ErdosBench: A Research-Mathematics Benchmark Built from Synthetic Erdős-Style Problems

ulam.ai

June 8, 2026

Abstract

Open mathematical problems are a natural stress test for advanced AI reasoning because progress requires literature search, theorem proving, error detection, computation, and sometimes formal verification rather than memorization. This white paper extends the evaluation thesis of *Open Mathematical Problems as an AI Reasoning Benchmark* [1] to a focused product benchmark: ErdosBench, a curated benchmark family built from synthetic Erdős-style candidate problems. The v0.1 source corpus contains 226 merged candidate problems. From these, we define 389 benchmark items across literature/novelty triage, proof-gap detection, exact finite-proxy solving, and research-progress tasks. We separate public demonstration material from private evaluation and holdout material, and we emphasize throughout that the source entries are *candidate* problems, not certified open problems without specialist review.

We also report a public-source smoke evaluation on fourteen problems from two public ulam.ai notes [2, 3]. In our audit, Codex outperformed Claude on decisive mathematical progress, with accepted smoke-level solved claims on Problems 1, 4, 5, and 7; Claude was more conservative and stronger on analytic-number-theory caveats. The experiment illustrates the intended use of ErdosBench: not merely binary scoring, but diagnosis of research behaviours such as finding hidden obstructions, applying known theorems, checking proof gaps, and avoiding overclaiming.

Contents

1 Motivation: from open-problem benchmarking to ErdosBench

The companion ulam.ai paper argues that open problems invert the usual benchmark dynamic: there is no accepted solution to memorize, so evaluation must focus on intermediate research behaviours such as synthesis, proof construction, error detection, and formalization [1]. That framing remains the foundation here. But a product-grade benchmark also needs controlled releases, repeatable scoring, leakage controls, and tasks that can be judged without requiring the evaluator to settle an open problem.

Existing research-mathematics benchmarks highlight complementary design choices. FrontierMath uses original, expert-vetted, mostly unpublished problems with automated verification to reduce contamination risk [4]. Riemann-Bench is a private benchmark of 25 expert-curated research-level problems with double-blind expert verification and programmatic verifiers [5]. SWE-bench Verified is not a mathematics benchmark, but it is instructive as a product benchmark because it shows the value of human validation and a filtered subset for reliable evaluation [6].

ErdosBench takes a different but compatible position. It is not a claim that 226 new open problems have been certified. Instead, it is a structured benchmark for measuring whether models can behave like useful mathematical research assistants: identify literature risk, detect proof gaps, solve finite proxies, produce correct local lemmas, run computations, and avoid inflated claims.

1.1 The core design principle

The benchmark distinguishes four things that are often conflated:

1. **A source candidate**: a synthetic Erdős-style problem statement with anchors, curation notes, and proof-program scaffolding.
2. **A public problem**: a statement intentionally released for marketing, community discussion, and smoke testing.
3. **A benchmark item**: a concrete task derived from a source candidate, with an answer format and scoring rule.
4. **A private holdout item**: a benchmark item whose statement, answer key, or verifier is not released publicly.

This separation lets ErdosBench be both clickable and credible: the public layer demonstrates mathematical flavour, while the private layer preserves evaluation value.

2 Source corpus

The v0.1 corpus is based on `AI-ERDOS-FINAL.json`, a merged set of 226 candidate problems. The source metadata records that the final merge used a strict 100-problem core, cleaned additions, rewritten refinements, expanded candidates, and 220-only additions. The metadata also

explicitly states the central caveat: no entry should be advertised as a certified open problem solely from the file; final openness requires specialist review.

Table 1: Curation buckets in the 226-problem source corpus.

Bucket	Count	Interpretation
Strict core	100	Second-pass keepers; closest to the intended Erdős-style target.
Cleaned potential additions	38	Added after cleaning; requires specialist literature check.
Rewritten refinements	42	Close variants rewritten/refocused to avoid duplicate claims.
Expanded top-30 candidates	30	Salvageable nonduplicates from the expansion pool.
Merged 220-only additions	16	Source-concept additions present only in the 220-problem expansion.

2.1 Domain coverage

The corpus spans number theory, graph theory, geometry, set systems, groups, probabilistic models, designs, dynamics, and mixed areas. Table 2 gives the split-level domain counts used in the package index. The large "unbucketed" component consists mostly of expansion candidates whose proof-program domain was not normalized in the source metadata; those items are still assigned to release splits and can be normalized in v0.2.

Table 2: Domain summary by release split.

Domain bucket	Total	Public	Dev	Private	Holdout
unbucketed	30	0	7	11	12
additive multiplicative number theory	22	4	4	7	7
probabilistic models	22	0	5	8	9
groups cosets	16	1	3	6	6
complex polynomial interpolation	14	0	3	5	6
multiplicative arithmetic functions	14	1	3	5	5
geometry incidence	14	3	2	4	5
recursive dynamics	14	1	3	5	5
diophantine metric discrepancy	13	1	3	5	4
hypergraphs set systems	11	1	2	4	4
graph ramsey	9	2	2	3	2
set theory compactness	9	0	2	3	4
general extremal	8	0	2	3	3
designs finite geometry	8	0	2	3	3
directed ordered graphs	8	0	2	3	3
additive combinatorics number theory	5	0	1	2	2
geometry incidence distance	3	0	1	1	1
other mixed	3	0	1	1	1

Domain bucket	Total	Public	Dev	Private	Holdout
groups cosets algebraic covers	2	1	1	0	0
multiplicative divisor functions	1	0	1	0	0

3 Benchmark item design

ErdosBench converts source candidates into tasks that can be judged without pretending that the full source problem has been solved. The four tracks are:

Literature and novelty triage. The model classifies a source candidate as solved/false, a direct duplicate, a refinement requiring specialist review, a candidate that passed internal audit but is not certified, or a candidate needing further check. Scoring penalizes fabricated citations and overconfident novelty claims.

Proof-gap detection. The model receives a plausible proof sketch and must identify the first fatal gap, a missing lemma, or a hidden assumption. This directly measures a failure mode visible in frontier systems: confident acceptance of attractive but invalid arguments.

Finite-proxy exact solving. The model solves a closed finite task derived from the candidate problem, often requiring search, coding, optimization, or a certificate. These items are the closest analogue of FrontierMath-style automated verification.

Research progress. The model is asked for an extended research attempt: propose approaches, prove local lemmas, run computations, check counterexamples, and state remaining gaps. These are manually graded with a rubric for verified progress rather than a binary pass/fail label.

Table 3: Default scorecard for an aggregate ErdosBench report.

Track	Weight	What counts
Finite-proxy exact solving	40%	Correct final answer or valid certificate checked by code.
Proof-gap detection	20%	Precise identification of a fatal mathematical gap.
Literature/novelty triage	20%	Correct risk classification, anchored prior art, no hallucinated citations.
Research progress	20%	Human-audited lemmas, computations, reductions, or complete solutions.

4 Release structure: what is public and what remains private

The public layer should be large enough to make the benchmark understandable and clickable, but small enough to protect the leaderboard. The public GitHub repository for this white paper releases only the fourteen public-source smoke problems from the two ulam.ai notes, plus the runner schema and smoke-run artifacts needed to reproduce the comparison. It does not release the full 226-problem source corpus, the dev/private/holdout manifests, private statements, exact finite-proxy answer keys, or private verifier logic.

Table 4: Public repository release boundary.

Layer	Public count	Release status
Smoke problem statements	14	Released publicly for reproducible examples, debugging, and community runs.
Smoke result artifacts	14 results per run	Released as public baseline diagnostics, not as private leaderboard evidence.
Runner/schema files	minimal	Released to let users produce comparable JSONL results.
Dev/private/holdout statements	0	Withheld from the public repository.
Answer keys and verifier internals	0	Withheld; distributed only through hosted or contractual evaluation.

Public material. Fourteen source problems from the two public ulam.ai notes are in the final 226-problem corpus and were used for the Claude/Codex smoke test. These are public-source and must not be treated as confidential holdout evidence. They are the only mathematical problem statements intended for the initial public GitHub repository.

Private material. The dev, private-eval, and holdout splits should not be dumped publicly with answers. For sales and beta evaluations, a customer can see the dev set and receive private score reports on the private-eval split; the holdout split should remain reserved for leaderboard stability and later re-evaluation.

Private verifiers and answer keys. Exact finite-proxy answers, private labels, and grader logic should be distributed only through a hosted evaluation API or under a strict data agreement. Publicly releasing the answer keys would convert the benchmark into training data.

5 Evaluation protocol

A credible ErdosBench run records model version, date, system prompt, temperature, token budget, tool access, wall-clock budget, evaluator version, retry policy, and failure policy. Three modes are recommended:

1. **Closed-book mode**: no web, code allowed, fixed token budget. This measures mathematical reasoning and computation.
2. **Research-agent mode**: web/search/code allowed. This measures tool use, citation hygiene, and literature triage.
3. **Long-horizon mode**: multiple independent runs per item. This estimates whether a model can eventually find the decisive move, not merely whether it finds it on pass@1.

For exact finite-proxy items, pass@1 and pass@k are appropriate. For proof-gap and research-progress items, independent human grading remains necessary. For public reporting, we recommend giving aggregate results and selected examples, while withholding private statements, solutions, and verifier internals.

6 Claude versus Codex: public-source smoke evaluation

We ran and audited two independent smoke attempts on the fourteen public-source problems from the two ulam.ai notes. This was not a private leaderboard run. It was designed to test whether the public problems are useful for diagnosing model behaviours.

6.1 Run provenance

The Claude run artifact is labelled `smoke_independent_claude-opus-4-8_2026-06-08` and reports 14 results: 1 solved and 13 partial. The Codex run artifact is labelled `smoke_self_2026-06-08` and reports 14 results: 4 solved and 10 partial, with Problem 10 conservatively left partial pending an independent lemma audit. Appendix C lists file hashes for the source and run artifacts.

Table 5: Smoke-run verdict counts.

Run	Solved	Partial	Solved problem numbers
Claude	1	13	7
Codex	4	10	1, 4, 5, 7
Audit-accepted solved claims	4	–	1, 4, 5, 7

6.2 Accepted smoke-level solved claims

Table 6 lists the four claims accepted by the audit. These should still receive normal mathematical review before being marketed as theorems; the point here is that the benchmark surfaced decisive, checkable mathematical moves rather than merely polished speculation.

Table 6: Accepted smoke-level solved claims.

#	Problem	Source	Audit conclusion
1	GCD-Sidon sets	Codex	Proposed $N^{1/2+o(1)}$ scale is false; smoke audit accepts $G(N) \le 1 + \max_{n \le N} \tau(n) = N^{o(1)}$.
4	Coprime graph representation	Codex	Correct order is $\exp(\Theta(n \log n))$, not $2^{\Theta(n)}$.
5	Squarefree divisor graph	Codex	Exact formula $\chi(D_N) = r(N) + 1$, where $r(N)$ is the largest primorial index with $p_1 \cdots p_r \le N$.
7	Pairwise coprime Beatty subsequences	Both	$C_\alpha(X) = (1/\alpha + o(1))\pi(X)$ via Vinogradov equidistribution and smallest-prime-factor partition.

6.3 Problem-level comparison

Table 7: Claude vs. Codex problem-level audit. Scores are internal 0–5 audit scores for this smoke run, not official benchmark scores.

#	Problem	Claude	Codex	Winner	Audit judgement
1	GCD-Sidon sets	partial (2.0)	solved (5.0)	Codex	Codex's divisor-function bound gives $G(N) \le N^{o(1)}$ and refutes the proposed square-root scale.
2	LCM-Sidon sets	partial (2.5)	partial (4.5)	Codex	Codex uses the Erdős–Ford multiplication-table bound to rule out positive density; exact order remains open.
3	Prime-intersection families	partial (2.0)	partial (4.0)	Codex	Codex applies Ray–Chaudhuri–Wilson layerwise to get $2^{o(N)}$, ruling out ordinary exponential growth.
4	Coprime representation of graphs	partial (3.0)	solved (5.0)	Codex	Codex gives matching $\exp(\Theta(n \log n))$ bounds, refuting $2^{\Theta(n)}$.
5	Squarefree divisor graph	partial (3.0)	solved (5.0)	Codex	Codex gives the exact formula $\chi(D_N) = r(N) + 1$.
6	Large prime factors of prime-indexed Beatty numbers	partial (2.0)	partial (1.8)	slight Claude	Neither proves the hard analytic claim; Claude gives better smooth-number caution.
7	Pairwise coprime Beatty subsequences	solved (5.0)	solved (5.0)	Tie	Both give the accepted Vinogradov plus smallest-prime-factor proof.
8	Primitive subsets of irrational Beatty sequences	partial (3.5)	partial (4.5)	Codex	Codex gives the stronger second-layer construction; the limit problem remains open.
9	Distinct-prime quotient chains in Beatty sequences	partial (3.0)	partial (3.0)	Tie	Both recover the primorial upper bound; Beatty lower bound remains open.
10	Bohr-prime quotient-free sets	partial (2.5)	partial (3.0)	Split	Codex has a promising density-1/2 route but leaves a finite-prime lemma unaudited; keep partial.
11	Prime reciprocal subset sums modulo one	partial (2.5)	partial (2.8)	slight Codex	Both prove the lower bound; Codex more cleanly records distinctness and Fourier structure.
12	Pairwise non-coprime Beatty values	partial (3.0)	partial (2.5)	Claude	Both give the 1/2 construction; Claude is better about why interval analogues do not transfer.
195	Coprime Schur-free sets	partial (3.5)	partial (3.0)	Claude	Both find the 2/3 construction; Claude has the cleaner residue/stability framing.
208	Non-squarefree Hamming-distance codes	partial (3.0)	partial (3.8)	Codex	Both prove Type-II lower bounds; Codex adds restricted upper-bound and exact small-N checks.

6.4 Skill comparison

The audit was more informative than a single score. Codex was better at finding decisive obstructions and using known theorems; Claude was better at caution, caveats, and not overclaiming in analytic number theory.

Table 8: Qualitative skill comparison from the public-source smoke audit. Scores are subjective 0–5 audit ratings.

Skill	Claude	Codex	Interpretation
Finding decisive obstructions	2.8	4.7	Codex found the hidden one-line or one-theorem moves on Problems 1, 4, and 5.
Using known theorems	3.1	4.4	Codex better exploited divisor maximal order, multiplication tables, and intersection theorems.
Proof hygiene and conservatism	4.4	3.7	Claude avoided overclaiming more reliably; Codex Problem 10 needs audit.
Computational evidence	3.6	4.0	Both supplied checks; Codex had stronger exact small cases in several rows.
Analytic-number-theory caution	4.1	3.5	Claude was stronger on Beatty/smooth-number caveats.
Product/benchmark usefulness	3.7	4.5	Codex produced more decisive, clickable research-progress headlines.

6.5 Interpretation

The smoke test produced three practical conclusions.

1. Public-source synthetic Erdős-style problems can expose meaningful differences between frontier systems, even when the problems are not private holdouts.
2. A benchmark report should not merely count solved problems. It should identify *how* a model made progress: hidden obstruction, known theorem, computation, proof checking, or literature triage.
3. Public problems are valuable for marketing and debugging but not for protected leaderboard claims. The private-eval and holdout splits remain necessary.

7 Product packaging

ErdosBench can be sold or deployed as a benchmark product in four layers.

Private model evaluation. A lab submits model outputs through a hosted runner or API and receives a score report: aggregate score, per-domain score, failure taxonomy, hallucinated-reference report, and selected examples.

Research-agent diagnostic suite. The proof-gap, literature-triage, and research-progress tracks are particularly useful for teams building long-horizon agents. They diagnose whether an agent can self-check, avoid false proofs, and use tools productively.

Verifier and data licensing. Exact finite-proxy items and private answer keys can be licensed under a no-training/no-redistribution agreement or exposed only through a hosted verifier.

Custom benchmark generation. The same pipeline can be offered for other mathematical or scientific domains: generate candidate problems, deduplicate, audit literature risk, derive verifiable finite proxies, and wrap them into private evaluation splits.

8 Conclusion

ErdosBench operationalizes the open-problem benchmark thesis for a sellable data product. It offers public examples for visibility, private splits for reliable evaluation, and a multi-track task design that rewards the behaviours we want in AI research mathematicians: literature awareness, proof hygiene, finite verification, and genuine local progress. The Claude/Codex smoke test shows that even a small public subset can generate useful diagnostic signal. The full value of the benchmark, however, lies in the private verifier-backed and expert-graded layers.

A Crosswalk for the released smoke problems

The public repository releases only the fourteen public-source smoke problems used in the Claude/Codex comparison. Table 9 maps their public-note numbering to the final corpus identifiers.

Table 9: Crosswalk from public-note problem numbers to released smoke problems.

Note	PDF #	PDF title	Final #	Final ID
10-ai-erdos.pdf	1	Coprime Schur-free sets	195	AI-ERDOS-195
10-ai-erdos.pdf	2	GCD-Sidon sets	1	AI-ERDOS-001
10-ai-erdos.pdf	3	LCM-Sidon sets	2	AI-ERDOS-002
10-ai-erdos.pdf	6	Prime-intersection families	3	AI-ERDOS-003
10-ai-erdos.pdf	7	Non-squarefree Hamming-distance codes	208	AI-ERDOS-208
10-ai-erdos.pdf	8	Coprime representation of graphs	4	AI-ERDOS-004
10-ai-erdos.pdf	9	Squarefree divisor graph	5	AI-ERDOS-005
10-ai-erdos2.pdf	1	Large prime factors of prime-indexed Beatty numbers	6	AI-ERDOS-006
10-ai-erdos2.pdf	2	Pairwise coprime subsequences of irrational Beatty sequences	7	AI-ERDOS-007
10-ai-erdos2.pdf	3	Primitive subsets of irrational Beatty sequences	8	AI-ERDOS-008
10-ai-erdos2.pdf	4	Distinct-prime quotient chains in Beatty sequences	9	AI-ERDOS-009
10-ai-erdos2.pdf	6	Bohr-prime quotient-free sets	10	AI-ERDOS-010
10-ai-erdos2.pdf	7	Prime reciprocal subset sums modulo one	11	AI-ERDOS-011
10-ai-erdos2.pdf	10	Pairwise non-coprime families inside Beatty sequences	12	AI-ERDOS-012

B Statements of the fourteen public-source smoke problems

These statements are public-source smoke problems. They are useful for examples, model debugging, and community discussion, but they should not be counted as private holdout items.

195. Coprime Schur-free sets (AI-ERDOS-195)

Let $C(N)$ be the largest size of $A \subseteq [N]$ with no Schur triple $x + y = z$ inside A whose two summands are coprime. Is $C(N) = (2/3 + o(1))N$? If so, prove stability: must every near-extremal A be close to a union of two local divisibility classes such as $\{n : 2 \mid n \text{ or } 3 \mid n\}$?

1. GCD-Sidon sets (AI-ERDOS-001)

Call $A \subseteq [N]$ GCD-Sidon if the values $\gcd(a, b)$ for unordered pairs $a < b$ in A are all distinct. Let $G(N) = \max |A|$. Is $G(N) = N^{1/2+o(1)}$? Does $G(N)/\sqrt{N}$ have a limit?

2. LCM-Sidon sets (AI-ERDOS-002)

Call $A \subseteq [N]$ LCM-Sidon if $\operatorname{lcm}(a, b) = \operatorname{lcm}(c, d)$ for $a < b$ and $c < d$ in A implies $\{a, b\} = \{c, d\}$. Let $L(N) = \max |A|$. Determine $L(N)$; in particular, can $L(N)$ have positive density?

3. Prime-intersection families (AI-ERDOS-003)

Let $\mathcal{F} \subseteq 2^{[N]}$ be prime-intersecting if $|A \cap B|$ is prime for every distinct $A, B \in \mathcal{F}$. Let $P(N) = \max |\mathcal{F}|$. Is $P(N)$ polynomial in N, or can it be exponential?

208. Non-squarefree Hamming-distance codes (AI-ERDOS-208)

Let $H(N)$ be the largest size of a family $\mathcal{F} \subseteq 2^{[N]}$ such that every nonzero Hamming distance $|A \triangle B|$, $A \neq B$, is non-squarefree. Is $H(N) = 2^{N/2+o(N)}$? Determine whether doubly-even self-dual codes are asymptotically optimal among all nonlinear families.

4. Coprime representation of graphs (AI-ERDOS-004)

For a finite graph G, let $\rho(G)$ be the least M such that vertices can be injectively labelled by integers in $[M]$ with $uv \in E(G)$ iff $\gcd(\ell(u), \ell(v)) = 1$. Let $R(n) = \max_{|V(G)|=n} \rho(G)$. Is $R(n) = 2^{\Theta(n)}$?

5. Squarefree divisor graph (AI-ERDOS-005)

Let D_N be the graph on $[N]$ where $a < b$ are adjacent iff $a \mid b$ and b/a is squarefree. Let $\chi(N) = \chi(D_N)$. Is $\chi(N) = \Theta(\log N / \log \log N)$? Determine the sharp constant if possible.

6. Large prime factors of prime-indexed Beatty numbers (AI-ERDOS-006)

For irrational $\alpha > 1$, define $M_\alpha(X) = \max_{p \le X \text{ prime}} P^+(\lfloor \alpha p \rfloor)$. Is $M_\alpha(X) \ge X^{1-o(1)}$? Equivalently, for every $\varepsilon > 0$ are there infinitely many primes p with $P^+(\lfloor \alpha p \rfloor) > p^{1-\varepsilon}$?

7. Pairwise coprime Beatty subsequences (AI-ERDOS-007)

Let $C_\alpha(X) = \max\{|A| : A \subseteq B_\alpha \cap [1, X] \text{ and } \gcd(a, b) = 1 \text{ for all } a \neq b\}$. Is $C_\alpha(X) = (1/\alpha + o(1))\pi(X)$ for every irrational $\alpha > 1$?

8. Primitive subsets of irrational Beatty sequences (AI-ERDOS-008)

For irrational $\alpha > 1$, let $P_\alpha(X)$ be the largest size of a primitive subset of $B_\alpha \cap [1, X]$. Does $\lambda(\alpha) = \lim_{X \to \infty} P_\alpha(X)/X$ exist, and what is it? Is the naive lower value $1/(2\alpha)$ ever sharp?

9. Distinct-prime quotient chains in Beatty sequences (AI-ERDOS-009)

Let $L_\alpha(X)$ be the largest k for which $b_0 \mid b_1 \mid \cdots \mid b_k$ all lie in $B_\alpha \cap [1, X]$ and the quotients b_i/b_{i-1} are distinct primes. Is $L_\alpha(X) = (1 + o(1)) \log X / \log \log X$?

10. Bohr-prime quotient-free sets (AI-ERDOS-010)

Fix irrational α and $0 < \eta < 1/2$. Let $P_{\alpha,\eta} = \{p \text{ prime} : \|\alpha p\| < \eta\}$. Let $F_{\alpha,\eta}(N)$ be the maximum size of $A \subseteq [N]$ with no pair $b = ap$ for $p \in P_{\alpha,\eta}$. Does $F_{\alpha,\eta}(N)/N$ have a limit, and what is it?

11. Prime reciprocal subset sums modulo one (AI-ERDOS-011)

Let $S_N = \{\sum_{p \le N} \varepsilon_p/p \bmod 1 : \varepsilon_p \in \{0, 1\}\}$ and let $\Delta(N) = \sup_{\xi \in \mathbb{R}/\mathbb{Z}} \min_{s \in S_N} \|\xi - s\|$. Is $\Delta(N) = 2^{-(1+o(1))\pi(N)}$, i.e. is the trivial cardinality lower bound essentially sharp?

12. Pairwise non-coprime Beatty values (AI-ERDOS-012)

Let $I_\alpha(N) = \max\{|A| : A \subseteq [1, N] \text{ and } \gcd(\lfloor m\alpha \rfloor, \lfloor n\alpha \rfloor) > 1 \text{ for all distinct } m, n \in A\}$. Is $I_\alpha(N) = (1/2 + o(1))N$ for every irrational $\alpha > 1$? Are near-extremals essentially one fixed prime divisor?

C Run artifact hashes

For reproducibility, the following files were used to prepare this white paper. Hashes are SHA-256 prefixes.

File	SHA-256 prefix	Role
AI-ERDOS-FINAL.json	e079c3c8a095…	226-problem source corpus
two-notes crosswalk	5079afed99c0…	crosswalk from the two public notes
Claude/Codex audit CSV	0daaec6f5e8d…	human audit of problem-level outcomes
Claude summary	0534e2fff2bf…	Claude run summary
Codex summary	4f7e31fd8f37…	Codex run summary

References

[1] Przemyslaw Chojecki. *Open Mathematical Problems as an AI Reasoning Benchmark*. ulam.ai research note, January 2026. https://www.ulam.ai/research/open-math.pdf.

[2] ulam.ai. *Ten New Erdős-style Problems in Number Theory and Combinatorics*. Public research note, 2026. https://www.ulam.ai/research/10-ai-erdos.pdf.

[3] ulam.ai. *Ten AI-Erdős Problems on Beatty Sequences, Primes, and Divisibility*. Public research note, 2026. https://www.ulam.ai/research/10-ai-erdos2.pdf.

[4] Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, and others. *FrontierMath: A Benchmark for Evaluating Advanced Mathematical Reasoning in AI*. arXiv:2411.04872, 2024. https://arxiv.org/abs/2411.04872.

[5] Suhaas Garre, Erik Knutsen, Sushant Mehta, and Edwin Chen. *Riemann-Bench: A Benchmark for Moonshot Mathematics*. arXiv:2604.06802, 2026. https://arxiv.org/abs/2604.06802.

[6] SWE-bench Team. *SWE-bench Verified*. 2024. https://www.swebench.com/verified.html.

[7] Dijen K. Ray-Chaudhuri and Richard M. Wilson. On t-designs. *Osaka Journal of Mathematics*, 12:737–744, 1975.

[8] Paul Erdős, A. W. Goodman, and Louis Pósa. The representation of a graph by set intersections. *Canadian Journal of Mathematics*, 18:106–112, 1966.

[9] Kevin Ford. The distribution of integers with a divisor in a given interval. *Annals of Mathematics*, 168(2):367–433, 2008.