

CROWDMATH: A Dataset of Crowdsourced Mathematical Research Discussions

Sherin Muckatira^{1,†,*} Jesse Geneson^{2,†} Slava Gerovitch³ Pavel Etingof³
Mikhail Gronas⁴ Anna Rumshisky^{1,5,◇}

¹University of Massachusetts Lowell ²San Jose State University

³Massachusetts Institute of Technology ⁴Dartmouth College ⁵Amazon AGI

Abstract

Large language models have made substantial progress on mathematical reasoning, but existing benchmarks typically evaluate well-specified problems with final answers, step-by-step solutions, or complete proofs. They do not capture collaborative open-problem solving: a setting in which participants propose partial arguments, identify gaps or errors in prior steps, repair flawed reasoning, and gradually synthesize incremental contributions into a proof. We introduce CROWDMATH, a dataset of 164 expert-annotated progress chains from the MIT PRIMES–Art of Problem Solving (AoPS) CrowdMath program (2016–2025), a collaborative research initiative whose discussions have led to peer-reviewed publications. Each chain traces a multi-participant forum discussion from an open-problem statement to a completed proof. Posts are labeled by their functional roles in the evolving solution process, including partial progress, proof completion, erroneous reasoning, and error identification. We define evaluation tasks and benchmark six frontier models. Models achieve 83–88% accuracy on next-post prediction, suggesting that they can follow the local flow of mathematical discussion. However, they struggle to identify the functional significance of individual contributions with the best model achieving only 0.42 macro-F1 on post-role classification. CROWDMATH exposes a gap between solving well-specified mathematical problems and understanding collaborative mathematical progress as it unfolds.

1 Introduction

Large language models (LLMs) have demonstrated impressive mathematical capabilities, with strong

performance on grade-school word problems, competition-style problem solving, and formal theorem proving (Cobbe et al., 2021; Hendrycks et al., 2021; Yang et al., 2023). These datasets and benchmarks have been valuable for measuring whether models can solve well-specified problems by producing a correct final answer, derivation, or machine-verifiable proof. Even when prior work exposes intermediate reasoning steps, it typically studies reasoning within a single solution attempt. This leaves open a complementary setting: collaborative open-problem solving, where mathematical progress is distributed across multiple posts and participants.

In collaborative open-problem solving, individual contributions may be incomplete in isolation but still essential to the eventual solution. A participant may propose a partial idea, test an example, identify an error, refine a conjecture, ask or answer a clarifying question, or connect earlier observations into a proof path. Understanding such discussions therefore requires more than checking whether a final answer is correct. A model must track where the discussion currently stands, how one contribution relates to earlier ones, whether a proposed step is valid or useful, and what kind of contribution would advance the argument.

We introduce CROWDMATH, a dataset of expert-annotated collaborative mathematical research discussions drawn from the MIT PRIMES–Art of Problem Solving (AoPS) CrowdMath program¹. CROWDMATH contains 164 progress chains tracing the development from open-problem statements to completed proofs across multi-participant forum threads. Each post is annotated with its functional role in the evolving solution process, including PROGRESS, PROOF, ERRONEOUS, FINDERROR, QUESTION, and ANSWER, with links to relevant prior posts and results.

[†] Equal contribution.

^{*} Corresponding author:

sherinbojappa_muckatira@student.uml.edu

[◇] Work done at University of Massachusetts Lowell.

¹<https://artofproblemsolving.com/polymath>

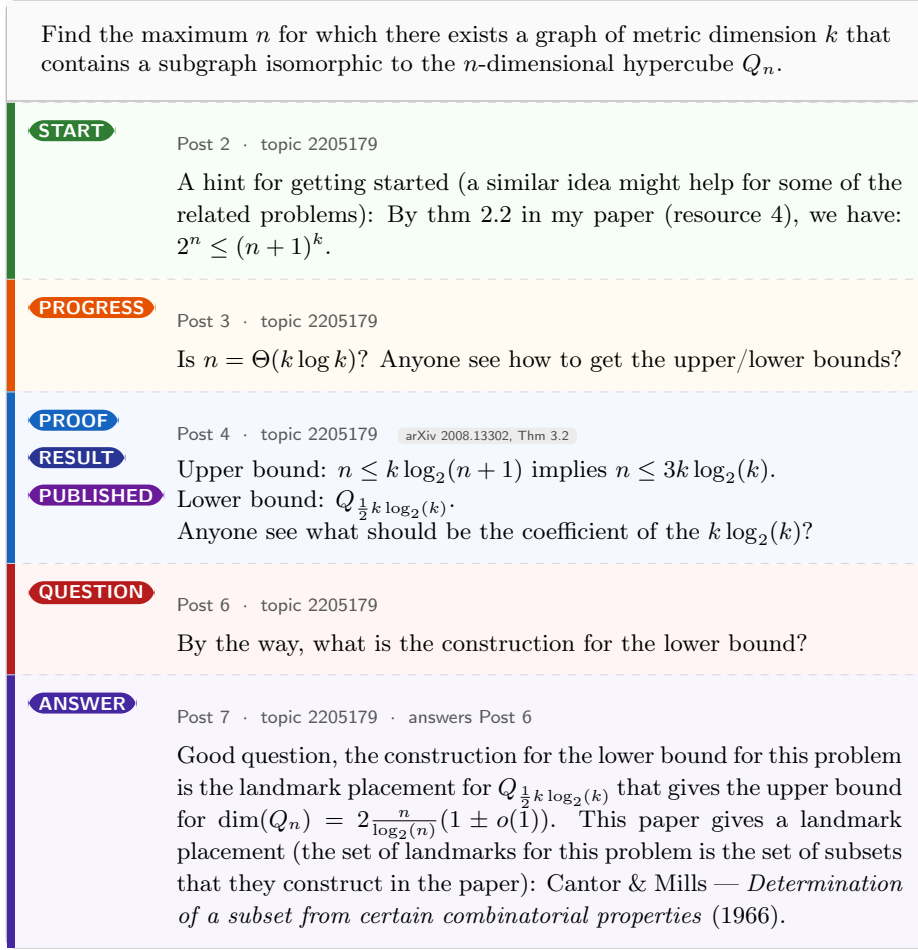


Figure 1: Example progress chain from CROWDMATH. The chain shows how an open problem discussion develops, including the start, partial progress, proof completion, and a follow-up question–answer exchange. The example is drawn from the metric-dimension and hypercube subgraph thread for Open Problem 2020-6.

CROWDMATH differs from prior resources in three key ways. First, it captures real-world research trajectories rather than textbook exercises or synthetic problems. Second, it annotates the process of mathematical progress, including partial ideas, errors, corrections, questions, answers, and proof-completing steps. Third, it preserves the collaborative structure of these open-problem discussions, enabling evaluation of whether models can track dependencies across participants and reason about how individual posts contribute to a shared mathematical result.

We evaluate frontier models on tasks derived from these annotations, including post-role classification and next-post prediction. Our results show that models can often follow the surface flow of mathematical discourse, but struggle to identify the functional significance of individual contributions,

especially when distinguishing proof-completing posts from partial progress. These findings suggest that current models remain limited in their ability to understand mathematical progress as an evolving collaborative process.

Our main contributions are as follows:

1. A dataset CROWDMATH of naturally occurring, multi-participant open mathematical problem-solving discussions from the MIT PRIMES CrowdMath program, with post-level annotations capturing each contribution’s role in the evolving solution process.
2. Evaluation tasks that test whether frontier models can classify proof steps and predict next contributions.
3. Baseline results showing that frontier models struggle to distinguish proof completions from partial progress, exposing a gap in process-level mathematical understanding.

2 Related Work

Grade School and Competition Style benchmarks There have been multiple math datasets that evaluate mathematical reasoning in language models with closed-form answers. Datasets such as (Ling et al., 2017; Amini et al., 2019; Cobbe et al., 2021; Miao et al., 2020) focus on grade-school and university-level mathematics with most of them having natural language rationale and a closed-form answer. Datasets such as (Hendrycks et al., 2021; Gao et al., 2025; He et al., 2024; Sun et al., 2026; Yue et al., 2024; Mao et al., 2024; Gulati et al., 2025) have harder settings that include high-school level competitions, Olympiad-style problems, and university-level contest problems. Dynamic and synthetic competition-style benchmarks further expand coverage by adding newly sourced or generated problems (Balunovic et al., 2026; Manem et al., 2025). While these datasets and benchmarks are well suited to problems with known solutions, they are less representative of complex open problems in mathematics. CrowdMath tries to capture the steps involved in solving open problems.

Research Level Math Reasoning datasets Recent datasets move beyond school and competition mathematics toward research-level reasoning. Glazer et al. (2025) introduce expert-crafted problems designed to be difficult for frontier models, while Zhang et al. (2026) draw problems from research papers and mathematical forums. Other works focus on open problems in specific mathematical areas, such as algebraic combinatorics and computational or applied mathematics (Chau et al., 2025; Wang et al., 2026). These benchmarks capture whether a model can arrive at a correct final answer, but not the process involving mathematical progress. CrowdMath captures how mathematicians collectively make progress on open problems.

Process supervision and proof-oriented datasets. Another set of datasets studies mathematical reasoning through process supervision, where supervision is provided at the level of intermediate reasoning steps rather than only the final answer. These datasets (Lightman et al., 2024; Wang et al., 2024) annotate individual steps in a solution with labels indicating correctness, using either human annotation or automatically generated supervision. Process-supervision datasets typically focus on evaluating the correctness of steps within a single

solution trajectory. In contrast, our dataset captures open mathematical problem solving as a broader collaborative process, where progress may arise from partial ideas, failed attempts, and corrections across multiple contributors. Formal and natural-language theorem-proving benchmarks evaluate whether models can prove mathematical statements, fill in missing proof steps, or translate informal mathematics into formal systems (Welleck et al., 2021; Zheng et al., 2022; Liu et al., 2023; Azerbayev et al., 2023; Li et al., 2021; Yang et al., 2023). While formal theorem-proving datasets provide rigorous benchmarks for machine-checkable reasoning, they do not capture the informal, exploratory, and collaborative processes that often characterize progress on open mathematical problems.

Collaborative mathematical discourse. A smaller body of work studies mathematics as dialogue. Macina et al. (2023) introduce teacher-student dialogue dataset between a human teacher and an LLM student with a focus on common student errors. Suresh et al. (2022) study classroom mathematics discussions through annotated K-12 lesson transcripts from video recordings. Unlike these datasets which study mathematical dialogue in educational settings, CrowdMath targets collaborative open problem solving and captures informal, multi-contributor discussions around difficult mathematical problems, where progress is often partial, non-linear, and distributed across participants.

Creativity and open-ended mathematical problem solving. Recent work has also begun to evaluate creativity in mathematical reasoning. Ye et al. (2025) study whether models can propose innovative solutions after seeing known solutions, and Chen et al. (2025) evaluate models on novel problems across areas such as algebra, geometry, and analysis. These works focus on assessing the creativity of model-generated solutions. CrowdMath provides a dataset for studying the collaborative process of open problem solving, where progress may emerge through partial ideas, errors, corrections, and counterexamples across multiple contributors.

3 CrowdMath Program

CrowdMath is a large-scale collaborative mathematics research program launched in 2016 by MIT PRIMES and the Art of Problem Solving (AoPS),

Label	Definition
Proof(y)	S_i completes the first accepted proof of result y .
NewProof(y)	S_i completes a better proof of y than the most recent accepted proof.
Start(y)	S_i initiates the thread of discussion leading to the first proof of y .
Progress(y)	S_i contributes to the proof of y (may be an idea, lemma, or question).
NewProgress(y)	S_i contributes to new progress of result y than the most recent accepted proof.
Question(y)	S_i asks a useful question relevant to result y .
Answer(y, j)	S_i answers question in S_j relevant to result y .
Erroneous(y)	S_i claims a proof of result y with a significant error.
FindError(y, j)	S_i finds an error in a claimed proof of result y in S_j .
Result	A new mathematical finding not already known in the literature. Clarifications, restatements of known theorems, or confirmations of known results are not considered new results.
Published-in-paper	Whether the results have been published in a paper; carries the arXiv ID as argument.
Theorem-number	Theorem number assigned in the published paper.
prev-ref	Post identifier of the previous post contributing to this result.

Table 1: CROWDMATH annotation schema for collaborative open-problem solving. Post-level labels capture functional roles such as proposing problems, making progress, completing proofs, errors, and identifying errors. Metadata fields record publication links, theorem numbers, and dependencies on prior posts.

inspired by the Polymath projects of Timothy Gowers and Terence Tao. It is open-enrollment and globally accessible, allowing high school and undergraduate participants to contribute without a formal selection process. Each annual cycle focuses on a small collection of original, unsolved problems, typically in combinatorics, graph theory, and related discrete areas, chosen to be approachable yet research-level. Participants work through them on persistent public AoPS forum threads under the guidance of mentors. Several projects have produced peer-reviewed publications in venues such as the Electronic Journal of Combinatorics and Discrete Applied Mathematics.

From a data perspective, CrowdMath provides an unusually rich record of mathematical discovery. Unlike curated problem sets or competition archives, the forum logs capture the full temporal evolution of reasoning: exploratory computations, partial proofs, counterexamples, errors, and corrections, all embedded in multi-participant discourse and tied to verifiable published outcomes. This combination of authentic open-problem research, collaborative interaction, and explicit intermediate

states makes CrowdMath particularly well-suited for constructing datasets that target fine-grained reasoning, error detection, and process-level supervision rather than final-answer accuracy alone.

4 Dataset

4.1 Annotation Process

Each post S_i is assigned one or more labels from the schema in Table 1. Because most labels are result-centric (taking a result identifier y , and sometimes a post identifier j , as arguments) a post cannot be fully labeled until the results it references have been identified. We therefore annotate each CrowdMath thread in two passes.

Pass 1: Result identification. We read the thread to identify every result established in its discussion, where a result is a new mathematical finding as defined in Table 1. Each result is given a unique identifier derived from its completion post: a result completed in post k of thread X is labeled $X-k$, with a trailing letter appended when a single post completes multiple results (e.g., $X-ka$, $X-kb$). When a result corresponds to

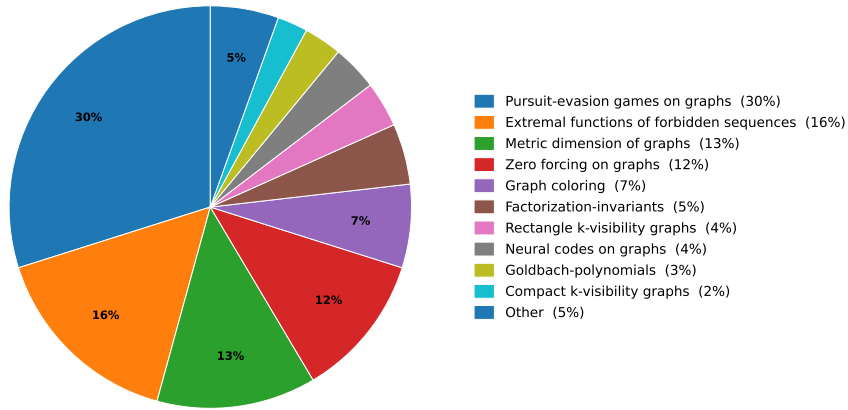


Figure 2: Distribution of the 164 CROWDMATH progress chains across mathematical topics. The largest areas are pursuit–evasion games on graphs, extremal functions of forbidden sequences, metric dimension of graphs, and zero forcing on graphs. Topics contributing fewer than 2% of chains are grouped as “Other.”

(A) Chains per year		(B) Posts per result			(C) Label distribution		
Year	Chains	Posts	Chains	%	Label	Count	%
2016	26	1	81	49.4	Proof	164	30.5
2017	59	2	22	13.4	Result	164	30.5
2018	6	3	25	15.2	Progress	28	5.2
2019	11	4	9	5.5	Question	42	7.8
2020	21	5	8	4.9	Start	60	11.2
2021	19	6–10	15	9.1	Answer	37	6.9
2022	8	≥11	4	2.4	Erroneous	16	3.0
2023	2	Mean		2.7	FindError	16	3.0
2024	5	Median		2	NewProgress	9	1.7
2025	7	Max		25	NewProof	2	0.4
Total	164				Total	538	

Table 2: Dataset statistics for CROWDMATH. (A) Number of annotated progress chains by year from 2016–2025. (B) Distribution of chain lengths, measured by the number of posts associated with each completed result. (C) Number of chains containing each label type.

a theorem in a published CrowdMath paper, we align the two and record the associated metadata (Published-in-paper, Theorem-number).

Pass 2: Post labeling. With results indexed, we revisit the thread and assign post-level labels. Labels such as $\text{Progress}(y)$, $\text{Proof}(y)$, and $\text{Erroneous}(y)$ reference a result y ; labels such as $\text{Answer}(y, j)$ and $\text{FindError}(y, j)$ additionally reference a prior post S_j .

Cross-thread results: A post in one thread occasionally contributes to a result completed in a different thread. Since result identifiers are determined by completion posts, labels referencing such a result cannot be finalized until the thread containing its completion has been processed. We main-

tain a global result registry across threads and defer finalization of these labels until the referenced completion post is indexed, ensuring all references are globally consistent.

Converting raw annotations into progress chains: The raw annotations consist of post-level labels that may reference results and, in some cases, prior posts. We convert these annotations into progress chains by first identifying the set of mathematical results established in each thread, then collecting all posts whose labels refer to each result. Each resulting chain contains the open-problem statement, any associated setup or resource text, and the sequence of posts that contribute to the corresponding result, ordered by post number. This construction allows the dataset to represent a proof trajectory rather than a flat collection of labeled

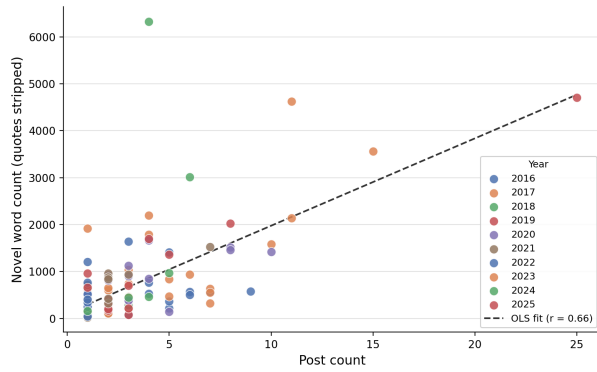


Figure 3: Post count versus text volume for CROWDMATH progress chains. Each point represents one completed result. The x-axis shows the number of posts in the chain, and the y-axis shows the total number of novel words after removing quoted prior-post content. Point color indicates the year of the completed result.

posts. The full chain-construction algorithm is given in Appendix A.1, and the post-level label schema is shown in Table 1.

Anonymization. Before we received the data, AoPS performed an initial metadata anonymization step: the dataset was provided as a CSV file with one row per post, in which the author username field was replaced by a hash string and exact timestamps were removed, with only chronological post order retained. Because this process anonymized metadata fields but did not systematically anonymize the contents of posts, some post bodies still contained names or other identifying references. We therefore performed an additional systematic scan of the post text to identify and replace remaining personally identifiable information, including real names, forum usernames, and other identifying references.

4.2 Dataset Statistics

CROWDMATH contains 164 annotated progress chains drawn from MIT PRIMES CrowdMath discussions between 2016 and 2025. The number of chains varies substantially by year. Table 2(A) shows the number of chains per year. The largest share of chains comes from 2017, with 59 chains, followed by 2016 with 26 chains, 2020 with 21 chains, and 2021 with 19 chains. Later years contribute fewer chains, with 2023, 2024, and 2025 contributing 2, 5, and 7 chains respectively.

The dataset covers a range of topics in discrete mathematics and combinatorics. As shown in Figure 2, the largest topic area is pursuit-evasion

games on graphs, accounting for 30% of progress chains. Other major areas include extremal functions of forbidden sequences (16%), metric dimension of graphs (13%), zero forcing on graphs (12%), graph coloring (7%), and factorization-invariants (5%). This distribution reflects the research focus of the CrowdMath program, where problems are chosen to be accessible to participants while still leading to publishable mathematical results.

Progress chains vary in length. Nearly half of the chains contain a single post associated with the completed result, 81 out of 164 chains (49.4%). Table 2(B) shows a detailed breakdown of the chain lengths. A substantial portion require multiple posts. The mean chain length is 2.7 posts, the median is 2 posts, and the longest chain contains 25 posts. This shows that CROWDMATH contains both compact proof trajectories and longer multi-step discussions in which progress accumulates across several contributions.

Every chain contains a PROOF label and a corresponding RESULT entry as the dataset is constructed around completed results, giving 164 instances of each. Intermediate contributions are also common. As shown in Table 2(C), PROGRESS appears in 28 chains times, QUESTION 42 times, START 60 times, and ANSWER 37 times. Error-related labels are less frequent but important for evaluating reasoning over flawed arguments: ERRONEOUS appears 16 times and FINDERROR appears 16 times, NEWPROGRESS appears 9 times and NEWPROOF appears 2 times. Each label type is counted only once in a chain. These labels allow the dataset to represent not only successful proof completion, but also partial progress, clarification, and error correction.

Figure 3 plots the number of posts in each chain against the total number of novel words per proved result after removing quoted prior-post content. Each point in the scatter plot represents a result with the color indicating the year in which the result was completed. Most progress chains are compact with fewer posts and fewer novel words. However, there are some progress chains with few posts but a large amount of novel mathematical text, indicating denser proof contributions rather than extended back-and-forth discussion.

5 Evaluations

We evaluate six frontier models (GPT-5.4, Claude Opus 4.6, Grok 4.20, DeepSeek V3.2, Gemini 2.5

Flash, Qwen3-235B) at temperature 0 and other default settings in OpenRouter² for these models. The model details can be found in the appendix table 4. The prompts used for this task are in Appendix D.

5.1 Frontier models struggle to identify proof completions.

To track how well frontier models predict the right post label in our dataset, we construct 275 instances from the dataset. Each ERRONEOUS and FIND_ERROR post becomes one instance; RESULT instances are the RESULT post of each proof chain; PROGRESS instances are sampled one per chain from posts labeled START, PROGRESS, or NEW-PROGRESS. The label counts are 164, 60, 28, and 23 for RESULT / PROGRESS / ERRONEOUS / FIND_ERROR respectively. The prompts used for this task are in Appendix D and contains one calibration example per class.

The first row of Table 3 shows the macro-F1 score across different frontier models. F1 scores are between 0.33 - 0.42, showing that frontier models struggle to identify the different steps in a proof. We report 95% bootstrap confidence intervals from 10,000 instance-level resamples. As baselines, we consider predicting the majority class (RESULT) and a position-only logistic regression classifier. We train the logistic regression classifier to predict the four labels from the number of preceding posts in the context, using 5-fold stratified cross-validation grouped by chain. The position-only baseline obtains macro-F1 scores of 0.186, showing that post position alone is insufficient to predict contribution type. Figure 4 shows the breakdown of model predictions across all 4 classes. Most models struggle to identify proof completions: rather than labeling these posts as result, they label them as progress. Precision on result is high, showing that when a model labels a post as result it almost always does complete the proof, but recall is low.

5.2 Frontier models perform well on next-post prediction.

To measure how well frontier models identify the next post in an ongoing mathematical discussion, we construct 81 multiple-choice instances from the dataset. Each proof chain with at least two posts is considered. All posts before the target post are provided as thread prefix to the model and the model needs to predict the next post out

of the 4 options. The remaining three distractor posts are sampled uniformly from other chains in the same year; chains with fewer than three same-year candidates are dropped. For the target post we prefer sampling posts labeled Start or Progress; if no such posts exist in the chain, we sample a post at random. Quote blocks are stripped from all texts because they would otherwise make the task trivial. Across 81 instances the target label distribution is 24 Progress, 13 Question, 8 FindError, 5 Answer, 3 Start, 2 NewProgress, 1 Erroneous, and 27 Proof+Result (of which 2 also carry FindError). We evaluate the same six frontier models at temperature 0 and other default settings in OpenRouter for these models with a 1-shot prompt containing one calibration example. The prompts used for this task are in Appendix D.

From table 3, we see that all six models perform well above the 25% random baseline, with accuracy ranging from 82.7% to 87.7%, suggesting that frontier models can perform reasonably well at predicting the next post in a mathematical discourse. We report 95% bootstrap confidence intervals from 10,000 instance-level resamples. We additionally compare against several simple baselines. The answer-position counts are A:18, B:17, C:23, and D:23, and the baselines select the most frequent answer position, the longest or shortest candidate, or the candidate with the highest TF-IDF cosine similarity to the discussion prefix. For the TF-IDF baseline, we fit a TF-IDF vectorizer using scikit-learn with default settings on the corpus containing the prefixes and four candidate posts for all 81 instances, and select the candidate with the highest cosine similarity to the prefix. The strongest baseline, TF-IDF cosine similarity, achieves 55.6% accuracy, substantially below the performance of the frontier models.

5.3 Qualitative Error Analysis

Identifying proof completions: To understand why RESULT posts over the six models are misclassified, we analyze the models' self-reported justifications (the reasoning field elicited by the prompt) using keyword patterns (we spot-check few of them manually) The patterns are not mutually exclusive.

1. Among the 582 RESULT to PROGRESS justifications across all frontier models, in 30% of these errors, the model judges the post against the wrong target: it says the post "does not

²<https://openrouter.ai/docs/quickstart>

(a) Task 1: Post-role classification			(b) Task 2: Next-post prediction	
Method / Model	Macro-F1 $\uparrow$	Accuracy $\uparrow$	Method / Model	Accuracy $\uparrow$
<i>Baselines</i>			<i>Baselines</i>	
Majority class (RESULT)	0.187	0.596	Uniform random	0.250
Position-only	0.186	0.589	Most frequent answer position	0.284
<i>Frontier models</i>			Longest candidate	0.247
GPT-5.4	0.362 [0.297, 0.418]	0.316 [0.262, 0.374]	Shortest candidate	0.321
Claude Opus 4.6	0.424 [0.357, 0.483]	0.371 [0.313, 0.429]	TF-IDF cosine similarity	0.556
Grok 4.20	0.396 [0.326, 0.459]	0.389 [0.331, 0.447]	<i>Frontier models</i>	
DeepSeek V3.2	0.354 [0.285, 0.417]	0.335 [0.280, 0.393]	GPT-5.4	0.877 [0.802, 0.938]
Gemini 2.5 Flash	0.326 [0.264, 0.382]	0.291 [0.240, 0.345]	Claude Opus 4.6	0.852 [0.765, 0.926]
Qwen3-235B	0.373 [0.311, 0.428]	0.316 [0.262, 0.371]	Grok 4.20	0.827 [0.741, 0.901]
			DeepSeek V3.2	0.877 [0.802, 0.938]
			Gemini 2.5 Flash	0.864 [0.790, 0.938]
			Qwen3-235B	0.827 [0.741, 0.901]

Table 3: Baselines and frontier-model performance on CROWDMATH evaluation tasks. Task 1 evaluates post-role classification using macro-F1 and accuracy over four classes: progress, result, erroneous, and find_error. Task 2 evaluates 4-way multiple-choice next-post prediction using accuracy. Values in brackets are 95% bootstrap confidence intervals from 10,000 instance-level resamples.

fully resolve the original problem." and does not consider that the result label is meant to be a self contained finding. For 17% of the posts, the model penalizes the informal writing style of the posts, calling the argument "not rigorous," "unproven," or "plausible but," even when the underlying mathematics is complete and correct.

- Models label RESULT posts as ERRONEOUS in 28% of the misclassifications. In 89% of these ERRONEOUS labels, the model points to a specific mistake it believes it found. In over half of them (55%), the post comes from a chain whose results were later published in a peer-reviewed paper.
- Finally, we check whether a model could identify RESULT posts simply by looking for phrases like "this completes the proof" or "QED." In 90% of the RESULT posts in the benchmark contain no such phrase, and of the 13 RESULT posts most models classify correctly, only 2 contain such a phrase. Thus, there is no wording shortcut and the model must reason about the actual mathematics in the post.

Next Post prediction We pool the errors across all models and analyze the models' self-reported justifications. The prompt elicits an assessment of every candidate, so for each error we can read why the model rejected the true next post. We use keyword patterns over these assessments, spot-checked a few manually and the categories overlap.

- In 38% of the errors, the models reject the true next post because it changes direction by

opening a new subproblem, asking a forward-looking question, or reporting a relevant external paper.

- In 17% the models reject the true next post as it is too brief or asking for a clarification.
- In 9% the models reject the true next post as mathematically wrong. These are target posts that correct or disagree with an earlier post.

6 Conclusion

We presented CROWDMATH, a benchmark for evaluating language models on collaborative open-problem solving. Existing mathematical reasoning benchmarks largely evaluate whether a model can produce a correct final answer or complete proof. In contrast, CROWDMATH evaluates whether models can understand the process by which mathematical results emerge: through partial progress, questions, errors, refinements, and proof completion across multi-post discussions. Our results reveal a gap between solving well-specified mathematical problems and understanding collaborative mathematical progress. While frontier models can often predict the local flow of a discussion, they struggle to judge the role and significance of individual posts, frequently confusing completed proofs with partial progress. These findings suggest that current models have not yet fully learned the discourse-level reasoning needed to track how mathematical ideas develop over time. By shifting evaluation from final-answer correctness to progress-aware reasoning, CROWDMATH opens a new direction for benchmarking mathematical intelligence. We hope it supports future work on models that can

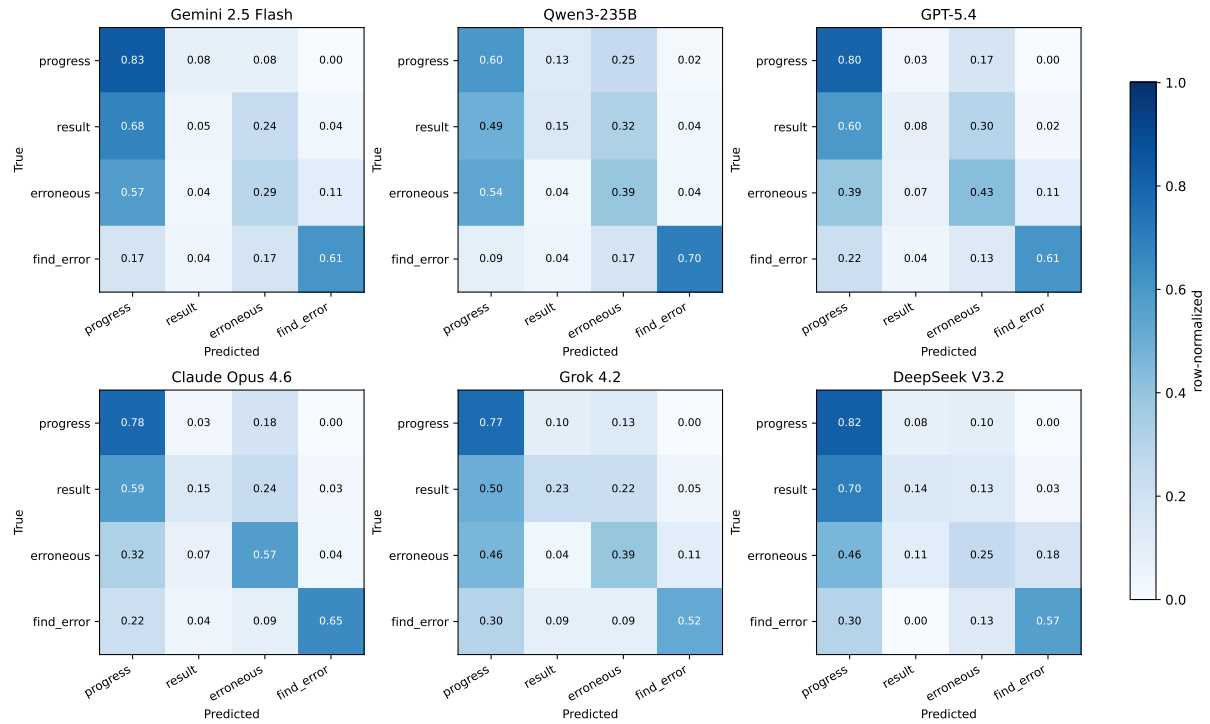


Figure 4: Row-normalized confusion matrices for Task 1 post-role classification across six frontier models. Each cell shows the fraction of posts with a given true label (rows) predicted as each class (columns) normalized along the row. Values along the diagonal are per-class recall.

participate more effectively in open-ended mathematical inquiry.

Ethical considerations

CROWDMATH is derived from discussion threads from the CrowdMath program, an initiative created by MIT PRIMES and AoPS. The study was reviewed by the Institutional Review Board (IRB) and determined to be exempt under Category 4 for secondary use research. The dataset is intended for studying collaborative mathematical problem-solving and process-level reasoning in mathematical discourse. Potential risks include misuse of the dataset outside its intended research context. As described in Section 4.1, the data was anonymized before analysis and release: usernames were replaced with hash strings, exact timestamps were replaced by chronological post order, and we performed an additional scan of post text to remove remaining identifying references. Because the original discussions are publicly accessible, anonymization does not fully eliminate re-identification risk. Users of the released dataset should not attempt to link anonymized posts back to original forum accounts or otherwise infer participant identities. The dataset and accompanying code are publicly avail-

able under the MIT License. The original forum participants were not recruited or compensated for this study as the dataset is derived from pre-existing public CrowdMath discussions. Expert annotators were recruited based on mathematical background and compensated for annotation work under standard project/institutional arrangements.

Limitations

The dataset has several limitations that bear on its use. First, it is result-centric: chains are built around completed results, so exploratory reasoning that does not contribute to a finalized result (for example, false starts, abandoned ideas, or dead-end computations) is not represented. This makes the dataset tractable but yields a filtered view of the problem-solving process, and the boundary between partial arguments and finished proofs reflects annotator judgment. Second, labels are assigned at the post level. A post that contains several logically distinct components receives all applicable labels, but the annotation does not record which spans of the text support each label, limiting fidelity for fine-grained reasoning tasks. Third, posts can contribute to more than one completed result and therefore appear in multiple chains. This re-

flects genuine mathematical structure, for example when a lemma supports several results, but means that naive train/test splits at the post or chain level can leak content across partitions; splitting at the level of results or connected components of the post-contribution graph avoids this. Annotations were produced by different annotators across years, with a single annotator per year. For the 2024 subset, we additionally measure inter-annotator agreement using independent double annotations and report them in Appendix C. However, for the remaining years, annotator-specific conventions are confounded with temporal variation, and inter-annotator agreement cannot be used to quantify uncertainty across the full dataset. The label space is coarse, distinguishing roles such as PROOF, ERRONEOUS, and QUESTION but not finer structure such as error types or degrees of rigor. Finally, CrowdMath is an online collaborative program aimed at high school and undergraduate participants, so the discourse, error patterns, and levels of rigor may not generalize to other mathematical settings such as professional research or formal proof systems.

Acknowledgments

This work was funded in part by an Amazon AGI research award to Anna Rumshisky. We thank Harold Polo for his contribution to the annotation.

References

- Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. [MathQA: Towards interpretable math word problem solving with operation-based formalisms](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. 2023. [Proofnet: Autoformalizing and formally proving undergraduate-level mathematics](#). *CoRR*, abs/2302.12433.
- Mislav Balunovic, Jasper Dekoninck, Ivo Petrov, Nikola Jovanović, and Martin Vechev. 2026. [Matharena: Evaluating LLMs on uncontaminated math competitions](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Herman Chau, Helen Jenne, Davis Brown, Jesse He, Mark Raugas, Sara C. Billey, and Henry Kvinge. 2025. [Machine learning meets algebraic combinatorics: A suite of datasets capturing research-level conjecturing ability in pure mathematics](#). In *Forty-second International Conference on Machine Learning*.
- Xiaoyang Chen, Xinan Dai, Yu Du, Qian Feng, Naixu Guo, Tingshuo Gu, Yuting Gao, Yingyi Gao, Xudong Han, Xiang Jiang, Yilin Jin, Hongyi Lin, Shisheng Lin, Xiangnan Li, Yuante Li, Yixing Li, Zhentao Lai, Zilu Ma, Yingrong Peng, and 12 others. 2025. [Deepmath-creative: A benchmark for evaluating mathematical creativity of large language models](#). *Preprint*, arXiv:2505.08744.
- 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. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. 2025. [Omni-MATH: A universal olympiad level mathematic benchmark for large language models](#). In *The Thirteenth International Conference on Learning Representations*.
- 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 Järvinen, Matthew Barnett, Robert Sandler, Matej Vrzala, Jaime Sevilla, Qiuyu Ren, Elizabeth Pratt, Lionel Levine, Grant Barkley, and 5 others. 2025. [Frontiermath: A benchmark for evaluating advanced mathematical reasoning in ai](#). *Preprint*, arXiv:2411.04872.
- Aryan Gulati, Brando Miranda, Eric Chen, Emily Xia, Kai Fronsdal, Bruno de Moraes Dumont, and Sanmi Koyejo. 2025. [Putnam-AXIOM: A functional & static benchmark for measuring higher level mathematical reasoning in LLMs](#). In *Forty-second International Conference on Machine Learning*.
- Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. 2024. [OlympiadBench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850, Bangkok, Thailand. Association for Computational Linguistics.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the math dataset](#). In *NeurIPS Datasets and Benchmarks*.

- Wenda Li, Lei Yu, Yuhuai Wu, and Lawrence C. Paulson. 2021. [Isarstep: a benchmark for high-level mathematical reasoning](#). In *International Conference on Learning Representations*.
- Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. [Let’s verify step by step](#). In *The Twelfth International Conference on Learning Representations*.
- Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. [Program induction by rationale generation: Learning to solve and explain algebraic word problems](#). In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 158–167, Vancouver, Canada. Association for Computational Linguistics.
- Chengwu Liu, Jianhao Shen, Huajian Xin, Zhengying Liu, Ye Yuan, Haiming Wang, Wei Ju, Chuanyang Zheng, Yichun Yin, Lin Li, Ming Zhang, and Qun Liu. 2023. [Fimo: A challenge formal dataset for automated theorem proving](#). *Preprint*, arXiv:2309.04295.
- Jakub Macina, Nico Daheim, Sankalan Chowdhury, Tanmay Sinha, Manu Kapur, Iryna Gurevych, and Mrinmaya Sachan. 2023. [MathDial: A dialogue tutoring dataset with rich pedagogical properties grounded in math reasoning problems](#). In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 5602–5621, Singapore. Association for Computational Linguistics.
- Chaitanya Manem, Pratik Prabhanjan Brahma, Prakamya Mishra, Zicheng Liu, and Emad Barsoum. 2025. [Sand-math: Using llms to generate novel, difficult and useful mathematics questions and answers](#). *Preprint*, arXiv:2507.20527.
- Yujun Mao, Yoon Kim, and Yilun Zhou. 2024. [CHAMP: A competition-level dataset for fine-grained analyses of LLMs’ mathematical reasoning capabilities](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 13256–13274, Bangkok, Thailand. Association for Computational Linguistics.
- Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. 2020. [A diverse corpus for evaluating and developing English math word problem solvers](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online. Association for Computational Linguistics.
- Yiyi Sun, Shawn Hu, Georgia Zhou, Ken Jiankun Zheng, Hannaneh Hajishirzi, Nouha Dziri, and Dawn Song. 2026. [OMEGA: Can LLMs reason outside the box in math? evaluating exploratory, compositional, and transformative generalization](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Abhijit Suresh, Jennifer Jacobs, Charis Harty, Margaret Perkoff, James H. Martin, and Tamara Sumner. 2022. [The TalkMoves dataset: K-12 mathematics lesson transcripts annotated for teacher and student discursive moves](#). In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 4654–4662, Marseille, France. European Language Resources Association.
- Erik Y. Wang, Sumeet Motwani, James V. Roggeveen, Eliot Hodges, Dulhan Jayalath, Charles London, Kalyan Ramakrishnan, Flaviu Cipcigan, Philip Torr, and Alessandro Abate. 2026. [Horizonmath: Measuring ai progress toward mathematical discovery with automatic verification](#). *Preprint*, arXiv:2603.15617.
- Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024. [Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9426–9439, Bangkok, Thailand. Association for Computational Linguistics.
- Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hanna Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. [Naturalproofs: Mathematical theorem proving in natural language](#). In *NeurIPS Datasets and Benchmarks*.
- Kaiyu Yang, Aidan M Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan Prenger, and Anima Anandkumar. 2023. [Leandojo: Theorem proving with retrieval-augmented language models](#). In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Junyi Ye, Jingyi Gu, Xinyun Zhao, Wenpeng Yin, and Grace Guiling Wang. 2025. [Assessing the creativity of llms in proposing novel solutions to mathematical problems](#). In *AAAI*, pages 25687–25696.
- Albert S. Yue, Lovish Madaan, Ted Moskowitz, DJ Strouse, and Aaditya K. Singh. 2024. [Harp: A challenging human-annotated math reasoning benchmark](#). *Preprint*, arXiv:2412.08819.
- Jie Zhang, Cezara Petrucci, Kristina Nikolić, and Florian Tramèr. 2026. [Realmath: A continuous benchmark for evaluating language models on research-level mathematics](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.
- Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2022. [minif2f: a cross-system benchmark for formal olympiad-level mathematics](#). In *International Conference on Learning Representations*.

A Algorithm

Progress chain construction The raw annotations are assigned at the post level, but many labels are result-centric: they indicate that a post proposes, advances, proves, or critiques a particular mathematical result. To construct the dataset used in our experiments, we convert these flat post annotations into result-centric progress chains. Each chain corresponds to one completed result and contains the open-problem statement, any associated setup or resource text, and the ordered sequence of posts that contribute to that result.

This conversion is necessary because a single forum thread may contain multiple results, and a single post may contribute to more than one result. We therefore first identify all result identifiers referenced by the annotations, then build one chain per result by collecting all posts whose labels refer to that result. Posts are ordered by their original post number, preserving the temporal structure of the discussion. When a post contributes to a result completed in a different thread, we resolve the reference using post numbers that are recorded globally.

Algorithm 1 gives the simplified procedure used to construct these progress chains from the annotated posts.

A.1 Model Details

We evaluate six frontier models through OpenRouter: GPT-5.4, Claude Opus 4.6, Grok 4.20, DeepSeek V3.2, Gemini 2.5 Flash, and Qwen3-235B-A22B. For reproducibility, Table 4 reports the exact OpenRouter slug used for each model, along with the provider and the closest available official documentation, model card, system card, or technical report. Several of these API models do not have archival technical papers; in those cases, we cite official provider documentation or model cards. All model evaluations are run at temperature 0 with the remaining OpenRouter settings left at their defaults.

B Research Relevance

The dataset reflects CrowdMath’s focus on research-level problems approachable to high-school and undergraduate students. ‘Result’ denotes a previously unknown finding established in the course of discussion. 88/164 chains are linked to 12 papers, including in the Electronic Journal of Combinatorics and Discrete Applied Mathematics.

Of the 88 chains linked to 12 published papers, 22 concern extremal functions of forbidden sequences (3 papers), 21 pursuit–evasion games on graphs (1), 16 metric dimension of graphs (1), 10 graph coloring (1), 7 zero forcing on graphs (1), 6 factorization invariants (1), 3 neural codes on graphs (1), 2 arithmetic of power monoids (2), and 1 powerful numbers (1).

C Annotation Quality

Two independent experts double-annotated all 298 posts in the 2024 subset. Both have combinatorics expertise and close familiarity with the threads. We compare exact per-post canonical label sets for the six core types (Progress, Proof, Question, Answer, FindError, and Erroneous), including arguments and referenced IDs. Post-level agreement is appropriate because chains are generated deterministically from these core labels. Across all posts, 294 agreed and 4 disagreed ($P_o = 98.66\%$, $P_e = 0.8800$, Cohen’s $\kappa = 0.8882$). Among the 20 posts receiving a core label from either annotator, 16 agreed ($P_o = 80.00\%$, $P_e = 0.0650$, $\kappa = 0.7861$). The same four posts account for the disagreements in both analyses. One is a Progress-versus-Answer disagreement, although both link it to the same result. The other three occur in the same thread; two may reflect a one-post numbering offset. No disagreement involved Proof, no disagreement was Proof versus Progress. These results show strong agreement for 2024

D Evaluation Prompts

We report the prompts used for post-role classification and next-post prediction. Template variables shown as {variable_name} are substituted with instance-specific content at evaluation time. Both tasks are evaluated at temperature 0 with default settings via OpenRouter. Post-role classification uses a 4-shot prompt (one calibration example per class); Next-post prediction uses a 1-shot prompt.

D.1 Post-Role Classification

The model is given all preceding posts in a proof chain as context and must classify a single target post into one of four functional roles: PROGRESS, RESULT, ERRONEOUS, or FINDERROR. The prompt asks the model to reason through three explicit steps before producing a structured JSON output.

Model	Provider	OpenRouter slug	Official documentation / model card
GPT-5.4	OpenAI	openai/gpt-5.4	OpenAI model documentation: https://developers.openai.com/api/docs/models/gpt-5.4
Claude Opus 4.6	Anthropic	anthropic/claude-opus-4.6	Anthropic system card: https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf
Grok 4.20	xAI	x-ai/grok-4.20	xAI model documentation: https://docs.x.ai/developers/models/grok-4.20
DeepSeek V3.2	DeepSeek	deepseek/deepseek-v3.2	DeepSeek technical report: https://huggingface.co/deepseek-ai/DeepSeek-V3.2/resolve/main/assets/paper.pdf
Gemini 2.5 Flash	Google	google/gemini-2.5-flash	Google DeepMind model card: https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-2-5-Flash-Model-Card.pdf
Qwen3-235B-A22B-Instruct-2507	Qwen	qwen/qwen3-235b-a22b-2507	Hugging Face model card: https://huggingface.co/Qwen/Qwen3-235B-A22B-Instruct-2507

Table 4: Models used in our frontier-model evaluations. We report the exact OpenRouter slug used for inference and link to the closest official documentation, model card, system card, or technical report available for each model.

Prompt template

Proof Step Classification

You are an expert in mathematical reasoning and discourse analysis. You will be given a sequence of posts from a collaborative mathematics research thread, followed by a single post to classify.

<category_definitions>

Classify the post into exactly one of these four categories:

PROGRESS - The post makes a meaningful mathematical contribution toward the result

but does not complete the proof. This includes: introducing a useful lemma, or observation, proposing a viable strategy; narrowing the problem; or providing a partial argument that advances the thread.

RESULT - The post provides a complete proof of the result, or supplies the final missing step that, together with prior posts, constitutes a full proof.

ERRONEOUS - The post itself contains a fatal mathematical error or incorrect claim.

FIND_ERROR - The post identifies a fatal mathematical error in a previous post. The current post is correct -- its contribution is to catch a fatal flaw in earlier reasoning.

</category_definitions>

<instructions>

Evaluate the post by working through these steps in order:

Step 1 -- Summarise the post

In one sentence, state what mathematical claim or action the post performs.

Step 2 -- Assess against the prior context
Using only the preceding posts as context, determine:

- Does this post introduce new correct mathematical content that advances the proof?
- Does it complete the proof given what came before?
- Does it contain a fatal error of its own?
- Does it identify a fatal error in a prior post?

Step 3 -- Assign category

Apply the definitions above to assign exactly one of:
PROGRESS, RESULT, ERRONEOUS, FIND_ERROR.

If the post both finds a fatal error and makes new progress, assign FIND_ERROR -- surfacing the fatal error is the primary contribution since it invalidates prior work.

If the post contains a fatal error but also partially advances the thread, assign ERRONEOUS.

</instructions>

<output_format>

Provide your classification inside <classification> and </classification> tags using this JSON structure exactly with no text outside the tags:

Algorithm 1 Progress Chain Construction (Simplified)

```
1: Input:  $\mathcal{S}$ : the annotated posts from the input file, each post carrying its full label set
2: Output: for each result  $y$  in  $ResultSet$ : a progress chain  $\text{Chain}(y)$  with fields problem_text,
   problem_resources, and posts
   Step 1 — Collect all results referenced in the input
3:  $ResultSet \leftarrow \emptyset$ 
4: for each post  $s \in \mathcal{S}$  do
5:   for each label  $\ell$  on  $s$  that references a result  $y$  do
6:     add  $y$  to  $ResultSet$ 
7:   end for
8: end for
   Step 2 — Build the chain for each result  $y \in ResultSet$ 
9: for each  $y \in ResultSet$  do
10:   $Posts(y) \leftarrow \{s \in \mathcal{S} \mid \text{some label on } s \text{ references } y\}$ 
11:   $ProbPost(y) \leftarrow \text{post in } Posts(y) \text{ carrying } \text{Problem}(y), \text{ if any}$ 
12:   $t\text{-}n \leftarrow \text{open-problem ref carried by any post in } Posts(y), \text{ if any}$ 
13:  if  $t\text{-}n$  found then
14:     $Setup \leftarrow \text{Setup section(s) from } \text{json\_object\_data}[t\text{-}n], \text{ if any}$ 
15:     $ProbSecs \leftarrow \text{Problem section(s) from } \text{json\_object\_data}[t\text{-}n], \text{ if any}$ 
16:     $Resources \leftarrow \text{Relevant Resources sections from } \text{json\_object\_data}[t\text{-}n], \text{ if any}$ 
17:     $\text{problem\_text}(y) \leftarrow Setup \parallel ProbSecs \parallel ProbPost(y).\text{text}$ 
18:     $\text{problem\_resources}(y) \leftarrow Resources \text{ if non-empty, else null}$ 
19:  else
20:     $\text{problem\_text}(y) \leftarrow ProbPost(y).\text{text} \text{ if } ProbPost(y) \text{ exists, else null}$ 
21:     $\text{problem\_resources}(y) \leftarrow \text{null}$ 
22:  end if
23:   $Rest(y) \leftarrow Posts(y) - \{ProbPost(y)\}$  sorted in ascending order of post_number
24:   $\text{Chain}(y) \leftarrow \{\text{problem\_text}(y), \text{problem\_resources}(y), Rest(y)\}$ 
25: end for
```

```
{
  "summary": "<one sentence describing what
the post does>",
  "reasoning": "<2-3 sentences explaining
why this category applies,
referencing the post
content and prior context>",
  "label": "progress" | "result" | "
erroneous" | "find_error"
}
</output_format>

<calibration_examples>
[Four examples, one per class -- see below.]
</calibration_examples>

Prior context:
{prior_context}

Post to classify:
{question}

<judgement>
</judgement>
```

Calibration examples

The prompt includes one example per class, drawn from the 2016 pattern-avoidance thread. We show all four below.

Example 1 -- RESULT:

```
<post>
If  $n$  is even and at least one of  $j$  and
 $k$  is odd, player 2 wins. Say  $k$  is
odd, then after player 1 flips  $(i, 2d+1)$ ,
player 2 flips  $(i, 2d+2)$ , or vice
versa. So after each move of player 2,
column  $2d+1$  and  $2d+2$  are identical
for
all  $d$ , and the matrix cannot start to
contain  $P$  which has an odd number of
identical columns after player 2's move.
So, to generalize: if  $n$  is even and  $P$  is
composed of an odd number of identical
rows or columns,  $\mathcal{W}(n, P)=2$ .
</post>
```

```
<judgement>
{
  "summary": "The post proves that Player 2
wins when  $n$  is even and  $P$  has an odd
```

```

        number of identical rows or
        columns, by constructing an explicit
        mimic strategy.",
"reasoning": "Post #13 asked whether a
mimic strategy could prove  $W(n,P)=2$  for
certain cases. This post
answers that question completely: it
constructs the strategy (
mirror columns  $2d+1$  and  $2d+2$ ), proves it
keeps the matrix  $P$ -free, and
states the general result. The proof
is self-contained and
contains no gaps.",
"label": "result"
}
</judgement>

```

Example 2 -- PROGRESS:

```

<post>
It depends on what exactly "this strategy"
is, in particular the way of mimic
employed by player 2. If player 2 always
flips  $(i, 2d+1)$  after player 1 flips
 $(i, 2d+2)$  and vice versa, then a pattern
 $P$  with even number of columns, along
with column 1 identical to column 2, column
3 to column 4, etc., guarantees victory
for player 1.
As for all the matrices  $P$  for which player
2 can guarantee victory with this
strategy when  $n$  is even, I haven't got
better idea than those with odd number of
identical rows or columns.

I also wonder for an odd  $n$ , suppose the
center of  $n$ -by- $n$  matrix has
coordinate
 $(0, 0)$ . What patterns lead to player 1's
victory if they first flip  $(0, 0)$  and
then flips  $(-i, -j)$  after player 2 flips  $(i, j)$ ?
</post>

```

```

<judgement>
{
"summary": "The post gives a partial
answer to the characterization question
by
        identifying one class of
matrices where Player 1 wins under the
mimic
        strategy, but does not
complete the characterization.",
"reasoning": "Post #16 asked for a full
characterization of matrices  $P$  where
        Player 2 wins. This post
advances that question by exhibiting a
concrete counter-class (even
columns with paired identical columns
gives Player 1 a win), but
explicitly admits it has no general
characterization beyond the
odd-identical-columns case already known.
        The proof is not complete --
the thread remains open.",
"label": "progress"
}
</judgement>

```

Example 3 -- ERRONEOUS:

```

<post>
A simple thought. All player indices below
are  $\mathbb{Z}_d$ -based.
For an even  $k$ , we should have loser  $L_2(n, P) \equiv L_k(n, P) \pmod{2}$ , where
the subscript means the number of players.
This is because if  $L_2(n, P)=1$ , then
players  $0, 2, 4, \dots, k-2$  can
collectively deploy player  $0$ 's winning
strategy
as if there are only 2 players to make sure
not to lose.
To generalize,  $L_d(n, P) \equiv L_{\lfloor n/d \rfloor}(n, P) \pmod{d}$ .
</post>

```

```

<judgement>
{
"summary": "The post claims that players
can collectively deploy a two-player
        winning strategy in a  $k$ -
player game, and generalizes this to
 $L_d(n, P) = L_{\lfloor n/d \rfloor}(n, P) \pmod{d}$ .",
"reasoning": "The argument assumes players
 $0, 2, 4, \dots, k-2$  will cooperate to
        execute player  $0$ 's two-
player strategy, but gives no
justification
        for why the other players
would do so -- each player acts
        independently and may
deviate. The collective strategy argument
is
        unverified, making the
generalization unsupported.",
"label": "erroneous"
}
</judgement>

```

Example 4 -- FIND_ERROR:

```

<post>
[Quoting post #24's argument.]
Is there an issue if any players among  $0, 2, 4, \dots, k-2$  do not deploy
player  $0$ 's two-player winning strategy?

Also, what happens for  $m$  players when
there are multiple  $d$ -player winning
strategies?
</post>

<judgement>
{
"summary": "The post identifies two gaps
in post #24's collective-strategy
        argument: non-cooperative
players and ambiguity when multiple
        winning
        strategies exist.",
"reasoning": "The post correctly pinpoints
that post #24's proof requires all
        grouped players to follow
the same strategy, which is not
        guaranteed
        since players act
independently. It also raises the valid

```

```

concern
    that the argument breaks
down when the d-player winning strategy
is
    not unique. This post
contains no errors of its own -- it is a
    correct critique of prior
reasoning.",
"label": "find_error"
}
</judgement>

```

D.2 Next-Post Prediction

The model is given a thread prefix and four candidate posts (A–D). Exactly one is the true next post; the other three are distractors drawn from different chains in the same year. The model reasons in three steps before selecting a candidate. A single calibration example is included in the prompt.

You are an expert in mathematical reasoning and discourse analysis. You will be given a research problem, a thread prefix showing the discussion so far, and four candidate posts (A, B, C, D). Exactly one is the actual next contribution to the proof. The other three are real posts drawn from different threads and are distractors.

Your task is to identify which candidate is the most likely next post in this thread.

<instructions>
Work through these steps in order before selecting your answer.

Step 1 -- Summarise the proof state
In 2-3 sentences, describe what has been established in the thread so far.

Step 2 -- Evaluate each candidate
For each of A, B, C, D, write one sentence assessing whether it fits as the next post.
Check:
- Does it address an open question or gap identified in the thread prefix?
- Does it use notation and concepts consistent with the thread?
- Is it at the right level -- neither repeating what is already done nor jumping past the current stage of the proof?

Step 3 -- Select the best candidate
Choose the single candidate that best fits as the next post. If multiple candidates seem plausible, select the one that best fits the evaluation criteria.
</instructions>

<success_criteria>
The correct next post:
1. Follows directly from what the thread

prefix has established or asked.
2. Uses notation already introduced in the thread.
3. Makes progress at a level consistent with the thread's current stage.
4. Does not introduce a topic unrelated to the current proof direction.
</success_criteria>

<output_format>
Provide your selection inside <selection> and </selection> tags using this JSON structure exactly -- no text outside the tags:
{
 "proof_state": "<2-3 sentences summarising what is established>",
 "candidate_assessments": {
 "A": "<one sentence>",
 "B": "<one sentence>",
 "C": "<one sentence>",
 "D": "<one sentence>"
 },
 "answer": "A" | "B" | "C" | "D"
}
</output_format>

<calibration_example>
<thread_prefix>
For what patterns $\$P\$$ is it true that any algorithm to determine whether $\$A\$$ contains $\$P\$$ must have $\Omega(h_A w_A)$ running time?
?

Post #31: The direction of this question sounds interesting, but I feel it may need some tweak. For h_A and w_A , it seems a scan of most of the h_A elements are required. We can always have an A with most of the elements 0 , and yet still contains P . Scanning the $\approx h_A w_A$ zeros seems inevitable when we are unlucky.

Post #33: Is it true that any algorithm to determine whether A contains P must check almost every entry of A if at least one entry of P is 1 ?
What if P has all zeroes?
</thread_prefix>

<candidates>
A. Lemma: If $n = 2$ and $k = 2$, then $W(n, P)$ will always be Player 2.
Proof: Let A be the 2×2 all-zeroes matrix and P be the 2×1 all-ones matrix. Player 1 must change one zero to a one. In either scenario for Player 2's response, Player 1 is forced to create the forbidden pattern P on the next move.
B. I think it's positive: the number of ones of a minimally non-linear binary matrix with k rows is $O(k)$. We can ignore rows with a single one, because they contribute $O(k)$ ones. We claim that

every column has no more than $2k$ ones, followed by the fact that there are at most $4k-2$ columns.

C. If A is $n \times n$, then any algorithm has to check at least $n^2 - \text{Ex}(n,P)$ entries to declare A avoids P , because if it skips $\text{Ex}(n,P)+1$ elements then in the worst case those skipped elements are all 1s and contain P . So if $\text{Ex}(n,P) = O(n)$ then the proportion of elements that must be checked before declaring A avoids P is close to 1.

D. I think this family also solves crowdmath open problem 2. The edge metric dimension of the k -th graph in this family is $3^k(1-o(1))$ and the metric dimension is k . To see why, it suffices to prove: if G has order n and some vertex of degree $n-1-x$ within distance 2 of all vertices, then $\text{dim}(G) \geq n-1-x-2^x$.

</candidates>

<selection>

```
{
  "proof_state": "Post #31 observes that scanning most of an n x n matrix seems unavoidable when checking for P-avoidance. Post #33 sharpens this into two specific questions: must every algorithm check nearly all entries when P has at least one 1, and what happens when P is all-zero? The all-zero sub-case and the general lower bound remain open.",
  "candidate_assessments": {
    "A": "Proves W(2,2)=Player 2 in a pattern avoidance game -- completely unrelated to algorithm complexity or the Ex(n,P) framework of this thread.",
    "B": "Bounds the number of ones in a minimally non-linear matrix -- a structural combinatorics result unrelated to the algorithmic question about checking time.",
    "C": "Directly answers Post #33's first question by proving a lower bound of n^2 - Ex(n,P) checks using a worst-case argument, using Ex(n,P) notation already established in the thread.",
    "D": "Claims a result about edge metric dimension and metric dimension of a graph family -- from a different thread on a different topic entirely."
  },
  "answer": "C"
}
```

</selection>

</calibration_example>

Identify the most likely next post in this thread.

<thread_prefix>
{thread_prefix}
</thread_prefix>

<candidates>
A. {candidate_a}

B. {candidate_b}

C. {candidate_c}

D. {candidate_d}
</candidates>

<selection>
</selection>