QEDBENCH: Quantifying the Alignment Gap in Automated Evaluation of University-Level Mathematical Proofs

Santiago Gonzalez [*1] **Alireza Amiri Bavandpour** [*1] **Peter Ye** [*1] **Edward Zhang** [*1] **Ruslans Aleksejevs** [2] **Todor Antić** [3] **Polina Baron** [4] **Sujeet Bhalerao** [5] **Shubhrajit Bhattacharya** [4] **Zachary Burton** [6] **John Byrne** [7] **Hyungjun Choi** [8] **Nujhat Ahmed Disha** [6] **Koppány István Encz** [9] **Yuchen Fang** [2] **Robert Joseph George** [10] **Ebrahim Ghorbani** [11] **Alan Goldfarb** [2] **Jing Guo** [12] **Meghal Gupta** [2] **Stefano Huber** [9] **Annika Kanckos** [13] **Minjung Kang** [14] **Hyun Jong Kim** [15] **Dino Lorenzini** [16] **Levi Lorenzo** [17] **Tianyi Mao** [18] **Giovanni Marzenta** [19] **Ariane M. Masuda** [20] **Lukas Mauth** [21] **Ana Mickovic** [22] **Andrés Miniguano-Trujillo** [23] **Antoine Moulin** [24] **Wenqi Ni** [25] **Tomos Parry** [26] **Kevin Ren** [8] **Hossein Roodbarani** [27] **Mathieu Rundström** [28] **Manjil Saikia** [29] **Detchat Samart** [30] **Rebecca Steiner** [31] **Connor Stewart** [32] **Dhara Thakkar** [33] **Jeffrey Tse** [34] **Vasiliki Velona** **Yunhai Xiang** [15] **Sibel Yalçın** [35] **Jun Yan** [34] **Ji Zeng** [36] **Arman Cohan** [1] **Quanquan C. Liu** [1]

Abstract

Self-evolving mathematical agents require reliable internal critics: a system that can generate conjectures or proofs but cannot verify them will optimize for persuasive mistakes. We introduce QEDBENCH, a benchmark and audit protocol for measuring whether frontier LLM judges align with expert mathematicians when grading upper-undergraduate and early-graduate natural-language proofs. QEDBENCH contains 272 proof problems across ten mathematical domains, more than 1,300 frontier-model-generated proofs, over 1,000 hours of expert human evaluation, and a fully crossed 7×5 evaluator-solver matrix under both expert and course-specific rubrics. We find a systematic alignment gap: several LLM judges accept flawed proofs at high rates, with `Llama 4 Maverick` reaching a 74.8% false-positive leniency rate and evaluators such as `Qwen 2.5 Max` and `Llama 4 Maverick` showing significant positive correlation between score and LaTeX density. Additional mixed-effects, binary-decision, contamination, and rubric-ablation analyses rule out simple confounds such as same-family self-preference, score discretization, and underspecified prompts. Our results show that current LLM judges are not yet reliable autonomous verifiers for mathematical agents, and provide a concrete benchmark for training and auditing the next generation of proof evaluation systems.

[*]Equal contribution [1]Department of Computer Science, Yale University, New Haven, CT, USA [2]UC Berkeley [3]Charles University [4]The University of Chicago [5]University of Illinois Urbana-Champaign [6]MIT [7]University of Delaware [8]Princeton University [9]USI, IDSIA [10]California Institute of Technology [11]Technische Universität Hamburg [12]Universität Regensburg [13]University of Helsinki [14]Illinois Institute of Technology [15]University of Western Ontario [16]University of Georgia [17]Fort Lewis College [18]Helm.ai [19]University of British Columbia [20]New York City College of Technology (CUNY) [21]University of Cologne [22]University of Amsterdam [23]King's College London [24]Universitat Pompeu Fabra [25]University of Edinburgh [26]Bilkent University [27]York University [28]University of Waterloo [29]Ahmedabad University, India [30]Burapha University, Thailand [31]Johannes Gutenberg University Mainz [32]CUNY Hunter College [33]Nagoya University [34]University of Oxford [35]Bursa Uludağ University [36]Alfréd Rényi Institute of Mathematics. Correspondence to: Quanquan C. Liu <quanquan.liu@yale.edu>.

The 3rd *AI for Math Workshop* at the 43rd International Conference on Machine Learning (ICML), Seoul, South Korea, 2026.
Copyright 2026 by the author(s).

1. Introduction

Recent mathematical reasoning systems are increasingly agentic: they search, propose, critique, and refine. The bottleneck is therefore shifting from *generation* to *verification*. If an agent's verifier rewards a proof because it is well formatted or rhetorically plausible, self-improvement loops can amplify exactly the behavior that mathematical systems must avoid: authoritative but invalid reasoning.

This risk is acute in natural-language university mathematics. Formal theorem proving in Lean or related systems offers machine-checkable guarantees (de Moura et al., 2015), but natural-language proof evaluation remains central to teaching, research communication, and hybrid autoformalization pipelines. Existing mathematical benchmarks often emphasize final answers (Hendrycks et al., 2021; Cobbe et al., 2021), formal statements, or generation performance. By contrast, QEDBENCH audits the *judges*: can frontier LLM evaluators grade full natural-language proofs the way

human experts do?

Contributions. We make four contributions aimed at reliable evaluation of mathematical reasoning.

- **A university-level proof-evaluation benchmark.** QEDBENCH includes 272 proof problems from ten upper-undergraduate and early-graduate domains, five frontier solver models, seven frontier judge models, and 1,000+ hours of expert evaluation.
- **A dual-rubric judge audit.** Each problem is evaluated under an Expert Rubric and a Course-Specific Rubric. This directly tests whether more explicit pedagogical constraints, such as forbidding advanced machinery, improve LLM judge alignment.
- **Evidence of a persistent alignment gap.** LLM judges show large domain-dependent false-positive behavior, especially in constructive discrete domains. The failure persists under stricter rubrics and binary Pass/Fail prompts.
- **Causal checks for common confounds.** We analyze same-family self-preference, proof length and LaTeX density, contamination, score discretization, and calibration settings. The main failure cannot be explained by these artifacts.

Why verifier alignment matters. Mathematical agents can only improve reliably when their critics remain more grounded than their generators. Our thesis is that the verifier is the bottleneck. A weak generator makes visible mistakes; a weak judge silently converts mistakes into reward. QEDBENCH therefore evaluates the component that future self-improving systems will most need: a critic that rejects persuasive but invalid proofs, distinguishes proof styles from proof correctness, and remains calibrated across mathematical domains.

2. Related Work

Mathematical reasoning benchmarks. GSM8K and MATH popularized large-scale evaluation of mathematical problem solving, but rely primarily on short-form answer checking (Cobbe et al., 2021; Hendrycks et al., 2021). Recent benchmarks move toward harder mathematics: FrontierMath targets expert-level closed-form problems (Glazer et al., 2024); IMProofBench studies research-level proof generation (Schmitt et al., 2025); MathConstruct emphasizes constructive objects (Balunović et al., 2025); and U-MATH includes university-level questions with a judge-consistency subset (Chernyshev et al., 2025). QEDBENCH occupies a complementary setting: full natural-language proof generation at university level, with human-verified proof scores and a primary focus on auditing evaluators rather than solvers.

Automated proof evaluation. LLM-as-a-judge systems are widely used for scalable evaluation (Zheng et al., 2023), including mathematical proof settings such as ProofBench and RefGrader (Ma et al., 2025; Mahdavi et al., 2025b). However, recent studies show that LLMs can overestimate proof quality, reward style, or fail to reject invalid mathematical reasoning (Dekoninck et al., 2025; Petrov et al., 2025; Mahdavi et al., 2025a;c; Stephan et al., 2025; Jain et al., 2025). QEDBENCH extends this line by isolating judge behavior across domains, solvers, and rubrics, and by adding confound analyses for self-preference and formatting bias.

Formal verification and autoformalization. Formal proof assistants can certify proofs once a theorem and proof have been correctly formalized. But autoformalization may preserve syntax while changing mathematical intent, and current systems can struggle with zero-shot formal proofs of advanced natural-language problems. Mathematical reasoning systems therefore need both formal tools and robust natural-language verifiers. Our benchmark targets the latter: the judge alignment layer that must decide whether an informal proof genuinely establishes the intended mathematical claim.

What is missing. The missing object is not another leaderboard of solvers, but a *calibration set for mathematical critics*: a dataset where judge decisions can be compared against expert proof-level ground truth, stress-tested across domains, and probed for causal failure modes such as style bias, self-preference, and rubric insensitivity. QEDBENCH is designed to fill that gap.

This focus also separates QEDBENCH from purely formal benchmarks. Formal verification asks whether a formal proof object checks against a formal theorem. Our setting asks whether a natural-language proof actually establishes the intended informal theorem. That semantic layer is unavoidable in teaching, research communication, autoformalization, and agentic pipelines that translate between informal and formal mathematics.

3. The QEDBENCH Benchmark

3.1. Problems and Solvers

QEDBENCH contains 272 proof-based problems across Analysis, Complex Analysis, Abstract Algebra, Discrete Math, Probability Theory, ODEs, Number Theory, Combinatorics, Algorithms, and Graph Theory. Problems were sourced from advanced textbooks and qualifying-exam style material, then manually rephrased into equivalent *ab initio* statements to reduce memorization risk.

We also ran an adversarial web-search audit. For each candi-

date, we asked `o3-deep-research` to distinguish exact or mathematically equivalent online solutions from merely analogous problems. The goal was not to claim perfect decontamination, but to prevent the benchmark from becoming a retrieval test. We later compare online-labeled and offline-labeled subsets and find no significant performance advantage for the online-labeled group.

We generated solutions with five frontier solvers: `o3-deep-research`, `GPT-5 Pro`, `Claude Sonnet 4.5`, `Gemini 3.0 Pro`, and `DeepSeek-Prover-V2`. This yields more than 1,300 full proofs. We allowed long-form generation with model-specific token budgets and retries. Failed generations after retries were scored as zero, reflecting inability to produce a usable proof under deployment constraints.

3.2. Expert Ground Truth

We recruited 48 expert evaluators with PhD-level training in mathematics or theoretical computer science and matched them to problems in their domains of expertise. Human evaluation required over 1,000 total hours. Each proof was graded on the shared scale $[0, 0.25, 0.5, 0.75, 0.9, 1.0]$, where 0.9 denotes a correct proof with minor expository oversight and 1.0 denotes a fully rigorous proof. Whenever possible, each problem was assigned to two independent experts; disagreements were adjudicated by a third expert. Graders were instructed to accept any logically valid proof path, including non-reference solutions.

The scale is intentionally coarse but semantically meaningful. Scores below 0.9 indicate that the proof should not be accepted as correct. A score of 0.75 corresponds to a promising proof with a local error or unjustified step; 0.5 indicates substantial partial progress with a serious gap; and 0 or 0.25 indicates little or no valid progress toward the target theorem. This lets QEDBENCH evaluate both strict correctness and partial mathematical understanding.

3.3. Dual-Rubric Judge Audit

For each problem, we construct two rubrics. **Expert Rubrics** grade mathematical correctness assuming standard expert common knowledge. **Course-Specific Rubrics** simulate undergraduate grading: they require textbook-level definitions, penalize unproved advanced machinery, and demand explicit derivations that a course instructor would expect. Human experts refined rubric drafts to match their grading standards.

Seven LLM judges evaluate every solver proof under both rubrics: `GPT-5.2 Pro`, `Claude Opus 4.5`, `Gemini 3.0 Pro`, `DeepSeek-V3`, `Grok 4`, `Llama 4 Maverick`, and `Qwen 2.5 Max`. All automated evaluations use temperature $T = 0.0$ for deterministic reproducibility. This forms a fully crossed 7×5 evaluator-solver matrix that separates solver quality from judge bias.

Table 1. QEDBENCH at a glance. The benchmark is designed to audit proof *judges*, not only proof solvers.

Problems	272 university-level proofs
Domains	10 math/TCS areas
Solver models	5 frontier systems
Judge models	7 frontier systems
Human evaluators	48 domain experts
Expert effort	1,000+ hours
Rubrics	Expert and course-specific
Main matrix	7 judges $\times$ 5 solvers

Table 2. Scale comparison with expert-reasoning benchmarks. QEDBENCH uniquely combines university-level natural-language proofs, expert human proof scores, and a multi-judge audit matrix.

Benchmark	Items	Target
GPQA Diamond (Rein et al., 2023)	198	PhD-level science QA
ProofNet (Azerbayev et al., 2023)	371	Undergraduate formalization
FrontierMath (Glazer et al., 2024)	350	Expert math problem solving
QEDBENCH	272	University-level proof-judge alignment

3.4. Metrics

Let H_i denote the human consensus score for proof i and J_i the score assigned by an LLM judge. We report mean score bias,

$$\Delta = \mathbb{E}_i[J_i - H_i],$$

and use the shared pass threshold $\tau = 0.9$ for binary metrics. A proof passes if its score is at least τ. The *Leniency Rate* is the false-positive rate

$$\Pr[J_i \geq \tau \mid H_i < \tau],$$

and the *Harshness Rate* is the false-negative rate

$$\Pr[J_i < \tau \mid H_i \geq \tau].$$

These metrics separate two very different failures: accepting invalid proofs and rejecting valid ones. For mathematical agents, leniency is especially dangerous because it creates a reward signal for invalid reasoning.

3.5. Scale Relative to Peer Benchmarks

QEDBENCH is smaller than high-school short-answer datasets, but it is large for expert-verified proof evaluation. Its size is best understood not only by problem count, but by proof-level annotations: each problem is solved by multiple frontier systems and judged by both humans and LLMs.

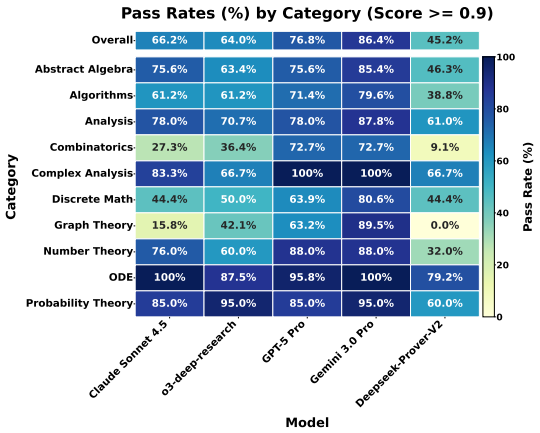

Figure 1. Human pass rates (score ≥ 0.9) by solver and discipline. Frontier solvers excel in procedural continuous domains but degrade in constructive discrete domains.

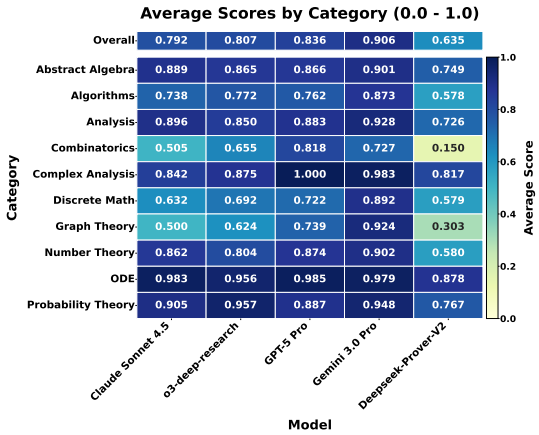

Figure 2. Average human scores by solver and discipline. Average score preserves partial-credit information that strict pass rate hides. Near-miss proofs often look mathematically sophisticated while still failing the target theorem.

3.6. Why Human Ground Truth Rather Than Lean Alone?

We considered Lean 4 formal verification early in the project. Formal checking is the right end state when the theorem and proof are faithfully formalized, but it is not a substitute for the semantic audit studied here. Autoformalization can be reward-hacked: a system can produce a valid proof of a weakened or shifted statement. Detecting that mismatch requires exactly the kind of natural-language judgment QEDBENCH measures. Moreover, zero-shot Lean proofs at this level would partly evaluate Lean proficiency rather than mathematical proof evaluation. We therefore use expert human annotation as ground truth while viewing formal methods as complementary.

4. Results

4.1. The Discrete-Continuous Divide

The strongest solver, `Gemini 3.0 Pro`, achieves an 86.4% overall pass rate and average human score 0.91. Yet performance is sharply domain-dependent. Procedural continuous domains such as ODEs are near-saturated: `Gemini 3.0 Pro` and `Claude Sonnet 4.5` both achieve 100.0% pass rates in ODEs. Constructive discrete domains are much harder: `Claude Sonnet 4.5` drops to 27.3% in Combinatorics and 15.8% in Graph Theory, while `o3-deep-research` reaches only 42.1% in Graph Theory.

This divide matters for autonomous mathematical reasoning because discrete proofs often require search over finite structures, construction of objects, and verification of global invariants. These are precisely the skills that self-evolving agents need when proposing new mathematical claims.

Partial-credit trap. Pass rates and average scores tell different stories. Some solvers produce many near-miss proofs: coherent arguments that earn 0.5 or 0.75 but fail because of an edge case, false implication, or missing construction. These proofs are important because they are the easiest failures for an automated judge to over-reward. In a self-training loop, partial-credit near misses can become a source of hallucinated rigor unless the critic learns to identify the exact broken dependency.

4.2. LLM Judges Inflate Scores

We next compare automated judges against expert human scores. Human experts establish a 67.7% pass-rate baseline. The strictest evaluator, `DeepSeek-V3`, passes 72.2% of proofs, while `Llama 4 Maverick` passes 90.2%. This is not merely a global strictness shift: biases are domain-dependent. In Combinatorics, `Llama 4 Maverick` inflates mean scores by $+0.36$ and `Qwen 2.5 Max` by $+0.30$ relative to human experts. In Complex Analysis, some judges are instead harsher than humans, penalizing implicit expert-standard steps.

We decompose binary errors using score ≥ 0.9 as pass. `Llama 4 Maverick` has a 74.8% *Leniency Rate*: among proofs humans reject, it accepts nearly three quarters. Even `GPT-5.2 Pro`, the best-calibrated judge, has a 38.0% leniency rate. These results show that current LLM judges can identify many correct proofs, but remain unreliable at rejecting plausible invalid ones.

The matrix view. The 7×5 matrix exposes a practical risk for benchmark design: a solver can look strong under a lenient judge and weak under a strict one. This is not a small leaderboard artifact. It changes which model appears to have made progress. A reliable mathematical-reasoning benchmark therefore needs either expert calibration, multi-

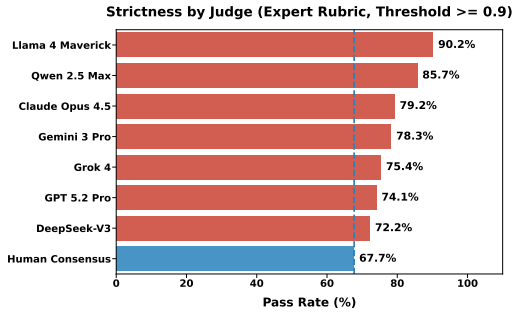

Figure 3. Evaluator strictness under the Expert Rubric. Pass-rate agreement with humans is not enough: a judge can have a similar average pass rate while making different false-positive and false-negative errors.

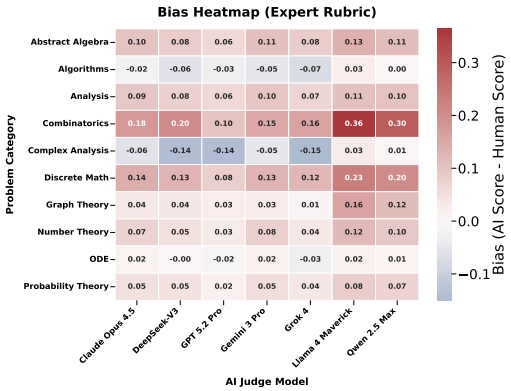

Figure 4. Evaluator score bias by domain under the Expert Rubric, $\Delta = S_{\text{AI}} - S_{\text{Human}}$. Positive values indicate AI leniency. Misalignment is domain-specific rather than a single global calibration offset.

ple independent judges with measured error profiles, or a judge trained explicitly against human proof-level annotations.

This also explains why final-answer benchmarks can look stable while proof benchmarks do not. When the answer is short, many evaluation errors collapse into exact-match success or failure. When the output is a proof, the evaluator must decide whether each local claim is true, whether the claims compose, whether the proof handles the correct quantifiers, and whether the argument proves the requested theorem rather than a nearby theorem. The matrix makes those hidden evaluator choices visible.

4.3. Rubric Specificity Does Not Fix Judges

The dual-rubric design tests a common hypothesis: perhaps judges fail because rubrics are underspecified. We find the opposite. Switching from the Expert Rubric to the Course-Specific Rubric changes scores only marginally. For `GPT-5.2 Pro`, Pearson correlation with human consensus changes from $r = 0.69$ to $r = 0.67$, and MAE

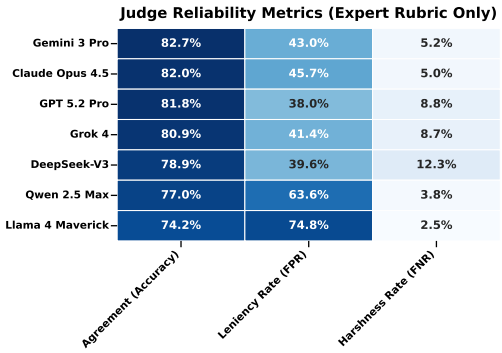

Figure 5. Judge error decomposition under the Expert Rubric. Leniency corresponds to false positives: proofs rejected by humans but accepted by an LLM judge.

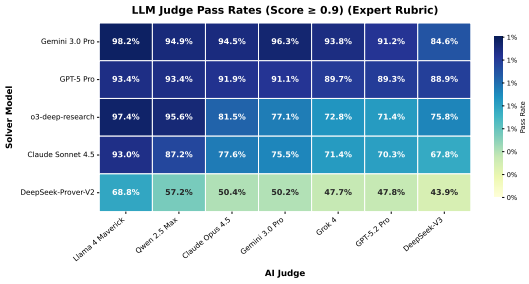

Figure 6. The evaluator-solver matrix under the Expert Rubric. Solver rankings depend strongly on the chosen judge, showing why single-judge leaderboards are fragile for mathematical proofs.

from 0.13 to 0.14. Course-specific constraints make some strong logic models slightly more punitive, but the overall evaluator-solver matrix remains structurally similar.

We call this **Rubric Insensitivity**: a judge's internal representation of mathematical correctness dominates explicit negative instructions. For educational and agentic settings, this is important. A natural-language instruction such as "do not use advanced theorems without proof" is insufficient if the judge cannot reliably enforce it.

4.4. Robustness to Confounds

We ran additional analyses to rule out simpler explanations for the alignment gap.

Self-preference. A mixed-effects regression controls for human proof quality, baseline judge strictness, and proof difficulty. Same-family effects are small: Claude -0.0024 ($p = 0.6505$), Gemini $+0.0332$ ($p < 10^{-4}$), DeepSeek -0.0141 ($p = 0.0088$), and GPT -0.0086 ($p = 0.0504$). These values are negligible relative to observed grade-inflation gaps.

Score discretization. We also force strict binary Pass/Fail decisions. Leniency trends persist, so the score scale itself is not the cause.

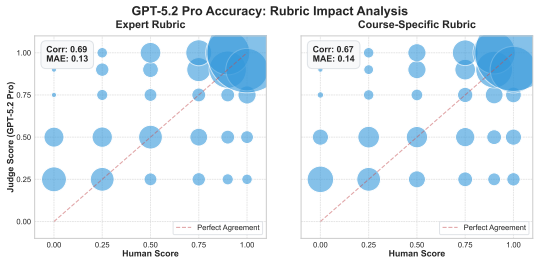

Figure 7. Rubric ablation for `GPT-5.2 Pro`. Course-specific constraints do not substantially improve alignment with human scores.

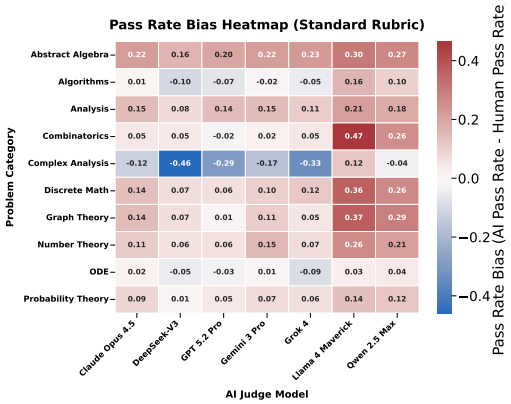

Figure 8. Pass-rate bias under the Expert Rubric. The binary version of the bias analysis mirrors the score-bias heatmap, showing that the alignment gap is not caused by partial-credit scoring.

Proof-by-intimidation. We correlate judge scores with LaTeX density, a proxy for visually dense mathematical presentation. `Qwen 2.5 Max` and `Llama 4 Maverick` show significant positive correlations ($r = 0.169$ and $r = 0.157$, $p < 10^{-4}$), while `GPT-5.2 Pro` shows a slight negative correlation ($r = -0.036$, $p = 0.0653$). Some judges are therefore susceptible to proof-by-intimidation: dense mathematical formatting can raise scores even when humans reject the proof.

Contamination. Among problems with high-confidence online/offline labels, online-labeled problems show only a small mean-score gap ($+0.017$) and pass-rate gap ($+0.011$), with no significant advantage. This supports the interpretation that the main effects are reasoning and judging failures rather than memorization artifacts.

Binary decisions preserve the failure. The pass-rate heatmap rules out a simple objection: perhaps judges only differ because they interpret partial credit differently. They do not. When reduced to a strict accept/reject decision, the same structure remains. Discrete constructive domains still show inflated acceptance, and continuous domains still show pockets of over-strictness. The problem is therefore not a cosmetic scoring mismatch; it is a failure to make the same mathematical accept/reject decisions as experts.

Table 3. Robustness checks. The alignment gap persists after controlling for common confounds, while formatting density explains part of the weakest judges' leniency.

Confound	Finding
Same-family bias	Coefficients range from -0.014 to $+0.033$ after controls.
Score discretization	Binary Pass/Fail prompts preserve leniency trends.
LaTeX density	Qwen and Llama correlate positively with symbolic density.
Temperature	All judge runs use $T = 0.0$ for deterministic grading.
Contamination	Online/offline problem split shows no significant advantage.

Table 4. Two causal probes from the rebuttal analysis. Same-family effects are too small to explain the main bias; LaTeX density reveals proof-by-intimidation in lenient judges.

Probe	**Effect**	p
Claude self-family	-0.0024	0.6505
Gemini self-family	$+0.0332$	$< 10^{-4}$
DeepSeek self-family	-0.0141	0.0088
GPT self-family	-0.0086	0.0504
Qwen LaTeX density	$r = +0.169$	$< 10^{-4}$
Llama LaTeX density	$r = +0.157$	$< 10^{-4}$
GPT LaTeX density	$r = -0.036$	0.0653

Why leniency is the central risk. Harshness is costly: it rejects valid work. Leniency is more dangerous for autonomous systems because it certifies invalid work. In a self-evolving agent, lenient judges promote flawed candidates into the training or search frontier. The resulting model can become better at producing proofs that look judge-approved while still being mathematically invalid. This is why QEDBENCH emphasizes false positives rather than only correlation or average error.

4.5. Qualitative Failure Modes

The most important failures are not small numerical calibration errors. They are failures of global proof verification.

Recognizing a flaw but still passing. In one Analysis problem, a solver had to construct a function satisfying a non-membership condition across all $q \neq p$. `GPT-5.2 Pro` correctly identified that the construction failed for the $q < p$ case, but still awarded 0.5/1.0 because the proof had partial structure.

Missing assumptions. In a Graph Theory problem about plane duals, human experts penalized proofs that omitted connectedness assumptions needed for the dual correspondence. LLM judges praised the same proofs as well structured.

Rewarding irrelevant or impossible proofs. Some solvers produced coherent mathematics for the wrong theorem,

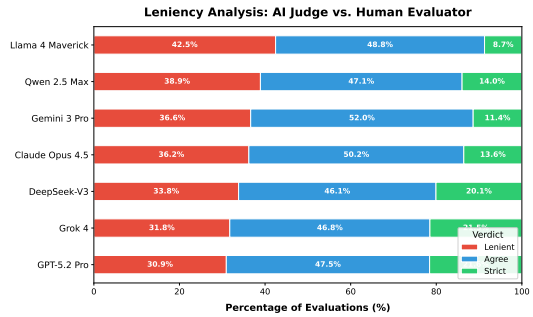

Figure 9. Per-judge distribution of score direction relative to humans. Lenient judgments are especially damaging because they turn rejected proofs into positive training signal.

Table 5. Representative qualitative failures. The common thread is failure to verify the global target theorem.

Failure mode	Observed behavior
Recognized fatal flaw	Judge identifies a failed case but still awards generous partial credit.
Missing assumption	Judge praises plane-dual proof while missing required connectedness.
Wrong theorem	Coherent but irrelevant proof receives nonzero credit.
Impossible object	Hallucinated construction is rewarded for surface rigor.

empty API-error outputs, or hallucinated constructions of impossible objects. Human experts assigned zero, while LLM judges sometimes granted nonzero credit for formatting, effort, or apparent rigor.

These examples suggest that LLM judges often grade a proof as a collection of locally plausible statements rather than tracing whether the chain of implications proves the target theorem.

The common pattern is target drift. A judge recognizes local mathematical language, but fails to maintain the invariant "does this prove exactly the requested statement?" That invariant is central to both proof assistants and human grading, and it is the behavior future LLM critics must learn.

5. Implications for Mathematical Agents

Self-improvement needs grounded critics. A mathematical reasoning system that trains on its own judge's feedback can amplify judge biases. If a verifier rewards dense but invalid proofs, the generator will learn to produce denser invalid proofs. QEDBENCH provides a stress test for this feedback loop.

Natural-language and formal verification should be complementary. Formal verification can certify a proof after correct formalization, but autoformalization can target the wrong or weakened statement. Natural-language judges must therefore verify semantic intent, not only local syntax. Our results show that current LLMs are not yet reliable enough to serve as sole semantic arbiters.

Better judges likely require process supervision. Rubric engineering, deterministic decoding, and binary prompts do not close the gap. Future judges may need training signals that supervise dependency tracking, theorem-target matching, false-premise detection, and adversarial rejection of plausible invalid proofs.

A benchmark for judge training. The same artifacts that expose failures also support progress: proof-level human scores provide supervised labels; textual expert comments identify local and global errors; dual rubrics separate mathematical correctness from pedagogical compliance; and the evaluator-solver matrix enables targeted training against false positives. This makes QEDBENCH a practical substrate for building critics for self-evolving mathematical agents.

6. What QEDBENCH Enables

Training rejection models. The most valuable negative examples in QEDBENCH are not nonsense outputs; they are plausible proofs that fail for subtle reasons. These examples can train critics to reject invalid reasoning without relying on superficial features such as length, equation density, or confident tone. The binary pass labels identify accept/reject decisions, while partial scores distinguish fatal gaps from incomplete but meaningful progress.

Auditing judge upgrades. Because QEDBENCH contains the same proofs evaluated by humans and multiple LLM judges, it can measure whether a new judge improves for the right reason. A genuine improvement should reduce leniency on human-rejected proofs without becoming excessively harsh on human-accepted proofs. It should also improve in the hardest domains, especially Combinatorics and Graph Theory, rather than simply shifting global strictness.

Stress-testing self-evolution loops. Agentic mathematical systems often use an evaluator to select candidates, revise proofs, or generate synthetic training data. QEDBENCH can be used as a pre-deployment test for that evaluator. If the judge rewards proof-by-intimidation or ignores course-specific constraints, the loop is likely to amplify those weaknesses. This makes judge alignment a safety and reliability problem, not just a benchmarking detail.

Connecting informal and formal mathematics. The benchmark also supports hybrid systems. A natural-language critic can first verify that the proof targets the intended theorem and identify likely gaps; a formal pipeline can then attempt faithful formalization. Failed formalization attempts can be compared against expert comments to distinguish syntax errors from genuine mathematical gaps.

Thus QEDBENCH links informal reasoning to formal verification.

7. Design Principles for Better Critics

Evaluate theorem-target matching, not only local plausibility. Many failures occur because the proof establishes something adjacent to the target theorem. A critic should explicitly track the statement being proved, the quantifiers and hypotheses in force, and the final dependency chain. This suggests training objectives that reward target preservation and penalize proofs of weakened, special-case, or shifted claims.

Separate mathematical correctness from presentation quality. Dense notation and fluent exposition are useful signals for humans, but they are not correctness. The LaTeX-density correlations show that some judges confuse mathematical appearance with proof validity. Future critics should be adversarially trained on pairs where the invalid proof is more polished than the valid one, forcing the evaluator to attend to logical dependencies.

Supervise the rejection decision. General-purpose LLM judges often produce a reasonable critique and then assign an overly generous final score. This means the failure is partly in decision calibration. A better critic should learn when a detected flaw is fatal. In particular, missing a required case, assuming a forbidden hypothesis, or proving the wrong theorem should trigger rejection even if the proof contains substantial correct material.

Audit across domains. A single aggregate correlation can hide domain failures. In QEDBENCH, the same judge can be lenient in Combinatorics and harsh in Complex Analysis. Best-practice evaluation should therefore report domain-conditioned false positives and false negatives, not only an overall score. This matters because future agents will work across heterogeneous mathematical areas.

These principles turn QEDBENCH from a static benchmark into a development loop. A new mathematical critic should be evaluated not by whether it is globally stricter, but by whether it rejects the right invalid proofs, preserves valid alternative proof styles, and improves in domains where current judges fail.

Table 6. Failure-to-training-signal map for future judge models.

Failure	Training signal
Nearby theorem	Track target and quantifiers.
Dense invalid proof	Penalize gaps despite polish.
Detected fatal flaw	Convert recognition into rejection.
Domain-specific bias	Calibrate errors by field.
Rubric violation	Enforce negative constraints.

8. Limitations

QEDBENCH focuses on English-language natural proofs and static expert ground truth. Some valid non-standard proofs may be undercounted despite multi-expert adjudication, although graders were explicitly instructed to accept any logically sound proof path. The benchmark is smaller than high-school short-answer datasets, but comparable to expert-reasoning benchmarks: GPQA has 448 graduate-level science questions with a 198-question Diamond subset (Rein et al., 2023), ProofNet has 371 undergraduate-level formalization examples (Azerbayev et al., 2023), and FrontierMath has 350 advanced problems (Glazer et al., 2024). Finally, QEDBENCH evaluates judges post hoc. The next step is to measure how these judge errors propagate inside actual self-improvement loops, where false positives may be repeatedly selected, refined, and distilled.

9. Conclusion

QEDBENCH shows that the central challenge for AI mathematical agents is no longer only generating plausible proofs, but verifying them. Across 272 university-level problems, 1,300+ generated proofs, 48 human experts, and a 7×5 judge-solver matrix, frontier LLM judges systematically diverge from human mathematical judgment. The failures persist under stricter rubrics, deterministic decoding, and binary prompts, and are partly explained by proof-by-intimidation effects. For self-evolving scientific agents to be trustworthy, their internal critics must become substantially more grounded than today's LLM-as-a-judge systems.

Takeaway. QEDBENCH provides a simple stress test: before a system is trusted to improve itself, its critic should reject the polished wrong proof, preserve the unconventional correct proof, and remain calibrated across domains. The current generation of LLM judges does not yet pass that test. Closing this gap is a concrete path toward mathematical agents that are not merely fluent, but reliably self-correcting.

Pedagogical alignment. The course-specific rubric exposes a second failure mode: judges must enforce not only theorem-level correctness, but also the permitted proof language. Human experts routinely penalized solvers that invoked advanced machinery without derivation, silently assumed missing hypotheses, or bypassed the requested constructive argument, while LLM judges often failed to enforce these negative constraints even when explicitly prompted. This matters for self-improving systems because a verifier that cannot reject out-of-scope shortcuts will train generators to retrieve authority rather than construct arguments. A useful critic must therefore accept sound alternative proof paths while rejecting persuasive violations of the target theorem and of the stated rules of inference.

References

Azerbayev, Z., Piotrowski, B., Schoelkopf, H., Ayers, E. W., Radev, D., and Avigad, J. ProofNet: Autoformalizing and formally proving undergraduate-level mathematics. *arXiv preprint arXiv:2302.12433*, 2023.

Balunović, M., Dekoninck, J., Jovanović, N., Petrov, I., and Vechev, M. T. MathConstruct: Challenging LLM reasoning with constructive proofs. In *International Conference on Machine Learning (ICML)*, 2025.

Chernyshev, K., Polshkov, V., Stepanov, V., Myasnikov, A., Artemova, E., Miasnikov, A., and Tilga, S. U-MATH: A university-level benchmark for evaluating mathematical skills in large language models. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM)*, 2025.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

de Moura, L., Kong, S., Avigad, J., van Doorn, F., and von Raumer, J. The Lean theorem prover (system description). In *International Conference on Automated Deduction (CADE)*, pp. 378–388. Springer, 2015.

Dekoninck, J., Petrov, I., Minchev, K., Balunović, M., Vechev, M., Marinov, M., Drencheva, M., Konova, L., Shumanov, M., Tsvetkov, K., Drenchev, N., Todorov, L. D., Nikolova, K., Georgiev, N., Kalinkova, V., and Ismoldayev, M. The open proof corpus: A large-scale study of LLM-generated mathematical proofs. *arXiv preprint arXiv:2506.21621*, 2025.

Glazer, E., Erdil, E., Besiroglu, T., Chicharro, D., Chen, E., Gunning, A., Olsson, C. F., Denain, J.-S., Ho, A., Santos, E. d. O., Järviniemi, O., Barnett, M., Sandler, R., Sevilla, J., Ren, Q., Pratt, E., Levine, L., Barkley, G., Stewart, N., Grechuk, B., Grechuk, T., and Enugandla, Shreepranav Varma, M. W. FrontierMath: A benchmark for evaluating advanced mathematical reasoning in AI. *arXiv preprint arXiv:2411.04872*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the MATH dataset. In *NeurIPS*, 2021.

Jain, S., Ahmed, U. Z., Sahai, S., and Leong, B. Beyond consensus: Mitigating the agreeableness bias in LLM judge evaluations. *arXiv preprint arXiv:2510.11822*, 2025.

Ma, W., Cojocaru, A., Kolhe, N., Louie, B., Sharif, R. S., Zhang, H., Zhuang, V., Zaharia, M., and Min, S. Reliable fine-grained evaluation of natural language math proofs. *arXiv preprint arXiv:2510.13888*, 2025.

Mahdavi, H., Hashemi, A., Daliri, M., Mohammadipour, P., Farhadi, A., Malek, S., Yazdanifard, Y., Khasahmadi, A., and Honavar, V. Brains vs. bytes: Evaluating LLM proficiency in olympiad mathematics. *arXiv preprint arXiv:2504.01995*, 2025a.

Mahdavi, H., Mahdavinia, P., Malek, S., Mohammadipour, P., Hashemi, A., Daliri, M., Farhadi, A., Khasahmadi, A., Mireshghallah, N., and Honavar, V. RefGrader: Automated grading of mathematical competition proofs using agentic workflows. *arXiv preprint arXiv:2510.09021*, 2025b.

Mahdavi, S., Kisacanin, B., Toshniwal, S., Du, W., Moshkov, I., Armstrong, G., Liao, R., Thrampoulidis, C., and Gitman, I. Scaling generative verifiers for natural language mathematical proof verification and selection. *arXiv preprint arXiv:2511.13027*, 2025c.

Petrov, I., Dekoninck, J., Baltadzhiev, L., Drencheva, M., Minchev, K., Balunović, M., Jovanović, N., and Vechev, M. T. Proof or bluff? evaluating LLMs on 2025 USA math olympiad. *arXiv preprint arXiv:2503.21934*, 2025.

Rein, D., Hou, B. L., Stickland, A. C., Petty, J., Pang, R. Y., Dirani, J., Michael, J., and Bowman, S. R. GPQA: A graduate-level google-proof Q&A benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

Schmitt, J., Bérczi, G., Dekoninck, J., Feusi, J., Gehrunger, T., Appenzeller, R., Bryan, J., Canova, N., de Wolff, T., Gaia, F., van Garrel, M., Hashemi, B., Holmes, D., Lopez, A., Jaeck, V., Jørgensen, M., Kelk, S., Kuhlmann, S., Kurpisz, A., Meroni, C., Metzler, I., Möller, M., Muñoz-Echániz, S., Nowak, R., Oberdieck, G., Platt, D., Possamaï, D., Ribeiro, G., Galán, R. S., Sun, Z., Teichmann, J., Thomas, R. P., and Vial, C. IMProofBench: Benchmarking AI on research-level mathematical proof generation. *arXiv preprint arXiv:2509.26076*, 2025.

Stephan, A., Zhu, D., Aßenmacher, M., Shen, X., and Roth, B. From calculation to adjudication: Examining LLM judges on mathematical reasoning tasks. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics (GEM)*, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E. P., Zhang, H., Gonzalez, J. E., and Stoica, I. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *NeurIPS*, 2023.