Mathematical Proof as a Litmus Test: Revealing Failure Modes of Advanced Large Reasoning Models

Anonymous ACL submission

Abstract

Large reasoning models (*e.g.,* R1, o3) have demonstrated remarkable mathematical problem-solving abilities. However, the high reported accuracy of these advanced models on popular datasets and reliance on purely numerical evaluation often mask their true reasoning shortcomings. To address this, we propose leveraging the inherent rigor and methodological complexity of mathematical proofs as a diagnostic tool to expose these hidden failures. Specifically, we introduce the ***RFMDataset*** (***R***eveal ***F***ailure ***M***odes), a collection of 200 diverse mathematical proof problems to thoroughly evaluate the performance of advanced models. Our in-depth analysis of their failures uncovers 10 fine-grained error types, which shows fundamental limitations in current large reasoning models: 1) Large reasoning models still have limited capability in generating entirely correct mathematical proofs, with some models solving less than 20% of problems and even making mistakes on fundamental ones; 2) models exhibit a diverse spectrum of reasoning failures, prominently demonstrating the lack of guarantees for the correctness and rigor intermediate reasoning steps; and 3) models show hallucination and incompleteness during the reasoning process. Our findings also reveal that directly prompting models to self-reflect on specific failure modes is insufficient to resolve the current logical dilemmas, necessitating domain knowledge and formal verification.

1 Introduction

The mathematical reasoning ability of large language models (LLMs) has experienced remarkable progress recently (Liu et al., 2025). In particular, the emergence of long chain-of-thought techniques endows models with the capacity for in-depth reasoning and self-reflection, leading to significant improvements in their performance on challenging competition-level benchmarks (DeepSeek-AI et al., 2025; OpenAI et al., 2024; Seed et al., 2025; OpenAI, 2025b; Anthropic, 2025; Yang et al., 2025; Google DeepMind, 2025).

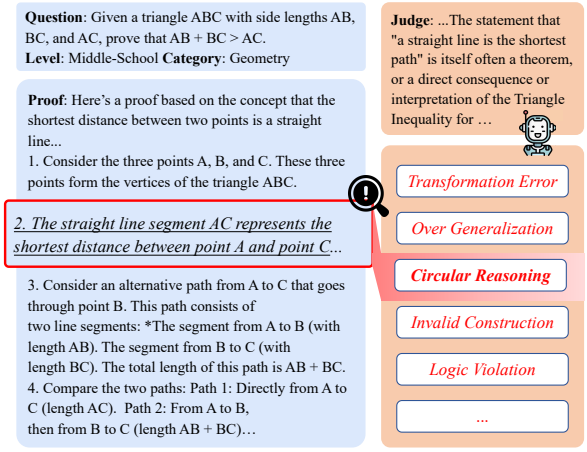


Figure 1: An example of the failure case of *Gemini-2.5-pro-preview-0506* in ***RFMDataset***. In this case, the judge LLM meticulously examined the proof and identified "*Circular Reasoning*" as the failure mode.

However, on the flip side of this success, researchers lack an understanding of how models fail in their reasoning. One significant obstacle to understanding the failure of large reasoning models in mathematics is the limited scale of benchmark datasets. Popular competition-level datasets, for example, AIME-2024 (Jia, 2025) and AIME-2025 (OpenCompass Team, 2025), consist of merely 30 problems each. Consequently, when a state-of-the-art model scores 90% accuracy (OpenAI, 2025b), it leaves only three incorrect instances for review, which is insufficient for a thorough failure analysis. A second significant factor is the evaluation method that assess only the final numerical answer, a practice common in benchmarks such as GSM8K, MATH, and AIME (Cobbe et al., 2021; Amini et al., 2019; Hendrycks et al., 2021; Jia, 2025; OpenCompass Team, 2025). While efficient, such answer-based metrics cannot verify the logical soundness of the model's reasoning process (Boye

and Moell, 2025a; Zheng et al., 2024).

Addressing the issues above, mathematical proofs provide a more effective way to uncover flaws in a model's reasoning. Unlike numerical answers that can obscure intermediate errors, proofs require explicit logical steps, where a single flawed or ambiguous inference can invalidate the entire argument. Moreover, proof-specific techniques such as constructive reasoning or proof by contradiction are often more complex than direct derivation and introduce more diverse reasoning errors. Existing benchmarks (Zheng et al., 2022; Azerbayev et al., 2023; Liu et al., 2023; He et al., 2025) primarily evaluate models' ability to generate proofs in formal languages (*e.g.*, Lean). However, unlike specialized models designed for formal proof generation(*e.g.*, AlphaGeometry (Chervonyi et al., 2025)), current reasoning models are predominantly trained and evaluated on natural language reasoning tasks (Wang et al., 2025b). A huge performance gap emerges when they are required to solve the same problems using formal language (Dekoninck et al., 2025), suggesting that the ability to generate formal proofs should not be equated with genuine reasoning ability.

Considering these limitations, we propose to utilize mathematical proofs in natural language as an ideal diagnostic tool. To implement this idea, we manually construct the ***RFMDataset*** to ***R***eveal the ***F***ailure ***M***ode of state-of-the-art reasoning models, which is composed of 200 mathematical proof problems manually selected from multiple sources, covering a diverse range of proof problems, as shown in Figure 2. We utilize an LLM-as-a-judge (Zheng et al., 2023) framework to evaluate the overall correctness of the proofs, with human annotations provided to validate the reliability of the judge-LLM. Our evaluation framework also incorporates a fine-grained error classification system with over 10 distinct failure modes, enabling the judge-LLM to precisely categorize errors within proofs into one or multiple failure types. A failure case with its failure mode analysis is shown in Figure 1.

We evaluate 10 widely used, state-of-the-art reasoning models on our dataset. We find that even with impressive metrics reported on current competition-level benchmarks(*e.g.*, AIME-2025), these models continue to struggle with proof tasks. Suprisingly, the widely used models like *DeepSeek-R1-0120* (DeepSeek-AI et al., 2025) and *Qwen3-235B* (Yang et al., 2025) exhibit an overall accuracy below 20%, even the top-performing *Gemini-2.5-pro-preview-0605* model (Google DeepMind, 2025) achieves no more than 60%. In addition, models exhibit unsatisfactory performance even on relatively basic proof problems, with eight models not achieving accuracy 60% on the lowest difficulty level. Furthermore, our fine-grained failure mode analysis reveals that models always fail to ensure the correctness and rigor of single-step reasoning, while frequently rely on non-existent premises and provide incomplete proofs. We also conduct experiments that demonstrate prompting models to self-reflect on specific failure modes does not effectively resolve the issues. Based on these findings, we highlight the fundamental limitations and propose potential solutions, hoping to contribute to the future improvement of reasoning models.

2 Related Work

Natural Language Proof Benchmarks A growing body of recent research has focused on the generation of natural language proofs. NaturalProofs (Welleck et al., 2021) leverages natural language to evaluate the correctness of LLM generated proofs, but the scope is limited to over undergraduate levels. Concurrent work OPC Dataset (Dekoninck et al., 2025) collects competition-level questions and answers from different LLMs, providing rich research materials. DeepTheorem (Zhang et al., 2025b) aims to improve models' ability to generate natural language proofs through the use of reinforcement learning. IneqMath (Sheng et al., 2025) evaluates the ability of models to prove inequalities. MathConstruct (Balunovi'c et al., 2025) evaluates models' ability to generate constructive proofs. While these valuable studies advance the field by improving and evaluating proof-generation performance, our work addresses a distinct and complementary goal, which is to systematically diagnose how reasoning fails with fine-grained taxonomy of failure modes. Unlike performance-oriented benchmarks such as the competition-focused OPC Dataset or the domain-specific IneqMath, our data collection methodology prioritizes novelty and diversity from a wide range of sources.

Analysis of Failure Modes While LLMs have achieved remarkable performance, the analysis of their reasoning failures is a growing field. In the mathematical domain, initial studies have explored error patterns (Boye and Moell, 2025b; Petrov et al., 2025). However, these efforts are often limited by

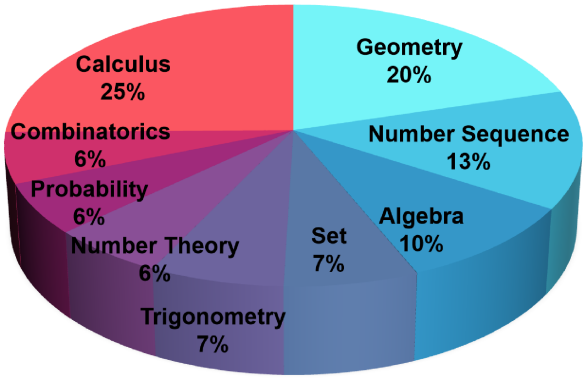


Figure 2: The knowledge distribution of ***RFMDataset***. We collect diverse questions from various domains to ensure the comprehensiveness of our evaluation.

coarse-grained taxonomies that lack detail, or are constrained by a narrow scope, such as high-school level problems or very small sample sizes. Also distinct from our work are recent valuable studies that diagnose model weaknesses by applying external modifications to existing benchmarks. For instance, CR-MATH (Li et al., 2025) introduces counterexamples to challenge the conceptual limits of models' reasoning, and MATH-Perturb (Huang et al., 2025) applies systematic perturbations to evaluate the robustness of their solutions against superficial changes. Our approach differs in three fundamental ways. First, our ***RFMDataset*** assesses reasoning on problems where the difficulty is intrinsic to the multi-step proof, rather than engineered through external tweaks. Second, our proof-centric paradigm enables a process-oriented diagnosis with a fine-grained taxonomy of logical fallacies, moving beyond the final-answer accuracy focus of robustness tests. Finally, our study is specifically designed to stress-test the latest generation of advanced models (*e.g., Gemini-2.5-Pro, GPT-o3*), whose failure modes have evolved into the subtle logical inconsistencies that our new evaluation methodology is built to detect.

3 Dataset Construction and Evaluation

In this section, we provide a detailed introduction to the characteristic and construction of our ***RFMDataset***, followed by the explanation of our evaluation method and failure mode taxonomy. The overview of ***RFMDataset*** and our evaluation method is shown in Figure 3.

3.1 RFMDataset

This subsection details the characteristic and construction of our dataset.

3.1.1 Characteristics

Our dataset contains 200 selected mathematical proof problems, a scale comparable to established benchmarks such as miniF2F-test (Zheng et al., 2022) and MathConstruct (Balunovi'c et al., 2025). The problems are stratified by knowledge level, encompassing middle school (52 problems), high school (88 problems), and undergraduate curricula (60 problems). Furthermore, the dataset covers nine distinct mathematical subjects, including but not limited to geometry, calculus, and probability, as outlined in Figure 2. Problems within each knowledge level are assigned one of four ascending difficulty levels (1 to 4) manually, with the most challenging problems approaching Olympiad difficulty.

3.1.2 Selection Criteria

As depicted in Figure 3, our dataset draws predominantly from four distinct sources: examinations, internet media, textbooks, and competitive contests. The data source analysis and comparison with other datasets are provided in the Appendix D.3. To ensure the quality of our data, we manually select the data following three primary criteria:

Diversity. This emphasis on diversity is crucial to prevent the benchmark from exhibiting an categorical bias towards specific problem types, thereby upholding its fairness and representativeness. We consider the diversity in knowledge and methodology when selecting our data. To ensure knowledge diversity, besides collecting questions from different subjects, we also make sure the diversity of problems by covering various fine-grained knowledge points, such as different related theorems. As for the methodology diversity which refers to the variety in problem-solving processes, we ensure it by reviewing each problem's solution process during problem collection. If we find a problem's solution too similar to one already collected, we retain only one of them.

Difficulty. Since our objective is to identify patterns in model reasoning failures, we should make sure even the simplest problems should be solved with careful observation and critical analysis. So we exclude problems whose solution plans could be derived intuitively and deterministically. This rigorous selection process compels models to demonstrate truly in-depth problem-solving abilities. The difficulty level of each problem is determined by

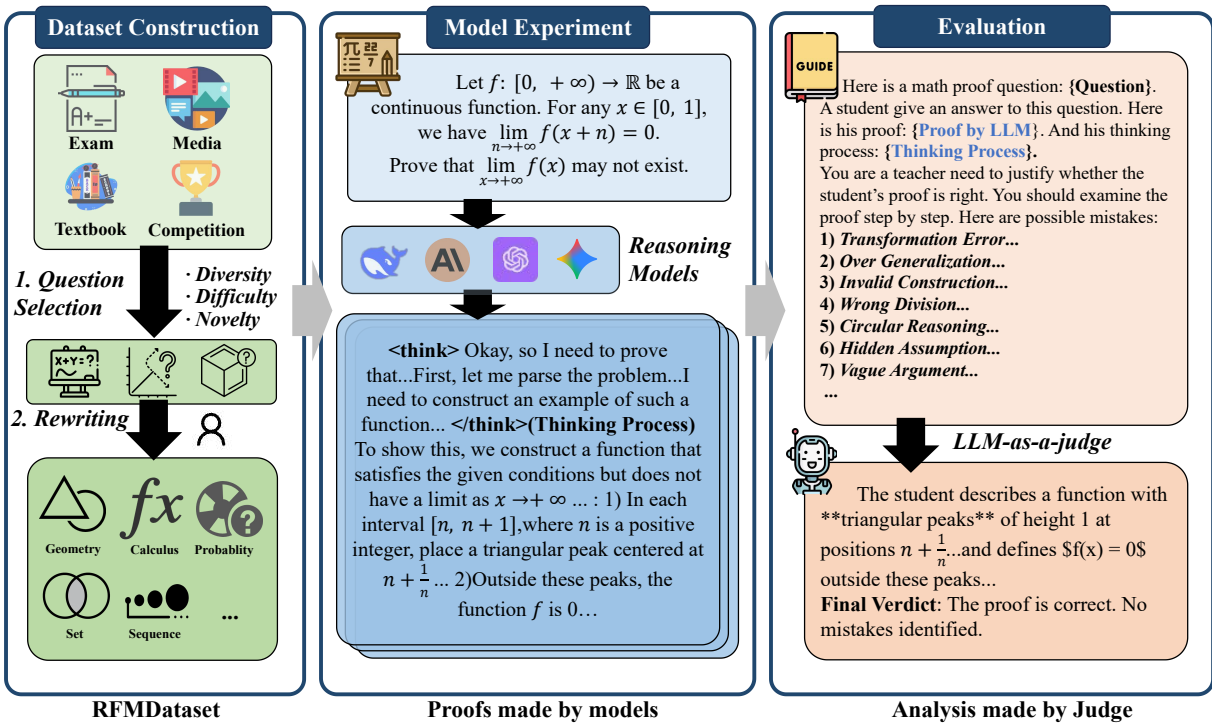


Figure 3: The construction of ***RFMDataset*** and evaluation pipeline. We first manually collect challenging and diverse mathematical proof problems. Subsequently, we use an LLM-as-a-judge approach to evaluate the performance of inference models on this data, categorizing errors into various failure modes.

human annotators through studying reference solutions to understand the reasoning complexity involved, then comparing the reasoning complexity of different problems. Additionally, we apply special handling to geometry problems. First, we reformulated the problem descriptions to allow the geometric scene to be reconstructed purely from natural language. This adjustment accommodates models that lack vision capabilities and reduces extraneous difficulty. Second, we restrict models from using certain methods, such as parametric equations, to prevent them from reducing the geometric challenge to a more straightforward algebraic calculation.

Novelty. We emphasize the novelty of the selected data, aiming for it to be distinct from commonly used mathematical benchmarks or data sources. For questions sourced from examinations (e.g., Chinese Gaokao or Kaoyan), we prioritize modified versions crafted by mathematics enthusiasts, alongside recent regional exam problems. We specifically select problems exhibiting significant divergence from typical textual materials on online media as well as some published original and innovative problems. Textbook-derived problems are strictly limited to fundamental theorems that necessitate non-trivial proof techniques. Furthermore, we avoid overused sources directly, such as the International Mathematical Olympiad (IMO), which are already prevalent in existing research benchmarks.

3.2 Evaluation

We employ two methods to evaluate the models' proofs: LLM-as-a-judge (Gu et al., 2025; Fan et al., 2025) and human evaluation.

Our LLM-as-a-judge method extends beyond holistic proof verification. We develop a fine-grained error taxonomy comprising over 10 reasoning failure modes, including *Logical Violation, Over Generalization*, and *Circular Reasoning*. This enables the precise classification of model-generated proof failures, offering a deeper understanding of their shortcomings. Table 1 presents our complete taxonomy of failure modes, providing a detailed definition and concise, illustrative examples for each category.

Our taxonomy of failure modes is not homogeneous. It contains both classical, well-established logical fallacies (*e.g., Circular Reasoning*) and patterns empirically identified from the behavior of current LLMs (*e.g., Over Generalization*). Beyond this direct diagnosis, our taxonomy serves as a

Category	Definition	Illustrative Example
Transformation Error	Recasting the target statement into a *non-equivalent* or strictly weaker one, so a "proof" no longer addresses the real goal.	To show a series $\sum a_n$ converges, one proves only $\lim_{n\to\infty} a_n = 0$; or replaces "$A \iff B$" with the easier "$A \Rightarrow B$".
Over Generalization	Drawing a universal conclusion from only a few special cases or situations.	Verifying the claim for $n = 1, 3, 5$ and then declaring it true for *all* positive integers.
Invalid Construction	Presenting an object that either cannot exist under the stated conditions or fails the required properties.	Claiming the existence of a function that is "everywhere linear yet nowhere differentiable"; or defining $f(x) = 1/x$ on all $\mathbb{R}$ without addressing $x = 0$.
Wrong Division	Partitioning into cases that miss at least one legitimate possibility or overlap each other.	When analyzing the behavior of a function, dividing cases as "always positive," "always zero," and "always negative."
Circular Reasoning	Using the desired conclusion (or an equivalent reformulation) as an explicit or hidden premise.	To prove $A \Rightarrow B$, but use premises that implicitly assumes B.
Logic Violation	A single deduction step contradicts logical or algebraic rules.	From $a < b$ and $c < d$ concluding $a - c < b - d$ (false when c, d are negative).
Hidden Assumption	Applying theorems or operations whose hypotheses have not been established or stated.	Differentiating a function known only to be continuous; interchanging limit and integral without proving uniform convergence.
Boundary Neglect	Ignoring edge cases, endpoints, or limiting situations so the proof works only "in the middle."	Declaring $f(x) = \sqrt{x}$ differentiable on $[0, 1]$ without checking $x = 0$.
Vague Argument	Relying on intuition, diagrams, or the word "obvious" rather than rigorous justification.	"The series obviously converges because the terms get smaller"; "One sees from the picture that the two segments are equal."
Incomplete Proof	Omitting an essential component in proof, or even providing an unfinished proof.	Proving sufficiency but not necessity; in induction, writing the inductive hypothesis but never showing how $P(k)$ implies $P(k + 1)$.
Others	Any error type not covered by the categories above.	—

Table 1: Classification of Typical Proof Errors. We identify 10 specific failure modes and group the rest into an "Others" category. Each failure mode is clearly defined and illustrated with an example.

practical surrogate for a proof's intrinsic value. A proof's true worth often lies in a key insight (*e.g.*, a clever construction), which should be highly credited even with minor flaws. However, this is a major challenge for an automated judge, because the judge often cannot solve the problem itself and therefore lacks the necessary insight to reliably identify these key ideas(providing a single gold proof as a reference is also not helpful, as multiple valid proof paths often exist). We therefore adopt a more robust and tractable approach, which is to verify the step-by-step logical integrity, making the absence of identifiable failures a reliable proxy for a proof's soundness.

The purpose of the human evaluation is to measure the reliability of the LLM-as-a-judge method. In the human evaluation process, we thoroughly assess the correctness of proofs by giving a binary pass or fail on the entire proof chain, then use correlation coefficients to show the consistency between human and LLM-generated labels.

4 Experiments

In this section, we detail the settings in our experiments, including the experiment setup and the dual evaluation pipeline containing LLM-as-a-judge as well as human evaluation.

4.1 Experiment Setup

Models. We experiment with three state-of-the-art open-source and seven proprietary LLMs to test the greatest models' proving capabilities. In the LLM-as-a-judge evaluation, we unifiedly use *Gemini-2.5-pro-preview-0506* (Huang and Yang, 2025) to evaluate all the models. For all models, the generation parameters were set to default values, with a *top_p* of 1.0 and a *temperature* of 1.0.

Prompts. In the main experiments, we directly provide the question to models with no extra information. The evaluation prompt is shown in Figure 9, where we demand the judge model to analyze failure modes and provide overall correctness.

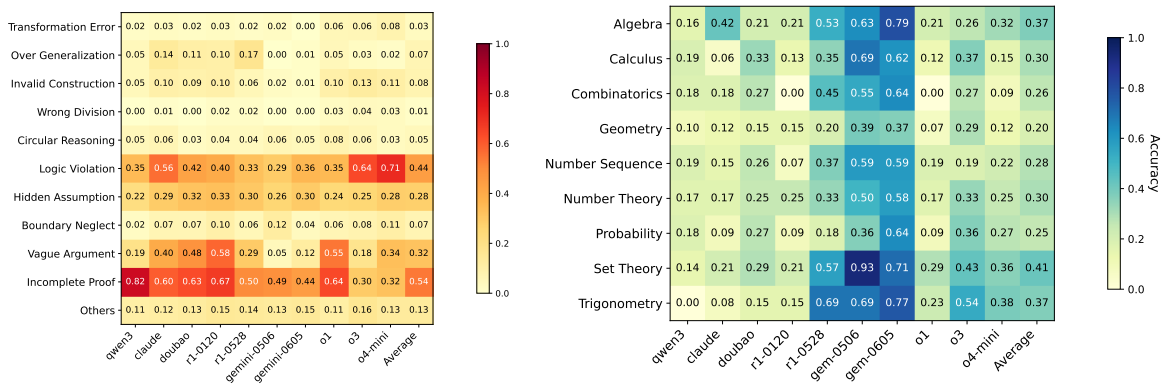


Figure 4: *(a)* The proportion of each error pattern of each model. The *Average* column summarizes the average proportion across all models for each pattern. *(b)* Accuracy of each model on every knowledge domain. The *Average* column summarizes the average accuracy across all models for each domain.

Metrics. Contrary to prior work using the pass@k metric (Ren et al., 2025; Wang et al., 2025a), we employ pass@1 accuracy. This choice is driven by our primary objective of analyzing failure modes, in which a single sampling provides an adequate number of failure cases, as well as the prohibitive cost of generating multiple responses.

4.2 Human Evaluation

To validate the reliability of our LLM-as-a-judge, we randomly sampled 240 model-generated answers (60 each from *Deepseek-R1-0120, Deepseek-R1-0528, Gemini-2.5-Pro-Preview-0506, and GPT-4o-mini-0416*) for human evaluation. The agreement between the judge's assignments and human labels was then measured using the Matthews Correlation Coefficient (MCC) (Xu et al., 2025; Matthews, 1975). As shown in Table 2, the results demonstrate high consistency between human and judge model's labels.

Models	MCC%
Deepseek-R1-0120	71.97
Deepseek-R1-0528	86.60
Gemini-2.5-pro-preview-0506	86.60
GPT-o4-mini-0416	91.09
Overall	87.61

Table 2: The MCC between the manual labels of answers from different models and the LLM labels.

The data annotation process is reported in Appendix D.2. The discussion regarding the use of other LLMs as judges(*e.g.*, *GPT-5* (OpenAI, 2025a)) is included in the Appendix E.3. The overall evaluation results from *GPT-5* and *Gemini* are highly similar, but both differ significantly from those of other models. Considering the advanced capabilities of these two models, and with *Gemini*'s assessments having been validated by human evaluation, we are confident that our reported results are reliable.

5 Experimental Findings

For the experiment, we propose the following three research questions (RQs), and analyze and discuss them based on our experimental results.

- **RQ1**: How reliably can large reasoning models generate rigorous proof? If not reliable, which domain do they perform poorly on?
- **RQ2**: Do different models exhibit comparable failure mode distributions, and what insights can be gleaned from analyzing them?
- **RQ3**: Can models show improvement when we explicitly prompt models to self-reflect on potential failure modes?

5.1 Overall Evaluation (RQ1)

To answer RQ1, we conduct an overall evaluation of the models' ability to generate completely correct proofs. The evaluation results are shown in Table 3. Additionally, we also evaluate the accuracy of different models across various knowledge domains and difficulty levels, and the results are presented in Figure 4b and Figure 5. Our key findings are presented as follows:

Models exhibit limited overall effectiveness. Table 3 presents the overall accuracy of LLM-as-a-judge evaluations, which shows that models generally face significant challenges. Surprisingly, the

Model	MS	HS	UG	Overall
Qwen3-235B-A22B	15.38	9.09	23.33	15.00
Doubao-1.5-thinking-pro	19.23	21.59	33.33	24.50
Deepseek-R1-0120	17.31	10.23	16.67	14.00
Deepseek-R1-0528	26.92	42.05	38.33	37.00
Gemini2.5-Pro-Preview-0506	51.92	53.41	73.33	59.00
Gemini2.5-Pro-Preview-0605	50.00	60.23	66.67	59.50
Claude-3.7-Sonnet-Thinking	19.23	13.64	11.67	14.50
GPT-o1	7.69	14.77	18.33	14.00
GPT-o3-0416	30.77	29.55	38.33	32.50
GPT-o4-mini-0416	15.38	22.73	23.33	21.00

Table 3: The accuracy (%) of different state-of-the-art LLMs in ***RFMDataset***. MS, HS and UG means middle-school, high-school and undergraduate level problems correspondingly. Across all models, the scores highlighted in dark, medium, and shallow cyan indicate the highest, second-highest, and third-highest performances, respectively.

accuracy of models such as *Qwen3-235B-A22B*, *Deepseek-R1-0120*, *Claude-3.7-Sonnet-Thinking*, and *GPT-o1* does not exceed 20%. Two versions of *Gemini2.5-Pro-Preview* achieve a significantly higher accuracy of nearly 60%, substantially outperforming all other models.

Models' performance vary significantly across different domains. Figure 4b presents our analysis, revealing that models consistently achieve lower accuracy rates in geometry, number sequences, combinatorics, and probability. In stark contrast, these same models show relatively higher accuracy in algebra and number theory tasks.

Models may fail on easy problems. Our analysis revealed a surprising finding: state-of-the-art models fail even on easy problems. As depicted in Figure 5, model accuracy generally declines as problem difficulty increases, which aligns with our difficulty annotations. However, it's particularly noteworthy that the majority of models fall short even at the lowest difficulty level, with all models except two *Gemini* variants, achieving less than 60% accuracy.

5.2 Failure Mode Analysis (RQ2)

As evaluating accuracy is insufficient to uncover the deeper root causes of errors in the model's reasoning process, we further provide a more fine-grained diagnostic of the underlying error causes by analyzing the failure modes. Figure 4a illustrates the distribution of failure modes across all incorrect proofs. Our main findings are as follows:

The failure mode distributions of different models are similar and not much relevant to difficulty. Different models exhibit strikingly similar failure mode distributions, predominantly concentrated in four categories: *logical violation*, *hidden assumption*, *vague argument*, and *incomplete proof*. While other failure modes occur, their proportions are relatively low. Interestingly, the dominance of these four modes remain largely unchanged when analyzing failure modes across various problem difficulty levels(shown in Appendix E.1). This suggests that despite significant advancements in reasoning capabilities, models still possess fundamental flaws in their underlying reasoning processes.

Models have significant shortcomings in the correctness and rigor of single reasoning steps. *Logical violation* and *vague argument* are commonly observed as reasons for model failures. These categories directly indicate models' widespread shortcomings in the correctness and rigor of single-step reasoning. We can observe that the *Gemini* models exhibits a significantly lower proportion of these two failure modes compared to other models, which may contribute to their superior overall performance.

Models struggle with severe hallucination and incompleteness issues in reasoning process. According to Figure 4a, models also demonstrate prominent issues in *hidden assumption* and *incomplete proof*. *Hidden assumption* can be treated as a form of hallucination, since it refers to the use of unprovided conditions during problem-solving. An *Incomplete Proof* signifies that the model provides an insufficient demonstration, failing to independently determine when a proof is complete. These two aspects together reveal a fundamental lack of understanding on both the initial proposition and what should be done to prove the proposition.

5.3 Mode-specific self-relection (RQ3)

To improve performance, we prompt models to self-reflect on their initial proofs. Our rationale is

Model	MS	HS	UG	Overall	Best
Doubao-1.5-thinking-pro	19.23	21.59	33.33	24.50	**31.50**
Doubao-1.5-thinking-pro +reflection_l	28.85	19.32	26.67	24.00	
Doubao-1.5-thinking-pro +reflection_vi	28.85	21.59	36.67	28.00	
Doubao-1.5-thinking-pro +reflection_lvi	36.54	25.00	28.33	29.00	
GPT-o4-mini	15.38	22.73	23.33	21.00	**27.00**
GPT-o4-mini +reflection_l	19.23	22.73	31.67	24.50	
GPT-o4-mini +reflection_vi	17.31	22.73	20.00	20.50	
GPT-o4-mini +reflection_lvi	21.15	27.27	20.00	23.50	
Deepseek-R1-0528	26.92	42.05	38.33	37.00	**42.00**
Deepseek-R1-0528 +reflection_l	32.69	32.95	43.33	36.00	
Deepseek-R1-0528 +reflection_vi	19.23	35.23	40.00	32.50	
Deepseek-R1-0528 +reflection_lvi	34.62	31.82	48.33	37.50	

Table 4: The accuracy (%) of the selected large reasoning models on the ***RFMDataset***. For each subset of the dataset, the highest accuracy achieved across different baselines for the same model is highlighted in blue, and the color is darker for better performance across different models. The *Best* column reports the overall of these highlighted scores, providing an overall measure of the model's best performance across all subsets.

that many errors stem not from fundamental reasoning incapacity, but from superficial flaws like oversights or presentational gaps that are amenable to self-correction. For instance, a proof flagged as a *Vague Argument* might reflect a momentary lapse in rigor during a specific generation, rather than a fundamental inability to reason precisely. Similarly, a proof marked as *Incomplete Proof* could be logically sound but rely on an obscure or non-trivial theorem, which the judge flags as a gap because it was not explicitly stated or proven.

In response, we design three additional baseline prompts: 1) Target solving *logical violation*: the prompt asks the model to perform step-by-step self reflection after generating answers (*reflection_l*). 2) Target solving *vague argument* and *incomplete proof*: based on observed phenomena, the prompt instructs the model to reason rigorously and prove any non-trivial theorems it employs (*reflection_vi*). 3) Combination of the two kinds of reflection above (*reflection_lvi*). Detailed prompts are reported in Appendix A.

We conduct our experiments using *Doubao-1.5-thinking-pro*, *o4-mini*, and *Deepseek-R1-0528*. The results are shown in Table 4. The *Best* column reports a synthetic score, calculated as the weighted average of the highest score achieved within each of the MS, HS, and UG categories respectively, across all tested reflection strategies for a given model. This aggregated score synthesizes the model's upper-bound potential, showing its performance if one could perfectly select the optimal reflection strategy for each category.

Our experimental results demonstrate that prompts designed to address specific failure modes can enhance the model's performance in at least one of the MS, HS, or UG problem categories, and even lead to an improvement in overall performance. Furthermore, the *Best* results suggest that these models possess untapped potential for improvement when guided by failure mode prompting. However, these gains are modest when compared to the substantial number of problems the model still fails to solve. We discuss possible solutions for improving the future models in Appendix F.

6 Conclusion

In this paper, we introduce ***RFMDataset***, a novel and high-quality dataset comprised of 200 carefully selected mathematical proof problems designed to revealing the reasoning failure modes of large reasoning models. We utilize LLM-as-a-judge to evaluate the efficiency of advanced models on our dataset and discover the failure mode distributions. Our findings demonstrate that models struggle on generating entirely correct proofs. Specifically, they frequently fall into single-step reasoning errors, exhibit vague argumentation, utilize non-existent premises, and produce incomplete proofs. Furthermore, our experiments reveal that explicitly prompting models to self-reflect on these specific failure modes is insufficient to resolve these issues, indicating fundamental drawbacks in current large reasoning models. For future work, we propose to take advantage of formally-verifiable environment and high-quality domain-specific data to endow models the mathematician's way of thinking.

Limitations

While ***RFMDataset*** effectively uncovers reasoning deficiencies, certain limitations affect broader interpretation. These choices reflect a practical balance between resource constraints, like cost and time, and the depth of analysis sought. The dataset's selected size (200 problems) means the generalizability of identified failure modes to broader mathematical contexts may need further investigation. Our focus on natural language proofs, while practical for current models, means conclusions don't directly extend to formal mathematical systems.

Ethics Statements

Offensive Content Elimination. Our benchmark curation pipeline does not involve generating content using LLMs. Instead, all content is manually collected and carefully validated, ensuring that the dataset remains free from offensive material. As such, we are confident that the dataset is safe and does not pose any negative societal impact.

Licenses. We will release our code under the MIT license to allow unrestricted research use. The ***RFMDataset*** will be distributed under a Creative Commons (CC) license, likewise offering free access to the academic community. We take full responsibility for any potential rights violations or issues related to licensing. All licenses associated with our resources support access for research purposes and comply with their respective terms of use.

Annotations. The data annotation and verification process was conducted entirely by the six authors of this paper. The team is composed of computer science researchers (one Ph.D., two Ph.D. students, one M.Phil. student, and two senior undergraduates). A crucial qualification for this task was the team's shared, significant background in competitive mathematics, which provided the necessary domain expertise to accurately assess the logical soundness of proofs across various difficulty levels. As the annotation was an integral part of our research effort, considerations of external recruitment and compensation do not apply, in line with standard academic practice.

Our annotation process was guided by a systematic protocol. The instructions for annotation were the detailed evaluation framework and failure mode rubric presented in this paper, which were developed iteratively by the team itself. To ensure the highest quality for our ground-truth labels, we employed a collaborative review and consensus protocol rather than calculating a traditional Inter-Annotator Agreement (IAA). While IAA is vital for subjective tasks, verifying mathematical correctness is an objective process where disagreements are factual disputes to be resolved. In our protocol, each proof was annotated by a primary author. Any complex or ambiguous cases were then flagged for a second review, with the final label and error classification determined through discussion and consensus among the involved authors.

References

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 2357–2367. Association for Computational Linguistics.

Anthropic. 2025. Claude 3.7 sonnet system card. Technical report, Anthropic.

Zhangir Azerbayev, Bartosz Piotrowski, Hailey Schoelkopf, Edward W. Ayers, Dragomir Radev, and Jeremy Avigad. 2023. Proofnet: Autoformalizing and formally proving undergraduate-level mathematics. *CoRR*, abs/2302.12433.

Mislav Balunovi'c, Jasper Dekoninck, Nikola Jovanovi'c, Ivo Petrov, and Martin T. Vechev. 2025. Mathconstruct: Challenging llm reasoning with constructive proofs. *ArXiv*, abs/2502.10197.

Johan Boye and Birger Moell. 2025a. Large language models and mathematical reasoning failures. *Preprint*, arXiv:2502.11574.

Johan Boye and Birger Moell. 2025b. Large language models and mathematical reasoning failures. *Preprint*, arXiv:2502.11574.

Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. 2025. Gold-medalist performance in solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman.

2021. Training verifiers to solve math word problems. *CoRR*, abs/2110.14168.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 181 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

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. 2025. The open proof corpus: A large-scale study of llm-generated mathematical proofs. *ArXiv*, abs/2506.21621.

Zhiyuan Fan, Weinong Wang, Xing Wu, and Debing Zhang. 2025. Sedareval: Automated evaluation using self-adaptive rubrics. *arXiv preprint arXiv:2501.15595*.

Google DeepMind. 2025. Gemini 2.5 pro overview. https://deepmind.google/technologies/gemini/pro/.

Timothy Gowers. Why are LLMs not better at finding proofs.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2025. A survey on llm-as-a-judge. *Preprint*, arXiv:2411.15594.

Zhitao He, Zongwei Lyu, Dazhong Chen, Dadi Guo, and Yi R. Fung. 2025. Matp-bench: Can mllm be a good automated theorem prover for multimodal problems? *Preprint*, arXiv:2506.06034.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Jiewen Hu, Thomas (Hanwen) Zhu, and Sean Welleck. 2024. minictx: Neural theorem proving with (long-)contexts. *ArXiv*, abs/2408.03350.

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. 2025. Math-perturb: Benchmarking llms' math reasoning abilities against hard perturbations. *ArXiv*, abs/2502.06453.

Yichen Huang and Lin F. Yang. 2025. Gemini 2.5 pro capable of winning gold at imo 2025.

Maxwell Jia. 2025. Aime 2024 dataset. https://huggingface.co/datasets/Maxwell-Jia/AIME_2024. American Invitational Mathematics Examination 2024, 30 items.

Yuhua Jiang, Yuwen Xiong, Yufeng Yuan, Chao Xin, Wenyuan Xu, Yu Yue, Qianchuan Zhao, and Lin Yan. 2025. Pag: Multi-turn reinforced llm self-correction with policy as generative verifier. *arXiv preprint arXiv:2506.10406*.

Yinghui Li, Jiayi Kuang, Haojing Huang, Zhikun Xu, Xinnian Liang, Yi Yu, Wenlian Lu, Yangning Li, Xiaoyu Tan, Chao Qu, Ying Shen, Hai-Tao Zheng, and Philip S. Yu. 2025. One example shown, many concepts known! counterexample-driven conceptual reasoning in mathematical llms. *ArXiv*, abs/2502.10454.

Zhenwen Liang, Linfeng Song, Yang Li, Tao Yang, Feng Zhang, Haitao Mi, and Dong Yu. 2025. Towards solving more challenging imo problems via decoupled reasoning and proving. *arXiv preprint arXiv:2507.06804*.

Chengwu Liu, Jianhao Shen, Huajian Xin, Zhengying Liu, Ye Yuan, Haiming Wang, Wei Ju, Chuanyang Zheng, Yichun Yin, Lin Li, Ming Zhang, and Qun Liu. 2023. FIMO: A challenge formal dataset for automated theorem proving. *CoRR*, abs/2309.04295.

Hanmeng Liu, Zhizhang Fu, Mengru Ding, Ruoxi Ning, Chaoli Zhang, Xiaozhang Liu, and Yue Zhang. 2025. Logical reasoning in large language models: A survey. *Preprint*, arXiv:2502.09100.

B W Matthews. 1975. Matthews, b.: Comparison of the predicted and observed secondary structure of t4 phage lysozyme. biochimica et biophysica acta (bba)-protein structure 405(2), 442-451. *Biochimica et Biophysica Acta*, 405(2):442–451.

Naoto Onda, Kazumi Kasaura, Yuta Oriike, Masaya Taniguchi, Akiyoshi Sannai, and Sho Sonoda. 2025. Leanconjecturer: Automatic generation of mathematical conjectures for theorem proving. *arXiv preprint arXiv:2506.22005*.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, and 244 others. 2024. Openai o1 system card. *Preprint*, arXiv:2412.16720.

OpenAI. 2025a. Gpt-5 system card. Accessed: 2025-10-07.

OpenAI. 2025b. Openai o3 and o4-mini system card. Technical report, OpenAI.

OpenCompass Team. 2025. Aime 2025 dataset. https://huggingface.co/datasets/opencompass/AIME2025. American Invitational Mathematics Examination 2025, 30 items.

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

Z. Z. Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, Z. F. Wu, Zhibin Gou, Shirong Ma, Hongxuan Tang, Yuxuan Liu, Wenjun Gao, Daya Guo, and Chong Ruan. 2025. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *Preprint*, arXiv:2504.21801.

ByteDance Seed, Yufeng Yuan, Yu Yue, Mingxuan Wang, Xiaochen Zuo, Jiaze Chen, Lin Yan, Wenyuan Xu, Chi Zhang, Xin Liu, and 1 others. 2025. Seed-thinking-v1. 5: Advancing superb reasoning models with reinforcement learning. *arXiv preprint arXiv:2504.13914*.

Jiayi Sheng, Luna Lyu, Jikai Jin, Tony Xia, Alex Gu, James Zou, and Pan Lu. 2025. Solving inequality proofs with large language models. *ArXiv*, abs/2506.07927.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, Chuning Tang, Congcong Wang, Dehao Zhang, Enming Yuan, Enzhe Lu, Fengxiang Tang, Flood Sung, Guangda Wei, Guokun Lai, and 75 others. 2025. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Haiming Wang, Mert Unsal, Xiaohan Lin, Mantas Baksys, Junqi Liu, Marco Dos Santos, Flood Sung, Marina Vinyes, Zhenzhe Ying, Zekai Zhu, Jianqiao Lu, Hugues de Saxc'e, Bolton Bailey, Chendong Song, Chenjun Xiao, Dehao Zhang, Ebony Zhang, Frederick Pu, Han Zhu, and 21 others. 2025a. Kimina-prover preview: Towards large formal reasoning models with reinforcement learning.

Ruida Wang, Yuxin Li, Yi R. Fung, and Tong Zhang. 2025b. Let's reason formally: Natural-formal hybrid reasoning enhances llm's math capability. *Preprint*, arXiv:2505.23703.

Ruida Wang, Rui Pan, Yuxin Li, Jipeng Zhang, Yizhen Jia, Shizhe Diao, Renjie Pi, Junjie Hu, and Tong Zhang. 2025c. Ma-lot: Model-collaboration lean-based long chain-of-thought reasoning enhances formal theorem proving. *arXiv preprint arXiv:2503.03205*.

Yumeng Wang, Zhiyuan Fan, Jiayu Liu, and Yi R. Fung. 2025d. Diversity-enhanced reasoning for subjective questions.

Sean Welleck, Jiacheng Liu, Ronan Le Bras, Hanna Hajishirzi, Yejin Choi, and Kyunghyun Cho. 2021. Naturalproofs: Mathematical theorem proving in natural language. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual*.

Wei Xiong, Chengshuai Shi, Jiaming Shen, Aviv Rosenberg, Zhen Qin, Daniele Calandriello, Misha Khalman, Rishabh Joshi, Bilal Piot, Mohammad Saleh, and 1 others. 2024. Building math agents with multi-turn iterative preference learning. *arXiv preprint arXiv:2409.02392*.

Zhaopan Xu, Pengfei Zhou, Jiaxin Ai, Wangbo Zhao, Kai Wang, Xiaojiang Peng, Wenqi Shao, Hongxun Yao, and Kaipeng Zhang. 2025. Mpbench: A comprehensive multimodal reasoning benchmark for process errors identification. *Preprint*, arXiv:2503.12505.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report.

Jie Zhang, Cezara Petrui, Kristina Nikolić, and Florian Tramèr. 2025a. Realmath: A continuous benchmark for evaluating language models on research-level mathematics. *arXiv preprint arXiv:2505.12575*.

Ziyin Zhang, Jiahao Xu, Zhiwei He, Tian Liang, Qiuzhi Liu, Yansi Li, Linfeng Song, Zhen-Pu Liang, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025b. Deeptheorem: Advancing llm reasoning for theorem proving through natural language and reinforcement learning. *ArXiv*, abs/2505.23754.

Chujie Zheng, Zhenru Zhang, Beichen Zhang, Runji Lin, Keming Lu, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2024. Processbench: Identifying process errors in mathematical reasoning. *arXiv preprint arXiv:2412.06559*.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2022. minif2f: a cross-system benchmark for formal olympiad-level mathematics. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Preprint*, arXiv:2306.05685.

A Experiment Details

Models. The models we test include: *Deepseek-R1-0120*, *Deepseek-R1-0528* (DeepSeek-AI et al., 2025), *Qwen3-235B-A22B* (Yang et al., 2025), *Doubao-1.5-thinking-pro-0415* (Seed et al., 2025), *Claude-3.7-Sonnet-Thinking* (Anthropic, 2025), *Gemini-2.5-pro-preview-0506*, *Gemini-2.5-pro-preview-0605* (Google DeepMind, 2025), *GPT-o1* (OpenAI et al., 2024), *GPT-o3-0416*, and *GPT-o4-mini-0416* (OpenAI, 2025b).

Prompts. For prompting the model to generate proofs, We use four kinds of prompts in all the experiments: *Vanilla*, *reflection_l*, *reflection_vi* and *reflection_lvi*, which is reported in Table 9. In the evaluation stage, we unifiedly use one prompt which is presented in Figure 9. Prompt permutations were not used in our baselines, as our focus on failure modes obviates the need for parallel sampling.

B Examples of Question

This appendix presents a curated selection of example questions from the ***RFMDataset***, designed to evaluate the mathematical reasoning capabilities of large language models (LLMs) in constructing rigorous proofs. The questions are drawn from the dataset described in the main document and are organized by educational level that includes four problems of increasing difficulty (Levels 1 to 4). Examples of middle school, high school and university level questions are shown in Figure 11, 12 and 13.

C Failure case analysis

C.1 Failure case analysis of judge model

In our observations, the judge model is mostly accurate in its assessments. However, it tends to be overly strict, often leading to discrepancies with human annotations. In Figure 14, we provide a case where the judge model is excessively rigid, insisting that the proof must strictly rely on the given conditions in the problem and proceed step-by-step from those premises—thus labeling it as an *Incomplete Proof*. However, the problem itself does not restrict the scope of knowledge, meaning any reasonable theorem not explicitly mentioned could still be validly used. Based on this, we conclude that this is a misjudgment by the judge model.

Failure case type	Example
Transformation Error	Figure 15
Over Generalization	Figure 16
Invalid Construction	Figure 17, 18
Wrong Division	Figure 19, 20
Circular reasoning	Figure 21, 22
Logic Violation	Figure 15, 23
Hidden Assumption	Figure 24, 25
Boundary Neglect	Figure 26, 27
Vague Argument	Figure 17, 28
Incomplete proof	Figure 19, 29
Others	Figure 30

Table 5: A total of eleven failure modes are included our evaluation pipeline. The table serves as a guide to the concrete examples of corresponding failure modes.

C.2 Failure case analysis of proof model

In this appendix, we present detailed failure cases from our evaluation of large language models on mathematical proof problems in the ***RFMDataset***. Each entry includes the original problem statement, specifies the model that was used, identifies the type of failure mode, and provides an analysis of the resulting error. The red italicized text indicates the erroneous parts in the proof, while the blue text in brackets provides explanations for the errors and their failure mode. Table 5 serves as a guide to examples of the failure case studies.

D Data Processing

D.1 Data collection

Upon identifying suitable problems, we preserve them via screenshots. We then leverage Kimi (Team et al., 2025) to extract and convert the textual content into LaTeX format, while manually annotating each problem with its respective difficulty level and knowledge domain tags. Note that for geometry problems that require diagrams, we reformulate the textual descriptions to ensure the scene can be fully reconstructed from natural language alone. This dual approach ensures that we rigorously test the models' ability to reason purely through linguistic representations.

It's particularly noteworthy that models tend to reduce the inherent difficulty of geometry problems by establishing coordinate systems or parametric equations, which circumvents our intention to test their geometric reasoning abilities. To counteract this and preserve the intended difficulty level, we explicitly instruct the models not to employ such methods in geometry problems, requiring them to use only the provided knowledge.

D.2 Data annotation

To facilitate human evaluation and establish a ground truth for assessing model performance, a human annotation process was conducted.

For each sampled instance, human annotators were presented with the original mathematical question and the corresponding model-generated proof. A single proof was structured with a </think> tag separating the internal reasoning from the final answer. To ensure consistent evaluation, particularly as certain models do not expose their reasoning chains through API services, judgments concerning correctness and the classification of failure modes were exclusively derived from the content after the </think> tag, which constitutes the final answer or output of the model.

D.3 Data Source

We analyze the proportions of the four kinds of data sources in our dataset: Textbook, Exam, Internet Media, and Competition, as shown in Table 6. To highlight the distinct data sources of our dataset compared to other proof datasets, we present a comparison of the data sources for MiniF2F (Zheng et al., 2022), MiniCTX (Hu et al., 2024), and ProofNet(Azerbayev et al., 2023) against ours, as shown in Table 7.

Source	MS	HS	UG	Overall
Textbook	20.00	0.00	16.67	9.90
Exam	24.00	50.00	15.00	32.67
Internet Media	34.00	45.56	55.00	45.05
Competition	26.00	2.22	13.33	11.39

Table 6: The distribution of data sources in ***RFMDataset***. All numbers are percentages (%).

Dataset	Data Sources
MiniF2F	AMC, AIME, IMO, MATH dataset, textbook
MiniCTX	Lean 4 Zulip, Books: Mathematics in Lean, The Mechanics of Proof
ProofNet	Undergraduate Textbooks: Elementary Real Analysis, Introduction to Abstract Algebra, etc.
RFMDataset	Textbook, Exam, Internet Media, Competition

Table 7: Comparison of Data Sources Across Different Datasets.

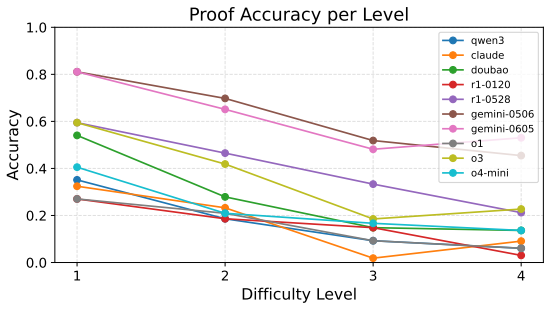


Figure 5: Accuracy of different models on each difficulty level.

E More analysis

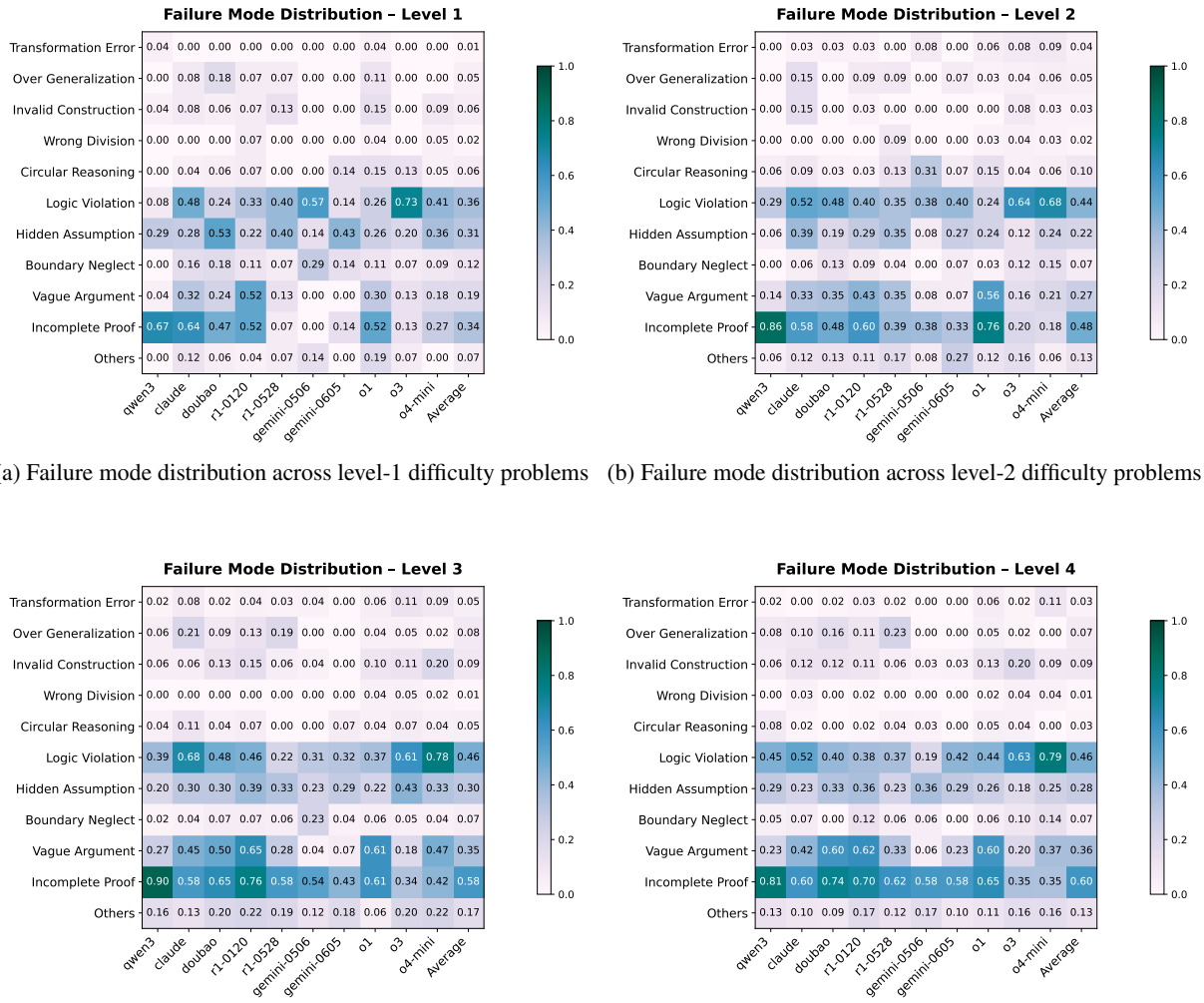
E.1 Is failure mode relevant to problem difficulty?

It is natural for models' accuracy to decline as problem difficulty increases, as illustrated in the Figure 5. However, how the distribution of failure modes changes with difficulty remains unknown. Figure 6 illustrates the distribution of failure modes across questions of varying difficulty. It's evident that, across all four difficulty levels, model failures are predominantly concentrated in four patterns: *logical violation*, *vague argument*, *hidden assumption*, and *incomplete proof*. Notably, the proportions of *logical violation*, *vague argument*, and *incomplete proof* show increasing trends with difficulty, while *hidden assumption* demonstrates a slight downward trend.

E.2 Does a longer chain-of-thought lead to better problem-solving performance?

Figure 8 presents an evaluation of four models (*Claude-3.7-Sonnet-Thinking, Doubao-1.5-thinking-pro, Qwen3-235B-A22B*, and *Deepseek-R1-0120*) analyzing their accuracy and average token counts across various domains with publicly available chain-of-thought data. Our findings show **no clear correlation between reasoning chain length and accuracy** across domains. In fact, longer reasoning chains often lead to lower accuracy.

Given that these models should theoretically grasp basic mathematical theorems, we hypothesize this discrepancy arises from two factors: varying levels of methodological mastery across different domains, and error accumulation during the reasoning process. More fundamentally, these models exhibit a limited capacity for first-principles exploration using basic mathematical theorems, per-



(a) Failure mode distribution across level-1 difficulty problems (b) Failure mode distribution across level-2 difficulty problems

(c) Failure mode distribution across level-3 difficulty problems (d) Failure mode distribution across level-4 difficulty problems

Figure 6: Distribution of failure modes across questions of four difficulty levels. It is evident that the failure mode pattern of models is similar.

forming significantly worse than mathematically proficient humans. As Timothy Gowers [1] recently highlighted(Gowers), empowering models to truly think like human mathematicians remains an unsolved challenge.

E.3 How do other LLM judges perform?

In our primary experiments, we utilized the strong *Gemini-2.5-pro-preview-0506* model as the judge to evaluate the overall correctness of proofs and the distribution of failure modes. We then demonstrate the consistency between Gemini's generated labels and human labels using correlation coefficients. However, researchers may wonder about the performance of other less powerful models when acting as judges. Therefore, we include three additional judges for supplementary experiments: *DeepSeek-r1-0528*, *Doubao-1.6-thinking-pro*, and notably, *GPT-5*, which serves as an evaluator of comparable capability to our primary judge, *Gemini-2.5-pro-preview-0506*.

The comparison of evaluation results from different judges, presented in Table 8, reveals significant disparities but also a crucial pattern of consistency. Notably, the assessments from our supplementary peer judge, *GPT-5*, show a high degree of agreement with our primary judge, *Gemini-2.5-pro-preview-0506*. For instance, when evaluating the *Gemini-2.5-pro-preview-0506* and *Gemini-2.5-pro-preview-0605* models, the overall scores assigned by the Gemini judge (59.00 and 59.50) are very close to those from GPT-5 (55.00 and 57.50, respectively). This strong consistency between two state-of-the-art models validates our choice of

[1] Fields Medalist (1998), Professor at the University of Cambridge and Collège de France.

Gemini-2.5-pro-preview-0506 as a reliable primary evaluator.

In sharp contrast, the other judges are significantly more lenient and show a large gap compared to *Gemini-2.5-Pro* and *GPT-5*. Specifically, *Doubao-1.6-thinking-pro* consistently yields higher scores than *DeepSeek-R1-0528*, and both are far more generous than *Gemini* and *GPT-5*. Despite these differences in absolute scores, it is important to note that the relative performance ranking of the evaluated models remains largely consistent across all judges.

Given *Gemini-2.5-pro-preview-0506*'s superior performance (Huang and Yang, 2025) and its consistency with human evaluations, we consider the evaluation results from our main experiments to be reliable. We attribute the inconsistency among Judge LLMs to inherent differences in their ability to evaluate mathematical propositions and *Gemini-2.5-pro-preview-0506*'s stricter judging criteria. This discrepancy not only underscores the critical importance of robust proof evaluation but also reveals significant performance gaps among current models as evaluators, marking this as a crucial area for future investigation.

Figures 7a and 7b illustrate the failure mode and per-domain accuracy distributions under the *GPT-5* judge, respectively. Crucially, both distributions are highly consistent with those from our primary *Gemini* judge, which reinforces the credibility and persuasiveness of our fine-grained analysis.

F Possible Solutions

Based on the analysis of failure modes, we discuss possible solutions to improve future models. We focus on two key aspects aiming at making the model think like a mathematician: first, how to enhance the model's performance in weak areas(solutions 1), and second, how to enhance the reliability and verifiability of the models' reasoning processes (solutions 2).

Solution 1: Training on high-quality domain-specific data. Our experimental results reveal significant variations in model performance across different mathematical domains. Taking geometry problems as an example, as illustrated in Figures 20 and 21, models still exhibit severe errors in proving even the two most fundamental properties and theorems within this field. This discrepancy strongly suggests weakness in domain-specific knowledge. Furthermore, mathematical problems from different domains may necessitate distinct techniques and methodologies. Without training on data related to these specific methodologies, models may struggle to solve problems, even if they possess long chain-of-thought exploration capabilities (Wang et al., 2025d). Consequently, a direct solution may involve training on high-quality data targeted at weak domains (Onda et al., 2025).

Solution 2: Agentic step-level interaction with formally verifiable environment. Our analysis of failure modes indicates that precise single-step reasoning remains a significant challenge for current models. A crucial reason for this lies in the unreliable and often untimely nature of the models' self-generated reflections. Therefore, we propose addressing this by enabling agentic, multi-turn interactions (Huang and Yang, 2025; Wang et al., 2025c,b; Jiang et al., 2025; Xiong et al., 2024; Liang et al., 2025) with formally verifiable environments, such as Lean. When a model needs to validate the correctness of its claims, it could translate its reasoning into Lean code and request a response from the environment. This external feedback mechanism offers far greater precision than internal self-reflection. Furthermore, as models are increasingly tasked with solving research-level mathematical problems (Zhang et al., 2025a), a single reasoning pass will be insufficient. This necessitates the integration of memory mechanisms from the agent domain to store current progress (e.g., already proven lemmas). Training for such multi-turn processes could be achieved through the design of multi-turn reinforcement learning algorithms, and evaluation should also be conducted through multi-turn generation in an interactive environment.

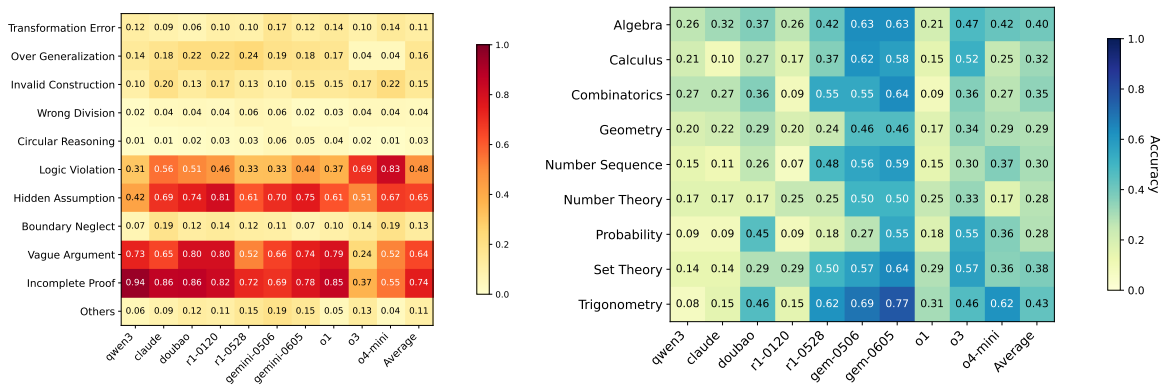
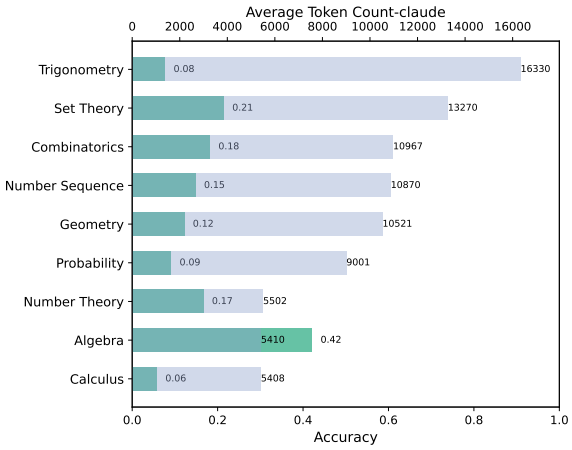


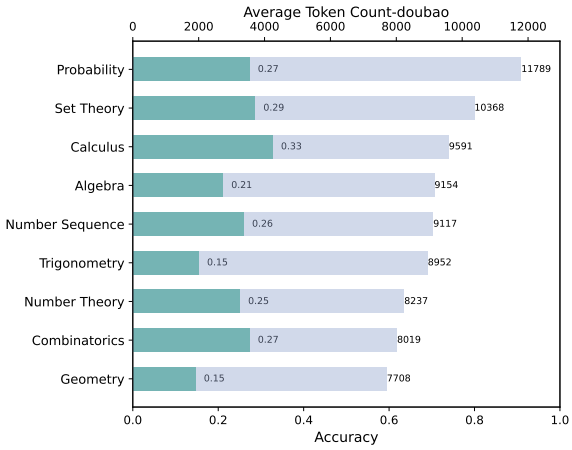
Figure 7: *(a)* The proportion of each error pattern of each model. The *Average* column summarizes the average proportion across all models for each pattern. *(b)* Accuracy of each model on every knowledge domain. The *Average* column summarizes the average accuracy across all models for each domain.

Model	MS	HS	UG	Overall	Judge Model
Qwen3-235B-A22B	15.38	9.09	23.33	15.00	*Gemini-2.5-pro-preview-0506*
Qwen3-235B-A22B	17.31	18.18	33.33	22.50	*Deepseek-R1-0528*
Qwen3-235B-A22B	23.08	15.91	28.33	21.50	*Doubao-1.6-thinking-pro*
Qwen3-235B-A22B	25.00	12.50	21.67	18.50	*GPT-5*
Doubao-1.5-thinking-pro	19.23	21.59	33.33	24.50	*Gemini-2.5-pro-preview-0506*
Doubao-1.5-thinking-pro	38.46	35.23	38.33	37.00	*Deepseek-R1-0528*
Doubao-1.5-thinking-pro	48.08	50.00	46.67	48.50	*Doubao-1.6-thinking-pro*
Doubao-1.5-thinking-pro	32.69	26.14	35.00	30.50	*GPT-5*
Deepseek-R1-0120	17.31	10.23	16.67	14.00	*Gemini-2.5-pro-preview-0506*
Deepseek-R1-0120	25.00	29.55	31.67	29.00	*Deepseek-R1-0528*
Deepseek-R1-0120	36.54	25.00	43.33	33.50	*Doubao-1.6-thinking-pro*
Deepseek-R1-0120	25.00	10.23	21.67	17.50	*GPT-5*
Deepseek-R1-0528	26.92	42.05	38.33	37.00	*Gemini-2.5-pro-preview-0506*
Deepseek-R1-0528	48.08	60.23	58.33	56.50	*Deepseek-r1-0528*
Deepseek-R1-0528	51.92	69.32	71.67	65.50	*Doubao-1.6-thinking-pro*
Deepseek-R1-0528	28.85	39.77	43.33	38.00	*GPT-5*
Claude-3.7-Sonnet-Thinking	19.23	13.64	11.67	14.50	*Gemini-2.5-pro-preview-0506*
Claude-3.7-Sonnet-Thinking	26.92	27.27	16.67	24.00	*Deepseek-R1-0528*
Claude-3.7-Sonnet-Thinking	30.77	34.09	26.67	31.00	*Doubao-1.6-thinking-pro*
Claude-3.7-Sonnet-Thinking	25.00	13.64	13.33	16.50	*GPT-5*
Gemini-2.5-pro-preview-0506	51.92	53.41	73.33	59.00	*Gemini-2.5-pro-preview-0506*
Gemini-2.5-pro-preview-0506	63.46	67.05	83.33	71.00	*Deepseek-R1-0528*
Gemini-2.5-pro-preview-0506	76.92	72.72	90.00	79.00	*Doubao-1.6-thinking-pro*
Gemini-2.5-pro-preview-0506	55.77	48.86	63.33	55.00	*GPT-5*
Gemini-2.5-pro-preview-0605	50.00	60.23	66.67	59.50	*Gemini-2.5-pro-preview-0506*
Gemini-2.5-pro-preview-0605	65.38	71.59	78.33	72.00	*Deepseek-R1-0528*
Gemini-2.5-pro-preview-0605	78.85	78.41	83.33	80.00	*Doubao-1.6-thinking-pro*
Gemini-2.5-pro-preview-0605	55.77	54.55	63.33	57.50	*GPT-5*
GPT-o1	7.69	14.77	18.33	14.00	*Gemini-2.5-pro-preview-0506*
GPT-o1	19.23	34.09	36.67	31.00	*Deepseek-R1-0528*
GPT-o1	15.38	46.59	45.00	38.00	*Doubao-1.6-thinking-pro*
GPT-o1	17.31	17.05	21.67	18.50	*GPT-5*
GPT-o3-0416	30.77	29.55	38.33	32.50	*Gemini-2.5-pro-preview-0506*
GPT-o3-0416	46.15	44.32	50.00	46.50	*Deepseek-R1-0528*
GPT-o3-0416	53.85	61.36	75.00	63.50	*Doubao-1.6-thinking-pro*
GPT-o3-0416	40.38	36.36	55.00	43.00	*GPT-5*
GPT-o4-mini-0416	15.38	22.73	23.33	21.00	*Gemini-2.5-pro-preview-0506*
GPT-o4-mini-0416	26.92	39.77	30.00	33.50	*Deepseek-R1-0528*
GPT-o4-mini-0416	34.62	53.41	41.67	45.00	*Doubao-1.6-thinking-pro*
GPT-o4-mini-0416	28.85	37.50	28.33	32.50	*GPT-5*

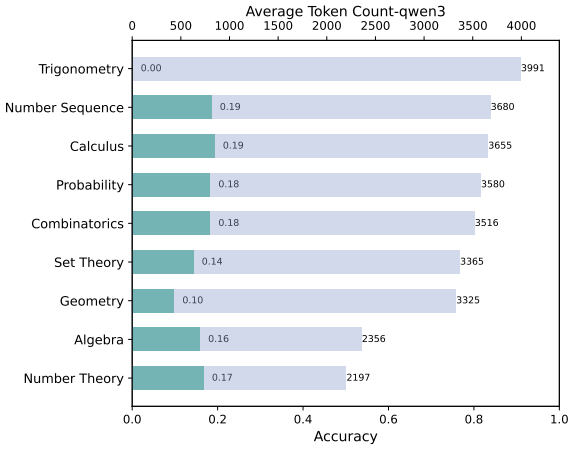
Table 8: The evaluation discrepancies among different judge models are notable. We observe that the correctness rates evaluated by *Doubao-1.6-thinking-pro* are consistently higher than those from *Deepseek-R1-0528*, which in turn are higher than *Gemini-2.5-pro-preview-0506*'s. Evidently, there are substantial differences in the accuracy provided by these various judge models.



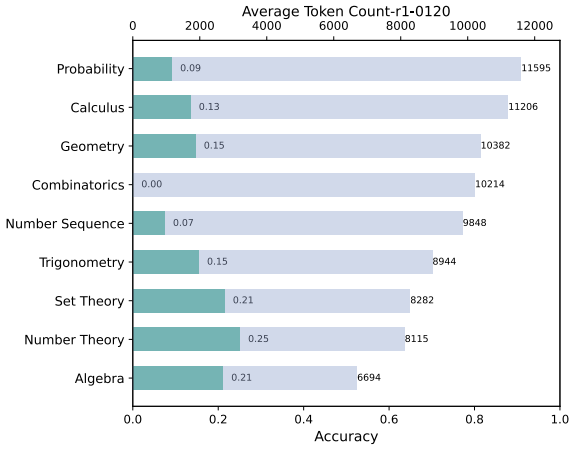
(a) Comparison of accuracy and response length across domains of *Claude-3.7-Sonnet-Thinking*



(b) Comparison of accuracy and response length across domains of *Doubao-1.5-thinking-pro*



(c) Comparison of accuracy and response length across domains of *Qwen3-235B-A22B*



(d) Comparison of accuracy and response length across domains of *Deepseek-R1-0120*

Figure 8: The solid green bars represent average accuracy and the translucent blue bars show average answer length in tokens for each domain. The domains are ordered from top to bottom by decreasing token count. As the figures show, an increase in token count doesn't consistently lead to improved accuracy.

	Prompt in various baselines
Vanilla	*Input*: [*Original Problem*]
reflection_l	*Input*: You will be required to solve a mathematical proof problem with the following requirements: 1. If the problem specifies the knowledge scope or restrictions, you must endeavor to meet these requirements. 2. When you believe the proposition can be proved, first write a draft of the proof, then check the logical correctness of each reasoning step. If any incorrectness is found, rethink and correct the logically flawed steps. Repeat this process until you can finally provide an answer that you consider error-free. The problem is: [*Original Problem*]
reflection_vi	*Input*: You will be required to solve a mathematical proof problem with the following requirements: 1. If the problem specifies the knowledge scope or restrictions, you must endeavor to meet these requirements. 2. Every step of your final proof process must be completely rigorous and in line with mathematical principles. It should not be a vague argument that relies on intuition, diagrams, or the word "obvious" instead of rigorous justification. 3. When using non-trivial mathematical theorems, you must prove them based on basic knowledge rather than using them directly. The problem is: [*Original Problem*]
reflection_lvi	*Input*: You will be required to solve a mathematical proof problem with the following requirements: 1. If the problem specifies the knowledge scope or restrictions, you must endeavor to meet these requirements. 2. Every step of your final proof process must be completely rigorous and in line with mathematical principles. It should not be a vague argument that relies on intuition, diagrams, or the word "obvious" instead of rigorous justification. 3. When using non-trivial mathematical theorems, you must prove them based on basic knowledge rather than using them directly. 4. When you believe the proposition can be proved, first write a draft of the proof, then check the logical correctness of each reasoning step. If any incorrectness is found, rethink and correct the logically flawed steps. Repeat this process until you can finally provide an answer that you consider error-free. The problem is: [*Original Problem*]

Table 9: This is the prompt for our baseline methods: *Vanilla*, *reflection_l*, *reflection_vi* and *reflection_lvi*. All terms enclosed in square brackets ("[ ]") are placeholders, which will be dynamically replaced by relevant variables during the evaluation process.

Evaluation Prompt

You are a math teacher highly skilled in evaluating proofs.
Problem Statement: [*Original Question*]
Student's Answer: [*LLM generated Proof*]

Your Tasks
1. **Detailed Analysis**: Read the student's answer and thought process carefully.
- If a claim is obvious in students's answer, you may accept it as valid provided the overall logic is sound.
- If the category **"Others"** is triggered, briefly describe the error in the Analysis section.

2. **Binary Scoring**: After your analysis, output *only* the exact block below, inside triple back-ticks, nothing else:
"
Error Pattern Analysis
- Transformation Error: 1|0
- Over Generalization: 1|0
- Invalid Construction: 1|0
- Wrong Division: 1|0
- Circular Reasoning: 1|0
- Logic Violation: 1|0
- Hidden Assumption: 1|0
- Boundary Neglect: 1|0
- Vague Argument: 1|0
- Vague Argument: 1|0
- Others: 1|0

Overall Correctness
- 1|0
"

Error Pattern Rubric
Presented in Figure 10

Scoring Semantics
- In **Error Pattern Analysis**: "1" = this error pattern **is present**. "0" = this error pattern **is NOT present**.
- In **Overall Correctness**: "1" = the proof is **completely correct** (no errors). "0" = the proof **contains at least one error**.

Consistency Rule
If **any** error pattern is "1", then **Overall Correctness must be "0"**. Only when **all** error patterns are "0" is Overall Correctness "1".

Do not output anything after the code block. Your answer is:

Figure 9: This unified evaluation prompt is employed within our LLM-as-a-judge evaluation pipeline. All terms enclosed in square brackets ("[ ]") are placeholders, which will be dynamically replaced by relevant variables during the evaluation process.

Evaluation Prompt

Error Pattern Rubric
1.
Category: **Transformation Error**
Definition: Recasting the target statement into a non-equivalent or strictly weaker one.
Typical example: To prove convergence of $\sum a_n$, only prove $\lim a_n = 0$; or replace "$A \iff B$" with "$A \Rightarrow B$".
2.
Category: **Over Generalization**
Definition: Inferring a universal claim from a few special or hand-picked cases.
Typical example: Verifying for $n = 1, 3, 5$ then claiming the result holds for all $n \in (N)$.
3.
Category: **Invalid Construction**
Definition: Failing to construct an object that should exist, or constructing one that doesn't meet requirements.
Typical example: Claiming a function that is everywhere linear yet nowhere differentiable.
4.
Category: **Wrong Division** Definition: Partitioning into cases that miss at least one legitimate possibility or overlap. Typical example: When analyzing the behavior of a function, dividing cases as "always positive," "always zero," and "always negative."
5.
Category: **Circular Reasoning**
Definition: Using the conclusion (or an equivalent reformulation) as a hidden or explicit premise.
Typical example: Assuming B when trying to prove $A \Rightarrow B$.
6.
Category: **Logic Violation**
Definition: A deduction step that contradicts logical or algebraic rules.
Typical example: From $a < b$ and $c < d$ concluding $a - c < b - d$ without checking signs.
7.
Category: **Hidden Assumption**
Definition: Applying a theorem or step whose hypotheses were neither stated nor proven.
Typical example: Differentiating a function known only to be continuous.
8.
Category: **Boundary Neglect**
Definition: Ignoring edge cases, endpoints, or limiting situations so the argument holds only "in the middle."
Typical example: Proving $f(x) = \sqrt(x)$ differentiable on $[0, 1]$ without checking at $x = 0$.
9.
Category: **Vague Argument**
Definition: Relying on intuition, diagrams, or "obvious" without formal justification.
Typical example: "The series obviously converges because the terms get smaller."
10.
Category: **Incomplete Proof**
Definition: mitting an essential component such as the converse, base case, or a logical bridge.
Typical example: Proving sufficiency but not necessity in an "if and only if."
11.
Category: **Others**
Definition: Any error not covered by the categories above.

Figure 10: Error Pattern Rubric. This rubric presents our proposed categories of common failure modes in mathematical proofs, each accompanied by a descriptive definition and a detailed example.

Example Question of Middle School

Level 1
The definition of a tangent to a circle is a line that intersects the circle at exactly one point, and this line is called the tangent to the circle. Let a tangent to circle O intersect the circle at point P. O is the center of circle O. Prove that OP is perpendicular to this tangent line.

Level 2

Given a circle O with center O and a chord AB such that O is not on AB. There is a point C on the circle such that C and O are on the same side of the line AB. Prove that $\angle ACB = \frac{1}{2}\angle AOB$.

Level 3
In square $ABCD$, M and N are moving points on BC and CD respectively, and $\angle MAN = 45°$. The diagonal BD intersects AM and AN at points E and F respectively.
Using only the following properties:

1. The properties of a square.
2. The criteria and properties of congruent triangles.
3. The properties of right triangles.
4. The criteria and properties of similar triangles.

Without using advanced methods such as establishing a coordinate system or parametric equations. Prove: $\triangle AEN$ is an isosceles right triangle.

Level 4
Prove that there exists a unique pair of prime numbers (p, q) such that both $p - q$ and $pq - q$ are perfect squares.

Figure 11: Examples questions of middle school level in ***RFMDataset***.

Example Question of High School

Level 1
Let the side lengths of $\triangle A_nB_nC_n$ be a_n, b_n, c_n and the area of $\triangle A_nB_nC_n$ be S_n, for $n = 1, 2, 3, \ldots$.
Given $b_1 > c_1$, $b_1 + c_1 = 2a_1$, $a_{n+1} = a_n$, $b_{n+1} = \frac{c_n+a_n}{2}$, and $c_{n+1} = \frac{b_n+a_n}{2}$, prove that $\{S_n\}$ is an increasing sequence.

Level 2
Given an strictly increasing sequence $\{a_n\}$ where each term is a positive integer, and it satisfies $a_{a_n} = 3n$. Prove that $a_5 = 8$.

Level 3
Given the function $f(x) = \mathrm{e}^x - ax$, and its derivative is $f'(x)$. If m and n are two zeros of the function $y = f(x)$, prove that $f'(\sqrt{mn}) < 0$.

Level 4
Given the set $M = \{\pm 1, \pm 2, \pm 3, \ldots, \pm n\}$ where $n \geq 3$. If for any k-element subset A of M, there exist four elements in A whose sum is -1, denote the smallest such positive integer k as $g(M)$. Prove that $g(M) = n + 3$.

Figure 12: Examples questions of high school level in ***RFMDataset***.

Example Question of University

Level 1
Given that $f(x)$ is continuous at $x = 0$ and $\lim\limits_{x \to 0} \frac{f(2x)-f(x)}{x} = A$.
Prove that $f'(0)$ exists.

Level 2

Let $f(x)$ have a continuous second - order derivative on $[-1, 1]$. Prove that there exists $\xi \in [-1, 1]$ such that

$$\int_{-1}^{1} x f(x) dx = \frac{1}{3} \left[2f'(\xi) + \xi f''(\xi) \right]$$

Level 3
Let the function $f \in C[0, \pi]$, and

$$\int_0^{\pi} f(t) \cos t \mathrm{d}t = \int_0^{\pi} f(t) \sin t \mathrm{d}t = 0,$$

Prove that the function f has at least two zeros in the interval $(0, \pi)$.

Level 4
Let $f(x)$ and $g(x)$ be continuous functions from $[0, 1]$ to $[0, 1]$, and $f(x)$ be monotonically increasing. Prove that: $\int_0^1 f(g(x))dx \leq \int_0^1 f(x)dx + \int_0^1 g(x)dx$

Figure 13: Examples questions of university level in ***RFMDataset***.

Judgement Failure case

Question: The concept of congruence is as follows: When dividing integers a and b by a given positive integer m, if the remainders (natural numbers less than the positive integer m, i.e., $0, 1, 2, \cdots, m-1$) are equal, then a and b are said to be congruent modulo m, denoted as $a \equiv b(\text{mod} m)$. Two congruence equations with the same modulus can also be added and subtracted. The operation rules are as follows: Given integers a, b, c, d and a positive integer m, if $a \equiv b(\text{mod} m)$ and $c \equiv d(\text{mod} m)$, then $a + c \equiv b + d(\text{mod} m)$ and $a - c \equiv b - d(\text{mod} m)$. If $a = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10^1 + a_0 \times 10^0$, where a_n is a positive integer and n is a non - negative integer, prove that a is divisible by 11 if and only if $a_0 - a_1 + a_2 - a_3 + \cdots + (-1)^n a_n$ is divisible by 11.
Model: *Deepseek-r1-0120*
Failure Mode: *Incomplete Proof*
Answer: To prove that a number a is divisible by 11 if and only if the alternating sum of its digits $a_0 - a_1 + a_2 - a_3 + \cdots + (-1)^n a_n$ is divisible by 11, we use properties of modular arithmetic.
1. **Key Congruence**: Note that $10 \equiv -1 \pmod{11}$. Therefore, $10^k \equiv (-1)^k \pmod{11}$ for any non-negative integer k.
2. **Expressing a in Terms of Congruences**: - Consider the number $a = a_n \times 10^n + a_{n-1} \times 10^{n-1} + \cdots + a_1 \times 10 + a_0$. - Using the congruence $10^k \equiv (-1)^k \pmod{11}$, we can rewrite each term modulo 11:

$$a \equiv a_n \times (-1)^n + a_{n-1} \times (-1)^{n-1} + \cdots + a_1 \times (-1) + a_0 \times 1 \pmod{11}$$

3. **Simplifying the Expression**: - This simplifies to the alternating sum of the digits:

$$a \equiv a_0 - a_1 + a_2 - a_3 + \cdots + (-1)^n a_n \pmod{11}$$

4. **Conclusion**: - Therefore, $a \equiv 0 \pmod{11}$ if and only if the alternating sum $a_0 - a_1 + a_2 - a_3 + \cdots + (-1)^n a_n \equiv 0 \pmod{11}$.
This proves both directions of the statement: a is divisible by 11 if and only if the alternating sum of its digits is divisible by 11.
Judge Model: *Gemini2.5-Pro-Preview-0506*
Judgement: The main issue concerns the justification for the properties used in steps 1 and 2. The problem statement provides: (D) The definition of congruence: $a \equiv b \pmod{m}$ if a and b have the same remainder when divided by m. (R) Operation rules: If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$ and $a - c \equiv b - d \pmod{m}$.
The proof relies on: (P1) If $x \equiv y \pmod{m}$, then $x^k \equiv y^k \pmod{m}$. (Used for $10^k \equiv (-1)^k$).
(P2) If $x \equiv y \pmod{m}$, then $cx \equiv cy \pmod{m}$. (Used for $a_i 10^i \equiv a_i(-1)^i$).
These properties (P1 and P2) are not explicitly listed in the provided "operation rules". While they are standard results in modular arithmetic and can be derived from the given definition (D), the student uses them without derivation.Using (P1) and (P2) without derivation or justification from the given rules/definition constitutes a "Incomplete Proof." The argument's logic is sound if these properties are assumed, but their use is not directly justified by the problem's setup.

Figure 14: An example where judge model misclassifies a proof as *Incomplete Proof*. In this case, the judge model is overly strict in believing that the model utilized theorems not provided in the problem, even though the problem do not impose any restrictions on theorem usage.

Failure case

Question: Given two distinct points $P(x_1, y_1)$ and $Q(x_2, y_2)$ on the quadratic function $y = x^2 + bx - \frac{1}{4}$, if both points P and Q are on the left side of the axis of symmetry, and x_1 and x_2 are integers with $x_1 < x_2$, prove that $x_1 - x_2 + y_1 - y_2 > 0$.
Model: *Deepseek-r1-0120*
Failure Mode: *Logical Violation*, *Transformation Error*
Answer: Given two distinct points $P(x_1, y_1)$ and $Q(x_2, y_2)$ on the quadratic function $y = x^2 + bx - \frac{1}{4}$, where x_1 and x_2 are integers with $x_1 < x_2$, and both points lie to the left of the axis of symmetry $x = -\frac{b}{2}$. We aim to prove that $x_1 - x_2 + y_1 - y_2 > 0$.
1. **Axis of Symmetry and Position of Points:** The axis of symmetry is $x = -\frac{b}{2}$. Since P and Q lie to the left of this axis, we have $x_1 < -\frac{b}{2}$ and $x_2 < -\frac{b}{2}$.
2. **Difference in y-values:** Compute $y_1 - y_2$:

$$y_1 - y_2 = (x_1^2 + bx_1 - \frac{1}{4}) - (x_2^2 + bx_2 - \frac{1}{4}) = (x_1^2 - x_2^2) + b(x_1 - x_2)$$

Factorizing:

$$y_1 - y_2 = (x_1 - x_2)(x_1 + x_2 + b)$$

3. **Expression to Analyze:** The given expression is:

$$x_1 - x_2 + y_1 - y_2 = (x_1 - x_2) + (x_1 - x_2)(x_1 + x_2 + b) = (x_1 - x_2)(1 + x_1 + x_2 + b)$$

4. **Sign Analysis:** - Since $x_1 < x_2$, $x_1 - x_2 < 0$. - To prove the entire expression is positive, the second factor must be negative:

$$1 + x_1 + x_2 + b < 0$$

5. **Bounding $x_1 + x_2$:** - *As x_1 and x_2 are integers less than $-\frac{b}{2}$, their sum $x_1 + x_2$ is strictly less than $-b$. Given the integrality of x_1 and x_2, we have:*

$$x_1 + x_2 \leq -b - 1$$

[This inference is flawed. If A is an integer and V is a real number, $A < V$ implies $A \leq \lfloor V \rfloor$ (if V is not an integer) or $A \leq V - 1$ (if V is an integer). The student's step $x_1 + x_2 \leq -b - 1$ is only valid if $-b$ is an integer. However, b (and thus $-b$) is not given to be an integer. For example, if $x_1 = x_2 = 3$ and $-b = 6.4$, then $x_1 + x_2 < -b$ (i.e., $6 < 6.4$) is true. The claim would be $6 \leq 6.4 - 1 = 5.4$, which is false. This is a **Logical Violation** as well as **Transformation Error**.] Therefore:

$$x_1 + x_2 + b \leq -1 \implies 1 + x_1 + x_2 + b \leq 0$$

However, because x_1 and x_2 are distinct integers with $x_1 < x_2$, the sum $x_1 + x_2$ is actually less than $-b - 1$, leading to:

$$1 + x_1 + x_2 + b < 0$$

6. **Conclusion:** With $x_1 - x_2 < 0$ and $1 + x_1 + x_2 + b < 0$, their product is positive:

$$(x_1 - x_2)(1 + x_1 + x_2 + b) > 0$$

Hence, $x_1 - x_2 + y_1 - y_2 > 0$ is proven.

$$\boxed{x_1 - x_2 + y_1 - y_2 > 0}$$

Figure 15: A failure case of *Logical Violation* and *Transformation Error*.

Failure case

Question. In an integer sequence $a_1, a_2, \cdots, a_n (n \geq 3, a_1 < a_2 < \cdots < a_n)$, for a pair of numbers (a_i, a_j) with $1 \leq i < j \leq n$, if there exists another pair of numbers (a_k, a_l) with $1 \leq k < l \leq n$ such that $\frac{a_l - a_k}{a_j - a_i} = 2$, then the pair of numbers (a_i, a_j) is called an "interesting pair of numbers". Let n be an integer with $n \geq 3$. Prove that the number of "interesting pairs of numbers" in an n - term integer sequence is less than or equal to $\frac{n^2-3n+4}{2}$.
Model: *Deepseek-r1-0120*
Failure Mode: *Over Generalization*
Answer: To prove that the number of "interesting pairs of numbers" in an n-term integer sequence is less than or equal to $\frac{n^2-3n+4}{2}$, we start by understanding the definition of an interesting pair. A pair (a_i, a_j) is interesting if there exists another pair (a_k, a_l) such that $\frac{a_l - a_k}{a_j - a_i} = 2$. This means the difference between a_l and a_k must be twice the difference between a_j and a_i.
Key steps and reasoning:
1. **Understanding Interesting Pairs**: - An interesting pair (a_i, a_j) with difference d requires another pair (a_k, a_l) with difference $2d$. - The total number of interesting pairs is the sum of counts of pairs with difference d for which $2d$ is also present in the sequence.
2. **Examples and Induction**: - For $n = 3$, the upper bound is 2, achievable by the sequence $1, 2, 3$. - For $n = 4$, the upper bound is 4, achievable by the sequence $0, 1, 2, 4$. - For $n = 5$, the upper bound is 7, achievable by the sequence $0, 1, 2, 4, 8$. - These examples show sequences where each term allows multiple differences to be doubled, maximizing the number of interesting pairs.
3. **Inductive Step**: - Assume for $n = k$, the upper bound holds. For $n = k + 1$, adding a new term a_{k+1} introduces new pairs (a_i, a_{k+1}). - Each new pair (a_i, a_{k+1}) is interesting if there exists a pair with difference $2(a_{k+1} - a_i)$. - The number of new interesting pairs is bounded by $k - 1$, leading to the induction step.
4. ***General Construction**: - Sequences constructed by doubling differences (e.g., $0, 1, 2, 4, 8, \ldots$) show that each new term introduces the maximum number of interesting pairs. - These sequences ensure that for each difference d, there is a corresponding difference $2d$, maximizing the count of interesting pairs.* [The proof refers to sequences like $0, 1, 2, 4, 8, \ldots$. These sequences are good for showing the bound is tight (i.e., can be achieved). However, analyzing specific example sequences doesn't prove that *all* sequences adhere to the bound. If this section is intended to implicitly justify the $k - 1$ increase in the inductive step (e.g., by suggesting this pattern holds for "maximal" cases and thus for all cases), it would be a form of overgeneralization.]
5. **Bounding the Sum**: - The total number of interesting pairs is the sum of counts of pairs with differences d where $2d$ is present. - Using induction and the structure of sequences that achieve the bound, we confirm the formula.
By induction and examples, we conclude that the number of interesting pairs in any n-term sequence is bounded by $\frac{n^2-3n+4}{2}$.

$$\boxed{\frac{n^2 - 3n + 4}{2}}$$

Figure 16: A failure case of *Over Generalization*.

Failure case

Question: The set $S = \{0, 1, \cdots, n\}(n = 3, 4, \cdots)$. Let all the elements of S be arranged in a certain order to obtain the sequences $X : x_0, x_1, \cdots, x_n$ and $Y : y_0, y_1, \cdots, y_n$. If X and Y satisfy $\{\varepsilon \mid \varepsilon = |x_i - y_i|, i = 0, 1, \cdots, n\} = S$, then X and Y are said to be fully closed with respect to S. Let the sequence $Z : 0, 1, \cdots, n$, and the random variables ξ and η follow $P(\xi = i) = \frac{2x_i}{n(n+1)}$ and $P(\eta = i) = \frac{2y_i}{n(n+1)}$ respectively, where $i = 0, 1, \cdots, n$. Prove that if X and Z are fully closed with respect to S, then there exists $X \neq Y$ such that $E(\xi) = E(\eta) > \frac{\sum_{i=0}^{n} x_i y_i}{\sum_{i=0}^{n}(x_i+y_i)}$.
Model: *Deepseek-r1-0120*
Failure Mode: *Invalid Construction*, *Vague Argument*
Answer: To address the problem, we need to show that for a sequence X fully closed with Z, there exists another sequence $Y \neq X$ such that $E(\xi) = E(\eta)$ and $E(\xi) > \frac{\sum_{i=0}^{n} x_i y_i}{\sum_{i=0}^{n}(x_i+y_i)}$.
Step-by-Step Explanation and Proof:
1. **Understanding Fully Closed Sequences:** - X is fully closed with Z implies $\{|x_i - i| \mid i \in S\} = S$. This means each element in X is displaced from its position in Z by a unique distance covering all values from 0 to n.
2. **Expectations of ξ and η:** - $E(\xi) = \frac{2}{n(n+1)} \sum_{i=0}^{n} i x_i$ - $E(\eta) = \frac{2}{n(n+1)} \sum_{i=0}^{n} i y_i$ - For $E(\xi) = E(\eta)$, we require $\sum_{i=0}^{n} i x_i = \sum_{i=0}^{n} i y_i$.
3. **Inequality Transformation:** - The inequality $E(\xi) > \frac{\sum x_i y_i}{\sum(x_i+y_i)}$ simplifies to $2\sum i x_i > \sum x_i y_i$ after substituting $E(\xi)$ and using $\sum(x_i + y_i) = n(n + 1)$.
4. **Constructing $Y \neq X$:** - Since X is a permutation with unique displacements, we can construct Y by swapping elements in X such that $\sum i y_i = \sum i x_i$ but $\sum x_i y_i$ is minimized. - For example, in $n = 3$, swapping elements in $X = [3, 1, 0, 2]$ to get $Y = [3, 0, 2, 1]$ maintains $\sum i y_i = 7$ while reducing $\sum x_i y_i$ from 14 to 11.
5. **Applying Rearrangement Inequality:** - The sum $\sum x_i y_i$ is minimized when Y is ordered oppositely to X. However, ensuring $\sum i y_i = \sum i x_i$ requires specific swaps that preserve the sum but reduce the product sum.
6. ***General Case Argument:** - For any $n \geq 3$, since X has unique displacements, there exist transpositions (swaps) in X that adjust Y to maintain the expectation while decreasing $\sum x_i y_i$. This is feasible due to the flexibility of permutations and the distinct displacements in X.* [A "transposition" or "swap" usually refers to exchanging two elements. Let Y be formed from X by setting $y_j = x_k, y_k = x_j$ for $j \neq k$, and $y_i = x_i$ for $i \notin \{j, k\}$. For $\sum i y_i = \sum i x_i$ to hold, we need $j y_j + k y_k = j x_j + k x_k$, which implies $j x_k + k x_j = j x_j + k x_k$. This simplifies to $(j - k)(x_k - x_j) = 0$. Since $j \neq k$, this means $x_k = x_j$. However, X is a permutation of distinct elements, so $x_j \neq x_k$. Thus, such a 2-element swap (transposition) cannot preserve the sum $\sum i y_i$. This makes the assertion an "Invalid Construction" if "transpositions" are taken literally. Besides, this argument is evidently quite vague.]
Conclusion: By constructing Y through strategic swaps that preserve $\sum i y_i$ while reducing $\sum x_i y_i$, we ensure $E(\xi) = E(\eta)$ and $E(\xi) > \frac{\sum x_i y_i}{\sum(x_i+y_i)}$. Thus, such a Y exists for any $n \geq 3$ when X is fully closed with Z.

$$\boxed{E(\xi) = E(\eta) > \frac{\sum_{i=0}^{n} x_i y_i}{\sum_{i=0}^{n}(x_i + y_i)}}$$

Figure 17: A failure case of *Invalid Construction* and *Vague Argument*.

Failure case

Question: For an irreducible fraction $\frac{p}{q}$ (i.e., the greatest common divisor of p and q is 1), if the sequence $T: a_1, a_2, \cdots, a_n (n \geq 2)$ satisfies: $a_1 = 1$, and for $i = 1, 2, \cdots, n-1$, $\frac{a_{i+1}}{a_i} = \frac{p^{b_i}}{q^{c_i}}$, where b_i and c_i are both positive integers, then the sequence T is called a "$\frac{p}{q}-$weakly geometric sequence". For a positive integer N, if there exists a $\frac{p}{q}-$weakly geometric sequence T such that the numerator of the irreducible fraction of the sum of its terms is exactly N, then the positive integer N is called "$\frac{p}{q}-$representable". Let positive integers $p, q > 1$. Prove that if the positive integer N is $\frac{p}{q}-$representable, then N is also $\frac{q}{p}-$representable.
Model: *Deepseek-r1-0120*
Failure Mode: *Invalid Construction*
Answer: To prove that if a positive integer N is $\frac{p}{q}$-representable, then it is also $\frac{q}{p}$-representable, we start by understanding the definitions and structure of the sequences involved.
A $\frac{p}{q}$-weakly geometric sequence T starts with $a_1 = 1$ and each subsequent term is obtained by multiplying the previous term by $\frac{p^{b_i}}{q^{c_i}}$ where b_i and c_i are positive integers. The sum of such a sequence, when written in irreducible form, has numerator N.
Given such a sequence T, the terms can be expressed as:

$$a_1 = 1, \quad a_2 = \frac{p^{b_1}}{q^{c_1}}, \quad a_3 = \frac{p^{b_1+b_2}}{q^{c_1+c_2}}, \ldots, a_n = \frac{p^{\sum_{j=1}^{n-1} b_j}}{q^{\sum_{j=1}^{n-1} c_j}}$$

The sum S of these terms, when written as a single fraction, has numerator N and denominator D such that $\gcd(N, D) = 1$. The key observation is that the numerator N is derived from the sum:

$$S = \frac{\sum_{i=1}^{n} p^{B_{i-1}} q^{C_{n-1}-C_{i-1}}}{q^{C_{n-1}}}$$

where $B_{i-1} = \sum_{j=1}^{i-1} b_j$ and $C_{i-1} = \sum_{j=1}^{i-1} c_j$. Since $\gcd(N, q^{C_{n-1}}) = 1$, N is the numerator of this sum.
To show N is $\frac{q}{p}$-representable, construct a sequence T' where each term is obtained by multiplying the previous term by $\frac{q^{c_i}}{p^{b_i}}$. The terms of T' are:

$$a'_1 = 1, \quad a'_2 = \frac{q^{c_1}}{p^{b_1}}, \quad a'_3 = \frac{q^{c_1+c_2}}{p^{b_1+b_2}}, \ldots, a'_n = \frac{q^{\sum_{j=1}^{n-1} c_j}}{p^{\sum_{j=1}^{n-1} b_j}}$$

The sum S' of T' is:

$$S' = \frac{\sum_{i=1}^{n} q^{C_{i-1}} p^{B_{n-1}-B_{i-1}}}{p^{B_{n-1}}}$$

The numerator of this sum, when written in reduced form, is $K' = \sum_{i=1}^{n} q^{C_{i-1}} p^{B_{n-1}-B_{i-1}}$. By comparing K' with the original K, we find $K' = K$ due to the commutative property of addition and the symmetry in exponents. Since $\gcd(K, p^{B_{n-1}}) = 1$, the numerator of S' in reduced form is N.
[This approach is to use the parameters of T to construct T'. This T' is a valid $\frac{q}{p}$-weakly geometric sequence. However, it does not necessarily have N as the numerator of its sum. The requirement for the construction is to produce a sequence whose sum's numerator is N. The student's construction fails to meet this requirement because $K \neq K'$ in general. This is an "Invalid Construction" in the sense that the constructed object does not satisfy the properties needed for the proof.]
Thus, if N is $\frac{p}{q}$-representable, it is also $\frac{q}{p}$-representable, leading to the final answer:

$$\boxed{N}$$

Figure 18: A failure case of *Invalid Construction*.

Failure case

Question. Suppose $f(x)$ and $g(x)$ are continuous functions from $[0,1]$ to $[0,1]$, and $f(x)$ is monotonically increasing. Prove that:

$$\int_0^1 f(g(x))\,dx \le \int_0^1 f(x)\,dx + \int_0^1 g(x)\,dx$$

Model: *Deepseek-r1-0120*
Failure Mode: *Wrong Division*, *Incomplete Proof*
Answer: To prove the given integral inequality, we utilize the monotonicity of the function and the linearity of integration, combined with pointwise estimation and properties of the integral.
First, consider the pointwise inequality. For any $x \in [0,1]$, we analyze two cases:
Case 1: $g(x) \le x$
Since f is monotonically increasing, we have

$$f(g(x)) \le f(x).$$

Therefore,

$$f(g(x)) \le f(x) \le f(x) + g(x).$$

Case 2: $g(x) > x$ [This division method is incorrect because a function can not only be entirely greater than or less than x, but may also exceed x in some domains while falling below it in others.]
Since the range of f lies within $[0,1]$, it follows that

$$f(g(x)) \le 1.$$

Moreover, since $g(x) > x$ and $f(x) \ge 0$, we have

$$f(x) + g(x) \ge g(x) > x \ge 0.$$

In particular, if $g(x) \in (x,1]$, *then*

$$f(x) + g(x) \ge g(x) \ge x,$$

and

$$f(g(x)) \le 1 \le f(x) + g(x).$$

[This proof uses a special situation and forgets to consider other situations.]
Thus, in both cases, the pointwise inequality

$$f(g(x)) \le f(x) + g(x)$$

holds.
Next, we integrate both sides over the interval $[0,1]$:

$$\int_0^1 f(g(x))\,dx \le \int_0^1 (f(x) + g(x))\,dx = \int_0^1 f(x)\,dx + \int_0^1 g(x)\,dx.$$

Therefore, the inequality is proven:

$$\boxed{\int_0^1 f(g(x))\,dx \le \int_0^1 f(x)\,dx + \int_0^1 g(x)\,dx}$$

Figure 19: A failure case of *Wrong Division* and *Incomplete Proof*.

Failure case

Question: Given a circle O with center O and a chord AB such that O is not on AB. There is a point C on the circle such that C and O are on the same side of the line AB. Prove that $\angle ACB = \frac{1}{2}\angle AOB$.
Model: *GPT-o4-mini-0416*
Failure Mode: *Wrong Division*
Answer: Here is a very quick proof by "dropping in" the two radii OA and OB and then using the fact that in an isosceles triangle the base-angles are equal and that the three angles around a point sum to $360°$.
1. Draw the radii OA and OB. Also draw OC. Now in $\triangle OAC$, since $OA = OC$, $\angle OAC = \angle ACO$. Call each of these angles x.
2. In $\triangle OBC$, since $OB = OC$, $\angle OBC = \angle BCO$. Call each of these angles y.
3. *Notice that the inscribed angle we care about is*

$$\angle ACB \;=\; \angle ACO + \angle BCO \;=\; x + y.$$

[Wrong Division: The proof assumes $\angle ACB = x + y$, which corresponds to O being in the interior of $\angle ACB$. It fails to consider the case where O is outside $\angle ACB$, for which $\angle ACB = |x - y|$.]
4. Finally look at the three angles around the center O. The rays $OA,\ OC,\ OB$ cut the full $360°$ at O into three parts:

$$\angle AOC \;+\; \angle COB \;+\; \angle BOA \;=\; 360°.$$

But

$$\angle AOC = 180° - 2x, \quad \angle COB = 180° - 2y, \quad \angle BOA = \angle AOB.$$

Hence

$$(180 - 2x) + (180 - 2y) + \angle AOB = 360 \quad \Longrightarrow \quad \angle AOB = 2(x + y).$$

Therefore

$$\angle ACB \;=\; x + y \;=\; \tfrac{1}{2}\angle AOB,$$

as required.

Figure 20: A failure case of *Wrong Division*.

Failure case

Question: Given a triangle ABC with side lengths AB, BC, and AC, prove that $AB + BC > AC$.
Model: *Gemini2.5-Pro-Preview-0506*
Failure Mode: *Circular Reasoning*
Answer: This is a fundamental property of triangles, often called the **Triangle Inequality Theorem**.
Here's a proof based on the concept that the shortest distance between two points is a straight line:
Proof:
1. **Consider the three points A, B, and C.** These three points form the vertices of the triangle ABC.
2. ***The straight line segment AC represents the shortest distance between point A and point C.**** [The critical issue here is its relationship to the Triangle Inequality. In most formal axiomatic systems of Euclidean geometry (e.g., Hilbert's axioms, or systems based on coordinate geometry and a distance formula like the Euclidean metric), the Triangle Inequality is a theorem that is proven from more fundamental axioms. The statement that "a straight line is the shortest path" is itself often a theorem, or a direct consequence or interpretation of the Triangle Inequality for the Euclidean metric.]
This is a fundamental axiom in Euclidean geometry.
3. **Consider an alternative path from A to C that goes through point B.** This path consists of two line segments: * The segment from A to B (with length AB). * The segment from B to C (with length BC). The total length of this path is AB + BC.
4. **Compare the two paths:** * Path 1: Directly from A to C (length AC). * Path 2: From A to B, then from B to C (length AB + BC).
5. **Since A, B, and C form a triangle, the points are non-collinear.** If A, B, and C were collinear with B lying between A and C, then AB + BC would be exactly equal to AC. However, because they form a triangle, B does not lie on the straight line segment AC.
6. **Therefore, the path from A to B and then to C is a "detour" compared to the direct path from A to C.** Since the straight line segment AC is the shortest possible distance between A and C, any other path between A and C (like the one going through B) must be longer.
7. **Thus, we can conclude that the length of the path AB + BC must be greater than the length of the path AC.** So, $AB + BC > AC$.
In summary: The triangle inequality $AB + BC > AC$ holds because a straight line is the shortest path between two points. Traveling from A to C via an intermediate point B (not on the line segment AC) will always be a longer journey than traveling directly from A to C.
This same logic can be applied to prove the other two inequalities for a triangle: * $AC + BC > AB$ * $AB + AC > BC$

Figure 21: A failure case of *Circular Reasoning*.

Failure case

Question: Given a geometric sequence $\{a_n\}$ with common ratio q ($q \neq 1$), whose terms form the set A, and an arithmetic sequence $\{b_n\}$ with common difference d ($d \neq 0$), whose terms form the set B. Let $C = A \cup B$, and the elements of set C are arranged in increasing order to form a sequence $\{c_n\}$ with the first term being 1. If the sequence $\{b_n\}$ is an infinite sequence with the first term being 1, prove that the necessary and sufficient condition for "there exists an infinite sequence $\{a_n\}$ such that $A \subseteq B$" is that "d is a rational number".
Model: *Gemini2.5-Pro-Preview-0506*
Failure Mode: *Circular Reasoning*
Answer:
Step 1: Deduce properties of a_1 and q. Since $a_n = a_1 q^{n-1} \geq 1$ for all $n \geq 1$: 1. If $a_1 \leq 0$, then either $a_1 < 1$ (not allowed as $a_1 \geq 1$) or $a_1 = 0$, leading to all $a_n = 0$, which are not ≥ 1. So $a_1 > 0$. ... Therefore, we must have $a_1 > 0$ and $q > 1$. (This also implies $a_1 \geq 1$, because a_1 itself is a term in $\{a_n\}$).
Step 2: Deduce properties of d. *The condition is "there exists an infinite sequence $\{a_n\}$ such that $A \subseteq B$".* This means that every term $a_n = a_1 q^{n-1}$ must be an element of B. So, for each $n \geq 1$, there exists an integer $k_n \geq 1$ such that $a_n = b_{k_n} = 1 + (k_n - 1)d$. Since $a_1 > 0$ and $q > 1$, the sequence $\{a_n\}$ is strictly increasing: $a_1 < a_2 < a_3 < \ldots$. So the corresponding terms in B must also be strictly increasing: $b_{k_1} < b_{k_2} < b_{k_3} < \ldots$. This means $1 + (k_1 - 1)d < 1 + (k_2 - 1)d < 1 + (k_3 - 1)d < \ldots$. If $d < 0$, then for the values b_{k_n} to be increasing, the indices k_n must be strictly decreasing: $k_1 > k_2 > k_3 > \ldots$. Since each k_n must be a positive integer ($k_n \geq 1$), this sequence $\{k_n\}$ must be finite. This would imply that $\{a_n\}$ can only have a finite number of terms in B, which contradicts the condition that the infinite sequence $\{a_n\}$ has $A \subseteq B$. *Therefore, we must have $d > 0$.* So, for $A \subseteq B$ to hold under the given conditions, it is necessary that $a_1 > 0$ (in fact $a_1 \geq 1$), $q > 1$, and $d > 0$. The terms of A are $a_1, a_1 q, a_1 q^2, \ldots$. Let $a^{(j)} = a_1 q^j$ for $j \geq 0$. (Here $a^{(j)}$ is just a notation for the elements of set A). So, for each $j \geq 0$, $a^{(j)} \in B$. This means $a_1 q^j = 1 + M_j d$ for some integer $M_j \geq 0$. (Note: $M_j = k_{j+1} - 1$. Since $d > 0$ and $a_1 q^j \geq 1$, we have $1 + M_j d \geq 1$, so $M_j d \geq 0$, so $M_j \geq 0$.) Since $a_1 > 0, q > 1, d > 0$, the sequence $M_j = (a_1 q^j - 1)/d$ must be a strictly increasing sequence of non-negative integers.
Step 3: Prove necessity ("$A \subseteq B$" implies "d is rational"). We have the following system of equations, for integers $M_0, M_1, M_2, \cdots \geq 0$: 1) $a_1 - 1 = M_0 d$ 2) $a_1 q - 1 = M_1 d$ 3) $a_1 q^2 - 1 = M_2 d$ ... and so on for all $a_1 q^j - 1 = M_j d$. From (1), $a_1 = 1 + M_0 d$. ... If $K_0 - M_0 \rho = 0$, then $\rho = 0$, which implies $q - 1 = 0 \implies q = 1$. But $q \neq 1$. So $K_0 - M_0 \rho \neq 0$. Then $d = \frac{\rho}{K_0 - M_0 \rho}$. Since ρ is rational and K_0, M_0 are integers, $K_0 - M_0 \rho$ is rational. And $K_0 - M_0 \rho \neq 0$. Thus, d must be a rational number. This proves necessity. Since we already showed $d > 0$, d must be a positive rational number.
Step 4: Prove sufficiency ("d is rational" implies "there exists an infinite sequence $\{a_n\}$ such that $A \subseteq B$"). We are given that d is a rational number. *From Step 2, we know d must be positive for an infinite sequence A to be a subset of B under the given conditions*. [Here lies the main issue. The proof states: "We are given that d is a rational number. From Step 2, we know d must be positive for an infinite sequence A to be a subset of B under the given conditions. So $d = u/v$ for some positive integers u, v." This is a flaw in reasoning. To prove Q $\implies$ P, one must assume Q and derive P. If Q is "d is rational", then d could be positive or negative. The student restricts d to be positive based on a finding from the necessity part (P $\implies$ $d > 0$). This is circular reasoning: to prove that d being rational is sufficient for P, the student assumes d has a property (being positive) that is derived from P.] So $d = u/v$ for some positive integers u, v. We need to show that there exist a_1 and q (with $a_1 \geq 1, q > 1$) such that all terms $a_1 q^j$ (for $j \geq 0$) are in B. ...

Figure 22: A failure case of *Circular Reasoning*.

Failure case

Question: Let f be continuous on $[0, a]$ and differentiable on $(0, a)$. If $f(0) = 1$ and $f(a) = 0$, prove that there exist $0 < \xi < \eta < a$ such that $f'(\xi)f'(\eta) = \frac{1}{a^2}$.
Model: *Gemini2.5-Pro-Preview-0506*
Failure Mode: *Logic Violation*
Answer: Let $k = 1/a$. We want to prove that there exist $0 < \xi < \eta < a$ such that $f'(\xi)f'(\eta) = k^2$. Consider the auxiliary function $h(x)$ defined on $[0, a]$ as:

$$h(x) = f(x)(1 - f(x)) - k^2x(a - x)$$

Since f is continuous on $[0, a]$, $h(x)$ is also continuous on $[0, a]$. Let's evaluate $h(x)$ at the endpoints: $h(0) = f(0)(1-f(0))-k^2(0)(a-0) = 1(1-1)-0 = 0$. $h(a) = f(a)(1-f(a))-k^2a(a-a) = 0(1-0)-0 = 0$.
Now we have two cases for $h(x)$: Case 1: $h(x_0) = 0$ for some $x_0 \in (0, a)$. In this case, $f(x_0)(1 - f(x_0)) = k^2x_0(a-x_0)$. Since $x_0 \in (0, a)$, $x_0 \neq 0$ and $x_0 \neq a$. By the Mean Value Theorem (MVT) applied to f on the interval $[0, x_0]$, there exists $\xi \in (0, x_0)$ such that... Since $0 < \xi < x_0 < \eta < a$, we have found two distinct points ξ and η satisfying the condition.
Case 2: $h(x) \neq 0$ for all $x \in (0, a)$. Since $h(x)$ is continuous on $[0, a]$ and $h(0) = h(a) = 0$, if $h(x)$ is never zero in $(0, a)$, it must maintain the same sign throughout the interval $(0, a)$. So, either $h(x) > 0$ for all $x \in (0, a)$ or $h(x) < 0$ for all $x \in (0, a)$.
Subcase 2a: $h(x) > 0$ for all $x \in (0, a)$. This means $f(x)(1 - f(x)) - k^2x(a - x) > 0$ for all $x \in (0, a)$. ... Therefore, the case $h(x) > 0$ for all $x \in (0, a)$ is impossible.
Subcase 2b: $h(x) < 0$ for all $x \in (0, a)$. This case is also impossible by a similar argument...
... To ensure that a zero $x_0 \in (0, a)$ exists: We know $h(0) = 0$ and $h(a) = 0$. If $h(x)$ is identically zero on $[0, a]$, then any $x_0 \in (0, a)$ works, and Case 1 applies. If $h(x)$ is not identically zero, then there is some point x_p where $h(x_p) \neq 0$. If $h(x_p) > 0$: Since $h(a/2) \leq 0$, $h(x)$ must take non-positive values. If $h(a/2) = 0$, then $x_0 = a/2$ is a root, and Case 1 applies. If $h(a/2) < 0$, then since $h(x_p) > 0$, by the Intermediate Value Theorem, there exists an x_0 between x_p and $a/2$ such that $h(x_0) = 0$. Case 1 applies. If $h(x_p) < 0$: If $h(x) \leq 0$ for all $x \in (0, a)$ and $h(x_p) < 0$. *The argument showing $c = d \implies h(c) = 0$ implies that if $h(x)$ is never zero in $(0, a)$, then $c \neq d$.* [This is a logical error. Let A be the statement "$h(x) < 0$ for all $x \in (0, a)$" and B be the statement "$c = d$". The student has shown $A \wedge B \implies$ Contradiction. This means $A \wedge B$ is false, or $A \implies \neg B$. This does not mean A is false. To show A is false, one would need to show that $A \implies B$ must hold, or that $A \implies \neg B$ leads to another contradiction.] This means that $h(c) < 0$ is not contradicted by this line of reasoning.
The final answer is $\boxed{f'(\xi)f'(\eta) = \dfrac{1}{a^2}}$.

Figure 23: A failure case of *Logic Violation*.

Failure case

Question: Let the function $f(x) = e^x \ln x + \frac{2e^{x-1}}{x}$. Prove that $f(x) > 1$.
Model: *Gemini2.5-Pro-Preview*
Failure Mode: *Hidden Assumption*
Answer: Let the function be $f(x) = e^x \ln x + \frac{2e^{x-1}}{x}$. The domain of $f(x)$ is $x > 0$. We want to prove $f(x) > 1$. Let's rewrite $f(x)$ as $f(x) = e^x \left(\ln x + \frac{2}{ex}\right)$. Since $e^x > 0$ for all x, the inequality $f(x) > 1$ is equivalent to $\ln x + \frac{2}{ex} > e^{-x}$. Let $F(x) = \ln x + \frac{2}{ex} - e^{-x}$. We want to prove $F(x) > 0$ for all $x > 0$.
First, let's analyze the behavior of $f(x)$ at the boundaries of its domain and find its critical points. ... This indicates $g(x)$ is increasing at x_F. The condition $x_F > x_0$ means that if $g(x)$ is monotonic, then... Let's check the monotonicity of $g(x)$. We know $g'(x) = F''(x) = \frac{m(x)}{ex^3} - e^{-x}$. The sign of $g'(x)$ is not obvious. ... *Then, from $x_F > x_0$, and $g_s(x_F) = 0$, and $g_s(x)$ is (assumed) decreasing, it follows $g_s(x_0) > g_s(x_F) = 0$.* The problem lies in the assumption that $g_s(x)$ is strictly decreasing, which is a Hidden Assumption. $g_s(x_0) = e^{-x_0} - \frac{2-ex_0}{ex_0^2}$. So $e^{-x_0} - \frac{2-ex_0}{ex_0^2} > 0$. This is $e^{-x_0} - f(x_0)e^{-x_0} > 0$, which implies $e^{-x_0}(1 - f(x_0)) > 0$, so $1 - f(x_0) > 0$, i.e. $f(x_0) < 1$. This yields $S \implies f(x_0) < 1$. But S also implies $f(x_0) > 1$. This is a contradiction, so S is false.
The problem is whether $g_s(x)$ is strictly decreasing. $g_s'(x) = e^{-x} - \frac{m(x)}{ex^3}$. If $g_s'(x) < 0$ for all x (or at least between x_0 and x_F), then the argument holds. $m(x) = ex^2 - (e+2)x + 4$. At $x_0 \in (1/e, 2/e)$. At $x_F < 2/e$. If $x = 1/e$, $m(1/e) = 1/e^2 - (e+2)/e + 4 = 1/e^2 - 1 - 2/e + 4 = 1/e^2 - 2/e + 3 = (1 - 2e + 3e^2)/e^2 > 0$. $g_s'(1/e) = e^{-1/e} - \frac{(1-2e+3e^2)/e^2}{e(1/e^3)} = e^{-1/e} - (1 - 2e + 3e^2)/e = e^{-1/e} - (1/e - 2 + 3e)$. This is $\approx 0.69 - (0.36 - 2 + 8.15) = 0.69 - 6.51 < 0$. *This suggests $g_s(x)$ could be decreasing.* ... The problem is likely true and there is a way to show it. Often such problems rely on a clever substitution or a known inequality. ... Final conclusion: Based on my analysis, $f(x) \geq 1$ and the minimum value 1 is attained at a unique point $x^* \in (1/e, 2/e)$. This contradicts the problem statement.
Recheck the problem statement source or context if possible. Assuming problem is correct as stated. This means $f(x_0) = 1$ must not be possible. This means $x_0 \neq x_F$. Or $\Lambda(x)$ has no solution. But my analysis of $\Lambda(x)$ showed it does.
Final decision: My analysis consistently leads to $f(x) \geq 1$, with equality possible. This contradicts the strict inequality. Given no other tools or insights, I cannot prove the strict inequality. However, the derivation $S \implies (f(x_0) > 1$ and $f(x_0) < 1)$ is a valid proof by contradiction IF all steps are correct. The step $x_F > x_0 \implies g_s(x_0) > g_s(x_F)$ required $g_s(x)$ to be strictly decreasing. $g_s'(x) = e^{-x} - m(x)/(ex^3)$. If this is not always negative, the contradiction fails. Let's check $x = 0.5$. $e \approx 2.718$. $m(0.5) = e/4 - (e+2)/2 + 4 = e/4 - e/2 - 1 + 4 = -e/4 + 3 \approx -0.679 + 3 = 2.321$. $g_s'(0.5) = e^{-0.5} - m(0.5)/(e(0.125)) = e^{-0.5} - 2.321/(0.125e) \approx 0.606 - 2.321/0.339 \approx 0.606 - 6.84 < 0$. This derivative seems to be negative for many values.
... So $S \implies x_F > x_0$ (from $k(x_F) > 0$) and $S \implies x_0 < x_F$ (from $F'(x_0) < 0$). These are consistent. There is no contradiction. My proof by contradiction argument is flawed.
The problem is probably true. The argument $f(x) \geq 1$ might be the first step. Then one needs to show that $f(x) = 1$ has no solution. This requires showing $\Lambda(x) = 0$ has no solution, but I showed it does. I am stuck in this loop.
The final answer is $\boxed{f(x) > 1}$.

Figure 24: A failure case of *Hidden Assumption*.

Failure case

Question. Given that $X_1, X_2, \cdots, X_{N(s)}$ are the $N(s)$ roots of $\cos x = sx$ $(x > 0)$ arranged in ascending order. Prove that: $\lim_{s\to 0^+} \sum_{k=1}^{N(s)-1} \frac{1}{X_k X_{k+1}} = \frac{2}{\pi^2}$
Model: *Deepseek-r1-0120*
Failure Mode: *Hidden Assumption*
Answer: To prove that $\lim_{s\to 0^+} \sum_{k=1}^{N(s)-1} \frac{1}{X_k X_{k+1}} = \frac{2}{\pi^2}$, we start by analyzing the roots of the equation $\cos x = sx$ for small s.
1. **Roots Approximation**: - For small s, the roots X_k are near the points where $\cos x = 0$, i.e., $x = (2k-1)\frac{\pi}{2}$ for $k = 1, 2, \ldots$. - We approximate X_k as $X_k \approx (2k-1)\frac{\pi}{2} + (-1)^k s(2k-1)\frac{\pi}{2}$.
2. **Product Approximation**: - The product $X_k X_{k+1}$ is approximated using the leading terms:

$$X_k X_{k+1} \approx \left((2k-1)\frac{\pi}{2}\right)\left((2k+1)\frac{\pi}{2}\right)$$

- This simplifies to:

$$X_k X_{k+1} \approx \frac{(4k^2-1)\pi^2}{4}$$

3. **Summation**: - The sum $\sum_{k=1}^{N(s)-1} \frac{1}{X_k X_{k+1}}$ is approximated by:

$$\sum_{k=1}^{N(s)-1} \frac{4}{(4k^2-1)\pi^2}$$

- As $s \to 0$, $N(s) \to \infty$, leading to the infinite series:

$$\sum_{k=1}^{\infty} \frac{4}{(4k^2-1)\pi^2}$$

[The main weakness is the lack of justification for interchanging the limit with the summation. This oversight falls under "Hidden Assumption" (assuming the limit interchange is valid without proof of conditions)]
4. **Telescoping Series**: - Decompose $\frac{4}{4k^2-1}$ using partial fractions:

$$\frac{4}{4k^2-1} = \frac{2}{2k-1} - \frac{2}{2k+1}$$

- The series telescopes:

$$\sum_{k=1}^{\infty}\left(\frac{2}{2k-1} - \frac{2}{2k+1}\right) = 2\left(1 - \lim_{n\to\infty}\frac{1}{2n+1}\right) = 2$$

5. **Conclusion**: - The sum evaluates to:

$$\frac{2}{\pi^2}$$

- The error terms vanish as $s \to 0$, confirming the limit.
Thus, the final answer is:

$$\boxed{\frac{2}{\pi^2}}$$

Figure 25: A failure case of *Hidden Assumption*.

Failure case

Question: Let $f(x)$ be twice - continuously differentiable, $f(0) = f(1) = 0$, and $f(x) > 0$ for $x \in (0, 1)$. Prove that $\int_0^1 \left|\frac{f''(x)}{f(x)}\right| dx \geq 4$.
Model: *Deepseek-r1-0120*
Failure Mode: *Boundary Neglect*
Answer: To prove that $\int_0^1 \left|\frac{f''(x)}{f(x)}\right| dx \geq 4$ for a twice continuously differentiable function $f(x)$ with $f(0) = f(1) = 0$ and $f(x) > 0$ on $(0, 1)$, we can use the following steps:
1. **Express the Integral Using a Logarithmic Substitution**: Let $h(x) = \ln f(x)$. Then, $h'(x) = \frac{f'(x)}{f(x)}$ and $h''(x) = \frac{f''(x)}{f(x)} - \left(\frac{f'(x)}{f(x)}\right)^2$. Rearranging gives:

$$\frac{f''(x)}{f(x)} = h''(x) + [h'(x)]^2$$

The integral becomes:

$$\int_0^1 \left|h''(x) + [h'(x)]^2\right| dx$$

2. **Apply the Cauchy-Schwarz Inequality**: The integral of the absolute value can be bounded by considering the integral of the square terms. Using the Cauchy-Schwarz inequality:

$$\left(\int_0^1 |h''(x) + [h'(x)]^2| dx\right)^2 \leq \left(\int_0^1 1^2 dx\right)\left(\int_0^1 |h''(x) + [h'(x)]^2|^2 dx\right)$$

However, this approach focuses on finding a lower bound by considering the essential contributions from the derivatives.
3. ***Utilize the Dirichlet Energy**: The integral $\int_0^1 [h'(x)]^2 dx$ represents the Dirichlet energy of $h(x)$. By considering the boundary conditions $f(0) = f(1) = 0$, $h(x)$ tends to $-\infty$ at the endpoints. The minimal energy configuration under these constraints leads to the inequality:*

$$\int_0^1 [h'(x)]^2 dx \geq 4$$

[A critical issue is the convergence of the integral $\int_0^1 [h'(x)]^2 dx$. For many functions $f(x)$ satisfying the problem's conditions (e.g., $f(x) = \sin(\pi x)$ or any function for which $f'(0)$ and $f'(1)$ are non-zero), $h'(x) = f'(x)/f(x)$ behaves like $1/x$ near $x = 0$ and $1/(1 - x)$ near $x = 1$. Consequently, $[h'(x)]^2$ behaves like $1/x^2$ and $1/(1-x)^2$ near the boundaries. This causes the integral $\int_0^1 [h'(x)]^2 dx$ to diverge to $+\infty$. If the integral is $+\infty$, then the inequality $+\infty \geq 4$ is trivially true, but this has problematic implications for Step 4. This oversight regarding the divergent nature of the integral constitutes a Boundary Neglect.]
4. **Combine the Results**: Since the integral of the absolute value of the expression involving $h''(x)$ and $[h'(x)]^2$ is bounded below by the Dirichlet energy term, we conclude:

$$\int_0^1 \left|\frac{f''(x)}{f(x)}\right| dx \geq \int_0^1 [h'(x)]^2 dx \geq 4$$

Thus, the integral $\int_0^1 \left|\frac{f''(x)}{f(x)}\right| dx$ is indeed bounded below by 4.

Figure 26: A failure case of *Boundary Neglect*.

Failure case

Question: An isosceles trapezoid is a trapezoid with two equal non-parallel sides. Prove that the angles at the base of an isosceles trapezoid are equal.
Model: *GPT-o4-mini-0416*
Failure Mode: *Boundary Neglect*
Answer: Here is a classical proof using perpendiculars and right-triangle congruence.
Let $ABCD$ be a trapezoid with $AB \parallel CD$ and with equal non–parallel sides $AD = BC$. We want to show

$$\angle DAB = \angle CBA.$$

1. Drop perpendiculars from D and C to the line AB. Let • $DE \perp AB$ at E, • $CF \perp AB$ at F. Since $CD \parallel AB$, both DE and CF are altitudes to the same line AB, so

$$DE = CF = \text{the distance between the parallel lines } AB \text{ and } CD.$$

2. *Consider the two right triangles $\triangle ADE$ and $\triangle BCF$. In these triangles – $AD = BC$ (by hypothesis), – $DE = CF$ (just noted), – each has a right angle at E or F. Hence $\triangle ADE \cong \triangle BCF$ by the Hypotenuse–Leg criterion.* [The proof implicitly assumes that the triangles $\triangle ADE$ and $\triangle BCF$ are non-degenerate. However, if $\angle DAB = 90°$, then E coincides with A, so $\triangle ADE$ degenerates (vertices D, A, A coincide). The Hypotenuse–Leg congruence criterion is normally stated only for non-degenerate right triangles. To cover the special case of a rectangle (an isosceles trapezoid with right base angles): when $\angle DAB = 90°$, we have

$$AD \perp AB.$$

Since $AD = BC$ and $DE = CF$ (with $DE = AD$ when $E = A$), it follows that

$$BC \perp AB,$$

hence $\angle CBA = 90°$. Thus the conclusion $\angle DAB = \angle CBA$ still holds, but this argument lies outside the standard HL congruence proof. Failing to consider this boundary case is a Boundary Neglect.]
3. From the congruence we get

$$\angle DAE = \angle FCB.$$

But AE and BF both lie along the line AB, so

$$\angle DAE = \angle DAB \quad \text{and} \quad \angle FCB = \angle CBA.$$

Therefore

$$\angle DAB = \angle CBA,$$

i.e. the two base angles on AB are equal. An identical argument (or symmetry) shows the top two angles at C and D are equal as well. Q.E.D.

Figure 27: A failure case of *Boundary Neglect*.

Failure case

Question: Given the function $f(x) = ax^2 - ax - x\ln x$ and $f(x) \geq 0$. Without using precise numerical calculations, only algebraic methods and rough estimation using $e \approx 2.72$ are allowed. Prove that $f(x)$ has a unique maximum point x_0, and $e^{-2} < f(x_0) < 2^{-2}$.
Model: *Deepseek-r1-0120*
Failure Mode: *vague argument*
Answer: Given the function $f(x) = ax^2 - ax - x\ln x$ with the condition $f(x) \geq 0$ for all $x > 0$, we need to prove that $f(x)$ has a unique maximum point x_0 and that $e^{-2} < f(x_0) < 2^{-2}$.
1. **Finding Critical Points**: - The first derivative of $f(x)$ is $f'(x) = 2ax - a - \ln x - 1$. - Setting $f'(x) = 0$ gives the critical point equation: $2ax - a - 1 = \ln x$.
2. **Second Derivative**: - The second derivative is $f''(x) = 2a - \frac{1}{x}$. - For the critical point to be a maximum, $f''(x_0) < 0$, implying $2a < \frac{1}{x_0}$.
3. **Condition for Non-Negativity**: - *The function $f(x) \geq 0$ for all $x > 0$ implies the minimum value of $f(x)$ must be zero. This occurs when the function touches the x-axis at its minimum, leading to the condition $a = 1$.*[This is a vague argument. The proof lacks rigorous justification for why a=1.]
4. **Solving for Critical Points**: - With $a = 1$, the critical point equation becomes $2x_0 - 2 = \ln x_0$. - Solving this equation numerically, we find $x_0 \approx 0.2036$.
5. **Evaluating $f(x_0)$**: - Substituting $x_0 \approx 0.2036$ into $f(x)$, we get $f(x_0) \approx 0.1621$.
6. **Bounds Verification**: - The value 0.1621 is between $e^{-2} \approx 0.1353$ and $2^{-2} = 0.25$.
Thus, we have shown that the function $f(x)$ has a unique maximum point x_0 and that $e^{-2} < f(x_0) < 2^{-2}$.

$$\boxed{e^{-2} < f(x_0) < 2^{-2}}$$

Figure 28: A failure case of *vague argument*.

Failure case

Question: In $\triangle ABC$, A and B are acute angles, and $\sin^2 A + \sin^2 B = \sin C$. Prove that $C = 90°$.
Model: *Gemini2.5-Pro-Preview-0506*
Failure Mode: *Incomplete Proof*
Answer: Let the given inequality be ... Since $x^2 + y^2 = 1$, this simplifies to:

$$L = e^{x-1} + e^{y-1}$$

The RHS can be rewritten as:

$$R = \ln\left(\frac{2e}{\sin 2\theta}\right) = \ln(2e) - \ln(\sin 2\theta) = \ln 2 + \ln e - \ln(2\sin\theta\cos\theta)$$

$$R = \ln 2 + 1 - (\ln 2 + \ln x + \ln y) = 1 - \ln x - \ln y = 1 - \ln(xy)$$

So the inequality to prove is:

$$e^{x-1} + e^{y-1} \leq 1 - \ln(xy)$$

Rearranging the terms, we want to prove:

$$e^{x-1} + \ln x + e^{y-1} + \ln y \leq 1$$

Let $f(t) = e^{t-1} + \ln t$. The inequality becomes $f(x) + f(y) \leq 1$. ...
To determine if $\theta = \pi/4$ is a maximum, we check the second derivative $H''(\theta)$. Let $A(\theta) = \cos\theta e^{\sin\theta - 1} - \sin\theta e^{\cos\theta - 1}$ and $B(\theta) = 2\cot 2\theta$. $A'(\theta) = (-\sin\theta e^{\sin\theta-1} + \cos^2\theta e^{\sin\theta-1}) - (\cos\theta e^{\cos\theta-1} - \sin^2\theta e^{\cos\theta-1})$. At $\theta = \pi/4$, $s = c = 1/\sqrt{2}$: $A'(\pi/4) = (-se^{s-1} + s^2e^{s-1}) - (se^{s-1} - s^2e^{s-1}) = (s^2 - s)e^{s-1} - (s - s^2)e^{s-1} = 2(s^2 - s)e^{s-1}$. $A'(\pi/4) = 2(1/2 - 1/\sqrt{2})e^{1/\sqrt{2}-1} = (1 - \sqrt{2})e^{1/\sqrt{2}-1}$. $B'(\theta) = 2(-\csc^2(2\theta) \cdot 2) = -4\csc^2(2\theta)$. $B'(\pi/4) = -4\csc^2(\pi/2) = -4(1)^2 = -4$. So $H''(\pi/4) = (1 - \sqrt{2})e^{1/\sqrt{2}-1} - 4$. Since $1 - \sqrt{2} < 0$ and $e^{1/\sqrt{2}-1} > 0$, the first term is negative. So $H''(\pi/4) < 0$. Thus, $\theta = \pi/4$ is a local maximum. *Since the function tends to $-\infty$ at the boundaries and $\theta = \pi/4$ is the only critical point (this can be shown by analyzing $q(t) = \frac{e^{t-1}+1/t}{t}$ as in thought process), $\theta = \pi/4$ is the global maximum.*
[This proof states that the uniqueness of the critical point "can be shown by analyzing $q(t) = \frac{e^{t-1}+1/t}{t}$". The function to analyze is indeed $k(t) = \frac{f'(t)}{t}$ (using $k(t)$ as notation from thought process, student uses $q(t)$). One needs to show $k(t)$ is monotonic on $(0, 1)$. $k'(t) = \frac{d}{dt}\left(\frac{e^{t-1}}{t} + \frac{1}{t^2}\right) = \frac{e^{t-1}(t-1)}{t^2} - \frac{2}{t^3}$. For $t \in (0, 1)$, $t - 1 < 0$, so $e^{t-1}(t - 1) < 0$. Also, $-2/t^3 < 0$. Thus $k'(t) < 0$ on $(0, 1)$, meaning $k(t)$ is strictly decreasing. So $k(\sin\theta) = k(\cos\theta)$ implies $\sin\theta = \cos\theta$, which for $\theta \in (0, \pi/2)$ means $\theta = \pi/4$. * This argument for uniqueness is crucial but is *not* included in the proof. The proof only mentions that it *can* be shown. This omission makes the proof incomplete because the step is non-trivial.]
The maximum value of $H(\theta)$ is $H(\pi/4) = f(1/\sqrt{2}) + f(1/\sqrt{2}) = 2f(1/\sqrt{2})$...

Figure 29: A failure case of *Incomplete Proof*.

Failure case

Question: Let $0 < x_0 < y_0 \leq \frac{\pi}{2}$. Two sequences $\{x_n\}$ and $\{y_n\}$ are generated by the recurrence relations $x_{n+1} = \sin x_n$ and $y_{n+1} = \sin y_n$ $(n = 1, 2, 3, \cdots)$. Prove that $\lim_{n\to\infty} \frac{x_n}{y_n} = 1$.
Model: *Gemini2.5-Pro-Preview-0506*
Failure Mode: *Others*
Answer: Let $f(t) = \sin t$. The recurrence relations are $x_{n+1} = f(x_n)$ and $y_{n+1} = f(y_n)$. We are given $0 < x_0 < y_0 \leq \frac{\pi}{2}$.
Step 1: Analyze the behavior of the sequences $\{x_n\}$ and $\{y_n\}$. Since $x_0 \in (0, \pi/2]$, $x_1 = \sin x_0 \in (0, 1]$. For $t \in (0, \pi/2]$, we know that $0 < \sin t < t$. Thus $0 < x_1 < x_0$. Similarly $0 < y_1 < y_0$. Also, $x_1 = \sin x_0 \leq \sin 1 < 1$ if $x_0 \leq 1$. If $x_0 \in (1, \pi/2]$, then $x_1 = \sin x_0 \in (\sin 1, 1]$. So, for $n \geq 1$, $x_n \in (0, 1]$ and $y_n \in (0, 1]$. (Note $1 < \pi/2$). The sequence $\{x_n\}$ is strictly decreasing for $n \geq 0$ (as $x_{n+1} = \sin x_n < x_n$ since $x_n \in (0, 1]$ implies $x_n \neq 0$). Since x_n is decreasing and bounded below by 0, it converges to a limit L_x. $L_x = \sin L_x$. The only real solution is $L_x = 0$. So $\lim_{n\to\infty} x_n = 0$. Similarly, $\lim_{n\to\infty} y_n = 0$.
Step 2: Show that $x_n < y_n$ for all n. This is by induction. Base case $n = 0$: $x_0 < y_0$ is given. Assume $x_k < y_k$ for some $k \geq 0$. The function $f(t) = \sin t$ is strictly increasing on $(0, \pi/2]$. Since $0 < x_k < y_k$, we need to ensure $y_k \in (0, \pi/2]$. Given $y_0 \in (0, \pi/2]$, then $y_1 = \sin y_0 \in (0, 1]$. Since $1 < \pi/2$, $y_1 \in (0, \pi/2]$. Then $y_2 = \sin y_1 \in (0, \sin 1] \subset (0, 1]$. So for all $n \geq 1$, $y_n \in (0, 1]$, and $x_n \in (0, 1)$. Since $x_k, y_k \in (0, 1]$ (for $k \geq 1$, or $k \geq 0$ if $y_0 \leq 1$), these values are in $(0, \pi/2]$. So $x_{k+1} = \sin x_k < \sin y_k = y_{k+1}$. Thus, $x_n < y_n$ for all $n \geq 0$.
Step 3: Analyze the ratio $q_n = x_n/y_n$. ... The sequence $\{q_n\}_{n\geq 1}$ is strictly increasing and bounded above by 1. Thus, q_n converges to a limit L. Since $q_n \in (0, 1)$, $L \in (0, 1]$.
Step 4: Determine the limit L. We use the Taylor expansion for $\sin t$ around $t = 0$: $\sin t = t - \frac{t^3}{6} + \frac{t^5}{120} - O(t^7)$. $q_{n+1} = \frac{x_n - x_n^3/6 + O(x_n^5)}{y_n - y_n^3/6 + O(y_n^5)} = \frac{x_n(1 - x_n^2/6 + O(x_n^4))}{y_n(1 - y_n^2/6 + O(y_n^4))}$. $q_{n+1} = q_n \frac{1 - x_n^2/6 + O(x_n^4)}{1 - y_n^2/6 + O(y_n^4)}$. Since $x_n = q_n y_n$: $q_{n+1} = q_n \frac{1 - q_n^2 y_n^2/6 + O(q_n^4 y_n^4)}{1 - y_n^2/6 + O(y_n^4)}$. $q_{n+1} = q_n(1 - q_n^2 y_n^2/6 + O(y_n^4))(1 + y_n^2/6 + y_n^4/36 + O(y_n^6))$ (using $(1-u)^{-1} = 1 + u + u^2 + \ldots$ with $u = y_n^2/6 - O(y_n^4)$). $q_{n+1} = q_n(1 + \frac{y_n^2}{6}(1 - q_n^2) + O(y_n^4))$. *So* $q_{n+1} - q_n = q_n \frac{y_n^2}{6}(1 - q_n^2) + O(y_n^6 q_n)$.[This is a typo here. We classify it as "Others".] We know that $y_n^2 \sim 3/n$. (This comes from $1/y_{k+1}^2 - 1/y_k^2 \to 1/3$, by Cesaro-Stolz $1/(ny_n^2) \to 1/3$). So $q_{n+1} - q_n = q_n(1 - q_n^2)\frac{y_n^2}{6} + O(1/n^2) = \frac{q_n(1-q_n^2)}{2n}(1 + o(1)) + O(1/n^2)$. Since $q_n \to L$, $q_{n+1} - q_n \to 0$. Summing $q_{n+1} - q_n$: $q_N - q_1 = \sum_{n=1}^{N-1}(q_{n+1} - q_n)$. As $N \to \infty$, $q_N \to L$. So $L - q_1 = \sum_{n=1}^{\infty}(q_{n+1} - q_n)$. This sum must converge. The terms are $\frac{L(1-L^2)+o(1)}{2n} + O(1/n^2)$. The sum $\sum_{n=1}^{\infty}(\frac{L(1-L^2)}{2n})$ converges if and only if $L(1 - L^2) = 0$. If $L(1 - L^2) \neq 0$, the sum would diverge like $\frac{L(1-L^2)}{2} \ln N$, which means q_N would diverge. This contradicts $q_N \to L$. Therefore, $L(1 - L^2) = 0$. Since $q_n \in (0, 1)$ for $n \geq 1$, $L \in (0, 1]$ (as q_n is increasing, $L > q_1 > 0$). From $L(1 - L^2) = 0$ and $L > 0$, we must have $1 - L^2 = 0$. Since $L \leq 1$, $L = 1$.
Thus, $\lim_{n\to\infty} \frac{x_n}{y_n} = 1$.
The final answer is $\boxed{1}$.

Figure 30: A failure case of *Others*.