Open Mathematical Problems as an AI Reasoning Benchmark

Przemyslaw Chojecki

ulam.ai

January 31, 2026

Abstract

Open mathematical problems provide a unique stress test for AI reasoning: there is no known solution to memorize, and progress requires search, synthesis, proof construction, error detection, and formal verification. We introduce the *UnsolvedMath* dataset as a benchmark for research-level mathematical work and evaluate frontier models (GPT-5.2, Gemini 3 Pro, Grok, Opus 4.5) across four dimensions: literature search, theorem proving, proof checking, and formalization. Key findings: (i) all models achieve strong literature search but none can autoformalize; (ii) GPT-5.2 Extended Thinking performs at PhD-student level on simpler problems; (iii) extended thinking time (20+ minutes) is the primary driver of reasoning quality; (iv) three capabilities—global mental picture, systematic self-verification, and analogical reasoning—remain absent and represent the frontier for autonomous mathematical AI.

Introduction

Mathematical reasoning has become a central arena for evaluating modern large language models (LLMs). Most widely-used benchmarks, however, contain problems with known solutions. This makes it difficult to rule out shortcut learning and training contamination, and it collapses a wide range of research behaviours into a single pass/fail label.

Open problems invert this dynamic: there is no accepted solution to overfit, and evaluation must focus on *intermediate* research behaviours. This paper discusses the *UnsolvedMath* dataset ([1]) and proposes a practical rubric for using open mathematical problems as an AI reasoning benchmark. We evaluate frontier models (GPT-5.2, Gemini 3 Pro, Opus 4.5, Grok) on PhD-level mathematical reasoning, with particular focus on Erdős open problems [2] [3], which got a lot of attention lately with AI-generated proofs and formal verifications.

The Landscape of AI Mathematical Reasoning

Benchmark Saturation and the Need for Open Problems

The trajectory of AI performance on mathematical benchmarks reveals both remarkable progress and fundamental limitations. Table 1 summarizes the current state.

The saturation of traditional benchmarks creates two problems. First, it becomes impossible to distinguish between models or measure incremental progress. Second, high scores may reflect memorization or pattern matching rather than genuine reasoning. As Terence Tao remarked after reviewing FrontierMath: "These are extremely challenging. I think that in the near term, basically the only way to solve them is by a combination of a semi-expert... paired with some combination of a modern AI and lots of other algebra packages" [6].

Benchmark	Best Model	Score	Human Baseline
GSM8K	GPT-5.2 Pro	99.2%	∼95%
MATH	Gemini 3 Pro	94%	∼90%
AIME 2025	GPT-5.2	100%	Median: 4–6/15
GPQA Diamond	GPT-5.2 Pro	93.2%	PhD: 65–74%
FrontierMath	GPT-5.2	40.3%	Hours–days
PutnamBench	Hilbert	70.0%	Top students

Table 1: Benchmark saturation across mathematical reasoning tasks. Traditional benchmarks are near-saturated, while research-level benchmarks reveal substantial gaps. Sources: OpenAI [23], Google [24], Epoch AI [6], Apple [9].

The Reasoning Model Revolution

A fundamental shift occurred in late 2024 with OpenAI's o1 and accelerated in January 2025 with DeepSeek-R1 [5]. These "reasoning models" differ from traditional LLMs in three key ways:

1. **Extended inference-time computation**: Rather than generating responses token-by-token, reasoning models engage in prolonged "thinking" phases that can extend to minutes or hours.

2. **Emergent reasoning behaviours**: DeepSeek-R1-Zero, trained via pure reinforcement learning without supervised fine-tuning, spontaneously developed self-verification, reflection, and backtracking—what the authors term an "aha moment" [5].

3. **Chain-of-thought as training signal**: Models are rewarded for correct final answers, not for specific reasoning patterns, allowing novel problem-solving strategies to emerge.

The DeepSeek-R1 paper demonstrates that on AIME 2024, pass@1 accuracy increased from 15.6% to 77.9%

through RL training alone, with response lengths naturally increasing as problems demanded more computation [5]. This "thinking time" scaling—where models improve by generating more intermediate reasoning tokens—has become a key axis of progress.

However, recent work from Anthropic raises concerns about chain-of-thought faithfulness [13]. In controlled experiments, reasoning models used embedded hints to arrive at answers but verbalized those hints less than 20% of the time. This suggests that the "reasoning" we observe may not fully reflect the model's internal computation, with implications for interpretability and safety monitoring.

Formal Verification as Ground Truth

The integration of LLMs with formal proof assistants has emerged as perhaps the most promising path toward verified mathematical reasoning. Key developments include:

AlphaProof [7]: Google DeepMind's system achieved silver-medal performance at IMO 2024 by combining LLM-generated proof candidates with verification in Lean 4. The key insight was using autoformalization to generate training data: natural language problems are translated to formal statements, and successful proofs provide reward signals for RL.

Aristotle [8]: This system achieved gold-medal equivalent performance on IMO 2025 by integrating informal reasoning (for generating high-level proof strategies and lemmas) with formal proof search in Lean 4. Notably, both Aristotle and DeepMind's Seed-Prover reached gold-medal performance simultaneously using different architectures—step-wise tactic generation versus whole-proof generation—suggesting multiple viable paths.

Hilbert [9]: Apple's framework achieves 99.2% on miniF2F and 70.0% on PutnamBench (462/660 problems) by recursively decomposing problems into subgoals that are solved by either informal reasoning or formal proving, with verifier feedback guiding refinement.

APOLLO [10]: This agentic framework achieves 84.9% on miniF2F with sub-8B parameter models by combining LLM proof generation with automated syntax repair, sublemma isolation, and iterative verification.

These results suggest that formal verification provides the "verifiable reward" signal that RL-based systems need to develop genuine mathematical reasoning, analogous to how game outcomes provide rewards for game-playing AI [7].

Dataset of Open Mathematical Problems

The saturation of traditional mathematical benchmarks creates a fundamental evaluation crisis. When models achieve 99% on GSM8K and 100% on AIME, we lose the ability to measure progress, distinguish capabilities, and identify remaining weaknesses. More critically, high performance on problems with known solutions cannot rule out sophisticated pattern matching against training data—a concern validated by the dramatic performance drop on FrontierMath, where even the best models solve fewer than half of research-level problems [6].

Open mathematical problems offer a principled solution to both challenges. By definition, there is no solution to memorize, no answer key that could leak into training corpora. Evaluation must therefore focus on *process* rather than *outcome*: the quality of literature synthesis, the soundness of intermediate reasoning, the ability to identify and correct errors, and the capacity to translate informal arguments into verifiable formal proofs. These process-oriented metrics capture precisely the capabilities that matter for AI systems to become genuine research partners rather than sophisticated lookup tables.

Furthermore, open problems provide a natural curriculum of difficulty that will not saturate as models improve. Unlike fixed benchmarks that become obsolete once solved, the space of unsolved mathematics is effectively infinite and self-renewing. A model that solves Problem A today faces Problem B tomorrow—and the mathematical community continuously generates new challenges at the frontier of human knowledge. This creates a benchmark that grows with AI capabilities rather than being left behind.

The UnsolvedMath dataset

We introduce *UnsolvedMath*, a curated collection of open mathematical problems distributed as a public dataset on Hugging Face [1]. Unlike FrontierMath, which uses privately held problems with numerical answers for automated verification, UnsolvedMath emphasizes *research-like workflows* on publicly accessible problems where partial progress, proof strategies, and intermediate results can be meaningfully evaluated.

Each entry is designed as a self-contained research starting point: a problem statement, minimal mathematical context, difficulty estimate, and (when available) references to prior partial results or related techniques. The dataset supports multiple evaluation modes of increasing complexity:

1. **Diagnostic prompts**: Short tasks testing specific capabilities (e.g., "identify the relevant technique family," "find related solved problems").

2. **Research episodes**: Extended sessions where models propose approaches, develop proof sketches, and validate intermediate claims (e.g., "propose and critique three potential proof strategies").

3. **Tool-augmented episodes**: Integration with symbolic algebra systems (SymPy, Mathematica), numerical computation, or formal proof assistants (Lean, Coq) to verify claims and explore conjectures.

The most significant evaluation target—and the most challenging—is genuine proof generation for open problems. While full solutions remain rare, we argue that measuring *verifiable partial progress* provides a more informative signal than binary pass/fail metrics on problems with known answers.

The current release contains 1,146 problems across 12 mathematical domains. The core is 632 Erdős problems—the largest machine-readable collection available. Additional problem sets include the Millennium Prize Problems, Hilbert's 23 Problems, Smale's 18 Problems, DARPA's 23 Mathematical Challenges, and Ben Green's 100 Open Problems in additive combinatorics. Problems are classified on a five-level difficulty scale and by domain, with number theory (43%), graph theory (19%), and combinatorics (17%) most represented. All statements include LaTeX notation, historical context, and references to partial results.

Benchmark tasks and judging criteria

We propose four primary judging dimensions:

- **Literature search** (0–5). Scale: 0 (non-existent) to 5 (human-level). Assesses correct retrieval of relevant prior art, accurate attribution, and synthesis into a viable approach.
- **Proving theorems** (0–5). Assesses the ability to produce correct lemmas, proof sketches, or complete proofs for *subclaims* derived from an open problem, with verification against injected or retrieved subgoals.
- **Proof check** (0–5). Assesses whether the model can identify major gaps or invalid steps in a proposed proof and (if possible) suggest repairs.
- **Formalization** (0/1). Assesses whether the model can translate the statement (and selected lemmas) into a formal system (e.g. Lean/Coq/Isabelle) with correct definitions and type-checking.

A key principle is **verifiability**: proving and proof-checking should be anchored to checkable subgoals (known lemmas, special cases, toy instances, or proof-assistant fragments). In addition, literature-search scoring should explicitly penalize fabricated citations.

Lessons from AI-assisted Erdős problems

A tractable subpart of UnsolvedMath is a collection of Erdős problems that also gained recently a lot of attention thanks to AI proofs as well as autoformalizations in Lean.

Genuine AI Milestone was achieved on Problem #728. GPT-5.2 produced a largely autonomous solution after feedback on an initial attempt. No previous work on this problem existed—confirmed as genuinely new. The initial proof contained minor errors and lacked context. Aristotle AI automatically corrected errors and produced a Lean-certified proof. A subsequent ChatGPT session generated a polished, publication-ready article with literature references.

This pattern—GPT-5.2 generating a proof, Aristotle AI verifying formally—has since repeated for Erdős problems #124, #397, and #729. Additionally, many genuinely new partial results have been obtained on various other problems.

A community summary of AI contributions to Erdős problems ([3]) suggests emerging patterns in how AI tools contribute to open-problem progress. AI involvement in mathematics appears on 4 different levels:

1. **Completely autonomous short solutions** to problems that largely follow a standard technique (often close to existing literature).
2. **AI-assisted modifications of existing solutions** that upgrade partial results, optimize parameters, or simplify arguments.
3. **Complex human–AI interactions** where AI tools provide crucial calculations or proofs of key steps, enabling moderately novel solutions.
4. **Primarily human research papers** where AI is used for secondary tasks such as code, numerics, references, or figures.

This spectrum motivates our benchmark design: rather than demanding full solutions, we score models for identifying relevant technique classes, making correct local improvements, delivering verifiable intermediate results that unlock human progress, and accelerating auxiliary research work.

Table 2 summarizes our systematic evaluation of frontier models across multiple dimensions. Our tests on dozens of Erdős problems (including #238, #397, #421, #729, #886, #1139 – see [2]) reveal a consistent pattern: models perform well on literature search, reliably identifying relevant papers and prior partial results. On theorem proving, results are mixed—models can generate informal proof sketches, identify relevant techniques, and reduce problems to simpler forms, but often produce arguments with subtle gaps or circular reasoning. Proof checking shows similar inconsistency; models sometimes catch errors but also validate flawed proofs. The starkest limitation is autoformalization: GPT-5.2, Gemini 3, Grok, and Opus 4.5 all scored zero, producing no complete machine-verifiable formalizations. We are unaware of any published results demonstrating this capability in general-purpose models. Autoformalization currently requires specialized systems like Aristotle AI, which combines reinforcement learning with Monte Carlo tree search specifically trained for Lean proof generation.

Despite these limitations, clear specializations emerged:

- **Constructive reasoning**: GPT-5.2 Extended Thinking and Pro modes generate elaborate, mostly correct arguments from a single prompt—often the best starting point for any problem.
- **Proof checking**: Grok excels at identifying errors and gaps in proofs generated by other models, making it valuable for critical review.
- **Formalization**: Aristotle AI successfully converts informal proof sketches into verified Lean code, and can occasionally work directly from problem statements.

This suggests an optimal pipeline: GPT-5.2 for generation, Grok for verification, Aristotle for formalization.

Criterion	GPT-5.2	Gemini 3 Pro	Grok	Opus 4.5
Literature search (0–5) Scale: 0 (non-existent) to 5 (human-level). Assesses correct retrieval of relevant prior art, accurate attribution, and synthesis into a viable approach.	5	5	5	5
Proving theorems (0–5) Assesses the ability to produce correct lemmas, proof sketches, or complete proofs for *subclaims* derived from an open problem, with verification against injected or retrieved subgoals.	3.5	2	2.5	2
Proof check (0–5) Assesses whether the model can identify major gaps or invalid steps in a proposed proof and (if possible) suggest repairs.	2	1	3	1
Formalization (0/1) Ability to produce machine-verifiable Lean code.	0	0	0	0

Table 2: Evaluation of Frontier Models on Advanced Mathematical Reasoning Capabilities

Analogy as a missing capability

A common view in mathematical practice is that progress often comes from transferring structure between problems. A quote attributed to Stefan Banach captures this succinctly:

> *Good mathematicians see analogies. Great mathematicians see analogies between analogies.*

Through this lens, current LLMs are not yet at the level of a *good* mathematician: they rarely discover useful analogies unaided. However, they can often be guided toward productive analogical moves—in a manner reminiscent of PhD students—when a human provides scaffolding (candidate analogies, similar theorems, or a technique family) and the model fills in intermediate details or explores variations.

The most capable reasoning model currently available, GPT-5.2 Extended Thinking (Table 2), performs roughly at the level of a strong PhD student. It can solve easier open problems when directed toward a potential strategy—much as a PhD advisor guides a student toward productive approaches. It also fills in gaps when shown a proof outline. However, this capability is limited to simpler problems or those admitting short solutions. Current models do not maintain a long-term view of an argument, do not automatically self-verify (verifying only when explicitly prompted after generating a proof), and consequently cannot produce a consistent mathematical paper longer than approximately five pages.

The key enabler of successful mathematical reasoning appears to be *extended thinking time*. GPT-5.2 excels here, with thinking sessions exceeding 20 minutes in Extended Thinking mode and over an hour in Pro Extended Thinking mode. As with human research, longer deliberation allows exploration of more strategic routes and deeper analysis of each.

Three capabilities remain missing if current models are to progress from PhD-student performance to that of working mathematicians:

- **Global mental picture.** Models do not maintain the comprehensive mental representation that mathematicians hold when pursuing a specific strategy. This leads to inconsistencies, especially in arguments with complicated technical details across multiple sections. Correcting error A often introduces error B, while fixing error B reintroduces error A—a failure mode familiar from LLM-managed codebases.

- **Systematic self-verification.** Models perform micro-level self-checks but lack consistent verification at each reasoning step. This produces a characteristic failure pattern: a model presents a "solution," then immediately identifies a critical flaw when prompted to review it—a flaw that invalidates the entire argument.

- **Analogical reasoning.** Current models do not seek analogous proofs from other areas of mathematics. Working mathematicians routinely borrow ideas across fields—as in Ngô Bao Châu's celebrated proof in number theory, which imported techniques from physics [4]. This cross-domain transfer remains absent in LLM reasoning.

Each limitation appears individually tractable. Extended thinking time combined with carefully structured prompts could, in principle, maintain a global mental picture. Self-verification can be addressed in agentic settings through systematic checking after each incremental step. Analogical reasoning can currently be elicited through human steering—often a single hint to "borrow ideas from this paper" or "consider techniques from this field" suffices.

The genuine difficulty lies in solving all three simultaneously. An autonomous AI mathematician must maintain coherent global structure, verify each step without external prompting, and discover cross-domain analogies—all within a single reasoning episode. This integration, rather than any individual capability, represents the current frontier.

Discussion

Open-problem benchmarks require careful design. The primary risk is rewarding confident fabrication: models generate plausible-looking proofs with invented lemmas, circular reasoning, or fictitious references. We therefore recommend pairing *UnsolvedMath* episodes with verification layers—unit tests for special cases, symbolic algebra checks, or proof-assistant fragments—and requiring explicit uncertainty quantification. This risk is acute for users without PhD-level training, who may not recognize subtle errors or unstated assumptions. A proof that "looks right" to a non-expert may contain fundamental flaws visible only to specialists.

This highlights a broader principle for LLM-assisted research: verification must be built into the workflow, not treated as optional. Mathematics offers an ideal test domain because formal proof assistants provide objective ground truth—a theorem either type-checks in Lean or it does not. The emergence of autoformalization systems like Aristotle suggests this verification paradigm could extend to other sciences. Physics, chemistry, and biology lack proof assistants, but analogous verification infrastructure—automated experiment replication, simulation consistency checks, formal specification of theories—may eventually play a similar role.

One behavioral quirk deserves attention: models often perform worse, or refuse entirely, when a problem is labeled "open." This appears to trigger excessive caution or learned helplessness. Overcoming this requires either removing such labels from prompts or explicitly providing candidate strategies for the model to evaluate. The framing "test whether this approach works" consistently outperforms "solve this open problem."

A second challenge is that "difficulty" is multidimensional. Some problems require a missing conceptual insight; others demand long technical chains with no single hard step. Our rubric separates *search and synthesis* (literature review), *constructive reasoning* (proof generation), and *critical reasoning* (verification), enabling more informative analysis than a single score. Even so, capturing difficulty fully requires tracking both quantity and quality of partial results—and "quality" remains inherently subjective, dependent on mathematical taste and assessments of what constitutes genuine progress.

The path forward is collaboration at two levels. First, mathematicians and AI researchers must work together to improve the models themselves: mathematicians contribute problem structure, evaluation criteria, and verification expertise; AI researchers bring training methods, architecture design, and scaling insights. Second, mathematicians working *with* models—providing high-level guidance, correcting misinterpretations, validating outputs—can achieve results neither could reach alone. The Erdős #728 solution exemplifies this partnership: human feedback clarified an ambiguous problem statement, AI generated the core proof strategy, and formal verification confirmed correctness. The result was genuinely new mathematics. This collaborative model, rather than fully autonomous AI, represents the most productive near-term paradigm.

Finally, the lack of analogical reasoning suggests a concrete research direction: benchmark episodes should include *analogy prompts* (e.g., "identify a similar solved problem," "map this to a known technique family"). Measuring performance with and without such guidance would distinguish autonomous reasoning from coached reasoning—and clarify how much of current AI mathematical capability depends on human steering versus genuine insight.

Methods (summary)

1. Dataset interface. We treat each *UnsolvedMath* entry as a benchmark seed. For each seed we derive (a) a short-context version of the statement, (b) a set of related known lemmas/results to retrieve, and (c) a set of verifiable subgoals (toy cases, parameter regimes, or literature lemmas).

2. Episode protocol. Each model run is an "episode": literature search $\rightarrow$ proposed approach $\rightarrow$ intermediate lemma attempts $\rightarrow$ proof check of its own output $\rightarrow$ optional formalization. Episodes are time- and tool-budgeted to support fair comparison across systems.

3. Verification layer. Subgoals are checked via deterministic tests (algebraic identities, numerics, finite cases) and/or proof-assistant fragments. Proof-check items include injected flawed proofs to measure error detection.

4. Scoring. Literature search is scored on a 0–5 scale; proof-check is scored 0–5; formalization is scored 0/1 (not able/able). Proving is scored on a 0–5 scale based on the number and difficulty of verified subgoals, with additional credit for generalization beyond retrieved templates.

5. Data and code availability. The *UnsolvedMath* dataset is publicly available ([1]).

6. Limitations. Our evaluation has several limitations. First, the Erdős problems tested represent a specific slice of mathematics (combinatorics, number theory); generalization to other fields remains untested. Second, human evaluation introduces subjectivity despite reasonable inter-rater reliability. Third, model capabilities evolve rapidly; scores reported here reflect January 2026 checkpoints.

References

[1] P. Chojecki, ulam.ai. *UnsolvedMath* dataset on Hugging Face. `https://huggingface.co/datasets/ulamai/UnsolvedMath` (accessed 26 Jan 2026).

[2] T. Bloom, Erdős Problems `https://www.erdosproblems.com`

[3] teorth and contributors. AI contributions to Erdős problems (community summary). `https://github.com/teorth/erdosproblems/wiki/AI-contributions-to-Erd%C5%91s-problems` (accessed 26 Jan 2026).

[4] Ngô B. Châu, Le lemme fondamental pour les algèbres de Lie, `https://arxiv.org/abs/0801.0446`

[5] DeepSeek-AI. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

[6] E. Glazer, E. Erdil, T. Besiroglu, et al. FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024.

[7] Google DeepMind. AlphaProof: Formal Mathematical Reasoning via Reinforcement Learning. Technical report, 2024.

[8] Harmonic AI. Aristotle: IMO-level Automated Theorem Proving. *arXiv preprint arXiv:2510.01346*, 2025.

[9] Apple Machine Learning Research. Hilbert: Recursively Building Formal Proofs with Informal Reasoning. In *NeurIPS MATH-AI Workshop*, 2025.

[10] NeurIPS. APOLLO: Automated Proof Repair via LLM and Lean Collaboration. In *NeurIPS*, 2025.

[11] A. Kumarappan, M. Tiwari, P. Song, et al. LeanAgent: Lifelong Learning for Formal Theorem Proving. In *ICLR*, 2025.

[12] K. Yang et al. LeanDojo: Theorem Proving with Retrieval-Augmented Language Models. In *NeurIPS*, 2023.

[13] Anthropic. Reasoning Models Don't Always Say What They Think. Technical report, 2025.

[14] OpenAI. OpenAI o3 System Card. Technical report, 2025.

[15] LLM-Stats. AIME 2025 Leaderboard. `https://llm-stats.com/benchmarks/aime-2025`, 2025.

[16] T. Johnson et al. Can Large Language Models Generalize Analogy Solving Like Children Can? *arXiv preprint arXiv:2411.02348*, 2025.

[17] X. Guan et al. MetaLadder: Ascending Mathematical Solution Quality via Analogical-Problem Reasoning Transfer. *arXiv preprint arXiv:2503.14891*, 2025.

[18] A. Novikov et al. AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery. Technical report, Google DeepMind, 2025.

[19] DARPA. Exponentiating Mathematics (expMath) Program. `https://www.darpa.mil/news/2025/math-ai-tomorrows-breakthroughs`, 2025.

[20] Renaissance Philanthropy. AI for Math Fund Announces $18 Million in Grants. Press release, September 2025.

[21] Google DeepMind. Accelerating Discovery with the AI for Math Initiative. `https://blog.google/technology/google-deepmind/ai-for-math/`, 2025.

[22] DeepSeek-AI. DeepSeek-Prover-V2: Formal Theorem Proving in Lean 4. Technical report, 2025.

[23] OpenAI. Introducing GPT-5.2. `https://openai.com/index/introducing-gpt-5-2/`, December 2025.

[24] Google DeepMind. Gemini 3: Introducing the latest Gemini AI model from Google. `https://blog.google/products/gemini/gemini-3/`, November 2025.

[25] D. Rein et al. GPQA: A Graduate-Level Google-Proof Q&A Benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

[26] K. Cobbe et al. Training Verifiers to Solve Math Word Problems. *arXiv preprint arXiv:2110.14168*, 2021.