MATHDB: A DYNAMIC EVALUATION PLATFORM FOR FRONTIER AI IN RESEARCH MATHEMATICS AT SCALE

Anonymous authors
Paper under double-blind review

ABSTRACT

As AI systems advance at mathematical reasoning, static benchmarks saturate quickly, and repeatedly constructing expert-curated replacements is costly and hard to scale. We introduce MATHDB, a continuously updated, structured database of more than 80,000 research-level mathematical problems with literature provenance, domain labels, status histories, verification evidence, and a knowledge graph of mathematical relationships. MATHDB grows through automated literature ingestion and community contributions, and its structured metadata supports the construction and later refresh of benchmark releases without resourcing an entirely new problem set from scratch.

As a first demonstration of this framework, we construct MATHDB-BENCH from 100 tasks adapted from 981 resolved MATHDB records, spanning five mathematical domains and four reasoning-demand dimensions: structural, symbolic, quantitative/computational, and geometric/spatial reasoning. The tasks use deterministic checkers reviewed by mathematicians for fidelity and resistance to guessing. Evaluating four frontier and open-weight models under a requested 15,000-token completion budget, pass@1 ranges from 13.5% to 69.0%. Token-budget truncation, rather than incorrect completed answers, dominates failures: all four models are correct 92–100% of the time when they produce a checkable answer. A targeted 30,000-token probe on 10 tasks that were truncated for every model and run at 15k further shows substantial budget sensitivity: GPT-6 Astra solves 6 of 10, while Kimi K3 solves 1 and Qwen3.8 Max solves none.

We release benchmark tasks, prompts, deterministic checkers, reasoning-demand annotations, and raw model transcripts, while MATHDB serves as the durable contribution: a growing, community-editable resource from which new, unsaturated evaluations can be constructed as existing benchmarks saturate.

1 INTRODUCTION

Mathematical reasoning benchmarks are increasingly limited by **saturation**. Frontier models now exceed 95% accuracy on GSM8K (Cobbe et al., 2021), solve more than 90% of MATH (Hendrycks et al., 2021), and a single frontier model has been reported at 98% on the 2026 USA Mathematical Olympiad within months of release (Dekoninck et al., 2026). Once scores approach the ceiling, fixed benchmarks stop separating systems. Public exposure creates a second problem: statements and solutions may enter training data, so strong performance can reflect memorization as well as reasoning. GSM1k, a fresh set matched to GSM8K in style and difficulty, found drops of up to 13 points for some model families, consistent with overfitting to the original benchmark (Zhang et al., 2024). Recent work moves toward formal proofs, expert-written questions, and newly released mathematics (Zheng et al., 2021; Tsoukalas et al., 2024; Glazer et al., 2024; Balunović et al., 2025; Dekoninck et al., 2026), but any fixed release can eventually face the same pressure. FRONTIERMATH, for example, held frontier models below 2% accuracy at its 2024 introduction (Glazer et al., 2024), while later models solved over 50% of its easier tiers (Epoch AI, 2026).

We introduce MATHDB, a continuously updated, **structured** database containing 87,372 records of research-level mathematical problems at the time of writing, with literature provenance, domain labels, status history, verification evidence, and typed relationships between problems (Figure 7,

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Appendix A). The corpus grows through automated literature extraction and community contributions (Figure 1, Appendix A), while its retrieval layer supports discovery by domain, status, and relationship.

MathDB maintains a continuously updated pool of mathematical problems, while each MATHDB-BENCH release is frozen once constructed. Future releases can draw new tasks from the updated resolved pool as older benchmark items become saturated or exposed.

MathDB records carry different status labels. For evaluation, we distinguish MATHDB-OPEN from MATHDB-SOLVED: the latter contains records with an established resolution and supporting evidence and serves as the source pool for MATHDB-BENCH. The live corpus contained 981 resolved-status records on 2026-09-21.

As a first demonstration, we curate a 100-task release spanning five mathematical domains. Pass@1 ranges from **13.5% to 69.0%** across four frontier and open-weight models, and **64 of 100 tasks** have non-uniform outcomes across model–run pairs. Under the fixed 15,000-token inference budget used here, the dominant distinction is whether a model reaches a checkable answer before exhausting its budget: conditional on producing a completed, checkable answer, the four models are correct 92–100% of the time. These results show that measured performance on research-level tasks can depend strongly on the available inference budget. In a targeted 30k-token probe on 10 tasks that were truncated for every model and run at 15k, GPT-6 Astra recovers 6 of 10, while Kimi K3 solves 1 and Qwen3.8 Max solves none. This targeted result indicates substantial budget sensitivity for the strongest evaluated model, while additional inference does not benefit all systems equally. Details of the DeepSeek V4 Pro probe, whose requested cap was not reliably enforced by the provider, are reported in Appendix B.

Our main contributions are:

- MATHDB, a versioned, community-growing corpus of **87,372** structured records of research-level mathematical problems with provenance, status, revision metadata, and typed relationships;
- a refreshable evaluation framework in which frozen MATHDB-BENCH releases are drawn from an evolving resolved pool, allowing saturated items to be retired without rebuilding the source corpus from scratch; and
- a verified 100-task release with deterministic checkers and two-run evaluations of **4** models, including completion-conditional and run-to-run analyses.

2 THE MATHDB DATASET

MathDB is a public database of research-level mathematical problems combining material from research papers, established problem collections, automated imports, and community submissions. Each record links a problem statement to sources, subject tags, status information, progress summaries, and revisions. Users can search and browse the corpus, follow literature references, and contribute problems, solutions, or corrections.

2.1 PROBLEM SELECTION, COLLECTION, AND SOURCES

The repository contains importers for arXiv papers, AIM problem lists, Erdős problems, selected Wikipedia entries, and other curated collections, alongside direct community submissions. For arXiv material, the pipeline reads LaTeX source, identifies candidate conjecture/problem statements, and uses a language-model rewrite stage while preserving the source passage. Automated checks cover quoted passages, required fields, and formatting. Stable import identifiers, title normalization, statement-similarity grouping, and reviewed merge tools reduce duplicates. Records may also expose typed relationships such as *generalizes* or *implied by*, used as retrieval signals. Further collection and correction details appear in Appendix A.

2.2 CLASSIFICATION AND DATASET STATISTICS

MathDB maps subject tags to broad topics, with overlap allowed across categories. The dated API snapshot used here contains 87,372 distinct records on 2026-09-21, spanning geometry/topology, combinatorics, algebra, number theory, analysis, and smaller areas including probability/statistics,

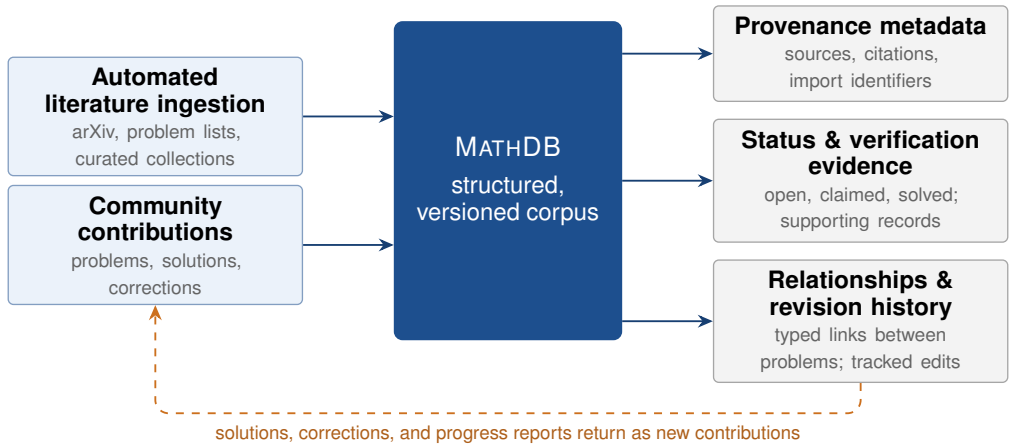


Figure 1: **Community-driven growth and maintenance of MATHDB.** Automated literature ingestion and direct community participation feed a structured corpus whose provenance, status evidence, relationships, and revision history evolve over time.

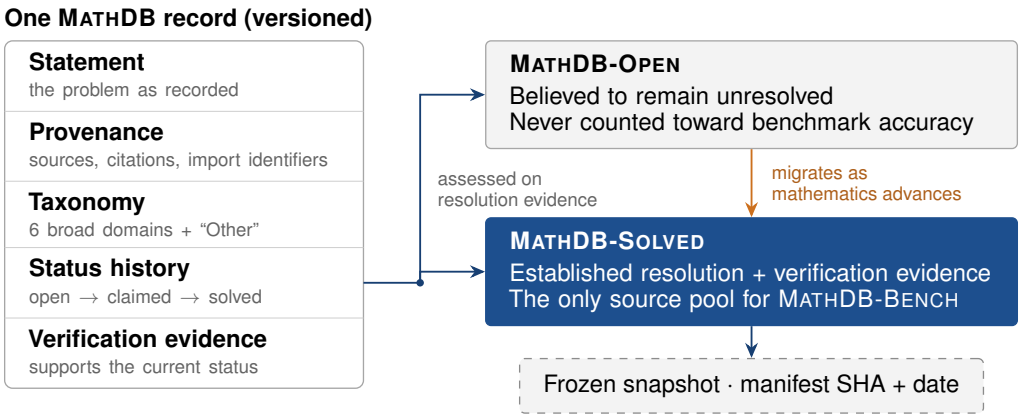


Figure 2: **A MathDB record and its use in evaluation.** A record brings together a problem statement, references, subject tags, and available progress information. The application also supports content revisions and separate records of verification. The MATHDB-OPEN/MATHDB-SOLVED distinction used in this paper requires an assessment of the evidence for a resolution; it is not a direct two-way split of the website's status filters.

theoretical computer science, optimization, mathematical physics, differential equations, and quantum information. Full topic counts are reported in Table 4 (Appendix A). These counts describe the live database rather than a frozen release, and the benchmark's five domain labels are assigned separately.

The website exposes several status categories, but a status label alone does not establish benchmark readiness. In the same dated observation, 84,655 records were open and 981 fell into the resolved-status group used to source MATHDB-SOLVED; those records retain supporting resolution evidence. These counts describe the live database on 2026-09-21.

2.3 QUALITY CONTROL AND REGULAR UPDATES

Quality control combines automated extraction checks, source retrieval, revision history, and curator review. The audit workflow compares proposed corrections against retrieved source passages and records review evidence for problem identity, mathematical content, attribution, chronology, and rendering.

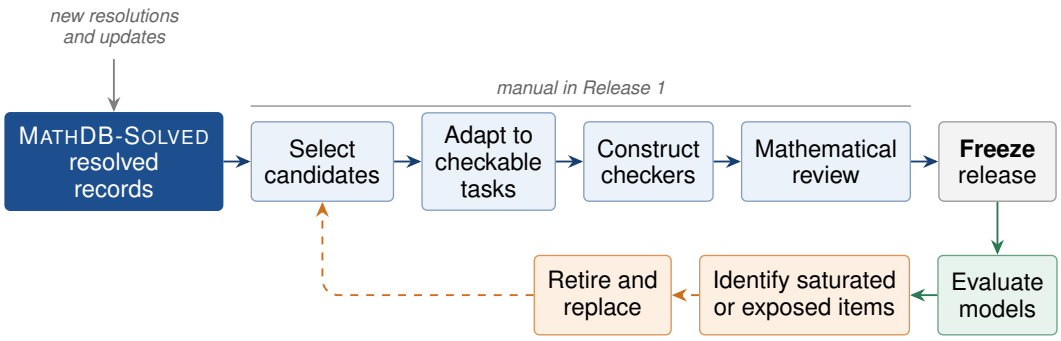


Figure 3: **Construction and refresh cycle of MATHDB-BENCH.** Each release selects resolved records, adapts them into checkable tasks, constructs deterministic verifiers, and freezes the reviewed benchmark before evaluation. Selection, adaptation, checker construction, and review are manual in Release 1. Future releases can retire saturated items and replace them from the growing MATHDB-SOLVED pool (orange, dashed) while preserving earlier frozen releases (gray).

Updates arrive through imports, community contributions, and progress searches. The application stores tracked revisions and supports consistent exports with manifests recording export time, scope, row counts, and file hashes, allowing benchmark releases to refer to frozen snapshots even while the live database continues to change.

An LLM-based spot-check of source fidelity. To estimate corpus-level source fidelity, we conducted an LLM-based spot-check of 200 records drawn from the general MathDB corpus across topic areas. For each record, a separate automated process fetched the cited source and classified the statement and attribution as matching, mismatched, misattributed but otherwise matching ("flag-naming"), or inconclusive. Of the 200 records, 163 (81.5%) matched, 8 (4.0%) had a naming/attribution issue, 19 (9.5%) mismatched their source, and 10 (5.0%) were inconclusive. This is an LLM-conducted source-fidelity check rather than a human mathematical audit; Appendix C gives the methodology and flagged records. The 100 benchmark tasks receive the stronger expert review described below.

2.4 MATHDB BENCH

MATHDB-BENCH uses selected problems from MathDB to evaluate mathematical reasoning under a fixed protocol. It is separate from the website's leaderboard, which summarizes recorded model contributions to individual problems. Here, models receive the same tasks and are evaluated using the same scoring rules.

2.4.1 TASK SELECTION AND RELEASE

The first release contains 100 tasks: 20 each from number theory, algebra, combinatorics, geometry/topology, and analysis. We use records with established resolutions and supporting evidence that can be adapted into checkable outputs; unresolved problems are excluded from the reported accuracy scores. Selection was manual, and preferentially drew on recently resolved records.

At the snapshot date, 981 resolved-status records already linked statements, sources, and resolution evidence, reducing discovery and provenance-tracking work during benchmark construction. Release 1 then manually adapts selected records into checkable tasks, constructs checkers, and reviews them before evaluation.

Selected problems are adapted, where needed, into tasks with explicit assumptions and outputs that can be checked deterministically. A task may ask for a construction, example, counterexample, exact expression, or witness. When an adaptation uses a special case or subproblem, its scope is stated explicitly; solving the adapted task is not presented as resolving the full source problem. Each benchmark task is adapted from a research-level source problem and is reviewed to preserve a nontrivial mathematical dependency on that source.

A worked example. MathDB record #315627 states Kourovka Problem 18.50: *given positive integers n and $1 \leq k \leq n!$, can n distinct group elements have exactly k distinct values among all their permuted products?* Posed by S. Kohl in 2013 and open for over a decade, it was resolved affirmatively in general in July 2026, shortly before this release's evaluation dates, via a central-extension construction. The adapted MATHDB-BENCH task fixes a single instance of this general question rather than asking for the general proof: work over the field $\mathbb{F}_{17}$; define a group on $\mathbb{F}_{17}^4 \times \mathbb{F}_{17}$ from a 4×4 matrix M via $(u, z) \cdot (v, w) = (u + v,\ z + w + u^\top M v)$; construct M so that the 24 permuted products of four standard generators take exactly 17 distinct values, and return M as a JSON array of integers in $\{0, \ldots, 16\}$. This turns the general existence question into a concrete construction task with many valid outputs. The checker recomputes the 24 products under the stated group law and verifies that exactly 17 distinct values occur, avoiding comparison against a single stored answer. The adaptation therefore makes verification cheap while retaining the central mathematical construction underlying the source result.

Each release freezes its task statements, source references, domain labels, prompts, and scoring rules before evaluation, together with stable problem and snapshot identifiers. Tasks, responses, and checkers are released for inspection and re-scoring.

2.4.2 ANSWER DESIGN AND VERIFICATION

The reported evaluation uses concrete outputs and deterministic checks. Its design follows two principles:

1. **Clear answer requirements.** Each task states what the model must produce and which conditions the answer must satisfy. To enable deterministic verification, we avoid open-ended long-form proof generation and instead ask, where possible, for an explicit object, value, construction, or witness. These outputs can be checked directly against the mathematical conditions specified by the task.
2. **Checks suited to the task.** Depending on the output, a checker may compare exact values, evaluate a symbolic expression, or test whether a submitted object satisfies the required conditions. Alternative valid outputs should be accepted when the task permits them.

Before release, all 100 adapted tasks and checkers were reviewed task-by-task by two mathematicians against both the MathDB record and its original cited source, with agreement required before inclusion.

Review covered source/status support, fidelity of the adaptation, nontrivial dependence on the source problem, unambiguous answer conditions, resistance to simple guessing, and checker behavior. Checker correctness was additionally tested with positive, negative, and recomputed-gold cases for all 100 tasks, alongside 33 public and 630 private tests of the adaptation and scoring pipeline. All task statements, checkers, inclusion decisions, and scoring rules were frozen before evaluated model outputs were inspected.

Reasoning-demand annotations. Each task additionally carries up to four overlapping reasoning-demand labels, reviewed and confirmed task-by-task by two mathematicians under a shared annotation guide. The labels support a descriptive breakdown of benchmark performance; full per-item annotations are reported in Appendix D.

3 EXPERIMENTS

We evaluate the first release of MATHDB-BENCH on four models. Our goal is to characterize the difficulty of the benchmark, compare model performance, and examine how outcomes vary across mathematical domains and individual problems.

3.1 EXPERIMENTAL SETUP

We evaluate 100 tasks from the current MATHDB-BENCH release, with 20 tasks in each of five mathematical domains. Each task is adapted from a MATHDB-SOLVED record with an established

Table 1: Models and reasoning-effort settings used through OpenRouter. Settings are provider-specific.

Model	Developer	Reasoning effort
GPT-6 Astra	OpenAI	`max`
Kimi K3	MoonshotAI	`max`
Qwen3.8 Max	Alibaba	`xhigh`
DeepSeek V4 Pro	DeepSeek	`max`

resolution into a format that permits deterministic programmatic checking. The evaluation excludes unresolved problems and solved records that cannot be adapted into this format.

We test GPT-6 Astra, Kimi K3, Qwen3.8 Max, and DeepSeek V4 Pro through OpenRouter. Each model uses the highest reasoning-effort setting available in our runs (Table 1).

Each model produces one scored response per task in each of two runs, giving $4 \times 100 \times 2 = 800$ scheduled evaluations. Every request starts with a fresh, zero-shot, closed-book context containing only the problem and answer-format instructions. We set a completion budget of 15,000 tokens, including reasoning and visible output, with 1,500 tokens reserved for output and a reasoning cap of 13,500 tokens. Responses must end with a single `FINAL ANSWER:` line.

A response that has not produced a `FINAL ANSWER:` line within 15,000 completion tokens is scored as *truncated*, even if the provider continues generating. Thus pass@1 is a budget-constrained success rate; Section 3.2 separates this from correctness conditional on completion.

All reported scores use deterministic, task-specific checkers. These check the submitted answer against the task's requirements, rather than grading every step of the written reasoning. We retain each response together with its token usage, latency, cost, finish reason, and provider-reported settings. Table 5 in Appendix B summarizes the protocol.

3.2 Overall Performance

Table 2 reports pass@1 accuracy averaged over the two runs, alongside the rate at which attempts hit the 15,000-token budget before producing a `FINAL ANSWER:` line. GPT-6 Astra scores highest at 69.0%, followed by Kimi K3 at 33.0%, Qwen3.8 Max at 20.5%, and DeepSeek V4 Pro at 13.5%. The scores remain below the benchmark ceiling and show clear differences among the evaluated models.

Truncation is the dominant unsuccessful outcome for every model: 30.5% of Astra's 200 attempts are truncated, versus 63.0% for Kimi, 79.5% for Qwen, and 86.5% for DeepSeek. Other non-`PASS` outcomes are rare (0.0–4.0% per model). Table 2 therefore separates budgeted success, completion, and correctness conditional on completion. Conditional correctness is 99.3% for Astra, 91.7% for Kimi, and 100.0% for both Qwen and DeepSeek. The large spread in pass@1 is consequently driven mainly by whether a model finishes within budget, not by wrong completed answers.

Appendix B makes this concrete using the token-usage logs already retained for every attempt (Section 3.1): Figure 9 there plots, for each model, the cumulative fraction of the 200 scheduled attempts that had reached a checkable `FINAL ANSWER:` line by a given completion-token count. Attempts that never reach one are right-censored at the requested cap rather than counted as completions there, so each curve ends at that model's completion rate (69.5%, 36.0%, 20.5%, and 13.5%) and the remaining mass is unobserved rather than observed to arrive at 15,000 tokens. Astra's curve is already at 63.0% by 10,000 tokens and gains only 6.5 further points over the last third of the budget, so most of its successes finish comfortably within budget rather than just under the wire; DeepSeek's curve stays close to flat across almost the entire range, so the great majority of its 200 attempts are still short of a checkable answer when the budget runs out, rather than failing shortly before it. The same appendix reports a targeted 30k-token probe on the tasks that were truncated for every model in both runs.

Outcomes were broadly similar across runs: model ordering is unchanged and per-task agreement ranges from 88% to 94%.

Table 2: Pass@1 accuracy across 100 tasks, averaged over two runs, alongside the fraction of the 200 scheduled attempts per model that were truncated at the 15,000-token budget or ended in some other non-PASS outcome (invalid, unparseable, empty, or API error), and the fraction of *completed* attempts (excluding truncated, empty, and API-error attempts) that passed the checker. "Solved" lists the counts for run 1 and run 2. The 95% confidence intervals use percentile bootstrap resampling over tasks, with 5,000 resamples. The 100 tasks are curated rather than randomly sampled, so the intervals describe sensitivity to task composition within this release, not population-level uncertainty.

Model	Solved	Accuracy	95% CI	Truncated / Other	Cond. correct
GPT-6 Astra	66 / 72	69.0%	[59.5, 77.5]	30.5% / 0.5%	99.3%
Kimi K3	32 / 34	33.0%	[24.5, 41.5]	63.0% / 4.0%	91.7%
Qwen3.8 Max	20 / 21	20.5%	[13.0, 28.0]	79.5% / 0.0%	100.0%
DeepSeek V4 Pro	14 / 13	13.5%	[7.5, 20.0]	86.5% / 0.0%	100.0%

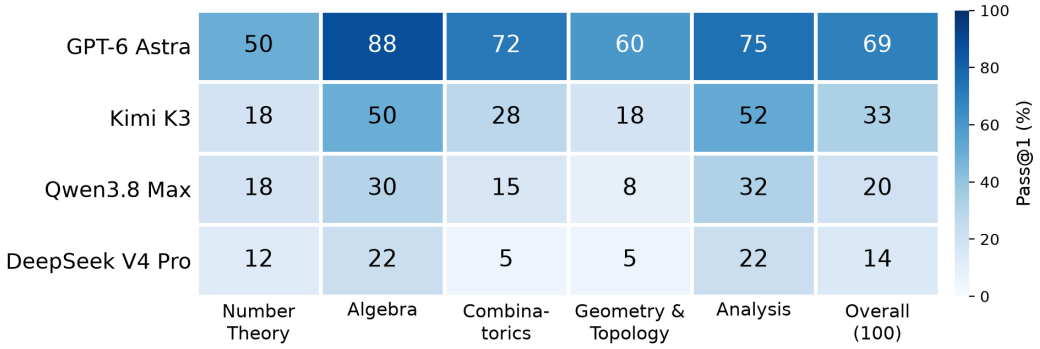


Figure 4: **Performance by mathematical domain.** Each cell gives pass@1 accuracy on the 20 tasks in one domain, averaged over two runs. The final column gives accuracy across all 100 tasks.

Provider-side enforcement of the requested 15,000-token cap was imperfect for two DeepSeek attempts; Appendix efapp:full-results gives the full accounting. Re-scoring the single over-budget passing attempt as truncated changes DeepSeek's pass@1 from 13.5% to 13.0%.

Attempts ending in an API error or empty response were resent once under a fresh run identifier; the retry outcome is the scored outcome. Appendix B gives the full accounting for all 12 such attempts.

3.3 Performance Across Mathematical Domains

Figure 4 breaks down budget-constrained success by subject. Number theory and geometry/topology have lower success rates in this release.

Reasoning-demand breakdown. We additionally annotate each task with up to four overlapping reasoning demands: abstract structural reasoning, symbolic manipulation, quantitative or computational reasoning, and geometric or spatial reasoning. Figure 5 shows performance by label. Because the labels overlap, this breakdown characterizes the composition of the release rather than estimating independent model abilities.

3.4 Item-Level Outcomes

Figure 6 shows which tasks each model solves. Eight tasks are solved by every model in both runs, while 28 receive no accepted answer from any model under the 15,000-token budget. Across all $28 \times 4 \times 2 = 224$ attempts on those 28 tasks, every attempt ends in truncation rather than a completed mathematical error. The remaining 64 tasks have mixed outcomes across model–run pairs.

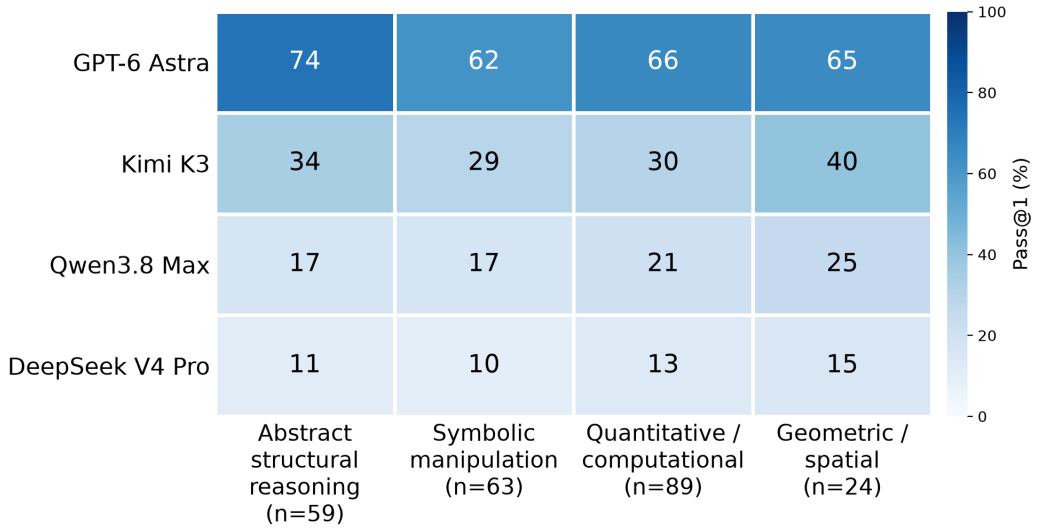


Figure 5: **Performance by reasoning demand.** Each cell gives pass@1 accuracy on tasks carrying the corresponding label, averaged over two runs. Labels may overlap across tasks.

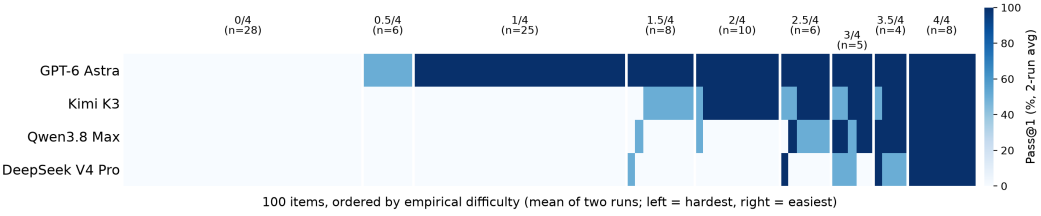


Figure 6: **Outcomes for individual tasks.** Each cell shows the checker result for one model and task, averaged over two runs: 0%, 50%, or 100%. Tasks are ordered by observed difficulty.

4 Related Work

Formal and competition benchmarks. MiniF2F (Zheng et al., 2021) and PutnamBench (Tsoukalas et al., 2024) evaluate theorem proving on Olympiad- and Putnam-level statements translated into proof assistants. Formalization gives machine-checkable correctness but restricts coverage to already formalized statements and ties evaluation to a proof language. MathDB-Bench instead keeps tasks in natural language and uses deterministic, item-specific checkers over concrete outputs.

Research-level and contamination-resistant benchmarks. FrontierMath (Glazer et al., 2024) commissions original, unpublished research-level problems from expert mathematicians, providing strong contamination resistance at substantial per-item expert cost. Two benchmarks target contamination through recency rather than research-level difficulty: MathArena (Balunović et al., 2025) evaluates newly released *competition* mathematics within hours of release, while ArXivMath (Dekoninck et al., 2026) sources *research-level* questions from recent arXiv papers as an ongoing platform, closer in scope to FrontierMath than to MathArena. MathDB follows a complementary model: it maintains a large public corpus first and constructs benchmark releases from its resolved subset, reusing provenance, status, and revision metadata while still requiring manual adaptation and verification.

Continuously maintained evaluation. Dynabench (Kiela et al., 2021) argued for benchmarks collected dynamically as models improve, and LiveCodeBench (Jain et al., 2024) demonstrates the value of rolling evaluation in programming by continuously incorporating newly released problems.

MathDB applies the same broad principle to research mathematics, with frozen releases drawn from an evolving database so that exposed or saturated items can later be retired and replaced.

Table 3 summarizes where MathDB-Bench sits along these axes relative to the systems discussed above.

Table 3: Positioning MATHDB-BENCH relative to related benchmarks. "Refresh" describes how the evaluation can incorporate new data or tasks over time.

Benchmark	**Problem level**	**Contamination strategy**	**Verification**	**Refresh**
MiniF2F / PutnamBench	Olympiad / Putnam	None (fixed, formalized)	Proof-assistant certification	Static
FrontierMath	Research (commissioned)	Withheld, unpublished problems	Expert-checked	Static per release
MathArena	Competition	Scored within hours of release	Reference-answer matching	Rolling, new competitions
ArXivMath	Research (arXiv-sourced)	Recent-paper recency	Answer matching	Ongoing platform
Dynabench	General NLP tasks	Fresh human-and-model-in-the-loop data	Task-specific evaluation	Dynamic, model-in-the-loop
LiveCodeBench	Competitive programming	Post-training-cutoff recency	Test-case execution	Rolling, continuous
MATHDB-BENCH (this work)	Research (database-sourced)	Recent resolutions	Deterministic, item-specific checkers	Frozen releases from an evolving pool

5 CONCLUSION, LIMITATIONS, AND FUTURE WORK

Corpus quality and benchmark scope. MathDB records have varying levels of review. The 200-record LLM-based source-fidelity audit found 81.5% matches and 9.5% mismatches in the general corpus, while the 100 benchmark tasks received task-by-task review by two mathematicians against their original sources. The release is manually curated, covers five domains, and evaluates programmatically checkable adaptations rather than open-ended proof writing or research mathematics as a whole.

Contamination and inference budget. Because MathDB records and many source statements are public, contamination cannot be ruled out; improved resolution-date metadata would enable stronger cutoff-based analysis. Pass@1 is also budget-dependent: the 15,000-token cap binds for many unsuccessful attempts, and the targeted 30k-token probe provides evidence of additional headroom for some models rather than a full scaling curve. Provider-specific reasoning settings mean that the evaluation fixes the protocol rather than underlying compute.

Future refresh. This paper evaluates the first frozen MATHDB-BENCH release. Future work will test the retire-and-refresh cycle empirically, expand audits of the resolved pool, improve contamination tracking, and extend the benchmark to proof-oriented tasks.

Conclusion. MathDB provides a growing, structured corpus of research-level problems with provenance, status, and verification evidence, and MATHDB-BENCH shows that frozen, deterministically checkable releases can be constructed from its resolved pool. On the first 100-task release, four models score 13.5–69.0% pass@1 under a 15,000-token budget with the same ordering across two runs; truncation rather than incorrect answers accounts for nearly all failures, so inference efficiency and termination are the main bottlenecks at this budget. Future releases will retire saturated items and draw replacements from the evolving resolved pool while preserving earlier snapshots for comparison.

AI USE STATEMENT

The authors used AI assistants (OpenAI Codex and Anthropic Claude) to assist with research planning, literature discovery, methodology review, drafting and editing parts of the text, LaTeX preparation, figure code, bibliography formatting, and draft figure and table placeholders. AI systems are also used inside the artifact itself, in two places that bear directly on the results and are reported where they occur: automated extraction of records from the literature into MathDB (Section 2.1), and the 200-record corpus spot-check, which is LLM-conducted end to end and is labeled as such throughout (Section 2.3, Appendix C). Final task selection was performed by humans. Reasoning-demand labels were initially drafted with AI assistance and then reviewed and jointly confirmed task-by-task by two mathematicians. All reported model outputs are scored by deterministic task-specific checkers. AI assistance was used in drafting some task statements and checker code. These materials were subsequently reviewed and revised by the authors, and all released tasks and task-specific checkers were manually inspected and tested before inclusion. The authors are responsible for checking AI-assisted text, claims, citations, code, and other materials included in the paper.

ETHICS STATEMENT

The release retains source attribution and follows applicable licensing requirements. It documents excluded material and distinguishes proposed solutions from accepted mathematical results. Reports of AI attempts likewise distinguish submitted artifacts, automated checks, human reviews, and the status of the underlying claim.

MathDB records are derived from publicly available mathematical literature and public problem lists, and every record carries a citation to its source. The released tasks are adaptations written for this benchmark rather than reproductions of source text, and they are distributed with their attributions. The study involves no human subjects and no personal data beyond names as they already appear in the cited literature and the identities community contributors choose to attach to their own public contributions. The benchmark tasks, checker code, and corpus exports are released under the MIT License. Community contributors are informed that submitted material may be incorporated into MathDB and publicly redistributed with attribution as part of future database and benchmark releases.

REPRODUCIBILITY STATEMENT

We estimate uncertainty by resampling benchmark tasks with replacement. Aggregate results use 5,000 resamples of the 100 tasks and percentile confidence intervals. Domain-level intervals use resampling within each 20-task domain. Exact McNemar tests compare paired binary outcomes between runs; these tests assess changes in success rates, rather than establishing that the runs are equivalent.

Run 1 took place on September 16–18, 2026, and run 2 on September 19–20. Both used the same evaluation parameters, fresh run identifiers, and separately submitted API requests. Reported scores average the two runs unless stated otherwise. The benchmark manifest has SHA-256 identifier `940eddf8...b86bee2f`.

Solver and evaluator commit identifiers, model-catalog and provenance filenames, and the full API-call and per-attempt record counts are recorded in the provenance files of the released archive. Checker/verifier code for all 100 tasks is released alongside the tasks and prompts, so released outputs can be independently inspected and re-scored, and the protocol can be rerun subject to provider/model availability; the reasoning-dimension labels (Figure 5) are published together with their shared annotation guide. The task set, prompts, checkers, per-attempt logs, and annotation guide accompany this submission as an anonymized supplementary archive whose manifest records the full SHA-256 hash; the public repository link accompanies the de-anonymized release.

REFERENCES

Mislav Balunović, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. MathArena: Evaluating LLMs on uncontaminated math competitions. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvaldsson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs. *arXiv preprint arXiv:2605.00674*, 2026.

Epoch AI. FrontierMath: Live results. https://epoch.ai/frontiermath, 2026. Accessed 2026-09-21.

Elliot Glazer, Ege Erdil, Tamay Besiroglu, Diego Chicharro, Evan Chen, Alex Gunning, Caroline Falkman Olsson, Jean-Stanislas Denain, Anson Ho, Emily de Oliveira Santos, Olli Järviniemi, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, Natalie Stewart, Bogdan Grechuk, Tetiana Grechuk, Shreepranav Varma Enugandla, and Mark Wildon. FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the MATH dataset. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Douwe Kiela, Max Bartolo, Yixin Nie, Divyansh Kaushik, Atticus Geiger, Zhengxuan Wu, Bertie Vidgen, Grusha Prasad, Amanpreet Singh, Pratik Ringshia, Zhiyi Ma, Tristan Thrush, Sebastian Riedel, Zeerak Waseem, Pontus Stenetorp, Robin Jia, Mohit Bansal, Christopher Potts, and Adina Williams. Dynabench: Rethinking benchmarking in NLP. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 4110–4124, 2021.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. PutnamBench: Evaluating neural theorem-provers on the Putnam mathematical competition. In *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2024.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Dylan Slack, Qin Lyu, Sean M. Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A careful examination of large language model performance on grade school arithmetic. *arXiv preprint arXiv:2405.00332*, 2024.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. MiniF2F: a cross-system benchmark for formal Olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*, 2021.

A Additional MathDB Dataset Details

This appendix gives detail on corpus collection, correction workflows, and topic statistics summarized in Section 2.

Collection and correction workflows. The default arXiv extraction path selects statements from recognized conjecture or problem environments; a language-model stage rewrites candidates using the surrounding context and expanded macros, with instructions to preserve the original assumptions and quantifiers, and its output includes the source passage used for the rewrite. Duplicate handling combines stable import identifiers, normalized-title comparison across sources, a batch process that groups repeated arXiv mentions by name and statement similarity, and reviewed merge tools that combine duplicate pages while preserving redirects from old identifiers. A correction manifest

requires approval of the formulation, problem identity, supporting evidence, and date information, with no recorded critical errors; importers preserve edited records so a later import does not overwrite reviewed wording. The LLM-conducted source-fidelity spot-check in Appendix C provides an empirical audit of the resulting records.

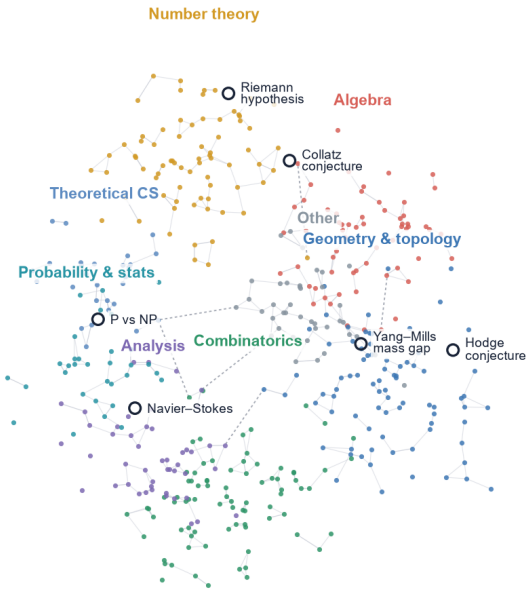


Figure 7: **The mathematics graph, schematically.** A stylized reconstruction, not a screenshot, of MathDB's public relationship view: records cluster by broad topic (color) with candidate typed edges (*generalizes*, *implied by*, *implies*, ...) drawn between them, and a handful of well-known open problems are marked as landmarks within their clusters (e.g. the Riemann hypothesis, the Hodge conjecture, P versus NP). Node positions and edges here are illustrative; the live site ranks the relationship graph by PageRank and node degree and lays out connected components for navigation.

Table 4: Distinct records returned by each topic filter on 2026-09-21, as reported in the dataset observation used for this paper. A record may appear under several topics, so the counts do not sum to the database total.

Topic	**Records**
Other	51,295
Geometry & topology	24,602
Combinatorics	17,882
Algebra	17,669
Number theory	10,615
Analysis	8,205
Probability & statistics	4,423
Theoretical computer science	1,325
Algorithms & optimization	1,223
Mathematical physics	766
Differential equations	588
Quantum information & computing	346
Total distinct records	**87,372**

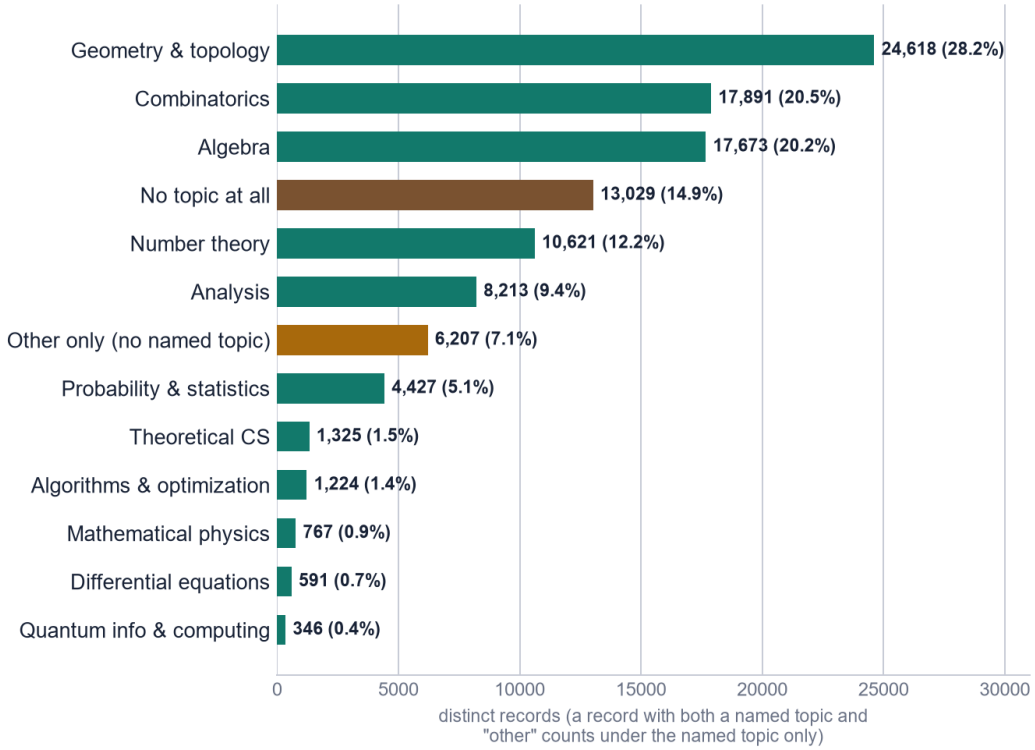


Figure 8: **Records by topic on 2026-09-21.** Counts are listed in Table 4. Topics overlap. The *Other* filter returns about 59% of records because it includes any problem carrying a tag assigned to that category, even when the problem also has tags in named areas.

B Additional Results for the Main 100-Task, Two-Run Evaluation

This appendix reports additional detail from the same two-run, four-model evaluation described in Section 3, at the level of individual domains and individual runs rather than the aggregates given there.

Table 5: Evaluation protocol.

Component	Setting
Tasks	100
Domains	5, with 20 tasks each
Models	4
Runs	Two independent pass@1 runs
Scheduled evaluations	800
Prompting	Zero-shot, closed-book
Completion budget	15,000 tokens: up to 13,500 for reasoning, with 1,500 reserved for output
Scoring	Deterministic task-specific checkers
Evaluation dates	September 16–20, 2026

Infrastructure and retry handling. A small number of requests were affected by provider-side failures or imperfect token-cap enforcement. Such cases were handled under a fixed retry and scoring policy, with detailed metadata retained in the released logs.

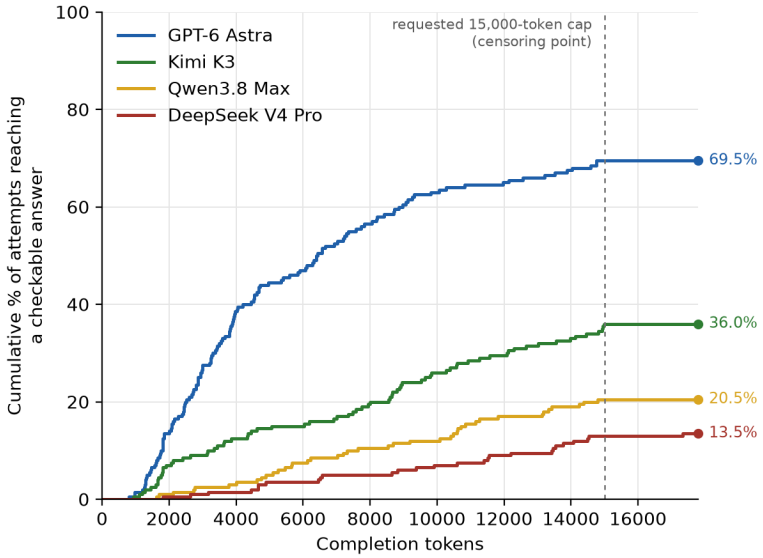


Figure 9: **When models reach a checkable answer.** Cumulative percentage of each model's 200 scheduled attempts that had produced a `FINAL ANSWER:` line by a given completion-token count, computed directly from retained token-usage logs. Attempts that never produce one are *right-censored* at the requested 15,000-token cap rather than recorded as events there, so each curve ends at that model's completion rate (69.5%, 36.0%, 20.5%, and 13.5%) instead of rising to 100%; the gap to 100% is censored mass, about which these data are uninformative. The single DeepSeek step past the cap is the one attempt whose provider overran the requested budget and still returned a checkable answer (Section 3.2).

Cost. The two runs cost $70.20 and $67.55 respectively across all four models (Astra $40.34/$39.86, Kimi $17.26/$15.29, Qwen $8.24/$8.15, DeepSeek $4.36/$4.25), consistent with Astra's higher per-call token consumption (Section 3.2) and highest reasoning-effort setting among the four models.

Rerun campaign. The 12 attempts that ended in `api_error` or an empty response — one for Astra in each run on task #315714, one each for Kimi on #339460, #382870, #392323 (both runs), and #405230, one each for Qwen on #344942 and #391484, and one each for DeepSeek on #376137, #383196, and #391462 — were each resent once under a fresh run identifier, and the retry's outcome is that attempt's scored outcome throughout this paper. Three retries resolved to a genuine, checker-verified PASS: both of Astra's attempts on #315714, and Qwen's first-run attempt on #391484. In each case the model's completion was well within the 15,000-token budget once the request succeeded, consistent with a transient provider-side fault rather than a modeling failure. Eight retries reproduced the identical non-completion — the model again generated no usable content within budget — and are scored as truncated (T) in Table 8. One retry (Kimi, #382870, second run) remained empty and is scored as such (N). Applying this single, uniform policy is the only change made to the original two-run data; Tables 2 and 8 and Section 3.4 reflect it throughout.

Budget-sensitivity probe. To test the budget-conflation limitation (Section 5) directly, we reran all four models at a doubled, 30,000-token completion budget on a fixed, pre-specified subset: the 10 tasks (#315800, #338094, #340334, #344070, #365581, #368387, #369591, #376123, #391552, #392113) that were truncated or errored in all eight attempts of the main two-run evaluation. This is a single run per model per task, not the two-run protocol used elsewhere in this paper, and the subset was chosen for being uniformly difficult rather than randomly sampled; we treat it as a targeted probe of the budget hypothesis, not as a systematic budget-scaling study. Correctness was checked with the same private evaluator used throughout. GPT-6 Astra solved 6 of the 10 tasks, and — restricting to the subset of its attempts that were neither truncated nor an API error — solved 6 of 6, i.e., every time Astra produced a checkable answer under the larger budget, that answer was correct. Kimi K3 solved 1 of 10; Qwen3.8 Max solved 0 of 10, and its only non-

truncated attempt was itself an API error. DeepSeek V4 Pro solved 0 of 10: on 7 of its 10 attempts (rerun up to three times each), the provider generated well past the requested 30,000-token cap (observed completion lengths up to 54,576 tokens, `finish_reason=length`) and returned no usable content, the same `cap_ignored_by_provider` behavior noted in Section 3.2; this null result reflects a provider-level inability to honor the requested budget on these tasks, not evidence about DeepSeek's mathematical performance under a larger budget. Taken together, doubling the budget recovers a substantial fraction of Astra's previously-truncated failures with no observed drop in correctness, while the same intervention does little for the two weaker models and cannot be meaningfully evaluated for DeepSeek given this provider-side constraint.

C Corpus Spot-Check: Methodology and Flagged Records

Section 2.3 reports an LLM-conducted spot-check of 200 records. This appendix gives the methodology and the full list of records flagged as a mismatch or a naming/attribution issue.

Methodology. We drew 200 records from the general MathDB corpus — all statuses, not restricted to MathDB-Solved — by a fixed pseudo-random seed, stratified across the corpus's topic areas. This sample's status composition (195 open, 2 solved, 1 claimed-solved, 2 claimed-progress) closely tracks the corpus-wide split reported in Section 2.2 (about 97% open), which is the expected result of sampling the general corpus rather than any status-specific subset. For each record, a separate automated process retrieved the record's cited source directly — an arXiv paper (via its HTML rendering, not the abstract page alone), a workshop-report PDF, or a public problem-list page — and compared the source's actual statement, resolution status, and attribution against the record's. Records were classified as **MATCH** (the source confirms the record's statement and attribution), **FLAG-NAMING** (the mathematical content matches, but an attribution or naming detail does not), **MISMATCH** (the source contradicts the record's statement, status, or points to different content entirely), or **INCONCLUSIVE** (the source could not be retrieved in usable form after repeated attempts — most often a paywalled or access-restricted workshop report — with no confirmed error). This is an LLM-conducted check: it establishes source-text agreement, not independent mathematical verification, and its classifications have not been separately audited by a human mathematician.

Result. 163 records (81.5%) matched, 8 (4.0%) were flag-naming, 19 (9.5%) were mismatches, and 10 (5.0%) remained inconclusive. Table 6 lists every record in the latter two categories with a short description of the issue found; full per-record reasoning is retained in the released audit data. Three of the mismatches (#395605, #399844, #403277) share a specific pattern worth separate note: in each, the cited source presents the statement as an already-proven theorem, while the MathDB record labels it an open problem, suggesting a recurring failure mode in the extraction pipeline (Section 2.1) rather than three independent errors. One mismatch (#345362) is of a different and more serious kind: after retrieving the complete text of the cited paper, we could not locate the record's stated conjecture anywhere in it, in any form; the record's claim appears to be a fabrication loosely inspired by language elsewhere in the source rather than a genuine misreading of it. Attribution- or status-only checks would not catch this class of error.

This 200-record general-corpus sample has no overlap with the 100 MathDB-Bench tasks. With 981 of 87,372 records (about 1.1%) carrying resolved status, such a sample is expected to contain few solved-status records — consistent with the 5 observed here — and to overlap this 100-task benchmark by under one record ($200 \times 100/87{,}372 \approx 0.23$).

#	Category	Issue
331601	Flag-naming	Source proves a narrower, related result; no matching numbered conjecture found under this framing.
348327	Flag-naming	Content matches an informal remark in the source, not a formally stated conjecture.
350660	Flag-naming	No formally numbered conjecture matches the precise stated bound; source gives only informal expectations.

continued on next page

#	Category	Issue
353429	Flag-naming	Content matches exactly; source attributes the result to different authors (Hasebe–Tsujie) than the record states.
367978	Flag-naming	Content matches exactly; source attributes the result to different authors than the record states.
370828	Flag-naming	Content matches exactly; source credits "Thomassen," not "Thomas" as the record states.
375602	Flag-naming	High-level framing matches; repeated fetches could not confirm the exact statement text.
378714	Flag-naming	Core formula and primary attribution match; an additional claimed attribution is not supported by the source.
335766	Mismatch	Stated group and order formula do not match the source, which states the analogous claim for a different (quotient) group.
335810	Mismatch	Cited arXiv identifier is a different paper entirely.
342704	Mismatch	An auxiliary function used in the record's formula is defined differently in the source.
343082	Mismatch	Source qualifies the conjecture as remaining open in general; the record's status does not reflect this qualification.
345362	Mismatch	No such conjecture appears anywhere in the cited source after a full-text read; the claim appears fabricated (see discussion above).
347974	Mismatch	Source presents this as a proven proposition, not an open conjecture.
349277	Mismatch	Cited paper concerns a different underlying equation than the one the record states.
355835	Mismatch	Source proves this only for a restricted special case, not the general claim stated in the record.
357891	Mismatch	Key terms and claimed content do not appear anywhere in the cited paper.
358538	Mismatch	Source's actual property ("tame") is weaker than the record's stated property ("finitely generated").
358887	Mismatch	Source paper disproves this conjecture with an explicit counterexample; the record does not reflect this.
363163	Mismatch	Source cites the referenced work as already proven, not as an open named conjecture.
365049	Mismatch	Cited arXiv identifier is a different, unrelated paper.
375195	Mismatch	No such conjecture found in the source after repeated targeted searches.
379306	Mismatch	Source proves this as an established theorem, not an open conjecture.
395605	Mismatch	Source presents this as an already-proven theorem (citing a 2007 proof), not an open problem.
397839	Mismatch	Record's formula omits an exponent present in the source's actual inequality.
399844	Mismatch	Source's own text describes this as an already-proven result.
403277	Mismatch	Source presents this as an already-proven theorem from the cited author's own prior paper.

Table 6: All 27 records flagged as a naming/attribution issue or a mismatch in the 200-record corpus spot-check (Appendix C).

D PER-ITEM REASONING-DEMAND ANNOTATIONS

Table 7 lists the reasoning-demand labels (Section 2.4.2) for all 100 tasks in the current MATHDB-BENCH release, identified by MATHDB problem number and grouped by domain. STR, SYM, QNT, and GEO abbreviate abstract structural reasoning, symbolic manipulation, quantitative or computational reasoning, and geometric or spatial reasoning; labels are non-exclusive, so a task can carry more than one.

Problem #	Reasoning demand	Problem #	Reasoning demand
Algebra			
392323	STR, SYM	315627	STR, SYM, QNT
315628	STR, QNT	315630	STR, SYM
315631	STR, QNT	315633	STR, SYM
392473	SYM, QNT, GEO	391938	STR, SYM
405223	STR, QNT	405162	SYM, QNT
405229	STR, GEO, SYM	381863	SYM, STR
369612	SYM, STR	358595	STR, SYM
338518	STR, QNT	369591	SYM, QNT
369590	SYM, QNT	357492	STR, SYM
329875	STR, QNT	356666	STR, QNT
Analysis			
315800	QNT, SYM	325018	SYM, QNT, GEO
344941	QNT, STR, SYM	344942	QNT, STR, SYM
345427	SYM, QNT	364465	GEO, QNT
381156	QNT, SYM	390720	GEO, QNT
390799	QNT, SYM	390802	QNT, SYM, GEO
390933	QNT, SYM	391016	QNT, STR
391020	GEO, QNT	391032	GEO, QNT, SYM
391035	QNT, SYM	391041	QNT, SYM
391451	STR, QNT	391484	GEO, SYM, QNT
391551	QNT, SYM	391552	QNT, SYM
Combinatorics			
372173	QNT, SYM, STR	372176	QNT, SYM, STR
372178	QNT, SYM, STR	372181	QNT, SYM, STR
375881	STR, QNT	375909	QNT, STR, SYM
376123	QNT, STR, SYM	376137	QNT, STR, SYM
376251	STR, QNT	378500	STR, QNT, SYM
382870	SYM, QNT	390692	STR, QNT
390803	STR, QNT	391196	STR, QNT
391253	STR, QNT	391264	STR, QNT
391297	STR, QNT	391349	QNT, STR
391369	STR, QNT	391462	STR, QNT
Geometry/Topology			
405230	SYM, STR, GEO	392128	GEO, QNT, SYM
383538	GEO, STR	383292	GEO, STR, QNT
405174	GEO, STR, QNT	392113	GEO, STR, SYM, QNT
315765	GEO, SYM, QNT	382954	STR, GEO, QNT
344070	GEO, STR, QNT	342558	STR, QNT, SYM
315714	SYM, STR, QNT	325178	GEO, QNT, SYM
344700	GEO, QNT, SYM	317901	GEO, STR, QNT
330524	GEO, QNT	376402	STR, SYM, QNT
315737	STR, SYM, QNT	370738	GEO, SYM, QNT
368387	GEO, QNT, SYM	405195	STR, QNT, SYM
Number Theory			
382449	QNT, SYM	370216	QNT
370289	QNT	370290	QNT
370291	QNT	383196	STR, SYM, QNT
352556	SYM, QNT	358100	SYM, QNT
338094	STR, SYM, QNT	363150	QNT, STR
346540	QNT	345258	QNT, SYM
339460	QNT, SYM	365581	STR, QNT, SYM
340334	STR, QNT, SYM	339692	STR, QNT, SYM
356045	STR, QNT, SYM	374009	SYM, QNT
315509	QNT	315531	QNT

continued on next page

Problem #	Reasoning demand	Problem #	Reasoning demand

Table 7: Per-item reasoning-demand labels for all 100 MATHDB-BENCH tasks, by MATHDB problem number.

E PER-ITEM, PER-MODEL RESULTS

Table 8 reports the checker outcome for every model on every task in both runs, in the same problem-number order as Table 7 so the two can be read side by side. Each cell gives *run 1/run 2* using: P (PASS, the checker accepted the submitted answer), T (truncated: the response hit the 15,000-token completion budget before a `FINAL ANSWER:` line was produced), I (invalid_answer: a final answer was given but failed the checker), U (unparseable_answer: no answer could be extracted from the response in the required format), E (api_error: the underlying request failed), and N (empty: an empty response). Summing PASS cells per model reproduces Table 2 exactly.

#	Astra	Kimi	Qwen	DeepSeek	#	Astra	Kimi	Qwen	DeepSeek
Algebra									
392323	P/P	T/N	T/T	T/T	315627	P/P	P/T	P/P	T/T
315628	P/P	T/T	T/T	T/T	315630	P/P	T/P	T/T	P/P
315631	P/P	P/I	P/P	P/P	315633	P/P	P/P	T/P	T/T
392473	P/P	T/P	T/P	T/T	391938	P/P	T/P	T/T	T/T
405223	U/P	T/T	T/T	T/T	405162	P/P	P/P	P/P	P/P
405229	P/P	P/P	T/T	T/T	381863	P/P	P/P	T/T	T/T
369612	P/P	P/P	T/T	T/T	358595	P/P	P/P	P/P	P/T
338518	P/P	T/P	T/T	T/T	369591	T/T	T/T	T/T	T/T
369590	P/P	T/T	T/T	T/T	357492	T/T	T/T	T/T	T/T
329875	P/P	T/T	T/T	T/T	356666	P/P	P/P	P/P	P/P
Analysis									
315800	T/T	T/T	T/T	T/T	325018	P/P	P/P	T/T	T/T
344941	P/P	T/T	T/T	T/T	344942	P/P	P/U	T/T	T/T
345427	P/P	T/T	T/T	T/T	364465	P/P	P/P	P/P	P/P
381156	P/P	P/P	P/P	T/T	390720	P/P	P/P	T/T	T/T
390799	P/P	T/T	T/T	T/T	390802	T/T	T/T	T/T	T/T
390933	P/P	T/I	T/T	P/T	391016	P/P	P/P	P/P	P/P
391020	P/P	P/P	P/P	T/T	391032	P/P	P/P	P/P	P/P
391035	P/P	P/P	P/T	P/T	391041	T/T	T/T	T/T	T/T
391451	P/P	P/P	T/T	T/T	391484	P/P	P/P	P/P	T/P
391551	T/T	T/T	T/T	T/T	391552	T/T	T/T	T/T	T/T
Combinatorics									
372173	P/P	P/P	T/P	T/T	372176	P/P	T/T	T/T	T/T
372178	P/P	P/P	P/T	T/T	372181	P/P	P/I	P/P	T/P
375881	P/P	T/T	T/T	T/T	375909	P/P	T/T	T/T	T/T
376123	T/T	T/T	T/T	T/T	376137	T/T	T/T	T/T	T/T
376251	T/T	T/T	T/T	T/T	378500	P/P	T/T	T/T	T/T
382870	P/P	T/N	T/T	T/T	390692	P/P	T/T	T/T	T/T
390803	P/P	T/P	T/T	T/T	391196	T/T	T/T	T/T	T/T
391253	P/P	T/T	T/T	T/T	391264	P/P	P/P	P/P	P/T
391297	T/P	T/T	T/T	T/T	391349	T/T	T/T	T/T	T/T
391369	P/P	T/P	T/T	T/T	391462	P/P	P/P	T/T	T/T
Geometry/Topology									
405230	P/P	T/T	T/T	T/T	392128	T/P	T/T	T/T	T/T
383538	P/P	T/T	T/T	T/T	383292	P/P	T/T	T/T	T/T
405174	T/T	T/T	T/T	T/T	392113	T/T	T/T	T/T	T/T
315765	P/P	T/T	T/T	T/T	382954	P/P	T/T	T/T	T/T
344070	T/T	T/T	T/T	T/T	342558	P/P	T/T	T/T	T/T
315714	P/P	T/T	T/T	T/T	325178	T/T	T/T	T/T	T/T
344700	T/T	T/T	T/T	T/T	317901	P/P	P/P	P/P	P/P
330524	P/P	P/P	P/T	T/T	376402	P/P	P/P	T/T	T/T
315737	P/P	I/P	T/T	T/T	370738	T/T	T/T	T/T	T/T

continued on next page

#	Astra	Kimi	Qwen	DeepSeek	#	Astra	Kimi	Qwen	DeepSeek
368387	T/T	T/T	T/T	T/T	405195	T/P	T/T	T/T	T/T
Number Theory									
382449	P/P	P/T	P/P	T/P	370216	P/P	P/P	P/P	P/P
370289	P/P	T/T	T/T	T/T	370290	T/P	T/T	T/T	T/T
370291	T/P	T/T	T/T	T/T	383196	T/T	T/T	T/T	T/T
352556	P/P	T/T	T/T	T/T	358100	T/T	T/T	T/T	T/T
338094	T/T	T/T	T/T	T/T	363150	P/P	P/P	T/T	T/T
346540	P/P	P/P	P/P	P/P	345258	T/T	T/T	T/T	T/T
339460	T/T	T/T	T/T	T/T	365581	T/T	T/T	T/T	T/T
340334	T/T	T/T	T/T	T/T	339692	P/P	T/T	T/P	T/T
356045	T/T	T/T	T/T	T/T	374009	T/T	T/T	T/T	T/T
315509	P/P	T/T	T/T	T/T	315531	P/P	T/I	T/T	T/T

Table 8: Per-item, per-model checker outcomes (run 1/run 2) for all 100 MATHDB-BENCH tasks.