini 3.1	79	65.8
	DeepSeek-R1	**62**	**51.7**
Gemini 3.1	GPT-5.5	37	30.8
	Gemini 3.1	**83**	**69.2**
	DeepSeek-R1	53	44.2
DeepSeek-R1[a]	GPT-5.5	30	27.3
	Gemini 3.1	57	51.8
	DeepSeek-R1	45	40.9

[a] DeepSeek-R1 failed to produce a parseable proof for 10 of the 120 lemmas; acceptance rates for this generator are computed over the 110 successfully generated proofs.

Table 4. **Proof quality – Batch of April 2026.** Three LLMs as generators, three as whole-proof judges. All proofs use filtered lemmas extracted by GPT-5.5. Experiment performed in May on lemmas extracted on the last week of April.

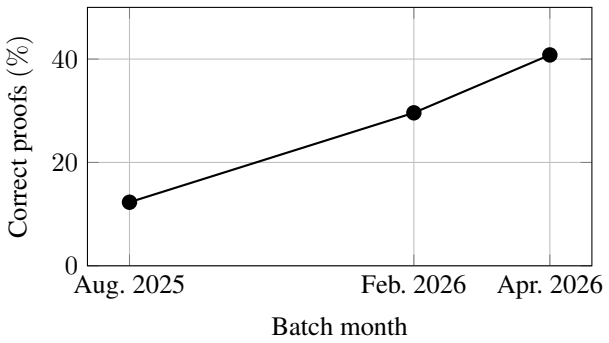

Figure 2. **Evolution of proof quality of the best performing model with time:** GPT-5 in Aug. 2025 and Feb. 2026, and GPT-5.5 in April 2026, according to the most critical judge (same model).

For the batch completed on the last week of April, we select 31 preprints for a total of 287 lemmas using GPT-5.5. Out of these, 120 lemmas remain, of which 49 lemmas (40.8%) are correctly proven according to the most critical judge (GPT-5.5) with the best model (also GPT-5.5). The results for the latest iteration of the benchmark in April 2026 are shown in Table 4 and the evolution with time of the performances of the best model (GPT-5 / GPT-5.5) is shown in figure 2. While evaluation of the human score is not yet available for these iterations of the benchmark, with the improving capabilities of general public LLMs in mathematics it seems reasonable to expect that the confidence score would be at least of the order of the one of September 2025.

Step-by-step evaluation We also performed an alternative experiment where the LLM-as-a-judge evaluates each step of the overall proof with a binary verdict (`True`, `False`), and labels the whole proof as correct if and only if all steps are correct. This mode of evaluation is likely too strict as the LLM-as-a-judge has a tendency to reject proofs with imprecisions that would be insignificant to humans (see Appendix G table 7 & H for an example) but can be used as a stricter lower bound and compare the models between each other. The results on Feb. 2026 can be found in Table 5. While the opinion of different LLMs judges may differ, GPT-5 is still seen by all judges as the best performing model.

Source	Generator	Judge	# Accepted	Rate (%)
Full-ctx (GPT-5)	GPT-5	GPT-5	**55**	**15**
		Gemini 3 pro	**67**	**19**
		Claude Opus 4.5	**125**	**35**
Full-ctx (GPT-5)	Gemini 3 pro	GPT-5	12	3
		Gemini 3 pro	57	16
		Claude Opus 4.5	104	29
Full-ctx (GPT-5)	Opus 4.5	GPT-5	6	2
		Gemini 3 pro	50	14
		Claude Opus 4.5	60	17

Table 5. **Proof quality – February 2026.** Acceptance counts and rates over 358 candidate lemmas, with three LLMs as generators and three LLMs as whole-proof judges. All proofs generated under full-context conditions from lemmas extracted by GPT-5.

6. Discussion

This paper presents a new approach to evaluate LLM mathematical capabilities in natural language. It consists in a live benchmark built directly from the latest research-level mathematical results proved by mathematicians on the arXiv. A key advantage is that it reduces the risk of data contamination over time, and avoids treating narrow, convenience-curated problem collections, whether competition exercises or research questions selected primarily for easy verifiability, as a proxy for mathematical research, where, by definition, data is relatively scarce (in contrast with mathematical competition exercises where in-distribution synthetic data has been massively generated in the last two years). The price to pay for this generality is the use of an LLM-as-a-judge to evaluate the qualities of the self-contained lemmas and the quality of the proof provided. Nevertheless, our evaluation by human mathematicians shows the reliability of LLM-as-a-judge in evaluating both the self-containedness of a statement and the validity of a proof. This can be explained as it is usually easier for a model to evaluate the validity of a proof than to actually produce a proof. The theorem proving capabilities of LLMs and their reliability as judges should improve as models get better in reasoning. We believe this is an additional motivation to use the type of benchmark introduced in this article. The quantitative results reported here reveal informative trends, and should be considered preliminary given the limited sample size (a few hundred problems) and the small number of reviewers (a dozen human mathematicians), and as an incitation to design such benchmarks directly grounded in the latest

research without restricting to specific problems. We plan to extend the evaluation protocol with more reviewers and more examples to improve the statistical robustness of the analysis.

Code Availability

The initial datasets and code are publicly available at LemmaBench.

Acknowledgements

The authors would like to thank Etienne Bernard, Urbain Vaes, Vincent Boulard, Alexandre Krajenbrinck, Ilan Zysman, David Renard, Baptiste Seraille, Thomas Cavallazzi, Hector Bouton, Gabriel Synnaeve, Claudio Dappiaggi.
This work was granted access to the HPC resources of IDRIS under the allocation 2024-AD011014527R1 made by GENCI. AH and AP were supported by the ANR-Tremplin StarPDE ANR-24-ERCS-0010, the PEPS Project JCJC 2025 from CNRS - INSMI and the Hi! PARIS and ANR/France 2030 program (ANR-23-IACL-0005).

References

Abouzaid, M., Blumberg, A. J., Hairer, M., Kileel, J., Kolda, T. G., Nelson, P. D., Spielman, D., Srivastava, N., Ward, R., Weinberger, S., et al. First proof. *arXiv preprint arXiv:2602.05192*, 2026.

Alexander, L., Leonen, E., Szeto, S., Remizov, A., Tejeda, I., Inchiostro, G., and Ilin, V. Semantic search over 9 million mathematical theorems. *arXiv preprint arXiv:2602.05216*, 2026.

Alfarano, A., Charton, F., and Hayat, A. Global lyapunov functions: a long-standing open problem in mathematics, with symbolic transformers. *Advances in Neural Information Processing Systems*, 37:93643–93670, 2024.

Amini, A., Gabriel, S., Lin, P., Koncel-Kedziorski, R., Choi, Y., and Hajishirzi, H. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

Aram H. Markosyan, Gabriel Synnaeve, H. L. Leanuniverse: A library for consistent and scalable lean4 dataset management. 2024.

Bai, Y., Lv, X., Zhang, J., Lyu, H., Tang, J., Huang, Z., Du, Z., Liu, X., Zeng, A., Hou, L., et al. Longbench: A bilingual, multitask benchmark for long context understanding. In *Proceedings of the 62nd annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 3119–3137, 2024.

Becker, L., de Frutos-Fernández, M. I., Diedering, L., van Doorn, F., Gouëzel, S., Jamneshan, A., Karunus, E., van de Meent, E., Monticone, P., Mulder-Sohn, J., et al. A blueprint for the formalization of carleson's theorem on convergence of fourier series. 2025.

Calvo Cortes, V., Stump, C., and Sturmfels, B. Benchmarks in leipzig, 2026. URL `https://math.sciencebench.ai/benchmarks/benchmarks-in-leipzig`. Hosted at the Max Planck Institute for Mathematics in the Sciences. A benchmark problem set of research-level problems compiled by 49 researchers to test large language models in mathematics research.

Castelvecchi, D. AI models solve maths problems at level of top students. *Nature*, 644(7):1, 2025.

Charton, F., Ellenberg, J. S., Wagner, A. Z., and Williamson, G. Patternboost: Constructions in mathematics with a little help from ai. *arXiv preprint arXiv:2411.00566*, 2024.

Chen, E., Cummins, C., Grubisic, D., Haller, L., Hong, L., Kurghinyan, A., Lau, K., Leather, H., Lee, S., Markosyan,

A., et al. Fel's conjecture on syzygies of numerical semigroups. *arXiv preprint arXiv:2602.03716*, 2026.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., et al. Advancing mathematics by guiding human intuition with ai. *Nature*, 600(7887):70–74, 2021.

De Santis, M., Eichfelder, G., and Porcelli, M. Objective-function free multi-objective optimization: Rate of convergence and performance of an adagrad-like algorithm. *arXiv preprint arXiv:2602.05893*, 2026.

Diez, C.-P., da Maia, L., and Nourdin, I. Mathematical research with gpt-5: a malliavin-stein experiment. *arXiv preprint arXiv:2509.03065*, 2025.

Dixit, A., Liang, T., and Telang, J. Project aletheia: Verifier-guided distillation of backtracking for small language models. *arXiv preprint arXiv:2601.14290*, 2026.

Fang, M., Wan, X., Lu, F., Xing, F., and Zou, K. Math-odyssey: Benchmarking mathematical problem-solving skills in large language models using odyssey math data. *Scientific Data*, 12(1):1392, 2025.

Feng, T., Trinh, T., Bingham, G., Kang, J., Zhang, S., Kim, S.-h., Barreto, K., Schildkraut, C., Jung, J., Seo, J., et al. Semi-autonomous mathematics discovery with gemini: A case study on the erd\h {o} s problems. *arXiv preprint arXiv:2601.22401*, 2026.

Gao, B., Song, F., Yang, Z., Cai, Z., Miao, Y., Dong, Q., Li, L., Ma, C., Chen, L., Xu, R., et al. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*, 2024.

Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C. F., Denain, J.-S., Ho, A., de Oliveira Santos, E., Järviniemi, O., Barnett, M., Sandler, R., Vrzala, M., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., Enugandla, S. V., and Wildon, M. Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai. 2025. URL https://arxiv.org/abs/2411.04872.

Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Huang, Y. and Yang, L. F. Gemini 2.5 pro capable of winning gold at imo 2025. *arXiv preprint arXiv:2507.15855*, 2025.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jang, U. and Ryu, E. K. Point convergence of nesterov's accelerated gradient method: An ai-assisted proof. *arXiv preprint arXiv:2510.23513*, 2025.

Li, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S., Rasul, K., Yu, L., Jiang, A. Q., Shen, Z., et al. Numinamath: The largest public dataset in ai4maths with 860k pairs of competition math problems and solutions. *Hugging Face repository*, 13(9):9, 2024.

LI, J., Beeching, E., Tunstall, L., Lipkin, B., Soletskyi, R., Huang, S. C., Rasul, K., Yu, L., Jiang, A., Shen, Z., Qin, Z., Dong, B., Zhou, L., Fleureau, Y., Lample, G., and Polu, S. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf), 2024.

Liu, C., Shen, J., Xin, H., Liu, Z., Yuan, Y., Wang, H., Ju, W., Zheng, C., Yin, Y., Li, L., et al. Fimo: A challenge formal dataset for automated theorem proving. *arXiv preprint arXiv:2309.04295*, 2023.

Liu, J., Zhou, Z., Zhu, Z., Santos, M. D., He, W., Liu, J., Wang, R., Xie, Y., Zhao, J., Wang, Q., et al. Numina-lean-agent: An open and general agentic reasoning system for formal mathematics. *arXiv preprint arXiv:2601.14027*, 2026.

Ma, J., Wang, G., Feng, X., Liu, Y., Hu, Z., and Liu, Y. Eternalmath: A living benchmark of frontier mathematics that evolves with human discovery. *arXiv preprint arXiv:2601.01400*, 2026.

MathArena Team. Arxivmath: Evaluating llms on mathematical research problems from recent arxiv papers. https://matharena.ai/arxivmath/, 2026. URL https://matharena.ai/arxivmath/.

Patel, N., Saha, R., and Flanigan, J. A new approach towards autoformalization. *arXiv preprint arXiv:2310.07957*, 2023.

Rastogi, A., Jiang, A. Q., Lo, A., Berrada, G., Lample, G., Rute, J., Barmentlo, J., Yadav, K., Khandelwal, K., Chandu, K. R., et al. Magistral. *arXiv preprint arXiv:2506.10910*, 2025.

Schmitt, J. Extremal descendant integrals on moduli spaces of curves: An inequality discovered and proved in collaboration with ai. *arXiv preprint arXiv:2512.14575*, 2025.

Son, G., Kim, S., Arnett, C., Ko, H., Lee, H., Kang, H., Longxi, J., Yun, J., Lee, J., Lee, K., et al. Soohak: A mathematician-curated benchmark for evaluating research-level math capabilities of llms. *arXiv preprint arXiv:2605.09063*, 2026.

Sothanaphan, N. Resolution of erd\h {o} s problem# 728: a writeup of aristotle's lean proof. *arXiv preprint arXiv:2601.07421*, 2026.

Sun, H., Min, Y., Chen, Z., Zhao, W. X., Fang, L., Liu, Z., Wang, Z., and Wen, J.-R. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *arXiv preprint arXiv:2503.21380*, 2025.

Team, T. H. Aristotle: Imo–level automated theorem proving. *arXiv preprint arxiv:2510.01346v1*, Oct 2025. URL https://arxiv.org/abs/2510.01346.

Tsoukalas, G., Lee, J., Jennings, J., Xin, J., Ding, M., Jennings, M., Thakur, A., and Chaudhuri, S. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *Advances in Neural Information Processing Systems*, 37:11545–11569, 2024.

Washburn, J. and Zlatanović, M. Uniqueness of the canonical reciprocal cost. *arXiv preprint arXiv:2602.05753*, 2026.

Xu, R., Wang, Z., Fan, R.-Z., and Liu, P. Benchmarking benchmark leakage in large language models. *arXiv preprint arXiv:2404.18824*, 2024.

Ying, H., Wu, Z., Geng, Y., Wang, J., Lin, D., and Chen, K. Lean workbook: A large-scale lean problem set formalized from natural language math problems. *Advances in Neural Information Processing Systems*, 37:105848–105863, 2024.

Zhang, J., Petrui, C., Nikolić, K., and Tramèr, F. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *arXiv preprint arXiv:2505.12575*, 2025.

Zheng, K., Han, J. M., and Polu, S. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

A. Extended Related works

With the increasing capabilities of LLM and LLM-based approaches in mathematics, several benchmarks have been developed in recent years. In natural language, one can cite for instance GSM8k (Cobbe et al., 2021), MathQA (Amini et al., 2019) or MATH (Hendrycks et al., 2021) which provided middle to high-school level questions. These benchmarks were often used to evaluate LLM capabilities to the point of being the staple of evaluation for mathematical abilities. This was until most models were able to reach very high to nearly perfect accuracies (Gao et al., 2024). These datasets regularly faced debate on contamination issues (see for instance (Xu et al., 2024)). Other datasets such as OlymMATH (Sun et al., 2025) later proposed problems up to college level coming from multilingual sources of exercises. A very large dataset of competition problems was created by the Numina project (LI et al., 2024). OmniMath introduced challenging Olympiad level problems (Gao et al., 2024). More recent examples like MathOdyssey (Fang et al., 2025) introduced a benchmark, still on high school to undergraduate level, curated by experts to limit the risk of data contamination. Nevertheless, given the accessibility of the dataset, contamination issues may arise in future generations of LLMs. FrontierMath (Glazer et al., 2025) had a similar expert-curated problems approach and to avoid any contamination issues, the dataset is purely closed-source. A larger similar benchmark was released with more than 400 problems curated by 64 mathematicians (Son et al., 2026), closed-source for the year 2026. Other benchmarks were recently released, such as First Proof (Abouzaid et al., 2026) where the statements of ten expert-curated questions with no publicly available solutions are provided during one-week for a one time evaluation of LLMs. Leipzig Tier-4 Benchmark present 102 research-level problems curated by mathematicians in open-source (Calvo Cortes et al., 2026). RealMath (Zhang et al., 2025) extract or convert problems from arXiv papers with verifiable answers, so as to focus on real human research too, see also (MathArena Team, 2026) that has a monthly update. While the approach is similar to ours, in particular for (MathArena Team, 2026)[4], it differs in that it restricts itself to problems that can be inherently formulated with a verifiable answers. This explains why it gives rise to less problems (40 problems over 2 months of arXiv against 400 lemmas for 2 weeks of arXiv) and why the performances of state-of-the-art LLMs differ (60% on the benchmark of (MathArena Team, 2026) compared to around 15% for our). EternalMath (Ma et al., 2026) is yet another similar approach where the questions are selected so as to propose an executable benchmark that can be code-verified.

It is worth mentioning that other benchmarks exist in formal language, also focusing on middle school to undergraduate level problems such as miniF2F (Zheng et al., 2021), FIMO (Liu et al., 2023), Lean-Workbook (Ying et al., 2024), PutnamBench (Tsoukalas et al., 2024).

[4]The first iteration of (MathArena Team, 2026) was released while we were preparing a revision of this manuscript.

B. Table of percentage results relative to the total lemmas retrieve for each method (A) and (B) and LLM

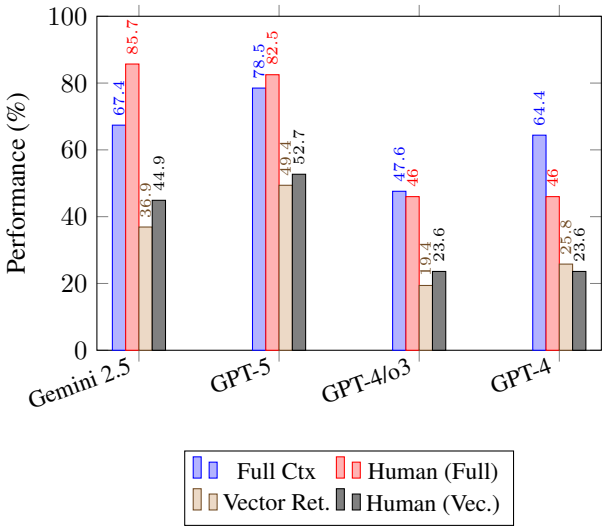

Figure 3. Percentage results relative to the total lemmas retrieved for both extraction modes (full context / vector retrieval) and models

C. Breakdown of lemmas per domain of mathematics

Domain code	Domain label	# Articles	#Lemmas
cs.IT	Computer Science	3	24
math-ph	Mathematical Physics	2	36
math.AP	Analysis of PDEs	3	46
math.CO	Combinatorics	5	60
math.CT	Category Theory	1	22
math.DG	Differential Geometry	1	9
math.DS	Dynamical Systems	2	18
math.FA	Functional Analysis	1	15
math.GR	Group Theory	1	5
math.GT	Geometric Topology	1	6
math.MG	Metric Geometry	1	2
math.NA	Numerical Analysis	2	13
math.NT	Number Theory	2	2
math.OC	Optimization and Control	4	19
math.PR	Probability	3	51
math.QA	Quantum Algebra	1	2
math.RT	Representation Theory	1	16
math.SG	Symplectic Geometry	1	2
math.ST	Statistics Theory	2	28

Table 6. Domain distribution in the dataset (one-week snapshot, August).

D. System Overview: Directed Acyclic Graph of the Pipeline

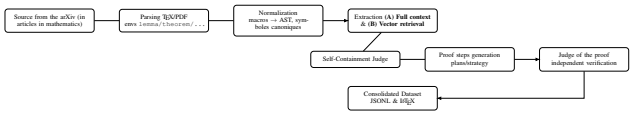

Figure 4. DAG : input → parsing → normalization → extraction (double mode) → self contained → proof steps → judge proofs → dataset.

E. Algorithms (Pseudo-Code)

Algorithm 1 Extraction & Self-Containment (Full context / vector retrieval)

Require: Article TEX/PDF, mode $\in$ {`full_context`, `vector_retrieval`}
for each `lemma` environment detected **do**
  $S \leftarrow$ **statement**(`lemma`)
  if mode = full_context **then**
    $C \leftarrow$ all paragraphs before lemma
  else
    $C \leftarrow$ **retrieve_topk**(S) paragraphs { embeddings}
  end if
  $(\hat{H}, \hat{D}) \leftarrow$ **LLM_extract**(S, C)
  $ok \leftarrow$ **SelfContainmentChecks**$(S, \hat{H}, \hat{D})$
  if ok **then**
    emit $(S, \hat{H}, \hat{D}, \texttt{self_contained} = 1)$
  else
    emit $(S, \hat{H}, \hat{D}, \texttt{self_contained} = 0)$
  end if
end for

Algorithm 2 Step Generation and Judgment

Require: (S, H, D) with `self_contained` $= 1$
steps $\leftarrow$ **LLM_prove**(S, H, D)
verdict $\leftarrow$ **LLM_judge_proof**(whole proof, H, D)
if verdict_ok **then**
  emit steps, verdict
else
  emit steps, verdict {partial or rejected}
end if

F. Example of a retrieved self contained lemma, with proof generated by gpt-5

Assumptions

- $L^2_{\alpha-1}(0,T;U) := \{u : (0,T) \to U \text{ measurable} \mid \int_0^T (T-s)^{\alpha-1}\|u(s)\|_U^2\, ds < \infty\}$, with inner product

$$\langle u, v\rangle_{L^2_{\alpha-1}(0,T;U)} := \int_0^T (T-s)^{\alpha-1}\langle u(s), v(s)\rangle_U\, ds.$$

  Assumptions: $\alpha \in (0,1)$, $T > 0$, $U = \mathbb{R}^m$.

- $L^2(\mathfrak{S}, H; \mu) := \{f : \mathfrak{S} \to H \text{ measurable} \mid \int_{\mathfrak{S}} \|f(\sigma)\|_H^2\, d\mu(\sigma) < \infty\}$. Assumptions: $(\mathfrak{S}, \mathcal{F}, \mu)$ is a probability space, $H = \mathbb{R}^n$.

- $\mathfrak{L}_T : L^2_{\alpha-1}(0,T;U) \to L^2(\mathfrak{S}, H; \mu)$ defined by

$$(\mathcal{L}_T u)(\sigma) := \int_0^T (T-s)^{\alpha-1} E_{\alpha,\alpha}\big((T-s)^\alpha A(\sigma)\big)\, B(\sigma)\, u(s)\, ds = \int_0^T \mathcal{K}_T(\sigma, s)\, u(s)\, ds,$$

  with kernel

$$\mathcal{K}_T(\sigma, s) := (T-s)^{\alpha-1} E_{\alpha,\alpha}\big((T-s)^\alpha A(\sigma)\big) B(\sigma) \in \mathbb{R}^{n\times m},$$

  and two-parameter Mittag-Leffler matrix function

$$E_{\alpha,\alpha}(M) := \sum_{k=0}^{\infty} \frac{M^k}{\Gamma(\alpha k + \alpha)}.$$

Lemma F.1. *$\mathcal{L}_T : L^2_{\alpha-1}(0,T;U) \to L^2(\mathfrak{S}, H; \mu)$ is a Hilbert–Schmidt operator.*

F.1. Proof steps

Identification of the domain as a standard L^2-space. Define the finite measure ν on $(0,T)$ by $d\nu(s) := (T-s)^{\alpha-1}\, ds$. Then

$$L^2_{\alpha-1}(0,T;U) = L^2((0,T), \nu; U).$$

With this notation the operator reads, for $u \in L^2((0,T), \nu; U)$,

$$(\mathcal{L}_T u)(\sigma) = \int_0^T (T-s)^{\alpha-1} E_{\alpha,\alpha}((T-s)^\alpha A(\sigma)) B(\sigma) u(s)\, ds = \int_0^T K_T(\sigma, s) u(s)\, ds,$$

where $K_T(\sigma, s) = (T-s)^{\alpha-1} E_{\alpha,\alpha}((T-s)^\alpha A(\sigma)) B(\sigma)$.

Rewriting as an integral operator with respect to ν. Since $d\nu(s) = (T-s)^{\alpha-1} ds$, we have $ds = (T-s)^{1-\alpha} d\nu(s)$. Therefore

$$(\mathcal{L}_T u)(\sigma) = \int_0^T \big[E_{\alpha,\alpha}((T-s)^\alpha A(\sigma)) B(\sigma)\big] u(s)\, d\nu(s).$$

Thus $\mathcal{L}_T$ is the integral operator with kernel

$$\tilde{K}(\sigma, s) := E_{\alpha,\alpha}((T-s)^\alpha A(\sigma)) B(\sigma) \in \mathbb{R}^{n\times m}.$$

Uniform bound on the Mittag-Leffler matrix factor. Let $M_A := \operatorname{ess\,sup}_{\sigma\in\mathfrak{S}} \|A(\sigma)\|$. Using the series expansion,

$$\|E_{\alpha,\alpha}((T-s)^\alpha A(\sigma))\| \le \sum_{k=0}^{\infty} \frac{(T^\alpha M_A)^k}{\Gamma(\alpha k + \alpha)} =: C_E < \infty.$$

L2-integrability of the kernel $\tilde{K}$. Identify $HS(U,H)$ with $\mathbb{R}^{n\times m}$ with Frobenius norm. Then

$$\|\tilde{K}(\sigma,s)\|_{HS} \le C_E \|B(\sigma)\|_{HS}.$$

Let $M_B := \operatorname{ess\,sup}_{\sigma\in\mathfrak{S}} \|B(\sigma)\|_{HS} < \infty$. Then

$$\int_{\mathfrak{S}} \int_0^T \|\tilde{K}(\sigma,s)\|_{HS}^2 \, d\nu(s)\, d\mu(\sigma) \le C_E^2 \frac{T^\alpha}{\alpha} M_B^2 < \infty.$$

Hilbert–Schmidt property. By the standard result, $\mathcal{L}_T$ is Hilbert–Schmidt and

$$\|\mathcal{L}_T\|_{HS}^2 \le \frac{T^\alpha}{\alpha} C_E^2 M_B^2.$$

Conclusion. $\mathcal{L}_T : L^2_{\alpha-1}(0,T;U) \to L^2(\mathfrak{S},H;\mu)$ is Hilbert–Schmidt, with

$$\|\mathcal{L}_T\|_{HS}^2 \le \frac{T^\alpha}{\alpha} C_E^2 \int_{\mathfrak{S}} \|B(\sigma)\|_{HS}^2 \, d\mu(\sigma).$$

G. Step-by-step judging

Proof model	Valid proofs	Pass rate
gpt-5.5	25/120	20.8%
gemini-3.1-pro	10/120	8.3%
gpt-5.4-mini	4/120	3.3%

Table 7. Step-by-step proof evaluation (judge: gpt-5.5), on (120 lemmas) judged self contained on the batch of April 2026. A proof passes if all of its steps are judged correct.

H. Over-strictness of the harshest judge when judging step-by-step

Across our Feb. 2026 batch, the three judges differ sharply in leniency when judging the proof step-by-step. Aggregated over all proofs each inspects, OPUS-4.5 accepts 26.2%, GEMINI-3 accepts 15.8%, and GPT-5 accepts only 6.3%: **GPT-5 is by far the strictest judge**, and it is the model most often in the minority when judges disagree. The example below is representative: GPT-5 rejects a *correct* proof because one step contains a single incorrect *parenthetical aside* that plays no role in the argument—a flaw GPT-5 itself describes as "not used later"—whereas GEMINI-3 and OPUS-4.5 accept the proof.[5]

The lemma. Let (X,G) and (Y,G) be equicontinuous minimal actions of a discrete group G on compact Hausdorff spaces, with uniformities $\mathbb{U}_X, \mathbb{U}_Y$ given by the neighbourhoods of the diagonals. For an entourage δ write $\delta[z] := \{z' : (z',z) \in \delta\}$, and for a point z and a set B write the hitting-time set $G(z,B) := \{g \in G : g.z \in B\}$.

[5]Lemma and proof extracted from arXiv:2602.05756 (Feb. 2026); the proof was generated by GPT-5.

Lemma H.1. *For $x \in X$ and $y \in Y$, there exists a factor map $\pi\colon X \to Y$ with $\pi(x) = y$ if and only if for every neighbourhood V of y there is a neighbourhood U of x with $G(x,U) \subseteq G(y,V)$.*

The candidate proof (generated by GPT-5). The model produced the proof below, which we submitted to each judge one step at a time. The single questionable clause—an unjustified, and ultimately unused, assertion in Step 3—is highlighted.

Candidate proof of Lemma H.1 (generated by GPT-5).

Step 0. We recall the setting. Let G be a discrete group. Let (X,G) and (Y,G) be equicontinuous minimal actions on compact Hausdorff spaces, with $g.x := \alpha_X(g)(x)$ and $g.y := \alpha_Y(g)(y)$. Denote by $\mathbb{U}_X$ and $\mathbb{U}_Y$ the uniformities given by the neighborhoods of the diagonals in X^2 and Y^2, respectively. For a minimal action (Z,G), a point $z \in Z$, and $B \subseteq Z$, define the hitting times set

$$G(z,B) := \{g \in G;\ g.z \in B\}.$$

We prove:

Lemma. Let $x \in X$ and $y \in Y$. There exists a factor map $\pi\colon X \to Y$ with $\pi(x) = y$ if and only if for any neighborhood V of y there exists a neighborhood U of x such that $G(x,U) \subseteq G(y,V)$.

Step 1. Proof of the easy direction ($\Rightarrow$). Assume there is a factor map $\pi\colon X \to Y$ with $\pi(x) = y$, i.e., π is continuous, surjective, and equivariant: $\pi(g.x) = g.\pi(x)$ for all $g \in G$, $x \in X$.

Let $V \subseteq Y$ be a neighborhood of y. By continuity of π at x, there exists a neighborhood $U \subseteq X$ of x such that $\pi(U) \subseteq V$. If $g \in G(x,U)$, then $g.x \in U$, so $\pi(g.x) \in V$. By equivariance,

$$\pi(g.x) = g.\pi(x) = g.y \in V,$$

hence $g \in G(y,V)$. Therefore $G(x,U) \subseteq G(y,V)$.

Step 2. Proof of the converse ($\Leftarrow$): stabilizer inclusion and definition on the orbit. Assume that for every neighborhood V of y there exists a neighborhood U of x with

$$G(x,U) \subseteq G(y,V).$$

First, we show the stabilizer inclusion $\mathrm{Stab}_G(x) \subseteq \mathrm{Stab}_G(y)$.

Indeed, if $h \in G$ satisfies $h.x = x$, then for any neighborhood V of y, choose U given by the hypothesis. Since $h.x = x \in U$, we have $h \in G(x,U)$, and thus $h \in G(y,V)$, so $h.y \in V$. As this holds for every neighborhood V of y, it follows that $h.y = y$. Hence

$$\mathrm{Stab}_G(x) \subseteq \mathrm{Stab}_G(y).$$

Define a map φ on the dense orbit $G.x \subseteq X$ by

$$\varphi(g.x) := g.y \quad \text{for } g \in G.$$

This is well-defined: if $g_1.x = g_2.x$, then $g_2^{-1}g_1 \in \mathrm{Stab}_G(x) \subseteq \mathrm{Stab}_G(y)$, hence $g_1.y = g_2.y$.

Moreover, φ is G-equivariant on $G.x$: for $h \in G$,

$$\varphi\big(h.(g.x)\big) = \varphi((hg).x) = (hg).y = h.(g.y) = h.\varphi(g.x).$$

Step 3. Uniform continuity of φ on $G.x$. We show that $\varphi\colon G.x \to Y$ is uniformly continuous with respect to the uniformities $\mathbb{U}_X$ and $\mathbb{U}_Y$ restricted to $G.x$ and Y, respectively.

Fix $\varepsilon \in \mathbb{U}_Y$. By equicontinuity of (Y, G), there exists $\delta_Y \in \mathbb{U}_Y$ such that for all $(y_1, y_2) \in \delta_Y$ and all $r \in G$,

$$(r.y_1, r.y_2) \in \varepsilon.$$

Let $V := \delta_Y[y] = \{y' \in Y;\ (y', y) \in \delta_Y\}$, a neighborhood of y with $V \times V \subseteq \delta_Y$.

By the hypothesis, choose a neighborhood U of x such that $G(x, U) \subseteq G(y, V)$.

Choose an entourage $E \in \mathbb{U}_X$ with $E[x] \subseteq U$. By equicontinuity of (X, G), there exists $\eta_X \in \mathbb{U}_X$ such that for all $(u, v) \in \eta_X$ and all $r \in G$,

$$(r.u, r.v) \in E.$$

We claim: if $g, h \in G$ satisfy $(g.x, h.x) \in \eta_X$, then $(\varphi(g.x), \varphi(h.x)) = (g.y, h.y) \in \varepsilon$.

Indeed, applying $r = h^{-1}$ to $(g.x, h.x) \in \eta_X$, we get

$$\big(h^{-1}g.x,\ h^{-1}h.x\big) = \big(h^{-1}g.x,\ x\big) \in E,$$

thus $h^{-1}g.x \in E[x] \subseteq U$, i.e., $h^{-1}g \in G(x, U) \subseteq G(y, V)$. Hence $h^{-1}g.y \in V$, which means $(h^{-1}g.y, y) \in \delta_Y$. Applying $r = h$ and the choice of δ_Y, we obtain

$$(g.y, h.y) = \big(h.(h^{-1}g.y),\ h.y\big) \in \varepsilon.$$

This establishes the desired uniform continuity of φ on $G.x$.

Step 4. Extension to a continuous map $\pi\colon X \to Y$. Since $G.x$ is dense in X (by minimality) and Y is compact Hausdorff (hence complete and Hausdorff as a uniform space), the uniformly continuous map $\varphi\colon G.x \to Y$ extends uniquely to a uniformly continuous (hence continuous) map

$$\pi\colon X \longrightarrow Y.$$

By construction, $\pi|_{G.x} = \varphi$, so in particular $\pi(x) = \varphi(x) = y$.

Step 5. Equivariance and surjectivity of π. For each fixed $h \in G$, the maps $X \to Y$, $z \mapsto \pi(h.z)$ and $z \mapsto h.\pi(z)$ are continuous and agree on the dense set $G.x$ (since φ is equivariant there). Hence they agree on all of X, and thus π is G-equivariant:

$$\pi(h.z) = h.\pi(z) \quad \text{for all } h \in G,\ z \in X.$$

Finally, π is surjective. Indeed, for any $g \in G$, $\pi(g.x) = g.\pi(x) = g.y$, so $\pi(G.x) \supseteq G.y$. Since (Y, G) is minimal, $\overline{G.y} = Y$. The image $\pi(X)$ is compact (hence closed) and contains the dense subset $G.y$, so $\pi(X) = Y$.

Thus π is a factor map with $\pi(x) = y$.

Step 6. Conclusion. We have shown: - If $\pi\colon X \to Y$ is a factor map with $\pi(x) = y$, then for every neighborhood V of y there exists a neighborhood U of x such that $G(x, U) \subseteq G(y, V)$. - Conversely, if for every neighborhood V of y there exists a neighborhood U of x with $G(x, U) \subseteq G(y, V)$, then there exists a continuous, equivariant, surjective map $\pi\colon X \to Y$ with $\pi(x) = y$.

This completes the proof of Lemma H.1.

How the three judges scored it. GEMINI-3 and OPUS-4.5 accept all seven steps and accept the proof; GPT-5 accepts six and rejects *Step 3*, hence rejects the whole proof (Table 8).

	GEMINI-3	GPT-5	OPUS-4.5
Step 0 (setup)	✓	✓	✓
Step 1 ($\Rightarrow$ direction)	✓	✓	✓
Step 2 ($\Leftarrow$: stabilizers)	✓	✓	✓
Step 3 (uniform continuity)	✓	×	✓
Step 4 (extension to X)	✓	✓	✓
Step 5 (equivariance, surjectivity)	✓	✓	✓
Step 6 (conclusion)	✓	✓	✓
Overall verdict	**accept**	**reject**	**accept**

Table 8. Per-step verdicts on the *same* proof of Lemma H.1. The only disagreement is Step 3, which GPT-5 alone rejects.

Why the rejection is unwarranted. The highlighted clause $V \times V \subseteq \delta_Y$ need not hold for an arbitrary entourage δ_Y, so it is indeed unjustified. **But it is never used.** In Step 3, V is used only through the hypothesis (to obtain U with $G(x, U) \subseteq G(y, V)$), and the entourage δ_Y is applied *directly* via equicontinuity at the end; deleting the clause leaves Steps 3–5 unchanged. The proof is therefore correct, as GEMINI-3 and OPUS-4.5 confirm.

The judge concedes the proof is correct, then rejects it. GPT-5's verbatim rationale grants that the *main argument is correct* and that the offending clause is *not used later*, yet converts this immaterial aside into a rejection of the entire proof:

GPT-5 — Step 3 — emitted verdict: `incorrect`

The main argument establishing uniform continuity is correct*, but the claim that $V := \delta_Y[y]$ satisfies $V \times V \subset \delta_Y$ is generally false and unjustified. $\delta_Y[y]$ is a neighborhood of y, but need not have $V \times V \subset \delta_Y$.* ***Although this inclusion is not used later****, the step contains an incorrect assertion, so it must be marked incorrect.*

Takeaway. *This failure differs from a judge that simply miscalculates: the local observation is correct—the parenthetical clause really is unjustified—but the step-level protocol has no notion of* materiality. *A single incorrect aside that contributes nothing to the deduction fails the step, and one failed step fails the whole proof. Holistic judging, which weighs whether the proof* as a whole *establishes the claim, does not exhibit this brittleness. Because* GPT-5 *is both the strictest judge and the most frequent lone rejecter, this bias disproportionately depresses the accept rates it reports; cross-judge disagreement is common, with split overall verdicts on* 20–35% *of proofs depending on the prover.*