MathArena: Evaluating LLMs on Uncontaminated Math Competitions

Mislav Balunović[1,2], **Jasper Dekoninck**[1], **Ivo Petrov**[2], **Nikola Jovanović**[1], **Martin Vechev**[1,2]
[1]ETH Zurich, [2]INSAIT, Sofia University
{mislav.balunovic,jasper.dekoninck,nikola.jovanovic,martin.vechev}@inf.ethz.ch,
ivo.petrov@insait.ai

https://matharena.ai/
https://github.com/eth-sri/matharena

Abstract

The rapid advancement of reasoning capabilities in large language models (LLMs) has led to notable improvements on mathematical benchmarks. However, many of the most commonly used evaluation datasets (e.g., AIME 2024) are widely available online, making it difficult to disentangle genuine reasoning from potential memorization. Furthermore, these benchmarks do not evaluate proof-writing capabilities, which are crucial for many mathematical tasks. To address this, we introduce MATHARENA, a new benchmark based on the following key insight: recurring math competitions provide a stream of high-quality, challenging problems that can be used for real-time evaluation of LLMs. By evaluating models as soon as new problems are released, we effectively eliminate the risk of contamination. Using this framework, we find strong signs of contamination in AIME 2024. Nonetheless, evaluations on harder competitions, such as CMIMC 2025, demonstrate impressive reasoning capabilities in top-performing models. MATHARENA is also the first benchmark for proof-writing capabilities. On IMO 2025, top models achieve slightly less than 40%, demonstrating both notable progress and significant room for improvement. So far, we have evaluated over 50 models across seven competitions, totaling 162 problems. As an evolving benchmark, MATHARENA will continue to track the progress of LLMs on newly released competitions, ensuring rigorous and up-to-date evaluation of mathematical reasoning.

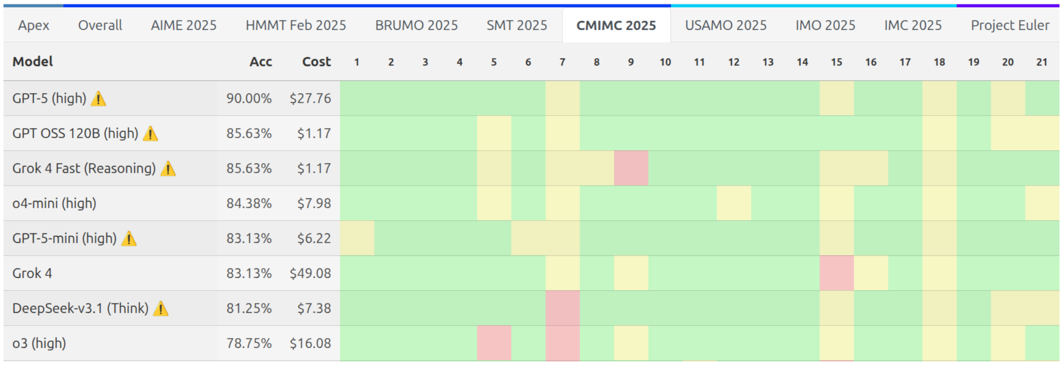
1 Introduction

Recent advances in the mathematical reasoning capabilities of large language models (LLMs) [20, 7] have raised the following three concerns about the adequacy of existing mathematics benchmarks:

1. Contamination risks: Many benchmarks are sourced from publicly available math competitions, which are accessible online and often used to train LLMs. This leaves them susceptible to data contamination, making it difficult to measure progress accurately. Data contamination can occur either through indirect inclusion of benchmark problems in training data or by using benchmark performance for hyperparameter tuning or model selection. For instance, we find that the popular AIME 2024 dataset is significantly contaminated by most leading LLMs, making the benchmark unsuitable for evaluating the models' capabilities.

2. High-cost, private benchmarks: To mitigate contamination, some leading benchmarks—such as FrontierMath [15] and HLE [29]—adopt a private, human-curated approach. While effective in avoiding data leakage, these datasets pose several major issues. First, their private nature raises concerns about reproducibility and transparency, making it impossible to verify the results accurately.

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Track on Datasets and Benchmarks.



Model	Acc	Cost	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
GPT-5 (high) ⚠	90.00%	$27.76	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟨	🟩
GPT OSS 120B (high) ⚠	85.63%	$1.17	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟨
Grok 4 Fast (Reasoning) ⚠	85.63%	$1.17	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟨	🟥	🟩	🟩	🟩	🟩	🟩	🟨	🟨	🟩	🟨	🟩	🟩	🟩
o4-mini (high)	84.38%	$7.98	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟩	🟨
GPT-5-mini (high) ⚠	83.13%	$6.22	🟨	🟩	🟩	🟩	🟩	🟨	🟨	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟩	🟩
Grok 4	83.13%	$49.08	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟥	🟨	🟩	🟨	🟩	🟩	🟩
DeepSeek-v3.1 (Think) ⚠	81.25%	$7.38	🟩	🟩	🟩	🟩	🟩	🟩	🟥	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟨	🟨
o3 (high)	78.75%	$16.08	🟩	🟩	🟩	🟩	🟥	🟩	🟥	🟩	🟨	🟩	🟩	🟩	🟩	🟩	🟨	🟩	🟩	🟨	🟩	🟨	🟩

Figure 1: MathArena leaderboard. The picture shows the CMIMC competition held in March 2025. Cell color denotes the pass rate of the model on the problem out of 4 attempts. Warning signs show a possible contamination risk due to the model being released after the competition.

Moreover, the benchmark creators may selectively grant access to certain organizations [10], creating an uneven playing field. And finally, the high cost of developing such datasets is prohibitive. For instance, HLE required a $500,000 prize pool to incentivize contributions.

3. Emphasis on final answers: Most existing benchmarks, including HLE and FrontierMath, primarily evaluate problems with single final answers. This can be misleading, as models may arrive at the correct answer through pattern recognition or brute-force enumeration, rather than genuine mathematical reasoning. Such benchmarks fall short of capturing the depth and rigor of problems found in mathematical olympiads, which often require detailed proofs and multi-step logic. Furthermore, most practical applications of LLMs in mathematics involve generating proofs or explanations, rather than simply providing final answers.

MathArena: A new benchmark for mathematical reasoning We introduce MathArena, a dynamic publicly available benchmark which addresses the above limitations by evaluating on newly released math competitions (see Fig. 1). Our core insight is that recurring math competitions produce a rich source of high-quality, uncontaminated problems. These problems are pre-vetted by the competition organizers for originality, ensuring that similar problems have not appeared previously and thereby reducing contamination risk. By evaluating models on competitions occurring after model release, MathArena eliminates the risk of contamination and offers a clean, forward-looking measure of progress. Furthermore, some of the included competitions (e.g., IMO 2025) have proof-based problems that are absent in other benchmarks. Unlike private or static benchmarks, MathArena is fully transparent, reproducible, and continuously updated throughout the year as new problems become available. This enables continuous adaptation to the evolving landscape of mathematical reasoning, ensuring that the included competitions remain relevant and challenging. We implement the entire MathArena pipeline for parsing, solving, and verifying problem solutions and release the code, data, and model responses as open source.

So far, we have evaluated over 50 models across seven competitions, totaling 162 problems. Our results indicate that GPT-5, Grok 4, and Gemini-2.5-Pro are the top-performing models on the included competitions, outperforming the top 1% of human participants. However, we also find room for improvement on proof-based competitions, with models scoring below 40% on the IMO 2025. This highlights the need for further research in this area.

Key contributions In summary, our key contributions are as follows:

- We introduce MathArena, a benchmark that leverages newly released competitions to evaluate LLMs, eliminating contamination while being fully transparent and reproducible.
- We propose a scalable evaluation pipeline for parsing, solving, and verifying problems from diverse competition formats, including final-answer and proof-based problems.
- We compare and thoroughly analyze the performance of state-of-the-art models on these competitions, highlighting significant progress made within the past year.

2 Related Work

In this section, we discuss the key prior approaches for evaluating mathematical reasoning.

Public answer-based benchmarks The most widely used benchmarks evaluate models by comparing their outputs to fixed ground-truth answers—typically numerical values or closed-form expressions. Early benchmarks such as GSM8K [6] and MATH [17] have largely been saturated by recent language models. Even more challenging competitions, like AIME 2024, have seen similar progress and are close to saturation. Omni-MATH [14], OlympiadBench [16], HARP [36], and OlymMATH [30] increase difficulty by incorporating harder problems from olympiad competitions. However, sourcing problems from past competitions, which have been available online for years, makes it difficult to track progress due to data contamination risks. This concern is supported by evidence in the case of GSM8K [37], and we confirm contamination of AIME 2024 in Section 4.

Private answer-based benchmarks FrontierMath [15] is a recently introduced private benchmark designed to be significantly more challenging, with problems that demand mathematical reasoning combined with a deep background in research-level mathematics. Similarly, Humanity's Last Exam [29] has collected a large number of private challenging problems across dozens of subjects. While their extreme difficulty makes an interesting target for frontier models, the private nature of these benchmarks makes standardized evaluation and fair model comparison difficult. Furthermore, this difficulty level makes tracking progress challenging, particularly for open source models and models on the Pareto frontier of cost-performance. Finally, the private nature of the benchmark raises concerns about reproducibility and transparency, as access to these benchmarks has been selectively granted to certain organizations [10].

Proof-based benchmarks Another line of evaluation focuses on verifying the correctness of reasoning traces rather than final answers. A common strategy is to require LLMs to generate formal proofs in systems like Lean, Coq, or Isabelle, enabling automatic verification. Datasets and benchmarks in this category include miniF2F [39], FIMO [21], PutnamBench [32], and LeanWorkbook [35]. However, these approaches often underutilize the natural language capabilities of LLMs and are limited by the models' ability to produce correct formal code. Concurrent work [22] reveals that models typically fail to generate fully rigorous proofs in natural language. Even for the correctly solved problems, the inclusion of the IMO shortlist problems likely leads to significant contamination, and the size of the benchmarks makes it infeasible to evaluate new models on all problems. GHOSTS [13] manually evaluate the proof-writing capabilities of GPT-4, but their benchmark is limited to just two older models and has not been updated since 2023.

Dynamic benchmarks To address contamination and adapt to evolving capabilities, some benchmarks are designed to be continuously updated with new problems. LiveBench [33], for instance, evaluates LLMs across domains including coding, data analysis, and mathematics. The mathematics portion in particular includes slightly harder than MATH-level problems, as well as fill-in-the-blank proof-based tasks, making it easier than MathArena, while also not evaluating rigorous proving capabilities. Another similar work to ours is LiveAoPSBench [23], which allows evaluating models on a snapshot of problems from a particular point in time. This can be seen as a retroactive simulation of live evaluation as performed in MathArena. However, the benchmark is not updated and does not contain problems from 2025, which precludes the evaluation of recent frontier models.

Perturbation-based benchmarks Another way to mitigate contamination risks is to generate new problems by perturbing existing ones [19, 24, 40]. While this strategy reduces overlap, it does not fully eliminate contamination: perturbed problems rely on the same underlying reasoning patterns. In contrast, our approach introduces entirely new problems that require new high-level reasoning strategies.

Other benchmarks Finally, some benchmarks adopt less conventional methods to evaluate mathematical reasoning. For example, MathTrap [38] evaluates logical consistency in model responses, while MathConstruct [8] focuses on problems that require constructive proofs. These approaches provide a more diverse view of the mathematical reasoning capabilities of the models. However, these benchmarks typically require expensive human data curation, which limits scalable evaluation.

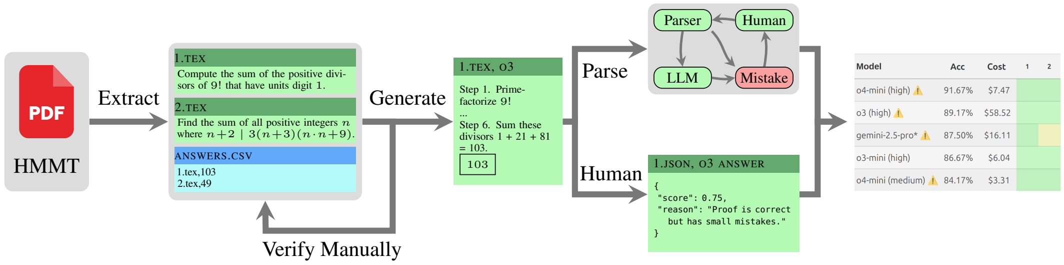

HMMT
PDF
Extract
1.tex
Compute the sum of the positive divisors of 9! that have units digit 1.
2.tex
Find the sum of all positive integers n where $n+2 \mid 3(n+3)(n \cdot n+9)$.
answers.csv
1.tex,103
2.tex,49
Verify Manually
Generate
1.tex, o3
Step 1. Prime-factorize 9!
...
Step 6. Sum these divisors 1 + 21 + 81 = 103.
103
Parse
Parser
Human
LLM
Mistake
Human
1.json, o3 answer
{
"score": 0.75,
"reason": "Proof is correct but has small mistakes."
}
Model
Acc
Cost
1
2
o4-mini (high) ⚠ 91.67% $7.47
o3 (high) ⚠ 89.17% $58.52
gemini-2.5-pro* ⚠ 87.50% $16.11
o3-mini (high) 86.67% $6.04
o4-mini (medium) ⚠ 84.17% $3.31

Figure 2: The pipeline for constructing MathArena. When a new competition is released, we first extract the problems and answers. We then query all selected models to obtain their responses. Depending on the problem type, we either use an automated parser or human graders for evaluation. Finally, we report scores on a public leaderboard with a GUI for viewing individual model answers

3 MathArena

In this section, we describe the pipeline used to construct MathArena, as shown in Fig. 2. The process begins by selecting a sufficiently challenging and reputable competition and extracting its problems and solutions (Section 3.1). Next, we evaluate a selected set of models on these problems, ensuring a fair comparison and avoiding data leakage (Section 3.2). Depending on the type of problem, either final-answer or proof-based, we use different methods for parsing and evaluation (Section 3.3). For final-answer problems, we use an automated rule-based parser to extract answers. For proof-based problems, human graders evaluate the model outputs. Finally, we compute leaderboard rankings and perform statistical post-processing to ensure accuracy and reliability (Section 3.4).

3.1 Competition Selection and Extraction

Table 1: Calendar of completed and planned competitions. N denotes the number of problems.

Competition	Type	Date	N	Current
AIME	Answer	Feb	30	✓
HMMT Feb.	Answer	Feb	30	✓
USAMO	Proof	Mar	6	✓
CMIMC	Answer	May	40	✓
BRUMO	Answer	Apr	30	✓
IMO	Proof	Jul	6	✓
Project Euler	Answer	-	20+	✓
MMATHS	Answer	Nov	TBD	✗
DMM	Answer	Nov	TBD	✗
PUMAC	Answer	Nov	TBD	✗
Putnam	Proof	Dec	12	✗

Competition selection To effectively repurpose high-quality math competitions for LLM evaluation, we carefully select which competitions to include in MathArena and ensure accurate formatting of each problem. Table 1 shows a calendar of competitions currently included in MathArena, along with additional competitions we plan to incorporate. At present, MathArena includes seven competitions comprising a total of 162 problems. We categorize competitions based on whether they consist of final-answer or proof-based problems. Final-answer competitions are easier to evaluate but tend to be less challenging. For these, we focus on high-difficulty competitions such as AIME (a qualifier for the USAMO) and several more difficult university-organized tournaments. We experimented with other well-known competitions, such as Kangaroo, and excluded them as they are already saturated by existing models.

Proof-based competitions pose a greater challenge and are more representative of deep mathematical reasoning. However, they also require manual evaluation, as scalable automated grading of proofs remains an open problem. To ensure high evaluation quality, we use human graders to evaluate proofs and focus on a small set of core competitions: USAMO (US high-school olympiad), IMO (International Math Olympiad), and the Putnam competition (US undergraduate level).

In addition to the standard mathematical competitions, we include problems from Project Euler [11], a popular online platform that emphasizes mathematical problem solving through code implementations. Unlike traditional competitions, Project Euler does not follow a fixed schedule or problem set. Instead, it maintains a continually expanding collection of problems. For evaluation, we focus only on the most recent problems and plan to update this subset regularly as new ones are released.

Problem extraction After selecting competitions, we extract the problems from their original sources and format them into a standardized template. We manually verify each problem for typographical errors, inconsistencies, or formatting issues.

3.2 Model Selection and Solution Generation

Model selection MATHARENA is continuously updated with newly released models. To avoid an overly cluttered leaderboard, we only select models that meet at least one of the following criteria: (i) the model competes for the top score in a given competition (e.g., GPT-5, GEMINI-2.5-PRO, GROK 4), (ii) the model competes for the top-performing open-weight option (e.g., DEEPSEEK-R1, QWEN3), or (iii) the model competes for a Pareto-optimal point on the cost-performance tradeoff curve (e.g., GROK 4 FAST, GPT-OSS-20B). We exclude non-reasoning models, as they consistently underperform reasoning models and do not satisfy any of the selection criteria.

Solution generation Each model is evaluated once per competition using the hyperparameters recommended by the model providers, without further tuning. This avoids overfitting and reduces the risk of information leakage. For answer-based competitions, we prompt the models to output their answer inside of a `boxed` environment, while for proof-based competitions, we prompt the models to output the entire proof. In App. D, we provide the prompts used for each competition. To account for stochasticity, each model generates four responses per question, and we report the average score across these runs. Models are evaluated close to the competition date, minimizing contamination risk. If a model was released after the competition date, this is clearly indicated on the leaderboard. Examples of model outputs and questions are shown in App. E.

Project Euler tools For Project Euler, we allow models to use tools to execute code, as this is often necessary to solve the problems. We provide a Python and C++ interpreter for this purpose. Models can generate code snippets that are executed in a secure sandbox environment, and the output can be used in subsequent reasoning steps. We limit the number of code executions to 20 per problem.

3.3 Solution Grading

Our grading strategy differs significantly between final-answer and proof-based problems. We outline details of both approaches below. These approaches are depicted in Fig. 2 with *Parse* (answer-based) and *Human* (proof-based) branches.

Answer-based competitions Answer-based competitions typically allow fairly accurate automated grading by extracting the final answer from `boxed` and using rule-based parsing on the extracted string. However, given the small size of these competitions, *fairly* accurate parsing is not good enough, as even minor parser errors can have a disproportionate impact. To this end, we develop a custom rule-based parser that converts arbitrary LaTeX strings into structured `sympy` expressions, capable of handling complex mathematical objects such as fractions, lists, and radicals. These expressions are then checked for equivalence with the ground truth answer using `sympy`. Since model outputs often vary in formatting, parser robustness is crucial. We implement two measures to ensure correctness.

First, we developed a GUI to support manual review of model answers, highlighting: (i) suspiciously short outputs, which may indicate truncation due to token limits, (ii) parser errors, and (iii) instances where the correct answer appears in the reasoning trace but is not successfully extracted. In the first case, if a model frequently exhibits this issue, we may consider re-running it with a different API provider, as the used provider likely limits the number of tokens per generation. In the other cases, we perform manual verification of all such flagged problems. Second, we incorporate an LLM-based judge, using the GEMINI-2.5-FLASH model, which evaluates whether the model's final answer is semantically equivalent to the ground truth. If the parser and judge disagree, we manually inspect the model response and update the parser as needed.

Proof-based competitions Automated grading is currently insufficient for proof-based problems, so we rely on expert human graders for precise grading. First, as competitions typically do not publish their grading scheme, expert graders develop a structured grading scheme meant to closely resemble the one used at the actual competition, e.g., rewarding points for partial progress. Next, graders receive anonymized solutions from the selected models and grade them according to the previously developed scheme. Two independent judges grade each solution, providing not only a final score but also a justification for their decision. We refer readers to [28] for further details of the procedure.

3.4 Leaderboard and Post-Processing

Once model outputs have been evaluated, we perform several post-processing steps to ensure the reliability of reported results. These include leaderboard construction and statistical variance estimation.

Table 2: The results of our numerical answer evaluation on the latest models evaluated on all competitions. Measured cost is the average cost to run a model one time on a single competition, and accuracy is the average accuracy across all 4 competitions. Green cells denote that the model was released after the competition date. Human performance is reported for the top 1% of participants in the AIME and HMMT competitions. For BRUMO and CMIMC, the human performance is not available.

Model	AIME	HMMT	BRUMO	CMIMC	Acc (avg)	Cost (avg)
GPT-5 (high)	95.0	88.3	91.7	90.0	91.3	4.83
Grok 4 Fast (Reasoning)	90.8	91.7	94.2	85.6	90.6	0.18
Grok 4	90.8	92.5	95.0	83.1	90.4	7.56
GPT OSS 120B (high)	90.0	90.0	91.7	85.6	89.3	0.21
DeepSeek-v3.2 (Think)	91.7	90.0	95.8	75.6	88.3	0.22
GPT-5-mini (high)	87.5	89.2	90.0	83.1	87.5	1.09
GLM 4.5	93.3	77.5	92.5	71.3	83.7	1.71
GPT OSS 20B (high)	89.2	75.0	85.0	72.5	80.4	0.22
gemini-2.5-pro	87.5	82.5	90.0	58.1	79.5	5.02
GPT-5-nano (high)	85.0	74.2	80.8	73.8	78.4	0.40
GLM 4.5 Air	83.3	69.2	90.0	70.6	78.3	0.90
Claude-Sonnet-4.5 (Think)	84.2	67.5	90.8	66.9	77.3	9.09
Human (Top 1%)	84.4	66.8	N/A	N/A	N/A	N/A

Leaderboard Results are published on a public leaderboard at `https://matharena.ai`. The interface is designed for ease of use, allowing users to navigate results, inspect individual model outputs, and verify parsing and grading decisions. This enables users to qualitatively analyze the models' performance and verify the correctness of our parser and grading process.

Variance estimation Due to the small size of most competitions, variance estimation is crucial for robust interpretation. We estimate variance for two key metrics: (1) model rank, important for comparative analysis, and (2) raw scores, which reflect absolute performance. To compute a confidence interval for model ranks, we use a paired permutation test to count the number of models significantly better or worse than a given model m_i at a significance level α, yielding a confidence interval for its rank. Details of the test can be found in App. C. To compute a confidence interval for the accuracy, we treat each answer as a Bernoulli trial with parameter $\hat{p}$ and compute variance as $\hat{p}(1-\hat{p})/N$, where N is the number of questions. $\hat{p}$ is estimated using model accuracy.

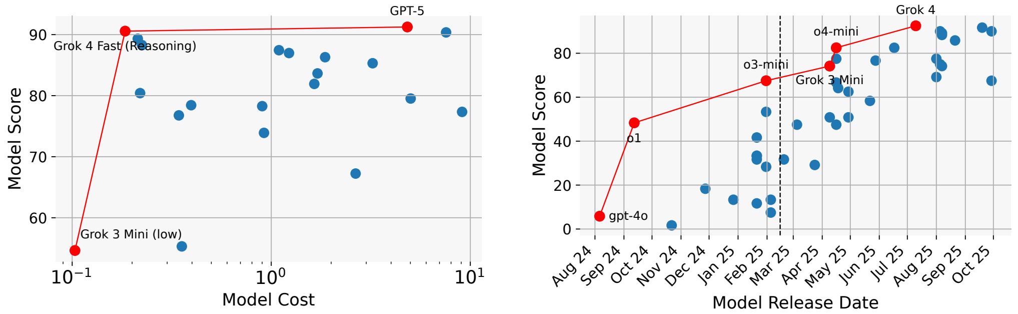
4 Evaluation

In this section, we present our evaluation of leading LLMs on MathArena. We also analyze the results to investigate data contamination, performance trends over time, and confidence intervals. Details on accessing the data and code used in our experiments, along with licensing information, are provided in App. A. To facilitate open research, we release all results and raw model responses on our website `https://matharena.ai`.

Setup We evaluated models on the following competitions from 2025: AIME [2, 3], HMMT [18],BRUMO [4], CMIMC [5], USAMO [26], IMO [12], and Project Euler [11]. Collectively, these competitions span 162 problems covering algebra, combinatorics, geometry, and number theory. USAMO and IMO are proof-based competitions, while the others require numerical final answers. We evaluated over 50 LLMs across all competitions, incurring approximately USD $2,000$ in API query costs for the experiments discussed in this paper, excluding development expenses.

4.1 Numerical Answer Competitions

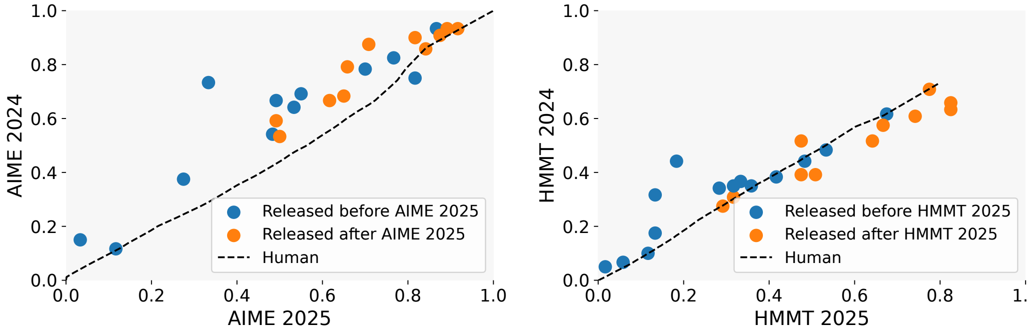
Our final-answer-based evaluation, excluding Project Euler, includes four competitions comprising 130 problems. We focus on non-deprecated models in this section and present full results in App. B. A model is deprecated once a strictly better version from the same provider is released (e.g., o3-mini is deprecated upon the release of o4-mini), after which it is excluded from future evaluations.



(a) Cost-Pareto frontier over all competitions

(b) Time-Pareto frontier for HMMT

Figure 3: Scores of models with respect to their release date and cost (in USD). Each dot represents a model; the red curves trace the Pareto frontier in both (a) cost vs. score for all competitions, (b) release-date vs. score for HMMT. The black dotted line indicates the release date of HMMT.



(a) Comparison between AIME 2024 and 2025

(b) Comparison between HMMT 2024 and 2025

Figure 4: Comparison between new and old competitions. The black dotted line indicates quantiles of human performance. Models above the human line are likely contaminated.

Main results Table 2 reports results for the best non-deprecated models at the time of writing. Following the evaluation protocol described in Section 3, each model was evaluated four times per problem, with accuracy computed using the pass@1 metric and no additional inference-time strategies (e.g., majority voting). Overall, the latest models demonstrate very strong performance. The best-performing models—GPT-5, Grok 4, and Grok 4 Fast—achieve accuracies of 91.3%, 90.6%, and 90.4%, respectively, with Grok 4 Fast being significantly cheaper. These models vastly outperform the top 1% of human participants in AIME and HMMT, indicating their capability to solve most problems correctly and compete with the best human contestants. Among open-source models, GPT-OSS-120B leads, closely followed by DeepSeek-v3.2 (Think).

Cost-accuracy Pareto frontier Fig. 3a shows the cost-accuracy Pareto frontier across all competitions. Cost reflects the money in USD needed to run a model on a full competition, averaged over all competitions. The frontier currently only includes three models from xAI and OpenAI.

Performance over time Fig. 3b illustrates the model scores on HMMT 2025 as a function of time. Each dot represents a model release, and the red line denotes the Pareto frontier of accuracy over time. The dashed vertical line marks the competition date, meaning models to the left of it are guaranteed to be uncontaminated. We show similar plots for other competitions in App. B. We observe that models released before September 2024 achieved less than 10% accuracy (e.g., GPT-4o). Performance significantly improved with the release of chain-of-thought reasoning models like o1 and continued to rise with subsequent iterations.

Data contamination of past competitions A key aim of our study is to evaluate the reliability of model performance on older competitions, particularly AIME 2024, where contamination may have occurred. Fig. 4a and Fig. 4b compare model scores on the 2024 and 2025 versions of AIME and HMMT. The x-axis shows performance on the 2025 version, while the y-axis shows the 2024 score. The dotted line represents human performance quantiles, enabling us to account for difficulty changes between the years, as the same human quantile is expected to yield similar performance across years. Most models lie above this line on AIME with a margin of $10\% - 20\%$, suggesting inflated performance on AIME 2024 due to data contamination. QwQ-Preview-32B is a notable outlier and outperforms the expected human-aligned performance by nearly 60%, indicating extreme contamination. In contrast, the discrepancy is much smaller for HMMT, indicating more trustworthy results—likely because HMMT is less prominent and less likely to be included in training datasets.

Another possible source for contamination of a new competition is that versions of problems from the new competition may have already appeared online, either in past contests or online forums. We investigate this for AIME 2025 and HMMT 2025 using DeepResearch [27], and find that 8 problems from AIME 2025 and 1 problem from HMMT 2025 can be found online in a similar form. We find that these are mostly easier problems that do not affect the overall results, but they underscore an interesting caveat of evaluating in future competitions. Details are provided in App. B.

Confidence intervals Most existing benchmarks for large language models rely on large datasets, raising concerns that the variance in a single competition may be too high to yield meaningful conclusions. In contrast, small competitions are often used to evaluate human participants, indicating that they can be reliable.

Using the methodology from Section 3.4, we compute 95% confidence intervals for rank and accuracy across all competitions. Table 3 shows these intervals averaged across competition, with per-competition intervals shown in App. B. Despite the smaller size, MathArena can reliably differentiate between most models. In particular, rank intervals are relatively small, with the top three models being GPT-5, Grok 4 Fast, and Grok 4, all within 1% of each other.

Table 3: Variance in the performance of models averaged for all competitions. 95% confidence intervals are shown for both rank and accuracy.

Model	**Rank**	**Acc (avg)**
GPT-5 (high)	1-4	91.25 ± 2.4
Grok 4 Fast (Reasoning)	1-5	90.57 ± 2.5
Grok 4	1-5	90.36 ± 2.5
GPT OSS 120B (high)	1-7	89.32 ± 2.6
DeepSeek-v3.2 (Think)	2-8	88.28 ± 2.6
GPT-5-mini (high)	4-9	87.45 ± 2.8
GLM 4.5	8-11	83.65 ± 3.0
GPT OSS 20B (high)	11-16	80.42 ± 3.4
gemini-2.5-pro	11-17	79.53 ± 3.2
GPT-5-nano (high)	12-17	78.44 ± 3.5
GLM 4.5 Air	12-17	78.28 ± 3.5
Claude-Sonnet-4.5 (Think)	12-17	77.34 ± 3.5

Repeating runs As a more intuitive approach to understanding variance, we follow Abdin et al. [1] and perform repeated evaluations. Specifically, we select several representative models (o4-mini (medium), Qwen3-30B-A3B, DeepSeek-R1-Distill-32B, and DeepSeek-R1-Distill-14B), sample 100 solutions per problem, and derive 25 score estimates per model using 4 per-problem samples as described in Section 3.4. We then fit kernel density estimates (KDEs) to these score distributions. The results show that the score distributions are sharp, validating our methodology of averaging accuracy over four runs.

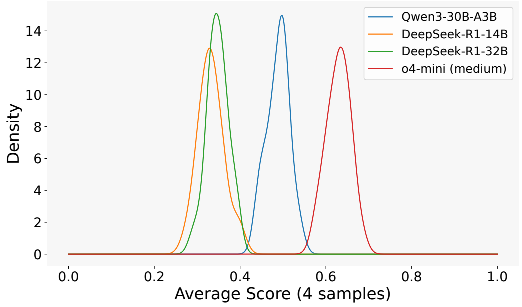


Figure 5: Distribution of the 4-sample accuracy estimates of several models for HMMT.

Cross-competition correlation We additionally compute the Spearman correlation between different competitions. A high correlation indicates consistent model rankings and suggests that a single competition is representative of overall performance. AIME, HMMT, and CMIMC all show correlations above 80%, clearly indicating that results from one competition generalize well to other similar competitions. The high overall correlation supports the conclusion that single-competition evaluations are generally robust.

4.2 Project Euler

Setup We evaluated six state-of-the-art models on Project Euler: GPT-5, o4-mini, Grok 4, Grok 4 Fast, Gemini-2.5-Pro, and Claude-Sonnet-4.5. These models were selected based on their strong performance in other competitions within MathArena. Since these problems typically require programming to solve, we allow models to use tools to execute code, as described in Section 3.

Results As shown in Table 4, GPT-5 achieved the highest accuracy of 55%, followed by Grok 4 and its faster and cheaper variant at 47.5%. Claude-Sonnet-4.5 and Gemini-2.5-Pro lag behind, achieving accuracies of 16.25% and 12.5%, respectively.

Table 4: Performance on Project Euler with tools. 95% confidence intervals are shown for both rank and accuracy. Cost reflects the money in USD needed to run a model on all 20 problems.

Model	Rank	Acc	Cost
GPT-5 (high)	1-3	55.00 ± 10.9	21.58
Grok 4 Fast	1-4	47.50 ± 10.9	2.22
Grok 4	1-4	47.50 ± 10.9	46.92
o4-mini (high)	2-4	43.75 ± 10.9	11.38
Claude-Sonnet-4.5	5-6	16.25 ± 8.1	16.93
gemini-2.5-pro	5-6	12.50 ± 7.2	6.90

4.3 Evaluating Natural Language Proofs

One of the core goals of MathArena is to evaluate models on proof-based math competitions, particularly the USAMO [26], IMO [12], and Putnam [25]. Of these, USAMO 2025 and IMO 2025 have occurred at the time of writing. We conducted evaluations immediately after problem release using the procedure described in Section 3. More details about the evaluation for USAMO 2025 can be found in our previous report [28]. In this section, we discuss the results of IMO 2025.

Model selection and evaluation We evaluated six state-of-the-art models: GPT-5, o3, o4-mini, Gemini-2.5-Pro, Grok 4, and DeepSeek-R1-0528. We applied the best-of-n selection strategy introduced by Dekoninck et al. [9], selecting the best proof from 32 samples per problem. In this process, the model itself serves as a judge in a bracket tournament between the generated proofs, choosing the winner of each round until a final proof is selected. Prompts for this procedure are provided in App. D.

Results GPT-5 achieved the highest score, with an average of 38% (16 points). Although this result may appear modest, especially given the 200 dollars spent to generate only 24 answers, it nonetheless represents strong performance given the exceptional difficulty of the IMO. However, 16 points fall short of the 19 required for a bronze medal (19/42). Full results are available on our leaderboard, where individual responses and judge feedback can be explored in detail. Several examples of model responses are given in App. E. Because the number of problems is small, the rank confidence intervals are wider than in numerical competitions. We therefore recommend caution when interpreting the results, particularly when comparing models with similar scores.

Qualitative analysis We highlight several qualitative findings from our evaluation. First, Grok 4 performed considerably below expectations. Many of its initial responses were extremely brief, often providing only a final answer without explanation. Similar patterns can be seen on other MathArena benchmarks, where Grok 4 frequently produces answers with little depth or justification. In contrast, Gemini-2.5-Pro shows a different issue: when it fails to find a valid proof, it often cites non-existent theorems. This is especially problematic because it misleads users by presenting false authority, thereby undermining trust in the model's reasoning. While this behavior was less common in the IMO responses compared to the USAMO [28], it remains a concern. On a more positive note, compared to earlier evaluations [28], we observed fewer formatting errors and fewer cases of models over-optimizing for final-answer styles, such as boxing entire proofs or assuming that every response must be numerical. This suggests progress in handling open-ended mathematical reasoning tasks more reliably. Finally, one of our judges briefly reviewed a subset of the 32 raw responses produced by the models before the best-of-n selection. They noted that many of these raw responses were very weak and estimated that, without filtering, model scores would likely have dropped below 10%. Interestingly, the judge also observed that some unselected answers appeared more coherent than the chosen ones, yet contained more factual errors.

5 Discussion

We briefly describe the limitations and broader impact of our work.

Limitations There are only a limited number of annual competitions that are sufficiently challenging to serve as effective benchmarks for state-of-the-art LLMs. As a result, the size of MathArena

Table 5: Main results of our evaluation. Problems are scored out of 7 points, with the maximum possible total score being 42. Listed scores are averaged over all four runs. We measure cost in USD, and report the average score across all generations and graders for each problem.

Model	**P1** (/7)	**P2** (/7)	**P3** (/7)	**P4** (/7)	**P5** (/7)	**P6** (/7)	**Total** (/42)	**Cost (avg)**
GPT-5 (high)	2.3	0.0	1.8	5.3	6.8	0.0	16.0	53.61
Gemini-2.5-Pro	1.0	0.0	5.0	3.3	4.0	0.0	13.3	107.99
o3 (high)	0.0	0.0	0.5	2.5	4.0	0.0	7.0	55.83
o4-mini (high)	1.1	0.0	0.4	3.3	1.3	0.0	6.0	25.84
Grok 4	0.9	0.3	1.3	0.9	1.8	0.0	5.0	131.96
DeepSeek-R1-0528	0.3	0.0	0.4	0.0	2.3	0.0	2.9	14.88

remains small, leading to relatively wide confidence intervals in our results. However, we expect this to improve over time as more competitions are added, gradually reducing uncertainty. Furthermore, current state-of-the-art models already solve nearly all but the most difficult questions in answer-based competitions. This suggests that such benchmarks may soon become saturated, possibly as early as 2026. To maintain meaningful evaluations, we anticipate the need to identify or design more challenging competitions. Unlike static benchmarks, however, the dynamic nature of MathArena allows it to evolve alongside model capabilities, ensuring continued relevance as the field progresses.

Further, there are some potential concerns about residual data contamination arising from the time gap between a model's release and the competition date. On our leaderboard, we clearly indicate models that were released after the competition data. However, since there is a time gap between the public release of a competition and our evaluation, it is theoretically possible that closed-source models could be updated with the new competition data before we evaluate them. In practice, however, our evaluations are conducted only a few hours to at most a few days after the competition concludes, while current training pipelines require much longer to incorporate new data. For these reasons, we believe that contamination risks in our setting are minimal.

Broader impact MathArena has already made a notable impact on the field. Several major model providers have cited MathArena results in their release notes, including Phi-4-Reasoning [1], Gemini-2.5-Pro [31], and Grok-3 [34]. In February, we were the first to demonstrate that the performance of reasoning-focused LLMs on older math competitions generalizes well to newer ones. Our work has gotten significant community interest, and we expect MathArena to remain a valuable and adaptive resource, supporting the ongoing evaluation of LLMs by keeping the benchmark both challenging and aligned with the evolving frontier of model capabilities.

6 Conclusion

We introduced MathArena, a benchmark designed to evaluate the mathematical performance of large language models (LLMs) using uncontaminated problems from human math competitions. The key insight was that such competitions generate a diverse set of challenging and naturally uncontaminated problems, making them ideal for rigorous evaluation. To support this, we developed a scalable pipeline that parses problems and answers, samples model solutions, extracts final answers, and verifies correctness. Using this framework, we evaluated over 50 LLMs on 162 problems from seven math competitions held in 2025. Our results show substantial progress in LLMs' mathematical capabilities while also confirming the impact of data contamination in prior benchmarks.

Acknowledgments

This research was partially funded by the Ministry of Education and Science of Bulgaria (support for INSAIT, part of the Bulgarian National Roadmap for Research Infrastructure). This project was supported with computational resources provided by Google Cloud Platform (GCP). This work has received funding from the Swiss State Secretariat for Education, Research and Innovation (SERI) (SERI-funded ERC Consolidator Grant).

References

[1] Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, Piero Kauffmann, Yash Lara, Caio César Teodoro Mendes, Arindam Mitra, Besmira Nushi, Dimitris Papailiopoulos, Olli Saarikivi, Shital Shah, Vaishnavi Shrivastava, Vibhav Vineet, Yue Wu, Safoora Yousefi, and Guoqing Zheng. Phi-4-reasoning technical report, 2025. URL `https://arxiv.org/abs/2504.21318`.

[2] Art of Problem Solving. 2025 aime i. Art of Problem Solving Wiki, 2025. URL `https://artofproblemsolving.com/wiki/index.php/2025_AIME_I`. Accessed: 2025.

[3] Art of Problem Solving. 2025 aime ii. Art of Problem Solving Wiki, 2025. URL `https://artofproblemsolving.com/wiki/index.php/2025_AIME_II`. Accessed: 2025.

[4] BRUMO. Brown university math olympiad 2025, 2025. URL `https://www.brumo.org/`. Accessed: 2025.

[5] CMIMC. Cmimc 2025, 2025. URL `https://cmimc.math.cmu.edu/`. Accessed: 2025.

[6] 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. *CoRR*, abs/2110.14168, 2021.

[7] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[8] Jasper Dekoninck, Mislav Balunovic, Nikola Jovanović, Ivo Petrov, and Martin Vechev. Math-construct: Challenging llm reasoning with constructive proofs. In *ICLR 2025 Workshop: VerifAI: AI Verification in the Wild*.

[9] Jasper Dekoninck, Ivo Petrov, Kristian Minchev, Mislav Balunovic, Martin T. Vechev, Miroslav Marinov, Maria Drencheva, Lyuba Konova, Milen Shumanov, Kaloyan Tsvetkov, Nikolay Drenchev, Lazar Todorov, Kalina Nikolova, Nikolay Georgiev, Vanesa Kalinkova, and Margulan Ismoldayev. The open proof corpus: A large-scale study of llm-generated mathematical proofs. *CoRR*, abs/2506.21621, 2025. doi: 10.48550/ARXIV.2506.21621. URL `https://doi.org/10.48550/arXiv.2506.21621`.

[10] Epoch. Openai and frontiermath. *Epoch AI Blog*, 2024. URL `https://epoch.ai/blog/openai-and-frontiermath`.

[11] Project Euler. Project euler, 2025. URL `https://projecteuler.net/`. Accessed: 2025.

[12] IMO Foundation. International mathematical olympiad, 2025. URL `https://www.imo-official.org/`. Accessed: 2025.

[13] Simon Frieder, Luca Pinchetti, Alexis Chevalier, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. Mathematical capabilities of chatgpt. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine, editors, *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL `http://papers.nips.cc/paper_files/paper/2023/hash/58168e8a92994655d6da3939e7cc0918-Abstract-Datasets_and_Benchmarks.html`.

[14] 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. Omni-math: A universal olympiad level mathematic benchmark for large language models. *CoRR*, abs/2410.07985, 2024.

[15] 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*, 2024.

[16] Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting AGI with olympiad-level bilingual multimodal scientific problems. In *ACL (1)*, pages 3828–3850. Association for Computational Linguistics, 2024.

[17] 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 *NeurIPS Datasets and Benchmarks*, 2021.

[18] HMMT. Hmmt 2025, 2025. URL `https://www.hmmt.org/`. Accessed: 2025.

[19] Kaixuan Huang, Jiacheng Guo, Zihao Li, Xiang Ji, Jiawei Ge, Wenzhe Li, Yingqing Guo, Tianle Cai, Hui Yuan, Runzhe Wang, Yue Wu, Ming Yin, Shange Tang, Yangsibo Huang, Chi Jin, Xinyun Chen, Chiyuan Zhang, and Mengdi Wang. Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations. *CoRR*, abs/2502.06453, 2025. doi: 10.48550/ARXIV.2502.06453. URL `https://doi.org/10.48550/arXiv.2502.06453`.

[20] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

[21] 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. FIMO: A challenge formal dataset for automated theorem proving. *CoRR*, abs/2309.04295, 2023.

[22] Hamed Mahdavi, Alireza Hashemi, Majid Daliri, Pegah Mohammadipour, Alireza Farhadi, Samira Malek, Yekta Yazdanifard, Amir Khasahmadi, and Vasant Honavar. Brains vs. bytes: Evaluating llm proficiency in olympiad mathematics. *arXiv preprint arXiv:2504.01995*, 2025.

[23] Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Christos Thrampoulidis, Leonid Sigal, and Renjie Liao. Leveraging online olympiad-level math problems for llms training and contamination-resistant evaluation. *CoRR*, abs/2501.14275, 2025.

[24] Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=AjXkRZIvjB`.

[25] Mathematical Association of America. 2025 putnam mathematical competition, 2025. URL `https://maa.org/putnam/`. Accessed: 2025.

[26] Art of Problem Solving. 2025 usa math olympiad, 2025. URL `https://artofproblemsolving.com/wiki/index.php/2025_USAMO`. Accessed: 2025.

[27] OpenAI. Deep research, 2025. URL `https://openai.com/index/introducing-deep-research/`.

[28] Ivo Petrov, Jasper Dekoninck, Lyuben Baltadzhiev, Maria Drencheva, Kristian Minchev, Mislav Balunović, Nikola Jovanović, and Martin Vechev. Proof or bluff? evaluating llms on 2025 usa math olympiad. *arXiv preprint arXiv:2503.21934*, 2025.

[29] Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Jason Hausenloy, Oliver Zhang, et al. Humanity's last exam. *arXiv*, 2025.

[30] Haoxiang Sun, Yingqian Min, Zhipeng Chen, Wayne Xin Zhao, Zheng Liu, Zhongyuan Wang, Lei Fang, and Ji-Rong Wen. Challenging the boundaries of reasoning: An olympiad-level math benchmark for large language models. *CoRR*, abs/2503.21380, 2025.

[31] Gemini Team. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261, 2025. doi: 10.48550/ARXIV.2507.06261. URL `https://doi.org/10.48550/arXiv.2507.06261`.

[32] 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. *CoRR*, abs/2407.11214, 2024.

[33] Colin White, Samuel Dooley, Manley Roberts, Arka Pal, Ben Feuer, Siddhartha Jain, Ravid Shwartz-Ziv, Neel Jain, Khalid Saifullah, Siddartha Naidu, et al. Livebench: A challenging, contamination-free llm benchmark. *arXiv preprint arXiv:2406.19314*, 2024.

[34] xAI Team. Grok 3 beta — the age of reasoning agents, February 2025. URL `https://x.ai/news/grok-3`. News post.

[35] Huaiyuan Ying, Zijian Wu, Yihan Geng, Jiayu Wang, Dahua Lin, and Kai Chen. Lean workbook: A large-scale lean problem set formalized from natural language math problems. *arXiv preprint arXiv:2406.03847*, 2024.

[36] Albert S. Yue, Lovish Madaan, Ted Moskovitz, DJ Strouse, and Aaditya K. Singh. HARP: A challenging human-annotated math reasoning benchmark. *CoRR*, abs/2412.08819, 2024.

[37] Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, William Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, et al. A careful examination of large language model performance on grade school arithmetic. *Advances in Neural Information Processing Systems*, 37:46819–46836, 2024.

[38] Jun Zhao, Jingqi Tong, Yurong Mou, Ming Zhang, Qi Zhang, and Xuanjing Huang. Exploring the compositional deficiency of large language models in mathematical reasoning. *arXiv preprint arXiv:2405.06680*, 2024.

[39] Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *ICLR*. OpenReview.net, 2022.

[40] Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL `https://openreview.net/forum?id=VOAMTA8jKu`.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: As we introduce a new benchmark in this paper, and we do provide the benchmark, it is clear that the claims made in the abstract and introduction are accurate. The only other claim we make is the strong signs of contamination in AIME 2024, which is supported by Fig. 4a.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: In Section 5 we discuss the limitations of MATHARENA.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: There are no theoretical results in this paper.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide all the information needed to reproduce the main experimental results in Section 3. Furthermore, our code contains detailed and clear instructions on how to reproduce the results.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results in App. A.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: In Section 3 we provide a clear description of the experimental setup. Furthermore, our code contains detailed and clear instructions on how to reproduce the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We dedicate paragraphs in Section 4 to the statistical significance of our results, clearly stating the method used to compute the confidence intervals. Furthermore, we provide all results with confidence intervals in App. B.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: In Section 4, we clearly indicate the costs for running each model on each competition, and mention the total cost of running all experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We confirm to have read and understood the NeurIPS Code of Ethics, and we confirm that our research is in accordance with it.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss the potential positive and negative societal impacts of our work in Section 5.

    Guidelines:

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: None of the datasets released in this paper pose a high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: We proactively contact the authors of the competitions to ask for permission to use their datasets. In accordance with the discussions with the authors, we released our datasets under the CC-BY-NC-SA 4.0 license. In App. A we describe the licenses of the datasets used in our work in more detail.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.

- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Our code and benchmark are well documented, and satisfy the requirements of the NeurIPS Benchmarks & Datasets track.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: This work does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: This work does not involve LLMs as any important, original, or non-standard components.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

A Code and Data Availability and Reprodicibility

This section outlines the availability of code and data used in our benchmark. Our code is publicly available at `https://github.com/eth-sri/matharena`. Regarding data availability, we typically publish datasets on HuggingFace at `https://huggingface.co/MathArena`. All data is available under the CC-BY-NC-SA 4.0 license, which allows for non-commercial use and modification, provided that the original source is credited. This license was chosen after consultation with the competition organizers. In particular, we reached out to all competition organizers to ensure that our data release complies with their policies. All organizers agreed to the use of their questions under the CC-BY-NC-SA 4.0 license.

B Additional Results

Full main results In Table 6 we include the complete results of our benchmark on many proprietary and open-source LLMs. We did not evaluate poorly performing models, or models that were superseded by better versions on the CMIMC or BRUMO competitions.

Domain-specific results While the best-performing models tend to show consistent results across different competitions, their performance varies significantly across mathematical problem domains. We manually classified each problem into one of four standard high-school competition categories: Algebra, Combinatorics, Geometry, and Number Theory, as shown in Table 7. Calculus problems were grouped under Algebra, while non-standard or word-based problems were categorized under Combinatorics.

Table 7: Distribution of problem types per competition. Some problems were assigned multiple domains, as they combined concepts from more than one area.

Competition	**Algebra**	**Comb.**	**Geo.**	**NT**
AIME	9	9	8	6
HMMT FEB.	7	10	11	4
CMIMC	8	14	14	5
BRUMO	7	10	8	5
TOTAL	31	43	41	20

As seen in Table 8, nearly all models struggle more with combinatorial and geometric problems—domains that typically require greater creativity capabilities. In the domain of Geometry, models consistently struggle with visualizing constructions or applying synthetic reasoning. Instead, most correct solutions rely on analytical approaches—typically brute-force coordinate methods. As a result, even weaker models can solve simpler problems using these methods. However, when compared to other mathematical domains, stronger models show relatively poorer performance on Geometry tasks, likely because the domain requires spatial intuition and reasoning that current models lack. In contrast, when problems require standard techniques or symbolic manipulation, as is often the case in Algebra and Number Theory, LLMs show significantly stronger performance.

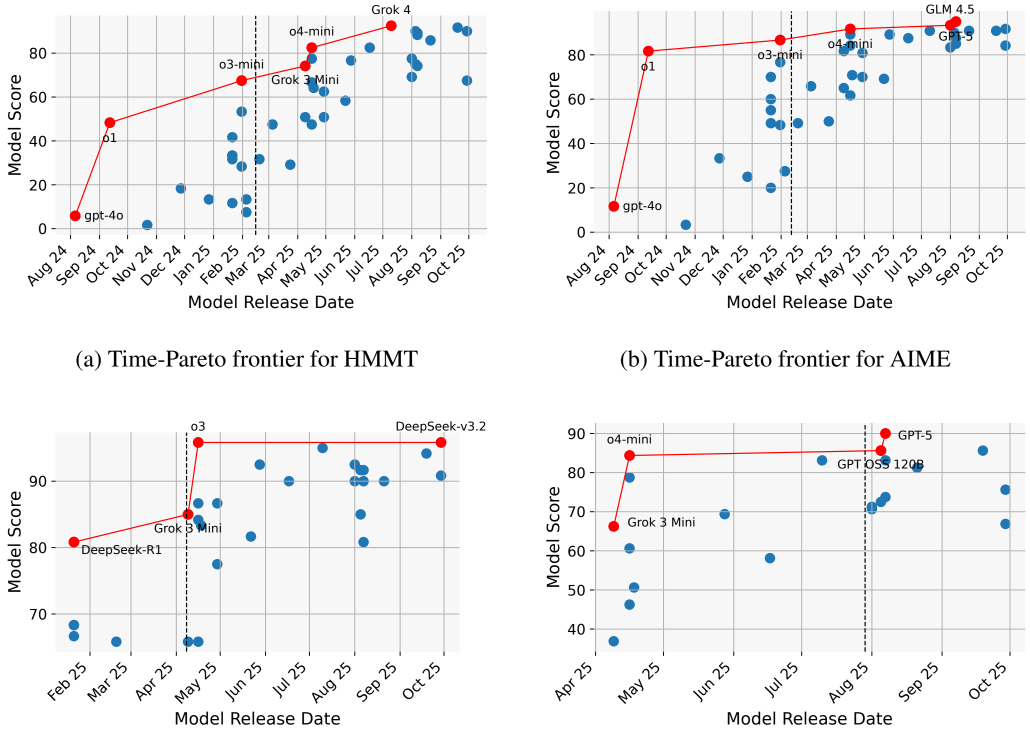
Confidence intervals per competition In Table 9,Table 10, Table 11, and Table 12 we show the confidence intervals for the results of each competition using the method described in Section 3.

Timeline for all competitions In Fig. 6 we show the Pareto frontiers for all competitions in function of time. The red curves trace the Pareto-optimal points in release-date vs. accuracy for all competitions. The black dotted lines mark the competition release dates.

Token usage per model As shown in Table 13, we also tracked the number of tokens used by almost all model during evaluation. This includes both prompt and response tokens, averaged over all problems from final-answer competitions. The higher number of input tokens for GPT-5 is due to caching of response tokens in some cases.

Data contamination of past competitions We used DeepResearch [27] to search the internet for problems similar to those in the AIME 2025 and HMMT 2025 competitions. We found the following sources that may be similar to the problems in the AIME 2025 and HMMT 2025 competitions:

- AIME 2025 Problem 1: `https://www.quora.com/In-what-bases-b-does-b-7-divide-into-9b-7-without-any-remainder`



(a) Time-Pareto frontier for HMMT

(b) Time-Pareto frontier for AIME

(c) Time-Pareto frontier for BRUMO

(d) Time-Pareto frontier for CMIMC

Figure 6: Accuracy of models with respect to their release date and cost. Each dot represents a released model; the red curves trace the Pareto-optimal points in release-date vs. accuracy for all competitions. Black dotted lines mark the competition release dates.

- AIME 2025 Problem 3: `https://math.stackexchange.com/questions/3548821/number-of-combination-of-incremental-numbers-for-a-number` (reported by Dimitris Papailiopoulos)

- AIME 2025 Problem 5: `https://math.stackexchange.com/questions/3146556/how-many-five-digit-numbers-formed-from-digits-1-2-3-4-5-used-exactly-once-a` (reported by Dimitris Papailiopoulos)

- AIME 2025 Problem 6: `https://www.wyzant.com/resources/answers/105893/isosceles_trapezoid_abcd_contains_inscribed_circle_o_the_area_of_the_trapezoid_is_510_square_centimeters_the_radius_of_the_circle_is_10_cm_find_ad`

- AIME 2025 Problem 10: `https://math.stackexchange.com/questions/1923331/how-many-different-ways-can-the-numbers-1-9-be-arranged-in-a-3x9-grid`

- AIME 2025 Problem 17: `https://math.stackexchange.com/questions/2526124/find-all-positive-integers-n-such-that-n-3-divides-n2-27#:~:text=4e`

- AIME 2025 Problem 22: `https://math.stackexchange.com/questions/2145033/counting-all-sets-with-the-same-least-common-multiple`

- AIME 2025 Problem 25: `https://math.stackexchange.com/questions/4824833/re-visit-combinatorics-involving-3-adjacent-objects`

- HMMT 2025 Problem 15: `https://puzzling.stackexchange.com/questions/116775/the-longest-path-of-edges-on-a-3x3-grid`

C Permutation Test for Rank Confidence Interval

Construction confidence interval To compute a confidence interval at significance level α for a model's rank, we use pairwise comparisons between models via a paired permutation test.

For a given model m_i, we compare it to every other model m_j:

- Let $N_{i,j}^{+}$ be the number of models m_j for which m_i performs significantly better.
- Let $N_{i,j}^{-}$ be the number of models m_j for which m_i performs significantly worse.

Given a total of N models, the rank confidence interval for model m_i is $[N_{i,j}^{+} + 1, N - N_{i,j}^{-}]$.

Paired permutation test A paired permutation test is a non-parametric method for testing whether two related samples have significantly different means. It requires:

- A set of paired observations: $(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)$
- A test statistic T computed over these pairs.

The null hypothesis is that x_i and y_i are exchangeable, i.e., swapping them does not affect the test statistic in expectation.

To test this hypothesis, we generate random permutations by flipping each pair (x_i, y_i) with a probability of 50%. For each permutation, we compute the test statistic. By repeating this process many times, we create a distribution of the test statistic under the null hypothesis. We then compare the unpermuted test statistic to this distribution to determine if it is significantly different. Specifically, if the quantile of the unpermuted statistic within this distribution is less than the significance level α, we reject the null hypothesis and conclude that there is a significant difference between the samples.

Paired permutation test for rank In our setting, each paired sample indicates the correctness of a single answer of models m_i and m_j on the same problem. For a single competition, the test statistic is the total difference in performance between the two models:

$$T((x_1, y_1), \ldots, (x_n, y_n)) = \sum_{i=1}^{n} (x_i - y_i).$$

When aggregating performance across multiple competitions, we weight each competition equally, regardless of the number of problems it contains. This results in a weighted sum in the test statistic, where each sample is weighted by the inverse of the number of problems in its respective competition. This ensures that the total weight associated with each competition is equal. If we denote the number of problems in competition c as N_c, and c_i as the competition of problem i, the test statistic becomes:

$$T((x_1, y_1), \ldots, (x_n, y_n)) = \sum_{i=1}^{n} \frac{(x_i - y_i)}{N_{c_i}}.$$

We can then apply the procedure described above to compute the rank confidence interval for model m_i.

Table 6: The full main results of our numerical answer evaluation, sorted by average score. We measure cost in USD, and report the average score across all generations for the 3 competitions.

Model	AIME	HMMT	BRUMO	CMIMC	Acc (avg)	Cost (avg)
GPT-5 (high)	95.0	88.3	91.7	90.0	91.3	4.83
Grok 4 Fast (Reasoning)	90.8	91.7	94.2	85.6	90.6	0.18
Grok 4	90.8	92.5	95.0	83.1	90.4	7.56
GPT OSS 120B (high)	90.0	90.0	91.7	85.6	89.3	0.21
DeepSeek-v3.2 (Think)	91.7	90.0	95.8	75.6	88.3	0.22
GPT-5-mini (high)	87.5	89.2	90.0	83.1	87.5	1.09
DeepSeek-v3.1 (Think)	90.8	85.8	90.0	81.3	87.0	1.23
o4-mini (high)	91.7	82.5	86.7	84.4	86.3	1.86
o3 (high)	89.2	77.5	95.8	78.8	85.3	3.23
gemini-2.5-pro-05-06	83.3	80.8	89.2	N/A	84.4	0.68
GLM 4.5	93.3	77.5	92.5	71.3	83.7	1.71
DeepSeek-R1-0528	89.2	76.7	92.5	69.4	81.9	1.65
GPT OSS 20B (high)	89.2	75.0	85.0	72.5	80.4	0.22
gemini-2.5-pro	87.5	82.5	90.0	58.1	79.5	5.02
GPT-5-nano (high)	85.0	74.2	80.8	73.8	78.4	0.40
GLM 4.5 Air	83.3	69.2	90.0	70.6	78.3	0.90
Claude-Sonnet-4.5 (Think)	84.2	67.5	90.8	66.9	77.3	9.09
o3-mini (high)	86.7	67.5	N/A	N/A	77.1	1.92
Grok 3 Mini (high)	81.7	74.2	85.0	66.3	76.8	0.34
Qwen3-235B-A22B	80.8	62.5	86.7	N/A	76.7	0.25
K2-Think	83.3	65.0	83.3	65.6	74.3	N/A
o4-mini (medium)	84.2	66.7	84.2	60.6	73.9	0.92
Qwen3-A22B-2507-Think	92.5	71.7	45.8	N/A	70.0	1.34
Claude-Opus-4.0 (Think)	69.2	58.3	81.7	N/A	69.7	34.26
gemini-2.5-flash (think)	70.8	64.2	83.3	50.6	67.2	2.65
Qwen3-30B-A3B	70.0	50.8	77.5	N/A	66.1	0.15
o3-mini (medium)	76.7	53.3	N/A	N/A	65.0	0.92
o1 (medium)	81.7	48.3	N/A	N/A	65.0	24.06
DeepSeek-R1	70.0	41.7	80.8	N/A	64.2	0.72
Phi-4-reasoning-plus	74.2	46.7	N/A	N/A	60.4	0.19
QwQ-32B	65.8	47.5	N/A	N/A	56.7	0.59
o4-mini (low)	61.7	47.5	65.8	46.3	55.3	0.36
Grok 3 Mini (low)	65.0	50.8	65.8	36.9	54.6	0.10
DeepSeek-R1-Distill-32B	60.0	33.3	68.3	N/A	53.9	0.16
DeepSeek-R1-Distill-70B	55.0	33.3	66.7	N/A	51.7	0.19
DeepSeek-R1-Distill-14B	49.2	31.7	68.3	N/A	49.7	0.08
Claude-3.7-Sonnet (Think)	49.2	31.7	65.8	N/A	48.9	10.89
OpenThinker-32B	56.7	36.7	N/A	N/A	46.7	N/A
gemini-2.0-flash-thinking	53.3	35.8	N/A	N/A	44.6	N/A
s1.1-32B	50.0	37.5	N/A	N/A	43.8	N/A
DeepSeek-V3-03-24	50.0	29.2	N/A	N/A	39.6	0.14
LIMO	49.2	30.0	N/A	N/A	39.6	N/A
o3-mini (low)	48.3	28.3	N/A	N/A	38.3	0.34
QwQ-32B-Preview	33.3	18.3	N/A	N/A	25.8	0.32
gemini-2.0-flash	27.5	13.3	N/A	N/A	20.4	0.04
DeepSeek-V3	25.0	13.3	N/A	N/A	19.2	0.10
gemini-2.0-pro	27.5	7.5	N/A	N/A	17.5	0.40
DeepSeek-R1-Distill-1.5B	20.0	11.7	N/A	N/A	15.8	0.11
Llama-4-Maverick	22.5	8.3	N/A	N/A	15.4	0.03
gpt-4o	11.7	5.8	N/A	N/A	8.8	0.26
Claude-3.5-Sonnet	3.3	1.7	N/A	N/A	2.5	0.26

Table 8: Average accuracy per model per problem type, sorted by average score.

Model	Algebra	Combinatorics	Geometry	Number Theory
GPT-5 (high)	100.0	91.3	81.1	94.0
Grok 4	96.8	87.8	86.6	88.1
GPT OSS 120B (high)	100.0	86.0	81.1	90.5
GPT-5-mini (high)	93.5	83.7	80.5	91.7
o4-mini (high)	96.0	86.6	75.6	86.9
o3 (high)	91.1	84.9	80.5	78.6
gemini-2.5-pro-05-06	97.8	77.6	75.0	91.7
GLM 4.5	94.4	77.3	75.6	85.7
DeepSeek-R1-0528	89.5	80.8	73.8	79.8
GPT OSS 20B (high)	93.5	75.0	71.3	82.1
gemini-2.5-pro	96.8	70.3	73.2	76.2
GLM 4.5 Air	87.1	69.8	75.0	84.5
GPT-5-nano (high)	86.3	73.3	70.1	88.1
o3-mini (high)	84.4	72.4	71.1	80.0
Grok 3 Mini (high)	83.9	76.2	67.1	78.6
Qwen3-235B-A22B	90.2	67.2	68.5	86.7
o4-mini (medium)	82.3	68.6	66.5	77.4
Claude-Opus-4.0 (Think)	75.0	65.5	63.0	80.0
Qwen3-A22B-2507-Think	87.0	57.8	59.3	78.3
gemini-2.5-flash (think)	83.1	56.4	58.5	73.8
Qwen3-30B-A3B	75.0	50.9	65.7	81.7
o3-mini (medium)	70.3	56.6	64.5	70.0
o1 (medium)	67.2	61.8	59.2	75.0
DeepSeek-R1	68.5	52.6	63.9	78.3
Phi-4-reasoning-plus	65.6	50.0	59.2	72.5
QwQ-32B	59.4	47.4	55.3	72.5
o4-mini (low)	61.3	52.9	45.1	70.2
Grok 3 Mini (low)	62.9	44.2	45.7	70.2
DeepSeek-R1-Distill-32B	56.5	42.2	54.6	71.7
DeepSeek-R1-Distill-70B	51.1	37.1	54.6	76.7
DeepSeek-R1-Distill-14B	50.0	37.1	50.9	73.3
Claude-3.7-Sonnet (Think)	53.3	41.4	43.5	61.7
OpenThinker-32B	54.7	26.3	53.9	65.0
gemini-2.0-flash-thinking	65.6	21.1	40.8	70.0
s1.1-32B	56.2	19.7	51.3	60.0
DeepSeek-V3-03-24	40.6	19.7	50.0	55.0
LIMO	43.8	23.7	42.1	62.5
o3-mini (low)	39.1	23.7	46.1	52.5
QwQ-32B-Preview	29.7	7.9	31.6	40.0
gemini-2.0-flash	20.3	5.3	26.3	30.0
DeepSeek-V3	17.2	3.9	28.9	30.0
gemini-2.0-pro	21.9	3.9	18.4	35.0
DeepSeek-R1-Distill-1.5B	9.4	5.3	18.4	35.0
Llama-4-Maverick	14.1	2.6	17.1	40.0
gpt-4o	4.7	0.0	10.5	27.5
Claude-3.5-Sonnet	1.6	0.0	3.9	5.0
Overall	64.5	48.8	55.5	68.6

Table 9: Results of the BRUMO competition with 95% confidence intervals.

Model	**Rank**	**Acc (avg)**
o3 (high)	1-10	95.83 ± 3.6
DeepSeek-v3.2 (Think)	1-9	95.83 ± 3.6
Grok 4	1-12	95.00 ± 3.9
Grok 4 Fast	1-14	94.17 ± 4.2
DeepSeek-R1-0528	1-14	92.50 ± 4.7
GLM 4.5	1-14	92.50 ± 4.7
GPT-5 (high)	1-17	91.67 ± 4.9
GPT OSS 120B (high)	1-17	91.67 ± 4.9
Claude-Sonnet-4.5 (Think)	1-17	90.83 ± 5.2
GPT-5-mini (high)	3-21	90.00 ± 5.4
DeepSeek-v3.1 (Think)	4-20	90.00 ± 5.4
gemini-2.5-pro	3-20	90.00 ± 5.4
GLM 4.5 Air	2-19	90.00 ± 5.4
gemini-2.5-pro-05-06	4-21	89.17 ± 5.6
Qwen3-235B-A22B	7-23	86.67 ± 6.1
o4-mini (high)	7-24	86.67 ± 6.1
Grok 3 Mini (high)	10-24	85.00 ± 6.4
GPT OSS 20B (high)	7-25	85.00 ± 6.4
o4-mini (medium)	9-25	84.17 ± 6.5
gemini-2.5-flash (think)	12-25	83.33 ± 6.7
K2-Think	11-25	83.33 ± 6.7
Claude-Opus-4.0 (Think)	15-25	81.67 ± 6.9
DeepSeek-R1	16-25	80.83 ± 7.0
GPT-5-nano (high)	15-25	80.83 ± 7.0
Qwen3-30B-A3B	18-25	77.50 ± 7.5
DeepSeek-R1-Distill-14B	26-31	68.33 ± 8.3
DeepSeek-R1-Distill-32B	26-31	68.33 ± 8.3
DeepSeek-R1-Distill-70B	26-31	66.67 ± 8.4
Grok 3 Mini (low)	26-31	65.83 ± 8.5
o4-mini (low)	26-31	65.83 ± 8.5
Claude-3.7-Sonnet (Think)	26-31	65.83 ± 8.5
Qwen3-A22B-2507-Think	32-32	45.83 ± 8.9

Table 10: Results of the CMIMC competition with 95% confidence intervals.

Model	**Rank**	**Acc (avg)**
GPT-5 (high)	1-4	90.00 ± 4.6
Grok 4 Fast	1-8	85.62 ± 5.4
GPT OSS 120B (high)	1-7	85.62 ± 5.4
o4-mini (high)	1-8	84.38 ± 5.6
Grok 4	2-8	83.12 ± 5.8
GPT-5-mini (high)	2-8	83.12 ± 5.8
DeepSeek-v3.1 (Think)	2-9	81.25 ± 6.0
o3 (high)	3-12	78.75 ± 6.3
DeepSeek-v3.2 (Think)	7-14	75.62 ± 6.7
GPT-5-nano (high)	8-15	73.75 ± 6.8
GPT OSS 20B (high)	8-17	72.50 ± 6.9
GLM 4.5	8-17	71.25 ± 7.0
GLM 4.5 Air	9-17	70.62 ± 7.1
DeepSeek-R1-0528	9-17	69.38 ± 7.1
Claude-Sonnet-4.5 (Think)	10-18	66.88 ± 7.3
Grok 3 Mini (high)	11-18	66.25 ± 7.3
K2-Think	11-18	65.62 ± 7.4
o4-mini (medium)	15-19	60.62 ± 7.6
gemini-2.5-pro	18-19	58.13 ± 7.6
gemini-2.5-flash (think)	20-21	50.62 ± 7.7
o4-mini (low)	20-21	46.25 ± 7.7
Grok 3 Mini (low)	22-22	36.88 ± 7.5

Table 11: Results of the AIME competition with 95% confidence intervals.

Model	Rank	Acc (avg)
GPT-5 (high)	1-9	95.00 ± 3.9
GLM 4.5	1-12	93.33 ± 4.5
Qwen3-A22B-2507-Think	1-12	92.50 ± 4.7
o4-mini (high)	1-15	91.67 ± 4.9
DeepSeek-v3.2 (Think)	1-15	91.67 ± 4.9
DeepSeek-v3.1 (Think)	1-16	90.83 ± 5.2
Grok 4	1-16	90.83 ± 5.2
Grok 4 Fast	1-18	90.83 ± 5.2
GPT OSS 120B (high)	1-18	90.00 ± 5.4
DeepSeek-R1-0528	2-18	89.17 ± 5.6
GPT OSS 20B (high)	2-18	89.17 ± 5.6
o3 (high)	2-18	89.17 ± 5.6
GPT-5-mini (high)	4-22	87.50 ± 5.9
gemini-2.5-pro	4-23	87.50 ± 5.9
o3-mini (high)	4-24	86.67 ± 6.1
GPT-5-nano (high)	6-24	85.00 ± 6.4
o4-mini (medium)	8-24	84.17 ± 6.5
Claude-Sonnet-4.5 (Think)	8-25	84.17 ± 6.5
K2-Think	13-25	83.33 ± 6.7
GLM 4.5 Air	13-25	83.33 ± 6.7
gemini-2.5-pro-05-06	13-25	83.33 ± 6.7
o1 (medium)	13-25	81.67 ± 6.9
Grok 3 Mini (high)	14-25	81.67 ± 6.9
Qwen3-235B-A22B	14-25	80.83 ± 7.0
o3-mini (medium)	18-30	76.67 ± 7.6
Phi-4-reasoning-plus	25-30	74.17 ± 7.8
gemini-2.5-flash (think)	25-32	70.83 ± 8.1
Qwen3-30B-A3B	25-33	70.00 ± 8.2
DeepSeek-R1	25-33	70.00 ± 8.2
Claude-Opus-4.0 (Think)	25-34	69.17 ± 8.3
QwQ-32B	27-34	65.83 ± 8.5
Grok 3 Mini (low)	27-35	65.00 ± 8.5
o4-mini (low)	28-37	61.67 ± 8.7
DeepSeek-R1-Distill-32B	30-37	60.00 ± 8.8
OpenThinker-32B	33-41	56.67 ± 8.9
DeepSeek-R1-Distill-70B	33-43	55.00 ± 8.9
gemini-2.0-flash-thinking	33-43	53.33 ± 8.9
DeepSeek-V3-03-24	35-43	50.00 ± 8.9
s1.1-32B	35-43	50.00 ± 8.9
LIMO	36-43	49.17 ± 8.9
DeepSeek-R1-Distill-14B	36-43	49.17 ± 8.9
Claude-3.7-Sonnet (Think)	35-43	49.17 ± 8.9
o3-mini (low)	35-43	48.33 ± 8.9
QwQ-32B-Preview	44-45	33.33 ± 8.4
gemini-2.0-pro	44-48	27.50 ± 8.0
gemini-2.0-flash	45-48	27.50 ± 8.0
DeepSeek-V3	45-49	25.00 ± 7.7
Llama-4-Maverick	45-49	22.50 ± 7.5
DeepSeek-R1-Distill-1.5B	47-49	20.00 ± 7.2
gpt-4o	50-50	11.67 ± 5.7
Claude-3.5-Sonnet	51-51	3.33 ± 3.2

Table 12: Results of the HMMT competition with 95% confidence intervals.

Model	Rank	Acc (avg)
Grok 4	1-6	92.50 ± 4.7
Grok 4 Fast (Reasoning)	1-7	91.67 ± 4.9
GPT OSS 120B (high)	1-7	90.00 ± 5.4
DeepSeek-v3.2 (Think)	1-7	90.00 ± 5.4
GPT-5-mini (high)	1-8	89.17 ± 5.6
GPT-5 (high)	1-10	88.33 ± 5.7
DeepSeek-v3.1 (Think)	2-10	85.83 ± 6.2
o4-mini (high)	6-13	82.50 ± 6.8
gemini-2.5-pro	5-14	82.50 ± 6.8
gemini-2.5-pro-05-06	7-16	80.83 ± 7.0
GLM 4.5	8-17	77.50 ± 7.5
o3 (high)	8-17	77.50 ± 7.5
DeepSeek-R1-0528	8-17	76.67 ± 7.6
GPT OSS 20B (high)	10-21	75.00 ± 7.7
Grok 3 Mini (high)	10-21	74.17 ± 7.8
GPT-5-nano (high)	9-21	74.17 ± 7.8
Qwen3-A22B-2507-Think	11-23	71.67 ± 8.1
GLM 4.5 Air	12-24	69.17 ± 8.3
o3-mini (high)	14-25	67.50 ± 8.4
Claude-Sonnet-4.5 (Think)	14-24	67.50 ± 8.4
o4-mini (medium)	15-25	66.67 ± 8.4
K2-Think	17-25	65.00 ± 8.5
gemini-2.5-flash (think)	17-25	64.17 ± 8.6
Qwen3-235B-A22B	18-26	62.50 ± 8.7
Claude-Opus-4.0 (Think)	21-28	58.33 ± 8.8
o3-mini (medium)	24-32	53.33 ± 8.9
Qwen3-30B-A3B	25-32	50.83 ± 8.9
Grok 3 Mini (low)	25-32	50.83 ± 8.9
o1 (medium)	26-33	48.33 ± 8.9
QwQ-32B	26-33	47.50 ± 8.9
o4-mini (low)	26-33	47.50 ± 8.9
Phi-4-reasoning-plus	26-33	46.67 ± 8.9
DeepSeek-R1	29-36	41.67 ± 8.8
s1.1-32B	33-39	37.50 ± 8.7
OpenThinker-32B	33-42	36.67 ± 8.6
gemini-2.0-flash-thinking	33-43	35.83 ± 8.6
DeepSeek-R1-Distill-70B	34-43	33.33 ± 8.4
DeepSeek-R1-Distill-32B	34-43	33.33 ± 8.4
DeepSeek-R1-Distill-14B	35-43	31.67 ± 8.3
Claude-3.7-Sonnet (Think)	34-43	31.67 ± 8.3
LIMO	35-43	30.00 ± 8.2
DeepSeek-V3-03-24	35-43	29.17 ± 8.1
o3-mini (low)	36-43	28.33 ± 8.1
QwQ-32B-Preview	44-46	18.33 ± 6.9
gemini-2.0-flash	44-48	13.33 ± 6.1
DeepSeek-V3	44-48	13.33 ± 6.1
DeepSeek-R1-Distill-1.5B	45-49	11.67 ± 5.7
Llama-4-Maverick	45-50	8.33 ± 4.9
gemini-2.0-pro	47-50	7.50 ± 4.7
gpt-4o	48-51	5.83 ± 4.2
Claude-3.5-Sonnet	50-51	1.67 ± 2.3

Table 13: Average number of input and output tokens used per model across all final-answer competitions.

	Input	Output
GPT OSS 20B (high)	151	44 214
GPT-5-nano (high)	186	31 651
GLM 4.5 Air	156	25 488
GLM 4.5	156	24 174
Claude-3.7-Sonnet (Think)	199	24 168
DeepSeek-R1-0528	141	23 855
GPT OSS 120B (high)	151	23 597
gemini-2.5-flash (think)	144	22 803
Grok 3 Mini (high)	141	21 917
DeepSeek-R1-Distill-1.5B	174	20 106
Claude-Sonnet-4.5 (Think)	203	19 157
Qwen3-A22B-2507-Think	158	18 749
DeepSeek-v3.1 (Think)	152	18 562
Phi-4-reasoning-plus	393	18 309
DeepSeek-v3.2 (Think)	152	17 426
Qwen3-30B-A3B	158	17 024
GPT-5-mini (high)	197	16 904
Grok 4	151	16 697
QwQ-32B	191	16 142
gemini-2.5-pro	156	16 133
GPT-5 (high)	1782	15 390
Claude-Opus-4.0 (Think)	199	15 189
o3-mini (high)	170	14 526
Qwen3-235B-A22B	158	14 063
o1 (medium)	172	13 324
Grok 4 Fast (Reasoning)	264	12 277
DeepSeek-R1	145	11 038
DeepSeek-R1-Distill-70B	148	10 566
QwQ-32B-Preview	191	8808
o3-mini (medium)	172	6892
o4-mini (medium)	189	6549
Grok 3 Mini (low)	140	6369
DeepSeek-V3-03-24	167	3828
gemini-2.5-pro-05-06	151	3712
gemini-2.0-flash	172	3040
gemini-2.0-pro	173	2644
o3-mini (low)	172	2511
DeepSeek-V3	167	2462
o4-mini (low)	148	2433
Llama-4-Maverick	170	1252
gpt-4o	173	814
Claude-3.5-Sonnet	193	542
gemini-2.0-flash-thinking	350	498

D Prompts

We provide the full set of prompts used in our evaluation below. In each prompt, {{problem}} is replaced with the problem statement.

Prompt AIME 2025

```
Please reason step by step, and put your final answer within \boxed{}.The answer is an integer between 0
and 999 inclusive.

{{problem}}
```

Prompt CMIMC and HMMT and BRUMO 2025

```
Please reason step by step, and put your final answer within \boxed{}.

{{problem}}
```

Prompt Project Euler

```
You are solving a problem from Project Euler.
As a tool you can use, you are given access to a code execution environment. Invoke that tool to execute
Python or C++ code. You can execute code up to 20 times, so you can use them quite liberally.
You can also use the tool to run code that helps you reason about the problem, but does not directly
compute the final answer.
Answers that consist of code are not accepted, I will only accept a final answer to the question that does
not require me to run any code you produce, as you should use the tool for this.
You are REQUIRED to use the tool to compute the final answer. It is impossible to solve this problem
without using the tool.
Instead, your code should compute the answer (via the tool), after which you should put your final answer
within \\boxed{{}}.
Your answer should be correctly formatted by putting your final answer within \boxed{}, i.e., end your
reply with "### Final answer: \\boxed{your_answer_here}".

{{problem}}
```

Tool Description Project Euler

```
function:
   description: 'Executes the code in the given language and returns the standard
      output and standard error. Your code is always executed as a self-contained
      script, and it does not have access to the previously executed code blocks!
      If you use python, your code will be run in an environment with the following
      libraries installed: pandas, numpy, scikit-learn, sympy, gmpy2'
   name: execute_code
   parameters:
   properties:
      code:
         description: 'The self-contained code to execute'
         type: string
      lang:
         description: 'The programming language of the code (python or cpp)'
         type: string
   required:
   -code
   -lang
   type: object
   type: function
```

Prompt USAMO 2025

```
Give a thorough answer to the following question. Your answer will be graded by human judges based on
accuracy, correctness, and your ability to prove the result. You should include all steps of the proof. Do
not skip important steps, as this will reduce your grade. It does not suffice to merely state the result.
Use LaTeX to format your answer.

{{problem}}
```

Prompt IMO 2025

```
Your task is to write a proof solution to the following problem. Your proof will be graded by human judges
for accuracy, thoroughness, and clarity. When you write your proof, follow these guidelines:

- You are creating a proof, not a proof outline. Each step should be carefully explained and documented. If
 not properly explained, the judge will assume that you cannot explain it, and therefore decrease your
grade.
- You can use general theorems and lemmas, but only if they are well-known. As a rule of thumb: if the
result has a name and is famous enough to have a Wikipedia page or something similar to describe it, it is
allowed. Any result from papers that would not be taught in high-school or low-level bachelor courses in
mathematics should not be used. Any use of such results will immediately give you a zero grade.
- Do not skip computation steps in your proof. Clearly explain what transformations were done and why they
are allowed in each step of a calculation.
- You should use correct LaTeX notation to write equations and mathematical symbols. You should encompass
these equations in appropriate symbols ("\\(" and "\\)" for inline math, "\\[" and "\\]" for block math) to
 enhance the clarity of your proof. Do not use any unicode characters.
- Your proof should be self-contained.
- If you are not sure about a specific step, or do not know how to prove an intermediate result, clearly
state this. It is much preferable to indicate your uncertainty rather than making incorrect statements or
claims.

{{problem}}
```

Prompt Best-of-n Selection IMO 2025

```
You are judging which of the two LLM-generated proofs for a given math problem is better.

### Input:

Your input will consist of the following components:
- **Problem Statement**: A mathematical problem that the proof is attempting to solve.
- **Proof Solution A/B**: The proofs that you need to evaluate. This proof may contain errors, omissions,
or unclear steps. Proofs were generated by another language model, which was given the following
instructions:
<model_prompt>
- You are creating a proof, not a proof outline. Each step should be carefully explained and documented. If
 not properly explained, the judge will assume that you cannot explain it, and therefore decrease your
grade.
- You can use general theorems and lemmas, but only if they are well-known. As a rule of thumb: if the
result has a name and is famous enough to have a Wikipedia page or something similar to describe it, it is
allowed. Any result from papers that would not be taught in high-school or low-level bachelor courses in
mathematics should not be used. Any use of such results will immediately give you a zero grade.
- Do not skip computation steps in your proof. Clearly explain what transformations were done and why they
are allowed in each step of a calculation.
- You should use correct LaTeX notation to write equations and mathematical symbols. You should encompass
these equations in appropriate symbols ("\\(" and "\\)" for inline math, "\\[" and "\\]" for block math) to
 enhance the clarity of your proof. Do not use any unicode characters.
- Your proof should be self-contained.
- If you are not sure about a specific step, or do not know how to prove an intermediate result, clearly
state this. It is much preferable to indicate your uncertainty rather than making incorrect statements or
claims.
</model_prompt>

### How the solution should be graded:
The following examples are small mistakes that should only be slightly penalised:
- Makes a small computational mistake that can be easily fixed
- Misses an edge case which can be easily proven/disproven
- Skips over a step that follows without much reasoning or manual work
On the other hand, a solution should should be severely penalised if:
- It marks a step as trivial, if it is not immediately obvious with little reasoning why this would be the
case.
- It omits algebra-heavy computational steps, regardless of whether or not it has outlined the methodology.
 Skipping shorter computations should be permitted.
- Generalizes over a pattern without rigorously describing the pattern, or without proving any relevant
properties.
- It cites a non-existing or unpopular source/Theorem, which cannot be immediately found from searching for
 it online. Thus, any theorems that can be immediately found and have a Wikipedia article are allowed.

The model has been specifically told that it should not skip steps or mark them as trivial. Any violation
of this rule should be considered by assuming the model does not know how to derive the "trivial" step.

### Further Potential Issues:

Here are some common types of issues to look for:
```

```
- **Overgeneralization**: The generated proof proceeds by proving the problem in one or more specific cases,
 and then concludes that the result holds in general. However, it does not provide a proof for the general
case.
- **Oversimplification**: The proof marks steps as trivial or obvious without proper justification.
- **Skipping Computation Steps**: Proofs that skip computation steps or do not explain transformations
clearly can lead to misunderstandings.
- **Citing Non-Standard Works or Theorems**: Some models may cite theorems or results that are not well-
known or are not typically taught in high-school or low-level bachelor courses. Such theorems are only
allowed if they are well known.
- **Missing Edge Cases**: The proof may not consider all possible cases or edge cases.

### Scoring instructions

You should compare the two proofs and determine which one is better. If you believe Proof A is better, end
your analysis with \boxed{A}. If you believe Proof B is better, end your analysis with \boxed{B}. If you
believe both proofs are equally good, end your analysis with \boxed{equal}.

### Problem Statement:
{{problem}}

### Proof Solution A:
{{solution_a}}

### Proof Solution B:
{{solution_b}}
```

E Example Problems and Model Solutions

E.1 AIME 2025 - Problem 15

Problem

Let N denote the numbers of ordered triples of positive integers (a, b, c) such that $a, b, c \leq 3^6$ and $a^3 + b^3 + c^3$ is a multiple of 3^7. Find the remainder when N is divided by 1000.

Solution by GPT-5

Let $3^7 = 2187$. For any $1 \leq n \leq 3^6$, write uniquely $n = u + 27v$ with $0 \leq u, v \leq 26$ (the pair $(u, v) = (0, 0)$ corresponds to $n = 3^6$). Then

$$n^3 \equiv u^3 + 81u^2v \pmod{2187}.$$

Thus for a triple (x, y, z) with (u_i, v_i) the corresponding pairs, the condition

$$x^3 + y^3 + z^3 \equiv 0 \pmod{2187}$$

becomes

$$A + 81B \equiv 0 \pmod{2187}, \quad \text{where } A = \sum u_i^3 \text{ and } B = \sum u_i^2 v_i.$$

Fix $u = (u_1, u_2, u_3)$ and write $d = \gcd(27, u_1^2, u_2^2, u_3^2)$. Then the linear congruence $\sum u_i^2 v_i \equiv -A/81 \pmod{27}$ has:

- 27^2 solutions if $d = 1$, provided $81 \mid A$;
- $27^2 \cdot 9$ solutions if $d = 9$, provided $729 \mid A$;
- 27^3 solutions if $d = 27$, provided $2187 \mid A$.

So it remains to count the number of u-triples in each case. Write u_i in base 3 as

$$u_i = a_i + 3b_i + 9c_i \quad \text{with } a_i, b_i, c_i \in \{0, 1, 2\}.$$

Then:

- u_i divisible by 3 iff $a_i = 0$; divisible by 9 iff $a_i = b_i = 0$.
- $u_i^3 \equiv a_i^3 + 9a_i^2 b_i + 27(\dots) \pmod{81}$;
- If $a_i = 0$, then $u_i^3 \equiv 27b_i^3 + 243b_i^2 c_i \pmod{729}$;
- If $a_i = b_i = 0$, then $u_i^3 \equiv 729c_i^3 \pmod{2187}$.

Count by cases:

1. $d = 1$ (at least one $u_i \not\equiv 0 \pmod 3$), requiring $81 \mid A$. Modulo 9, $a_i^3 \in \{0, 1, 8\}$, and $\sum a_i^3 \equiv 0 \pmod 9$ with some $a_i \neq 0$ forces $\{a_1, a_2, a_3\} = \{0, 1, 2\}$. For such a pattern, the further mod 81 condition reduces to a single congruence modulo 9 in the (b_i, c_i). For $a_i = 1$ or 2, the contribution is uniformly distributed over $\mathbb{Z}/9$ as (b_i, c_i) vary; for $a_i = 0$ it is in $\{0, 3, 6\}$ equally often. Hence for each of the $3! = 6$ assignments of $(0, 1, 2)$ to the three places, exactly 81 of the 9^3 choices of (b_i, c_i) work. Thus $N_1 = 6 \cdot 81 = 486$.
2. $d = 9$ (all $u_i \equiv 0 \pmod 3$ but not all $\equiv 0 \pmod 9$), requiring $729 \mid A$. With $a_i = 0$, $u_i^3 \equiv 27b_i^3 + 243b_i^2c_i \pmod{729}$, so
$$729 \mid A \iff \sum b_i^3 + 9\sum b_i^2 c_i \equiv 0 \pmod{27}.$$
Among the 9^3 choices of (b_i, c_i), exactly 81 satisfy this (by a simple check or character sum), of which 27 have all $b_i = 0$ (these correspond to $d = 27$). Therefore $N_9 = 81 - 27 = 54$.
3. $d = 27$ (all $u_i \equiv 0 \pmod 9$), requiring $2187 \mid A$. Then $u_i = 9c_i$ and $A = 729\sum c_i^3$; so
$$2187 \mid A \iff \sum c_i \equiv 0 \pmod 3.$$
There are $N_{27} = 3^2 = 9$ such triples.

Therefore the total number of (x, y, z) is
$$N = 27^2 N_1 + 27^2 \cdot 9N_9 + 27^3 N_{27} = 729 \cdot 486 + 729 \cdot 9 \cdot 54 + 27^3 \cdot 9 = 885735.$$
Hence $N \pmod{1000}$ is 735.

$$\boxed{735}$$

E.2 HMMT Feb 2025 - Problem 20

Problem

The circumference of a circle is divided into 45 arcs, each of length 1. Initially, there are 15 snakes, each of length 1, occupying every third arc. Every second, each snake independently moves either one arc left or one arc right, each with probability $\frac{1}{2}$. If two snakes ever touch, they merge to form a single snake occupying the arcs of both of the previous snakes, and the merged snake moves as one snake. Compute the expected number of seconds until there is only one snake left.

Solution by Grok 4 Fast

$$\boxed{\frac{448}{3}}$$

Explanation

The problem involves 15 snakes on a circle of 45 arcs, initially placed every third arc, resulting in 15 gaps of 2 empty arcs each. Define $h_i = G_i/2$ for each gap G_i, so initially all $h_i = 1$ and $\sum h_i = 15$. The process tracks the number of clusters k (equal to the number of gaps) until $k = 1$.

At each step, k independent directions $d_1, \ldots, d_k \in \{+1, -1\}$ are chosen with equal probability. The updates are
$$\varepsilon_i = \frac{d_{i+1} - d_i}{2}, \quad i = 1, \ldots, k \text{ (cyclic)},$$
so
$$h_i' = h_i + \varepsilon_i.$$
A gap is removed if $h_i' \leq 0$, which occurs only if $h_i = 1$ and $\varepsilon_i = -1$, reducing k by the number of such events. The sum of the remaining h_i remains 15.

Thus the state evolves as a Markov chain on partitions of 15, where the number of parts is k.

Computations for small n (number of snakes) yield expected times:

$$e_2 = 2, \quad e_3 = \tfrac{16}{3}, \quad e_4 = 10.$$

These fit the quadratic form

$$e_n = \frac{2}{3}(n^2 - 1),$$

verified as

$$e_2 = \tfrac{2}{3}(4-1) = 2, \quad e_3 = \tfrac{2}{3}(9-1) = \tfrac{16}{3}, \quad e_4 = \tfrac{2}{3}(16-1) = 10.$$

For $n = 15$,

$$e_{15} = \frac{2}{3}(225-1) = \frac{2}{3} \times 224 = \frac{448}{3}.$$

This pattern arises from the coalescing dynamics, where the expected time scales quadratically with n due to the symmetric random walk behavior of the gaps.

E.3 IMO 2025 - Problem 1

Problem

A line in the plane is called
emphsunny if it is not parallel to any of the x-axis, the y-axis, or the line $x + y = 0$.
Let $n \geq 3$ be a given integer. Determine all nonnegative integers k such that there exist n distinct lines in the plane satisfying both of the following:

- for all positive integers a and b with $a + b \leq n + 1$, the point (a, b) lies on at least one of the lines; and
- exactly k of the n lines are sunny.

Solution by o4-mini

Proof. Fix an integer $n \geq 3$. Let

$$S_n \;=\; \{(a,b) \in \mathbb{Z}^2 : \; a \geq 1, \; b \geq 1, \; a + b \leq n + 1\},$$

so $|S_n| = \frac{n(n+1)}{2}$. We seek n distinct lines in the plane, exactly k of which are "sunny," that together contain every point of S_n. Recall a line is "sunny" if it is not parallel to any of

- the x–axis,
- the y–axis,
- the line $x + y = 0$.

Equivalently, "non–sunny" lines are exactly the horizontals, verticals, or antidiagonals $x + y =$ const.
Write $s = n - k$ for the number of non–sunny lines, and k for the sunny lines. We will show:

1. One can cover S_n with n lines having exactly $k \in \{0, 1, 3\}$ sunny lines.
2. No other value of k is possible.

—

Step 1 - Reduction to a smaller triangular grid. Let $\mathcal{L}$ be any collection of n lines covering S_n, with s of them non–sunny and $k = n - s$ sunny. Since non–sunny lines are parallel to one of the three directions (horizontal, vertical, or antidiagonal $x + y =$ const), each non–sunny line covers points in at most one of the three "grid directions." One checks easily that to maximize the number of points of S_n covered by s non–sunny lines, one should take them all horizontal:

$$y = 1, \; y = 2, \; \ldots, \; y = s.$$

Indeed any other choice covers no more than this. These s horizontal lines cover exactly

$$\sum_{i=1}^{s} \bigl|\{(a,i) : a \geq 1, \; a + i \leq n + 1\}\bigr| = \sum_{i=1}^{s}(n + 1 - i) = s(n+1) - \tfrac{s(s+1)}{2}.$$

Since $|S_n| = \frac{n(n+1)}{2}$, the number of points not yet covered is

$$\frac{n(n+1)}{2} - \left[s(n+1) - \tfrac{s(s+1)}{2}\right] = \frac{(n-s)(n-s+1)}{2} = \frac{k(k+1)}{2}.$$

One checks that the uncovered points are precisely

$$T_k \;=\; \{(a,b)\in\mathbb{Z}^2 :\; a\ge 1,\; b\ge s+1,\; a+b\le n+1\}.$$

If we set $b' = b - s$, then $b' = 1, 2, \dots, k$ and

$$a+b\le n+1 \iff a+b'\le k+1,$$

so the uncovered set is congruent to the standard "triangular grid"

$$S_k = \{(a,b')\in\mathbb{Z}^2 :\; a\ge 1,\; b'\ge 1,\; a+b'\le k+1\},$$

which has exactly $k(k+1)/2$ points. To finish covering S_n we must cover T_k, i.e. cover S_k, by the remaining k lines—and these k must be sunny, because we have already used up all s non–sunny lines.

Thus the original problem for (n,k) is solvable if and only if the smaller problem

> "Cover the triangular grid S_k by k lines, none parallel to the three forbidden directions"

admits a solution.

—

Step 2 - Characterization of which k work for S_k. Write

$$S_k = \{(a,b)\in\mathbb{Z}^2 :\; a\ge 1,\; b\ge 1,\; a+b\le k+1\},$$

and let

$$B = \{(a,b)\in S_k :\; a=1 \;\vee\; b=1 \;\vee\; a+b=k+1\}$$

be the boundary lattice points of the little triangle. One checks

$$|B| = (k)+(k)+(k)-3 = 3k-3$$

because the three corners $(1,1),(1,k),(k,1)$ are each counted twice.

Now observe:

- A line that is not parallel to any of the three boundary–directions (horizontal, vertical, or antidiagonal) meets the perimeter of the convex triangle in exactly two points (entry and exit). Hence it can pass through at most two boundary lattice points of S_k.

Therefore k such "sunny" lines can together cover at most $2k$ of the $3k-3$ boundary points. To cover all of S_k in particular they must cover all boundary points, so one needs

$$2k \;\ge\; 3k-3 \quad\Longrightarrow\quad k\le 3.$$

Hence for any $k\ge 4$, it is impossible to cover S_k by k sunny lines.

It remains only to check $k=0,1,2,3$:

$k=0$. Then S_0 is empty, and 0 lines do cover it trivially.

$k=1$. Then $S_1 = \{(1,1)\}$. A single non–horizontal, non–vertical, non–antidiagonal line through $(1,1)$ covers it.

$k=2$. Then

$$S_2 = \{(1,1),(1,2),(2,1)\}.$$

But any two of these three points lie on a line of slope 0, ∞, or -1, all of which are forbidden. Hence no one sunny line can ever pass through two of them; two distinct sunny lines could cover at most two points of S_2, so can never cover all three. Thus $k=2$ is impossible.

$k=3$. Here

$$S_3 = \{(1,1),(1,2),(1,3),(2,1),(2,2),(3,1)\}.$$

One checks directly that the three lines

$$L_1:\; y=-2x+5 \quad \text{(through } (1,3),(2,1)\text{)},$$
$$L_2:\; y=-\tfrac{1}{2}\,x+\tfrac{5}{2} \quad \text{(through } (1,2),(3,1)\text{)},$$
$$L_3:\; y=x \quad \text{(through } (1,1),(2,2)\text{)}$$

have slopes -2, $-\frac{1}{2}$, 1, none of which is $0, -1$, or ∞, and together they pass through all six points of S_3. Hence $k=3$ is possible.

Combining these cases, one sees that S_k can be covered by k sunny lines if and only if

$$k \in \{0, 1, 3\}.$$

—

Step 3 - Lifting back to the original n. From Step 1 the existence of an n–line cover of S_n with exactly k sunny lines is equivalent to the existence of a k–line sunny cover of S_k. By Step 2 the latter happens precisely for $k = 0, 1, 3$. Finally one checks directly that for every $n \geq 3$ one can realize

$$k = 0: \quad \text{take horizontals } y = 1, 2, \dots, n,$$

$$k = 1: \quad \text{take } y = 1, 2, \dots, n-1 \text{ (horizontals) plus any one sunny line through } (1, n),$$

$$k = 3: \quad \text{take } y = 1, 2, \dots, n-3 \quad \text{and then the three sunny lines } \begin{cases} y = -2x + (n+2), \\ y = -\frac{1}{2}\,x + \frac{2n-1}{2}, \\ y = x + (n-3), \end{cases}$$

which cover exactly the top little triangle of size 3 as in Step 2, and hence together cover all of S_n.

Answer. For each integer $n \geq 3$, there exist n distinct lines covering all points (a, b) with $a, b \geq 1,\ a + b \leq n + 1$ and having exactly k sunny lines if and only if

$$k \in \{0, 1, 3\}.$$

No other nonnegative integer k is possible. This completes the proof. □

Grading by human evaluator

Reaching the answer $k = 0, 1, 3$ for all n
Description: One point should be given for just finding the answer $k = 0, 1, 3$.
Points: 1 / 1
Comment: Correct.

Describing a construction for $k = 0, 1, 3$ for $n = 3$
Description: For rigorously describing a construction for $n = 3$. Should prove that $k = 2$ is impossible.
Points: 0 / 1
Comment: $k = 2$ has gaps.

Reducing the problem from n to $n - 1$ given a boundary line
Points: 0 / 1
Description: Stating and proving that if a line contains one of the three perimeter sides, the problem can be reduced for $n - 1$ without changing the answer.
Comment: "One checks easily that to maximize the number of points of S_n covered by s non–sunny lines, one should take them all horizontal:" — no.

Making an argument about the boundary points' covering
Description: Stating and proving that the leftmost and bottommost points are covered by n or $n - 1$ lines.
Points: 0 / 1
Comment: Not done for boundary for n, but for k which does not help having in mind the incorrect Step 1.

Origin in the n boundary lines case
Description: Arguing that if n lines cover the leftmost and bottommost points, then the origin is not covered.
Points: 0 / 1
Comment: No such argument.

Hypotenuse points in the $n-1$ boundary lines case
Description: Arguing that if $n-1$ lines cover the leftmost and bottommost points, then the inside points of the hypotenuse cannot be among the covered ones.
Points: 1 / 1
Comment: Done for k instead of n, but the argument is the needed one.

Finishing
Description: Finishing by showing that the aforementioned points cannot be covered by one line.
Points: 0 / 1
Comment: Not present.

Total: 2 / 7 points.

E.4 IMO 2025 - Problem 3

Problem

Let $\mathbb{N}$ denote the set of positive integers. A function $f\colon \mathbb{N} \to \mathbb{N}$ is said to be bonza if

$$f(a) \quad \text{divides} \quad b^a - f(b)^{f(a)}$$

for all positive integers a and b.
Determine the smallest real constant c such that $f(n) \leqslant cn$ for all bonza functions f and all positive integers n.

Solution by Gemini-2.5-Pro

An elegant solution to the problem.
We are looking for the smallest real constant c such that $f(n) \leqslant cn$ for all bonza functions f and all positive integers n. This constant c is given by

$$c = \sup_{f,n} \frac{f(n)}{n},$$

where the supremum is taken over all bonza functions f and all $n \in \mathbb{N}$.
Let the given condition be

$$P(a,b) : f(a) \mid b^a - f(b)^{f(a)}.$$

Part 1: Initial properties of bonza functions

Lemma 1: For any bonza function f, we have $f(1) = 1$.
Proof: Using $P(1,1)$, we have

$$f(1) \mid 1^1 - f(1)^{f(1)},$$

which simplifies to $f(1) \mid 1 - f(1)^{f(1)}$. Since $f(1) \geq 1$, we know $f(1)^{f(1)}$ is divisible by $f(1)$. As $f(1)$ divides both $f(1)^{f(1)}$ and $1 - f(1)^{f(1)}$, it must divide their sum, which is 1. Thus $f(1) = 1$.

Lemma 2: For any bonza function f and any $a \in \mathbb{N}$, we have $f(a) \mid a^a$.
Proof: Using $P(a,a)$, we have $f(a) \mid a^a - f(a)^{f(a)}$. Since $f(a) \geq 1$, $f(a)^{f(a)}$ is divisible by $f(a)$. As $f(a)$ divides both $f(a)^{f(a)}$ and $a^a - f(a)^{f(a)}$, it must divide a^a. Thus, $f(a) \mid a^a$.
An important consequence of Lemma 2 is that any prime divisor of $f(a)$ must also be a prime divisor of a.

Part 2: Analysis based on prime values

Let p be a prime number. By Lemma 2, $f(p) \mid p^p$. This means $f(p)$ must be of the form p^k for some integer $k \geq 0$.

Lemma 3: If p is a prime such that $f(p) > 1$, then $f(b) \equiv b \pmod{p}$ for all $b \in \mathbb{N}$.
Proof: Suppose $f(p) = p^k$ with $k \geq 1$. The condition $P(p, b)$ states $f(p) \mid b^p - f(b)^{f(p)}$. Thus $b^p - f(b)^{f(p)} \equiv 0 \pmod{p}$. By Fermat's Little Theorem, $b^p \equiv b \pmod{p}$. Hence

$$b - f(b)^{p^k} \equiv 0 \pmod{p}.$$

But $f(b)^{p^k} \equiv f(b) \pmod{p}$ by repeated application. Thus $f(b) \equiv b \pmod{p}$.
Let S be the set of prime numbers p for which $f(p) > 1$. We analyze the structure of S.

Case 1: S contains two or more primes
Suppose $p, q \in S$ with $p \neq q$. From Lemma 3 applied to p, $f(q) \equiv q \pmod{p}$. Since $f(q) = q^j$ for some $j \geq 1$, we have $q^j \equiv q \pmod{p}$, i.e., $p \mid q^{j-1} - 1$. If S is infinite, this forces $j = 1$. Hence $f(q) = q$ for all $q \in S$. Thus $f(p) = p$ for all primes p. Then Lemma 3 gives $f(n) \equiv n \pmod{p}$ for all primes p. So $f(n) = n$. This $f(n) = n$ is a bonza function. For this function, $f(n)/n = 1$, so $c \geq 1$.

Case 2: S finite and non-empty
Let $S = \{p_1, \ldots, p_m\}$ and $P = p_1 p_2 \cdots p_m$. For $q \notin S$, we have $f(q) = 1$. Lemma 3 gives $1 \equiv q \pmod{p_i}$ for each $p_i \in S$. So $q \equiv 1 \pmod{P}$. By Dirichlet's theorem, if $P > 2$, there are infinitely many primes $q \not\equiv 1 \pmod{P}$. Contradiction. Thus $P \leq 2$. So S is either empty or $\{2\}$.

Case 3: S empty
If $f(p) = 1$ for all primes p, then for any $n > 1$, $f(n)$ must divide $p^n - 1$ for all primes p. If a prime q divides $f(n)$, take $p = q$. Then $q \mid q^n - 1$, impossible. Thus $f(n) = 1$ for all n. This $f(n) = 1$ is a bonza function, giving $f(n)/n = 1/n \leq 1$.

Part 3: The case $S = \{2\}$ and the value of c
Now let $S = \{2\}$.

- $f(2) > 1$. By Lemma 2, $f(2) \mid 4$, so $f(2) \in \{2, 4\}$.
- $f(p) = 1$ for all odd primes p.
- By Lemma 3, $f(b) \equiv b \pmod{2}$. So $f(b)$ is odd for odd b, and even for even b.

For odd n, $f(n) = 1$. For even $n = 2^k m$ with m odd, $f(n)$ must be a power of 2, say 2^s, and must divide $3^n - 1$. It can be shown that $s \leq v_2(3^n - 1) = k + 2$. So $f(n) \leq 2^{v_2(n)+2}$. Then

$$\frac{f(n)}{n} \leq \frac{2^{k+2}}{2^k m} = \frac{4}{m}.$$

Since $m \geq 1$, this is at most 4. Thus $c \leq 4$.

Part 4: Construction of a bonza function achieving the bound

Define

$$f(n) = \begin{cases} 1 & \text{if } n \text{ odd}, \\ 4 & \text{if } v_2(n) = 1, \\ 2^{v_2(n)+2} & \text{if } v_2(n) \geq 2. \end{cases}$$

It can be checked case by case that this is a bonza function. For $n = 4$, $f(4) = 16$, giving $f(4)/4 = 4$.

Conclusion

We have shown that for any bonza function f and any n, $f(n) \leq 4n$, so $c \leq 4$. We have also constructed an example with $f(4)/4 = 4$, so $c \geq 4$. Therefore the smallest real constant c is

$$\boxed{4}.$$

Grading by human evaluator

$f(p) \neq 1$ **is true for finitely many** p
Description: Showing that there are only a finite number of primes p, for which $f(p) \neq 1$.
Points: 1 / 1
Comment: Correct.

$f(2k+1) = 1$
Description: Showing that if f is not the identity, then $f(a) = 1$ for any odd a.
Points: 2 / 2
Comment: Correct.

$f(2k) = 2^m$
Description: Proving that for any even a, $f(a)$ is a power of 2.
Points: 1 / 1
Comment: Correct.

$\nu_2(f(a)) \leq \nu_2(a) + 2$
Description: Showing that the power of 2 in $f(a)$ does not exceed the power of 2 of a by more than 2.
Points: 1 / 1
Comment: Correct.

Construction for $c = 4$
Description: Deriving a construction for f for which the inequality for $c = 4$ is satisfied tightly.
Points: 2 / 2
Comment: Correct.

Total: 7 / 7 points.