MATP-BENCH: Can MLLM Be a Good Automated Theorem Prover for Multimodal Problems?

Anonymous ACL submission

Abstract

Theorem proving in fields like geometry often relies on visual reasoning with combined text and diagrams. While Multimodal Large Language Models (MLLMs) have shown potential in mathematics, their application in multimodal automated theorem proving remains a largely unexplored area. In this paper, we introduce the **M**ultimodal **A**utomated **T**heorem **P**roving benchmark (**MATP-BENCH**), a novel Multimodal, Multilevel, and Multilanguage benchmark designed to evaluate MLLMs in this role as multimodal automated theorem provers. MATP-BENCH consists of 1056 multimodal theorems drawn from high school, university, and competition-level mathematics. All these multimodal problems are accompanied by formalizations in Lean 4, Coq and Isabelle, thus making the benchmark compatible with a wide range of theorem-proving frameworks. MATP-BENCH requires models to integrate sophisticated visual understanding with mastery of a broad spectrum of mathematical knowledge and rigorous symbolic reasoning to generate formal proofs. Our evaluation of advanced MLLMs reveals that existing methods solve only a limited number of problems, establishing MATP-BENCH as a significant open challenge for the future of automated theorem proving.[1]

1 Introduction

Recent advancements in large language models (LLMs) have driven their application in automated theorem proving (ATP) (Zheng et al., 2021; Yang et al., 2023b; Azerbayev et al., 2023a; Wang et al., 2024c; Xin et al., 2024; Wang et al., 2025c; Lin et al., 2025; Wang et al., 2025a), where LLMs are trained to generate formal proofs that are verified for logical soundness by proof assistants like Lean 4 (Moura and Ullrich, 2021), Isabelle (Wenzel et al., 2008), and Coq (Chlipala, 2013). However, despite substantial progress has been made in automated theorem proving, existing studies remain largely confined to text-based inputs (Zheng et al., 2021; Azerbayev et al., 2023a; Tsoukalas et al., 2024; Yu et al., 2025), leaving the potential of multimodal theorem proving unexplored. This limitation is particularly acute in domains like geometry, where visual elements such as diagrams are often indispensable for understanding and reasoning, purely textual descriptions are frequently insufficient. While recent works touch upon this area, they do not fill the specific gap of autonomous multimodal proving. For instance, LeanEuclid (Murphy et al., 2024) focuses on auto-formalization—translating existing human-written proofs into Lean 4, rather than on autonomous proof generation. Similarly, AlphaGeometry (Seo et al., 2015) can solve Olympiad geometry problems, but it operates on symbolic inputs, not visual diagrams, and its solutions are not verifiable in proof assistants such as Lean 4.

To this end, we present **M**ultimodal **A**utomated **T**heorem **P**roving benchmark (**MATP-BENCH**), a multimodal, multilevel, and multilanguage benchmark for multimodal automated theorem proving (MATP). Our benchmark consists of 1056 multimodal theorems drawn from high school, university, and competition-level mathematics. Each theorem is accompanied by formal theorems in Lean 4 (Moura and Ullrich, 2021), Isabelle (Wenzel et al., 2008), and Coq (Chlipala, 2013), making the benchmark compatible with a wide range of theorem-proving frameworks. As shown in Figure 1, each data sample in MATP-BENCH consists of an image, a natural language theorem statement, and formal theorem statements in three different languages. Compared to traditional ATP, MATP requires the integration of reasoning across both language and visual modalities. This is particularly crucial in geometry-related mathematical problems, which often rely on mathematical structures conveyed through images, such as topological relations, that are typically difficult to express precisely

[1] Our code will be released publicly.

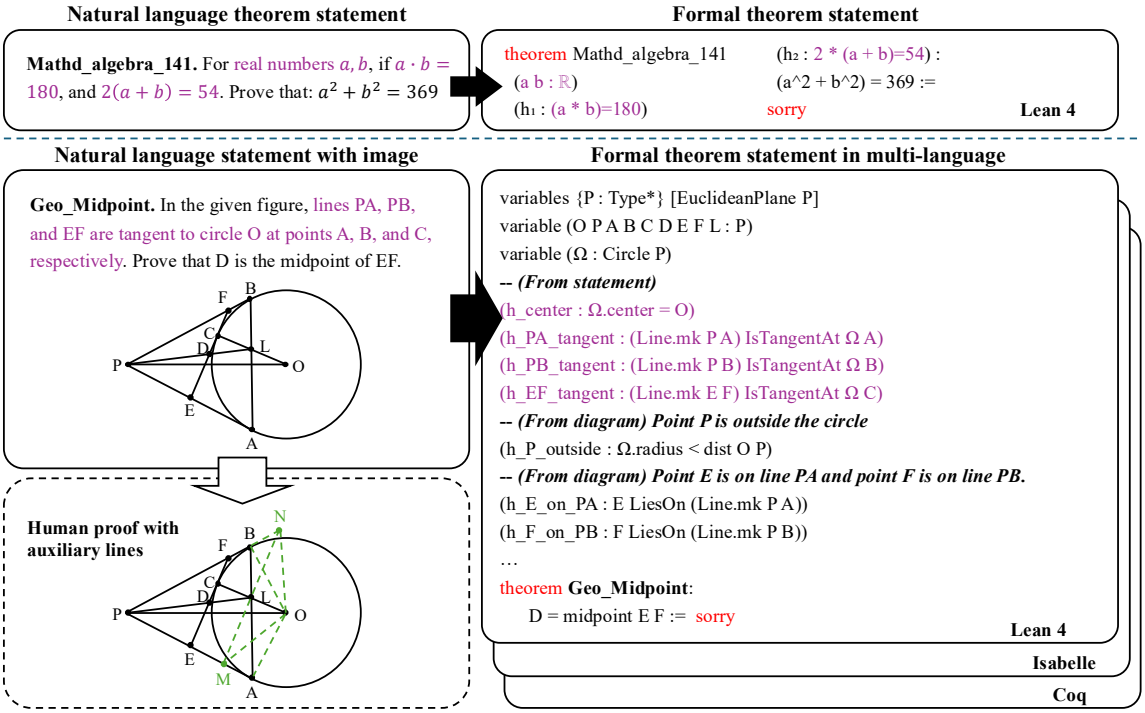

Figure 1: In traditional ATP, such as the miniF2F benchmark (top), theorem formalization relies solely on text statements, whereas our MATP task setting (bottom) requires model to also extract critical premises not explicitly expressed in the text by analyzing accompanying diagrams (see ***From diagram*** on the right), as inspired by human cognition. Purple indicates premises derived from the original statement, and we provide formalized versions of all multimodal theorems in Lean4, Coq, and Isabelle.

in natural language. For humans, the reasoning process in such tasks involves not only drawing intuitive insights from visual representations but also constructing auxiliary diagrams and identifying implicit structural relations.

We use MATP-BENCH to evaluate a variety of multimodal language models. Our findings indicate that even current state-of-the-art models can only solve a limited number of problems, particularly when generating proofs in the Lean 4 language, where they perform poorly even for problems of only high school difficulty. Detailed analysis further reveals that: (1) All evaluated models commonly exhibit issues with incomplete understanding of problem information and the generation of invalid formal proof steps, indicating deficiencies in their visual-symbolic joint reasoning capabilities; (2) While models demonstrate some ability in converting natural language questions and geometric figures into formal statements, they still face significant challenges in subsequent complex logical reasoning and the construction of correct formal proofs; (3) Models can introduce steps or concepts involving auxiliary constructions (e.g., auxiliary lines) in proofs, but they fail to effectively utilize these constructions to substantively advance the proof process. Our contributions are as follows:

- We introduce the Multimodal Automated Theorem Proving benchmark (MATP-BENCH), which contains 1056 multimodal theorems drawn from high school, university, and competition-level mathematics. Each problem is accompanied by formalizations in Lean 4, Coq and Isabelle.
- We conduct extensive experiments on MATP-BENCH with six advanced multimodal language models of varying sizes. The results show that even current state-of-the-art models can only solve a limited number of problems, particularly when generating proofs in Lean 4.
- Our detailed analysis indicates that the primary bottleneck in MATP task lies in symbolic reasoning and the construction of correct formal proofs. Although models have demonstrated an initial ability to introduce auxiliary constructions (e.g., auxiliary lines), they fail to utilize these constructions to effectively advance the proof process.

2 Related Work

Multimodal Math Benchmarks. Various benchmarks have been created to assess the mathematical reasoning capabilities of LLMs (Amini et al., 2019; Cobbe et al., 2021; Mishra et al., 2022; Frieder et al., 2023; Hendrycks et al., 2020, 2021; Ling et al., 2017), with a growing number of specialized evaluations for MLLMs (Lu et al., 2021; Masry et al., 2022; Lu et al., 2023; Wang et al., 2024b).

Benchmark	Size	Verifiable	Theorem Proving	Theorem formalization	Multi-modal	Multi-level	Lean	Isabelle	Coq
miniF2F (Zheng et al., 2021)	488	✓	✓			✓	✓	✓	
ProofNet (Azerbayev et al., 2023a)	371	✓	✓				✓		
Fimo (Liu et al., 2023a)	149	✓	✓				✓		
Geometry3K-test (Lu et al., 2021)	601	✓			✓				
LeanEuclid (Murphy et al., 2024)	173	✓		✓	✓	✓	✓		
PutnamBench (Tsoukalas et al., 2024)	640	✓	✓				✓	✓	✓
AlphaGeometry-test (Trinh et al., 2024)	30	✓	✓		✓				
ProverBench (Ren et al., 2025)	325	✓	✓				✓		
GeoTrust-test (Fu et al., 2025)	240	✓			✓	✓			
MATP-Bench (ours)	1056	✓	✓	✓	✓	✓	✓	✓	✓

Table 1: Comparison of existing related benchmarks. MATP-BENCH is a **Multimodal, Multilevel, and Multilanguage** benchmark designed to evaluate MLLMs as automated theorem provers.

GeoQA+ (Cao and Xiao, 2022), UniGeo (Chen et al., 2022), GEOS (Seo et al., 2015), and Geometry3K (Lu et al., 2021) provide standardized benchmarks focused on plane geometry problem-solving. MATHVERSE (Zhang et al., 2024) is a visual math benchmark with 2,612 core problems, generating 15K test samples across plane geometry, solid geometry, and functions. MATHVISTA (Lu et al., 2023) includes 6,141 examples to assess mathematical and visual reasoning across diverse tasks. MATH-Vision (Wang et al., 2024b) consists of 3,040 carefully selected problems with visual contexts, drawn from 19 real-world math competitions and spanning 12 grade levels. MV-MATH (Wang et al., 2025b) is a multimodal benchmark featuring 2,009 multi-image questions, categorized into three question types. Although significant progress has been made, the exploration of automated theorem proving in multimodal settings remains limited. To address this gap, we propose MATP-BENCH, a benchmark that evaluates the ability of MLLMs to integrate visual perception, mathematical reasoning, and symbolic manipulation to construct rigorous formal proofs.

Automated Theorem Proving. Automated theorem proving (ATP) has been a long-standing challenge in symbolic reasoning (Robinson and Voronkov, 2001; Bibel, 2013; Zheng et al., 2021; Liu et al., 2023a), with substantial progress made in developing automated theorem provers (Polu and Sutskever, 2020; Polu et al., 2022; Lample et al., 2022; Thakur et al., 2023; Azerbayev et al., 2023b; Jiang et al., 2022; Wang et al., 2024a, 2025d). In recent years, multiple benchmarks (Jiang et al., 2021; Ying et al., 2024; Lin et al., 2025; Yu et al., 2025) have been proposed for formal mathematical proof. MINIF2F (Zheng et al., 2021) features a diverse collection of 488 problems, each formalized in mainstream languages such as Lean 3 and Isabelle. ProofNet (Azerbayev et al., 2023a) comprises 371 parallel formal and natural language theorem statements with proofs, designed to evaluate autoformalization and theorem proving. PutnamBench (Tsoukalas et al., 2024) is a multilanguage benchmark for theorem provers, featuring 1,692 formalizations of 640 Putnam Competition problems. We push the boundaries of formal verification by exploring richer multimodal automated theorem proving with multilanguage formalization proofs. The comparison between related benchmarks and MATP-BENCH is shown in Table 1.

3 Problem Formulation

Automated Theorem Proving (ATP). In the ATP task (Zheng et al., 2021; Azerbayev et al., 2023a; Tsoukalas et al., 2024; Liu et al., 2023a), the system takes a ***formalized theorem statement*** FT as input, as shown in the *upper part* of Figure 1. The goal is to generate a formal proof FP:

$$\texttt{Prover}_{ATP}(FT) \rightarrow FP \tag{1}$$

Then the proof FT of theorem FP is checked for correctness by proof assistants such as Lean 4.

Multimodal Automated Theorem Proving (MATP). In the MATP task, the input to the MATP system is a pair (MI, NT), where:

- MI is the ***multimodal input***;
- NT is the ***natural language theorem***.

As shown in the *bottom part* of Figure 1, the natural language theorem NT and the information from the multimodal input MI are complementary, together forming a complete theorem. Hence, the

model must first generate the complete formal theorem FT. Then, generates a valid formal proof FP. The entire MATP task can be summarized as:

$$\texttt{Prover}_{MATP}(MI, NT) \rightarrow (FT, FP) \quad (2)$$

Then the theorem FT and the proof FP are verified by proof assistants.

Preventing Modality Leakage. To mitigate modality leakage by the proofer model, the MATP task provides the raw natural language theorem NT as input, instead of the formalized theorem. This encourages the model to interpret and reason over multimodal inputs, as humans do when proving multimodal theorems. Furthermore, we incorporate a formal theorem verification task in the experimental setup to ensure that the formal theorems automatically generated by the multimodal model are consistent with the problem statement, rather than fabricating simple theorems arbitrarily.

4 MATP-BENCH

MATP-BENCH is a new multimodal, multilevel, and multilanguage benchmark designed to evaluate MLLMs as automated theorem provers, which consists of 1056 multimodal theorems drawn from high school, university, and competition-level mathematics. Each theorem is accompanied by formalization in multiple formal languages, making the benchmark compatible with diverse theorem-proving frameworks.

Multimodal Context and Multilanguage Theorem. Compared to previous automated theorem proving datasets, which typically contain only plain-text theorems, MATP-BENCH introduces multimodal context to jointly evaluate models on visual understanding, mathematical reasoning, and symbolic manipulation. As shown in the lower part of Figure 1, each theorem consists of an image and a corresponding natural language description. Our benchmark provides formalizations of these multimodal theorems in Lean 4, Isabelle, and Coq. To the best of our knowledge, MATP-BENCH is the first multimodal automated theorem proving benchmark covering all three of these languages. Our benchmark presents the following challenges: (1) ***Visual Understanding***: The model must accurately extract key information from theorem-related images, akin to human perception, to construct formal theorem statements; (2) ***Mathematical Reasoning***: It requires rigorous mathematical reasoning to derive complete proofs based on the provided natural language descriptions and images; (3) ***Neural-Symbol Proof Generation***: The model must be proficient in these formal languages and capable of strictly translating the mathematical reasoning process into verifiable formal proof.

	Category	Count	Percentage
Level	High School	472	44.7%
	College	468	44.3%
	Competition	116	11.0%
Type	Plane Geometry	937	88.7%
	3D Geometry	73	6.9%
	Analytic Geometry	46	4.4%
Topic	Segment Relationships	355	33.6%
	Angle Relationships	282	26.7%
	Area Relationships	222	21.0%
	Circles and Tangents	86	8.1%
	Parallel and Perpendicular Lines	38	3.6%
	Similarity and Proportionality	25	2.4%
	Cyclic Quads & Common Points	20	1.9%
	Other	28	2.7%

Table 2: Statistics summary of MATP-BENCH. Counts and percentages are provided for each category.

Hierarchy and Diversity. Existing widely used benchmarks such as ProofNet (Azerbayev et al., 2023a) mainly focus on basic mathematical problems from early undergraduate courses, FIMO (Liu et al., 2023a) is limited to high school mathematics, while PutnamBench (Tsoukalas et al., 2024) targets advanced mathematical reasoning at the undergraduate level. In contrast, as shown in Table 2, the problems in MATP-BENCH span three distinct educational stages, high school, university, and competitions, systematically covering a wide range of difficulty levels from elementary to advanced. Specifically, the high school and university problems are collected from publicly available multimodal math problem datasets (Lu et al., 2023; Wang et al., 2024b, 2025b; Lu et al., 2021), while the competition problems are sourced from open mathematical competitions at the high school and university levels in China. We also manually annotate the formal statements of each problem in three formal languages, to enable a thorough examination of models' reasoning capabilities across different levels of complexity and mathematics problems. Furthermore, the multimodal theorems in MATP-BENCH are primarily centered around the domains of geometry, such as plane geometry, 3D geometry, and analytic geometry. These theorems not only require models to demonstrate a solid grasp of fundamental geometric knowledge, but also emphasize cross-modal understanding, complex reasoning, spatial modeling, and multi-step logical deduction, aiming to systematically evaluate models' overall performance and depth of reasoning in

structured mathematical tasks.

Task Formulation. As mentioned in Section 3, we aim to achieve end-to-end multimodal automated theorem proving (Task 1), where the input is a natural language theorem and an image, and the output is a formal theorem and its proof, i.e. $\texttt{Prover}_{\text{MATP}}(MI, NT) \rightarrow (FT, FP)$. To prevent the model from generating formal theorems that do not align with original problems, we separately set up multimodal theorem formalization (Task 2) for verification, which aligns with LeanEuclid (Murphy et al., 2024). Thus, we divide the task into two progressively challenging sub-tasks:

- **Task 1: Multimodal Automated Theorem Proving**: This task aims to achieve end-to-end multimodal automated theorem proving similar to human provers, i.e. $\texttt{Prover}_{\text{Task1}}(MI, NT) \rightarrow (FT, FP)$. For example, the input is a natural language statement with an image, as shown in Figure 1. The expected output is a formal theorem and a formalized valid proof. This presents a significant challenge as the model must first accurately formalize the theorem, and subsequently construct a valid proof.
- **Task 2: Multimodal Theorem Formalization**: To prevent the model from generating formal theorems that do not align with original problems, the model is required to formalize original natural language problems into a precise theorem T, formally denoted as $\texttt{Prover}_{\text{Task2}}(MI, NT) \rightarrow FT$. This task evaluates model capability to correctly understand and formalize information from both text and visual modalities.

Formalization Effort and Challenges. Our formalization team consists of two doctoral students and several undergraduate students with backgrounds in advanced mathematics, computer science, and prior experience with formal proof assistants. The problems cover multiple question types that we manually formalized case by case. Specifically: (1) Multiple-choice questions: by extracting key information from the image and incorporating the correct answer, the problem is transformed into a concrete theorem; (2) Fill-in-the-blank questions: the correct answer is directly inserted into the problem statement, and the theorem is formalized by integrating visual information; (3) Open-ended questions: interrogative forms are converted into declarative statements based on the given answer, combined with image cues to construct a complete formal theorem. On average, fully formalizing a high school problem, a university problem, and a competition problem takes approximately 25, 30, and 18 minutes respectively (in one language). Among these, the descriptions of high school and university problems are relatively concise, with rich information contained in the images requiring preprocessing, while competition problems are more detailed and thus easier to formalize. Each formalization is reviewed by at least one other team member. In contrast to prior formalization tasks based solely on textual theorems, a central challenge in our setting lies in the incompleteness of the original natural language descriptions. Many essential assumptions are conveyed exclusively through diagrams. As a result, the formalization process requires manual identification and extraction of visual information, such as geometric structures, to reconstruct a complete and rigorous formal statement.

5 Experiments

5.1 Experimental settings

Methods: we conduct extensive experiments on a wide variety of advanced multimodal large language models with different sizes. Specifically, the reasoning models (Li et al., 2025; Qu et al., 2025; Chen et al., 2025) include ***o1*** (OpenAI, 2024b), `Claude-3.7-Sonnet-Thinking` (***Claude-3.7***) (Anthropic, 2024), and `Gemini-2.0-Flash-Thinking` (***Gemini-2.0***) (Doshi, 2025); while the non-reasoning models include ***GPT-4.1*** (OpenAI, 2024a), `Qwen2.5-VL-Instruct-70B` (***Qwen2.5VL***) (Team, 2025), and `Llama3.2-Vision-Instruct-11B` (***Llama3.2V***) (Liu et al., 2023b).

Metrics: For **Task 1**, which requires the model to generate both a correct formal theorem and its proof, we follow prior studies (Zheng et al., 2021; Tsoukalas et al., 2024; Liu et al., 2023a) and adopt **pass@n** (n=1,5,10) as the evaluation metric. This metric evaluates whether the prover can successfully complete a valid proof within n attempts in the formal proof environment. For **Task 2**, which requires the model to formalize the multimodal theorem, we use GPT-4o as the judge to assess whether the formal theorem generated by the model is consistent with the our annotated ground truth. We also adopt **pass@n** (n=1,5,10) as the evaluation metric. Details of the prompts for different tasks and evaluation can be found in Appendix F.

	Task 1	Lean4				Coq				Isabelle			
		H.	U.	C.	Avg	H.	U.	C.	Avg	H.	U.	C.	Avg
pass@10	o1	7.63	**4.70**	**3.45**	**5.26**	**28.37**	11.73	5.45	**19.43**	**11.23**	**7.48**	0.86	**6.83**
	Claude-3.7	7.20	3.85	1.72	5.11	22.47	**12.39**	4.31	16.92	8.90	5.34	1.72	5.90
	Gemini-2.0	8.47	2.14	0.86	4.82	14.76	4.27	1.72	8.71	7.84	3.63	0.00	4.11
	GPT-4.1	**9.32**	2.99	2.45	4.92	28.25	6.62	**9.48**	16.64	10.48	4.49	**2.59**	6.39
	Llama3.2V	3.58	1.92	0.00	2.46	6.96	4.91	0.17	7.37	3.60	3.21	0.00	2.45
	Qwen2.5VL	2.12	1.50	0.00	1.61	7.59	5.34	0.86	3.59	4.03	2.78	0.00	2.27
pass@5	o1	4.03	2.78	**1.72**	2.84	18.78	6.84	**8.62**	**11.43**	**7.84**	**4.70**	0.86	**4.47**
	Gemini-2.0	3.39	1.71	0.86	2.27	16.24	**8.12**	1.72	11.08	5.30	3.42	1.72	3.48
	Claude-3.7	5.51	1.50	0.00	3.12	8.65	3.85	0.86	5.67	4.45	2.14	0.00	2.20
	GPT-4.1	**5.08**	**3.38**	0.86	**3.11**	**20.89**	4.27	1.72	8.96	6.62	3.82	**2.59**	4.35
	Llama3.2V	2.54	1.72	0.00	1.61	4.43	3.21	0.00	3.40	1.91	2.35	0.00	1.42
	Qwen2.5VL	1.48	0.86	0.00	1.04	3.58	2.56	0.00	2.65	2.54	1.92	0.00	1.49
pass@1	o1	2.75	1.50	0.00	1.89	**10.13**	**4.91**	5.42	**6.82**	4.24	**3.28**	0.00	2.51
	Gemini-2.0	2.54	0.85	**0.86**	1.52	6.54	3.63	0.00	4.54	3.60	2.14	**1.72**	1.91
	Claude-3.7	3.18	0.43	0.00	1.61	3.59	1.28	0.86	2.27	2.97	0.85	0.00	1.27
	GPT-4.1	**3.39**	**2.85**	0.00	**2.56**	6.96	2.56	**7.62**	5.71	**4.87**	2.14	0.86	**2.62**
	Llama3.2V	1.48	0.64	0.00	0.95	3.16	1.44	0.00	2.08	0.85	1.28	0.00	0.71
	Qwen2.5VL	0.85	0.43	0.00	0.57	1.48	1.07	0.00	1.13	1.27	1.07	0.00	0.78

Table 3: Experimental results of **Multimodal Automated Theorem Proving** (Task 1), which requires model to generate both formalized theorem and proof. We adopt **pass@n** (n=1,5,10) as the evaluation metric here. **H. U. C.** represents high school, university, and competition-level, respectively.

5.2 Main results

Lean 4. In the Lean4 experiments, GPT-4.1 and o1 demonstrate closely matched performance as the leading models. The highest score achieved is 9.32% by GPT-4.1 on high school problems under the pass@10 metric. A consistent trend is observed where all models' scores decrease as the problem difficulty rises from high school to university and competition levels. Competition-level problems prove most challenging, where models like Llama3.2V and Qwen2.5VL frequently hit the minimum score of zero. This highlights a clear performance gap, with the top model o1's average pass@10 score of 5.26% being over three times higher than the lowest-scoring model Qwen2.5VL's average of 1.61%. Gemini-2.0 also performs well, notably securing the best score on the most difficult competition problems at pass@1.

Coq. The Coq experiments are defined by o1's dominant performance, which achieves an exceptional score of 28.37% on high school problems at pass@10, the maximum score in the entire study. The performance trend shows a particularly steep drop in success rates as the difficulty level increases from foundational high school tasks to more complex university and competition problems. On these harder tasks, models like Llama3.2V hit the minimum score of zero. The performance disparity among models is vast, the leading model o1's average pass@10 score of 19.43% is over five times greater than that of the lowest-performing model Qwen2.5VL, which averages 3.59%.

Isabelle. Isabelle proves to be a significant challenge for all models, with generally lower success rates. The maximum score is 11.23%, achieved by o1 on high school problems at pass@10. While o1 and GPT-4.1 constitute the top tier of performers, a clear downward trend in performance is visible for all models as problem difficulty increases. The difficulty is most pronounced on competition-level problems, where most models score the minimum of zero, underscoring the extreme challenge of these tasks. The performance gap is still significant; at pass@10, the top-performing model o1's average score of 6.83% is three times higher than the lowest-scoring model Qwen2.5VL's 2.27%. More analysis is in Appendix B and C.

5.3 Error Analysis

We analyze the types of errors generated by multimodal models in theorem proving. As shown in Figure 6, the results indicate that different mod-

	Task 2	Lean4				Coq				Isabelle			
		H.	U.	C.	Avg	H.	U.	C.	Avg	H.	U.	C.	Avg
pass@10	o1	**53.12**	**61.28**	63.32	**59.24**	**42.65**	45.64	63.37	41.31	51.88	**63.50**	**63.08**	**60.14**
	Claude-3.7	55.07	60.42	61.64	57.26	39.07	**51.22**	**65.20**	**49.65**	**49.78**	62.22	58.28	56.21
	Gemini-2.0	44.02	56.11	59.40	51.05	27.15	30.00	54.17	31.76	35.79	54.86	42.59	44.97
	GPT-4.1	47.58	58.19	**65.27**	57.01	37.03	43.89	68.35	43.13	42.10	65.73	44.27	52.56
	Llama3.2V	15.50	21.81	28.58	19.72	11.79	16.39	31.38	15.97	17.42	23.89	19.61	20.52
	Qwen2.5VL	26.84	33.33	38.66	31.46	16.73	20.14	38.66	20.64	21.94	31.81	25.63	26.66
pass@5	o1	**49.64**	**60.14**	**60.52**	**56.77**	**38.95**	45.83	**63.58**	**49.45**	43.16	**61.67**	**57.72**	**52.39**
	Claude-3.7	47.58	58.89	58.84	53.82	31.40	**46.89**	60.52	39.91	42.37	60.83	54.91	50.91
	Gemini-2.0	39.36	52.36	55.47	46.88	23.45	25.83	53.23	27.77	30.44	48.61	37.54	39.26
	GPT-4.1	42.92	55.00	52.11	49.27	31.27	40.69	65.04	39.14	**45.52**	57.36	38.10	45.46
	Llama3.2V	13.58	21.27	25.90	20.29	9.87	14.31	30.26	14.07	15.63	21.67	16.81	18.43
	Qwen2.5VL	24.70	30.83	36.42	28.38	13.54	17.22	35.86	18.06	19.34	29.03	21.45	23.96
pass@1	o1	**33.05**	45.61	**43.15**	**40.24**	**27.66**	**35.49**	57.26	**34.62**	23.68	**51.19**	30.26	35.04
	Claude-3.7	30.72	**47.82**	39.78	38.21	18.92	25.28	47.63	24.88	**26.31**	49.86	31.70	**35.96**
	Gemini-2.0	24.14	35.14	36.98	30.41	18.10	12.92	34.18	17.57	15.77	27.36	21.85	21.56
	GPT-4.1	26.19	37.22	33.06	31.82	16.46	28.61	**58.72**	26.36	16.73	53.33	**37.34**	35.80
	Llama3.2V	10.68	13.80	19.05	11.67	8.68	5.36	12.61	8.35	8.78	13.47	8.95	10.87
	Qwen2.5VL	13.71	20.56	24.66	17.94	11.38	8.89	21.85	11.43	10.97	18.19	12.89	14.38

Table 4: Experimental results of **Multimodal Theorem Formalization** (Task 2), which only require model to generate formalized theorem. We use GPT-4o as the judge and adopt **pass@n** (n=1,5,10) as the evaluation metric. **H. U. C.** represents high school, university, and competition-level, respectively.

els exhibit both common errors and specific issues. For the Claude model, the most significant errors are concentrated in invalid/unproven proof steps (24.9%), missing preconditions/hidden assumptions (23.8%), and missing unformalized information (underutilizing multimodal information, 23.3%); these three categories collectively account for approximately 72% of its total errors. Similarly, the error distribution of the GPT-4.1 model is very similar. This suggests that even for relatively better-performing models, the core challenges lie in complex logical reasoning and identifying and utilizing all implicit and explicit information required by the theorem. The error distribution of the Qwen2.5 model, however, differs. While missing unformalized information (20.9%) and missing preconditions/hidden assumptions (18.7%) are also major errors, the proportion of invalid/unproven proof steps is relatively lower (15.6%), while more fundamental formalization errors such as missing/incorrect library imports (11.3%) and incorrect/undeclared variables/definitions (10.6%) are more prominent. This might indicate that while Qwen struggles with proof step errors, it also faces significant issues in generating basic code that conforms to the formal language specification. Overall, all models commonly exhibit problems with incomplete information understanding and broken reasoning chains.

5.4 Main Bottleneck in Multimodal Automated Theorem Proving

As shown in Figure 2, the analysis of the pass@n performance of multimodal models in the Coq language shows that for both complete theorem formalization and proof generation (Task 1) and formalization only (Task 2), allowing models to generate more candidates (from pass@1 to pass@10) generally increases the success rate. However, the pass@n success rate for Task 1 is significantly lower than the pass@n success rate for Task 2 across all difficulty levels and models. For example, at the high school difficulty level, the highest pass@10 for Task 1 reaches 28.27% for GPT-4.1, while the pass@10 for Task 2 for the same model reaches 37.03%. Notably, the pass@n values for Task 2, especially at the competition difficulty level, can even remain at a relatively high level (68.35% for GPT-4.1's pass@10). This may be attributed to the fact that the problem descriptions for competition-level theorems are often more complete and precise, providing clearer formalization basis for the models, even if the proof process heavily relies on complex steps like constructing auxiliary lines (which is precisely why the Task 1 success rate is extremely low at this difficulty). This large performance gap between Task 1 and

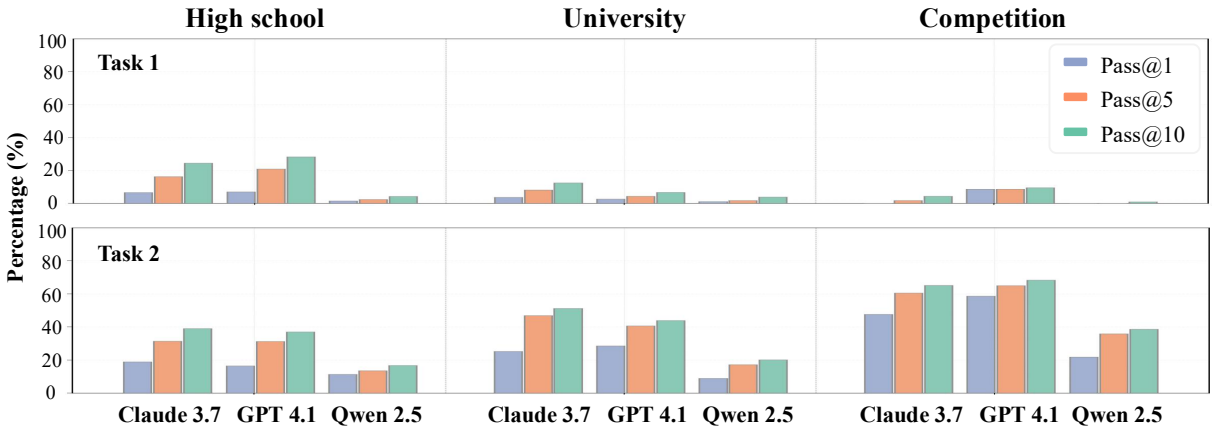

Figure 2: Performance of different MLLMs (Gemini-2.0-flash-thinking, GPT4.1, and Qwen2.5-VL-Instruct-70B) on multimodal theorem automated proving (Task 1) and theorem formalization (Task 2), across varying difficulty levels.

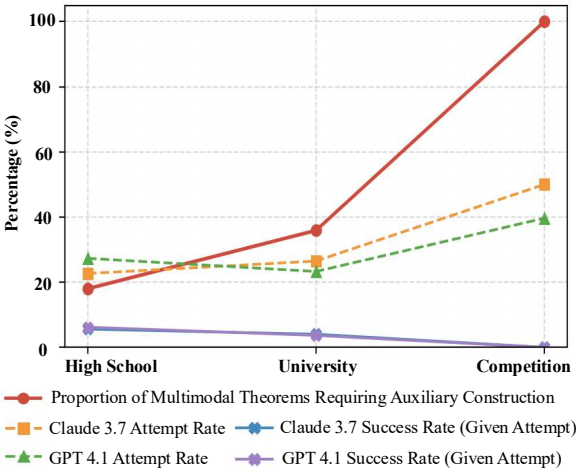

Figure 3: Auxiliary construction analysis by question difficulty level and model, evaluated using *Pass@10*.

Task 2 is consistent across all pass@n settings, demonstrating that models have shown a certain ability in converting natural language descriptions and geometric figures into Coq formal statements (Task 2 pass@n is relatively high), but still face significant challenges in the subsequent complex logical reasoning and constructing formal proofs.

5.5 Model Capability in Generating Auxiliary Constructions for Multimodal Theorems

A characteristic distinguishing multimodal theorem proving from pure text theorem proving is that many theorems require the construction of auxiliary lines to aid thinking, especially problems at the competition level. Therefore, we further investigate the models' ability to construct auxiliary lines during the proof process. As shown in Figure 3, the proportion of problems requiring auxiliary constructions rises significantly with increasing theorem difficulty, confirming the importance of auxiliary construction for solving high-difficulty geometric theorems. Claude 3.7 and GPT 4.1, which perform best in Task 1 and Task 2, attempt auxiliary constructions when generating proofs, and this attempt rate increases with difficulty, indicating a degree of autonomous auxiliary construction capability and awareness. However, the success rate of these proofs containing auxiliary constructions (Figure 3 blue and purple solid lines) is very low. This prominently indicates that while models can introduce steps or concepts involving auxiliary constructions in proofs, they cannot effectively utilize these constructions to advance the proof process. Recent research shows that multimodal models can be augmented by visual prompts (Shtedritski et al., 2023; Yang et al., 2023a; Hu et al., 2024; Wang et al., 2025e). For example, Visual Sketchpad (Hu et al., 2024) demonstrates this potential by providing MLLMs with a sketching interface and drawing tools, which offers a promising direction for multimodal theorem proving.

6 Conclusion

In this paper, we introduce MATP-BENCH, a new multimodal, multilevel, and multilanguage benchmark designed to evaluate MLLMs as automated theorem provers. MATP-BENCH consists of 1056 multimodal theorems drawn from high school, university, and competition-level mathematics. All these multimodal problems are accompanied by formalizations in Lean 4, Coq and Isabelle. Experimental results demonstrate significant variability in the performance of various mainstream MLLMs. Our findings highlight the capabilities and limitations of advanced MLLMs in automated multimodal theorem proving, and reveals the primary bottleneck is constructing correct formal proofs.

Limitations

This paper evaluates the capabilities of various mainstream Multimodal Large Language Models (MLLMs) in multimodal automated theorem proving. We select three different formal languages (Lean 4, Coq, and Isabelle) for testing, and compare the performance (primarily using pass@10 as the metric) of general MLLMs, including o1, Claude-3.7, Gemini-2.0, GPT-4.1, Qwen2.5-VL, and Llama3.2-Vision, on problems of varying difficulty levels. Although this study provides an exploration into the application of MLLMs in the domain of formal proof, it also has several limitations. Firstly, the testing of MLLMs in this paper primarily involves one-shot generation of formal theorems and proofs, and does not explore multi-step or interactive proof generation capabilities. Secondly, the analysis of the model is primarily based on its final results, without delving into its internal mechanisms or the specific impact of different reasoning steps on performance. Future work could consider employing more comprehensive datasets, exploring richer evaluation scenarios such as multi-step and interactive proof generation, and conducting a more in-depth mechanistic analysis of the models' proof generation process.

Ethical Considerations

All natural language mathematical problems in our dataset are sourced exclusively from publicly available resources. The high school and college-level problems originate from existing open-access multimodal mathematics datasets, while the competition-level problems are collected from publicly available Chinese high school and university mathematics competitions. We manually formalized all problems and conducted rigorous human verification. This meticulous creation process guarantees that the formal proofs are novel and have never been exposed to existing models, thereby preventing data leakage and enabling a pristine evaluation of reasoning capabilities.

Licenses Our dataset is distributed under a Creative Commons (CC) license, providing free access to the academic community. All use of our resources must comply with the terms of the respective licenses and be for research purposes.

References

Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*.

Anthropic. 2024. Claude Sonnet. https://www.anthropic.com/claude/sonnet. Accessed: 2025-04-29.

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

Zhangir Azerbayev, Hailey Schoelkopf, Keiran Paster, Marco Dos Santos, Stephen McAleer, Albert Q Jiang, Jia Deng, Stella Biderman, and Sean Welleck. 2023b. Llemma: An open language model for mathematics. *arXiv preprint arXiv:2310.10631*.

Wolfgang Bibel. 2013. *Automated theorem proving*. Springer Science & Business Media.

Jie Cao and Jing Xiao. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th international conference on computational linguistics*, pages 1511–1520.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. 2022. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. *arXiv preprint arXiv:2212.02746*.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Adam Chlipala. 2013. *Certified programming with dependent types: a pragmatic introduction to the Coq proof assistant*. MIT Press.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Tulsee Doshi. 2025. Start building with gemini 2.5 flash. https://developers.googleblog.com/en/start-building-with-gemini-25-flash/. Accessed: 2025-04-29.

Simon Frieder, Luca Pinchetti, Ryan-Rhys Griffiths, Tommaso Salvatori, Thomas Lukasiewicz, Philipp Petersen, and Julius Berner. 2023. Mathematical capabilities of chatgpt. *Advances in neural information processing systems*, 36:27699–27744.

Daocheng Fu, Zijun Chen, Renqiu Xia, Qi Liu, Yuan Feng, Hongbin Zhou, Renrui Zhang, Shiyang Feng, Peng Gao, Junchi Yan, and 1 others. 2025. Trustgeogen: Scalable and formal-verified data engine for trustworthy multi-modal geometric problem solving. *arXiv preprint arXiv:2504.15780*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

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. *arXiv preprint arXiv:2103.03874*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *arXiv preprint arXiv:2406.09403*.

Albert Q Jiang, Sean Welleck, Jin Peng Zhou, Wenda Li, Jiacheng Liu, Mateja Jamnik, Timothée Lacroix, Yuhuai Wu, and Guillaume Lample. 2022. Draft, sketch, and prove: Guiding formal theorem provers with informal proofs. *arXiv preprint arXiv:2210.12283*.

Albert Qiaochu Jiang, Wenda Li, Jesse Michael Han, and Yuhuai Wu. 2021. Lisa: Language models of isabelle proofs. In *6th Conference on Artificial Intelligence and Theorem Proving*, pages 378–392.

Guillaume Lample, Timothee Lacroix, Marie-Anne Lachaux, Aurelien Rodriguez, Amaury Hayat, Thibaut Lavril, Gabriel Ebner, and Xavier Martinet. 2022. Hypertree proof search for neural theorem proving. *Advances in neural information processing systems*, 35:26337–26349.

Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, and 1 others. 2025. Perception, reason, think, and plan: A survey on large multimodal reasoning models. *arXiv preprint arXiv:2505.04921*.

Yong Lin, Shange Tang, Bohan Lyu, Jiayun Wu, Hongzhou Lin, Kaiyu Yang, Jia Li, Mengzhou Xia, Danqi Chen, Sanjeev Arora, and 1 others. 2025. Goedel-prover: A frontier model for open-source automated theorem proving. *arXiv preprint arXiv:2502.07640*.

Wang Ling, Dani Yogatama, Chris Dyer, and Phil Blunsom. 2017. Program induction by rationale generation: Learning to solve and explain algebraic word problems. *arXiv preprint arXiv:1705.04146*.

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. 2023a. Fimo: A challenge formal dataset for automated theorem proving. *Preprint*, arXiv:2309.04295.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. 2023b. Improved baselines with visual instruction tuning. *Preprint*, arXiv:2310.03744.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

Swaroop Mishra, Matthew Finlayson, Pan Lu, Leonard Tang, Sean Welleck, Chitta Baral, Tanmay Rajpurohit, Oyvind Tafjord, Ashish Sabharwal, Peter Clark, and 1 others. 2022. Lila: A unified benchmark for mathematical reasoning. *arXiv preprint arXiv:2210.17517*.

Leonardo de Moura and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language. In *Automated Deduction–CADE 28: 28th International Conference on Automated Deduction, Virtual Event, July 12–15, 2021, Proceedings 28*, pages 625–635. Springer.

Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. 2024. Autoformalizing euclidean geometry. *arXiv preprint arXiv:2405.17216*.

OpenAI. 2024a. GPT-4.1. `https://openai.com/index/gpt-4-1/`. Accessed: 2025-04-29.

OpenAI. 2024b. Introducing o3 and o4-mini. `https://openai.com/index/introducing-o3-and-o4-mini/`. Accessed: 2025-04-29.

Stanislas Polu, Jesse Michael Han, Kunhao Zheng, Mantas Baksys, Igor Babuschkin, and Ilya Sutskever. 2022. Formal mathematics statement curriculum learning. *arXiv preprint arXiv:2202.01344*.

Stanislas Polu and Ilya Sutskever. 2020. Generative language modeling for automated theorem proving. *arXiv preprint arXiv:2009.03393*.

Xiaoye Qu, Yafu Li, Zhaochen Su, Weigao Sun, Jianhao Yan, Dongrui Liu, Ganqu Cui, Daizong Liu, Shuxian Liang, Junxian He, and 1 others. 2025. A survey of efficient reasoning for large reasoning models: Language, multimodality, and beyond. *arXiv preprint arXiv:2503.21614*.

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, and 1 others. 2025. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*.

Alan JA Robinson and Andrei Voronkov. 2001. *Handbook of automated reasoning*, volume 1. Elsevier.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1466–1476.

Aleksandar Shtedritski, Christian Rupprecht, and Andrea Vedaldi. 2023. What does clip know about a red circle? visual prompt engineering for vlms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11987–11997.

Qwen Team. 2025. Qwen2.5-vl.

Amitayush Thakur, George Tsoukalas, Yeming Wen, Jimmy Xin, and Swarat Chaudhuri. 2023. An in-context learning agent for formal theorem-proving. *arXiv preprint arXiv:2310.04353*.

Trieu H Trinh, Yuhuai Wu, Quoc V Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482.

George Tsoukalas, Jasper Lee, John Jennings, Jimmy Xin, Michelle Ding, Michael Jennings, Amitayush Thakur, and Swarat Chaudhuri. 2024. Putnambench: Evaluating neural theorem-provers on the putnam mathematical competition. *arXiv preprint arXiv:2407.11214*.

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 Saxcé, 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. *Preprint*, arXiv:2504.11354.

Haiming Wang, Huajian Xin, Zhengying Liu, Wenda Li, Yinya Huang, Jianqiao Lu, Zhicheng Yang, Jing Tang, Jian Yin, Zhenguo Li, and 1 others. 2024a. Proving theorems recursively. *arXiv preprint arXiv:2405.14414*.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024b. Measuring multimodal mathematical reasoning with math-vision dataset. *Advances in Neural Information Processing Systems*, 37:95095–95169.

Peijie Wang, Zhong-Zhi Li, Fei Yin, Xin Yang, Dekang Ran, and Cheng-Lin Liu. 2025b. Mv-math: Evaluating multimodal math reasoning in multi-visual contexts. *arXiv preprint arXiv:2502.20808*.

Ruida Wang, Yuxin Li, Yi R. Fung, and Tong Zhang. 2025c. 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. 2025d. Ma-lot: Multi-agent lean-based long chain-of-thought reasoning enhances formal theorem proving. *arXiv preprint arXiv:2503.03205*.

Ruida Wang, Jipeng Zhang, Yizhen Jia, Rui Pan, Shizhe Diao, Renjie Pi, and Tong Zhang. 2024c. Theoremllama: Transforming general-purpose llms into lean4 experts. *arXiv preprint arXiv:2407.03203*.

Yaoting Wang, Shengqiong Wu, Yuecheng Zhang, Shuicheng Yan, Ziwei Liu, Jiebo Luo, and Hao Fei. 2025e. Multimodal chain-of-thought reasoning: A comprehensive survey. *arXiv preprint arXiv:2503.12605*.

Makarius Wenzel, Lawrence C Paulson, and Tobias Nipkow. 2008. The isabelle framework. In *International Conference on Theorem Proving in Higher Order Logics*, pages 33–38. Springer.

Huajian Xin, ZZ Ren, Junxiao Song, Zhihong Shao, Wanjia Zhao, Haocheng Wang, Bo Liu, Liyue Zhang, Xuan Lu, Qiushi Du, and 1 others. 2024. Deepseek-prover-v1. 5: Harnessing proof assistant feedback for reinforcement learning and monte-carlo tree search. *arXiv preprint arXiv:2408.08152*.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023a. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *arXiv preprint arXiv:2310.11441*.

Kaiyu Yang, Aidan Swope, Alex Gu, Rahul Chalamala, Peiyang Song, Shixing Yu, Saad Godil, Ryan J Prenger, and Animashree Anandkumar. 2023b. Leandojo: Theorem proving with retrieval-augmented language models. *Advances in Neural Information Processing Systems*, 36:21573–21612.

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

Zhouliang Yu, Ruotian Peng, Keyi Ding, Yizhe Li, Zhongyuan Peng, Minghao Liu, Yifan Zhang, Zheng Yuan, Huajian Xin, Wenhao Huang, Yandong Wen, Ge Zhang, and Weiyang Liu. 2025. Formalmath: Benchmarking formal mathematical reasoning of large language models. *Preprint*, arXiv:2505.02735.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, and 1 others. 2024. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pages 169–186. Springer.

Kunhao Zheng, Jesse Michael Han, and Stanislas Polu. 2021. Minif2f: a cross-system benchmark for formal olympiad-level mathematics. *arXiv preprint arXiv:2109.00110*.

A Examples of questions at different levels

In Figure 4 and Figure 5, we show high school and university-level problems respectively, with Figure 1 featuring competition-level questions.

Example of a High School Level Question

Proof Coplanar squares $ABGH$ and $BCDF$ are adjacent, with $CD = 10$ units and $AH = 5$ units. Point E is on segments AD and GB. The area of triangle ABE, in square units, is $\frac{25}{3}$.

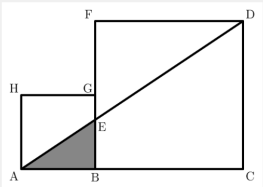

Figure 4: An example of a high school level mathematics problem, requiring the calculation of a triangle's area within a configuration of two adjacent squares (ABGH and BCDF) of differing side lengths.

Example of a University Level Question

Proof A regular icosahedron is a 20-faced solid where each face is an equilateral triangle and five triangles meet at every vertex. The regular icosahedron shown below has one vertex at the top, one vertex at the bottom, an upper pentagon of five vertices all adjacent to the top vertex and all in the same horizontal plane, and a lower pentagon of five vertices all adjacent to the bottom vertex and all in another horizontal plane. Find the number of paths from the top vertex to the bottom vertex such that each part of a path goes downward or horizontally along an edge of the icosahedron, and no vertex is repeated is 810.

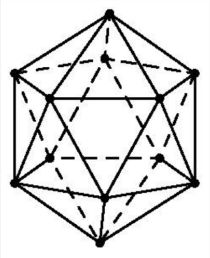

Figure 5: An example of a university level mathematics problem, requiring the determination that the number of non-repeating paths from the top to bottom vertex of a regular icosahedron, under downward or horizontal movement constraints, is 810.

B Multimodal Automated Theorem Proving

Tables 3 present the experimental results for Task 1, evaluating multimodal automated theorem proving using pass@5 and pass@1 metrics, respectively, across Lean 4, Coq, and Isabelle formal languages and three difficulty levels. Comparing the two tables, it is evident that providing models with more attempts (pass@5 vs pass@1) generally leads to higher success rates across all models, formal languages, and difficulty levels, highlighting the benefit of multiple decoding attempts in this task. Analyzing the pass@5 results in Table 5, among the individual models, GPT-4.1 consistently ranks among the top performers under pass@5, showing notable strength in handling higher difficulty levels. Gemini-2.0, Qwen2.5-VL, and Llama3.2 generally achieve lower pass rates across most tasks and difficulty levels under pass@5.

The pass@1 results in Table 6, which assesses the model's ability to generate a correct proof on the very first attempt, are considerably lower across the board. The relative ranking of models shifts for some languages under this stricter metric. GPT-4.1 still demonstrates relative strength at the competition level even at pass@1 in Coq and Isabelle, suggesting some capability for direct high-difficulty solutions. The performance difference between pass@5 and pass@1 highlights that while all models benefit from retries, some models, appear to leverage multiple attempts more effectively to find a successful proof compared to their initial attempt performance, whereas others, like o1 in Coq and Isabelle, are relatively stronger at generating a correct proof on the first try.

C Multimodal Theorem Formalization

Tables 4 present the experimental results for Multimodal Theorem Formalization (Task 2), which evaluates models on generating formalized theorems using pass@5 and pass@1 metrics across Lean 4, Coq, and Isabelle. As expected, pass@5 scores consistently exceed pass@1 scores, indicating that multiple attempts improve formalization accuracy. However, compared to full proof generation (Task 1), the relative increase from pass@1 to pass@5 appears less dramatic for Task 2, suggesting that models capable of formalizing a theorem often do so successfully on earlier attempts. Overall, for theorem formalization, models achieve considerably higher pass rates than for proof generation, highlighting that generating the correct theorem statement is a less challenging task than generating the complete proof.

Analyzing the results, o1 demonstrates particu-

lar strength in first-attempt formalization (pass@1), frequently leading in Coq and Isabelle. Models like Claude-3.7 and GPT-4.1 show competitive performance in specific languages or difficulty tiers, while others generally trail. These results indicate that while current models are significantly better at theorem formalization than full proof generation, their ability to accurately formalize theorems still varies depending on the specific formal language and problem complexity, with o1 showing notable capabilities in this task.

D Performance Comparison Across Formal Languages

Figure 7 illustrates the performance of Claude 3.7, GPT 4.1, and Qwen 2.5 on multimodal theorem proving tasks across Lean 4, Coq, and Isabelle, evaluated by Pass@1, Pass@5, and Pass@10 metrics across varying difficulty levels. The results consistently demonstrate that model performance significantly decreases with increasing task difficulty and improves with a greater number of allowed attempts, with Pass@10 achieving the highest pass rates. Among the evaluated models, GPT 4.1 generally exhibits the strongest performance across most formal languages and difficulty levels, particularly excelling in challenging scenarios. Claude 3.7 typically ranks as the second-best performer, while Qwen 2.5 consistently shows the lowest pass rates. Performance also varies by formal language, with models often achieving higher success rates in Coq compared to Lean 4 and Isabelle. The substantial difference between Pass@1 and Pass@10 highlights the models' ability to find correct proofs with multiple tries, although overall performance remains low on complex competition-level problems for all evaluated models.

E Prompts

Figures 8 and 9 outline the prompt for multimodal automated theorem proving task, which aims to achieve end-to-end multimodal automated theorem proving similar to human provers, by directly generating a formalized theorem and its proof from multimodal informal input. Figure 10 presents the prompt for multimodal theorem formalization task. The prover receives the multimodal question, and is required to formalize it into a precise theorem.

F AI Usage

This paper introduces MATP-BENCH, a benchmark for evaluating MLLMs in automated theorem proving, featuring multimodal theorems from high school, university, and competition levels. For the preparation of this manuscript, we utilized AI-powered tools exclusively for writing assistance, such as grammar correction and phrasing refinement, to enhance the overall clarity of the paper.

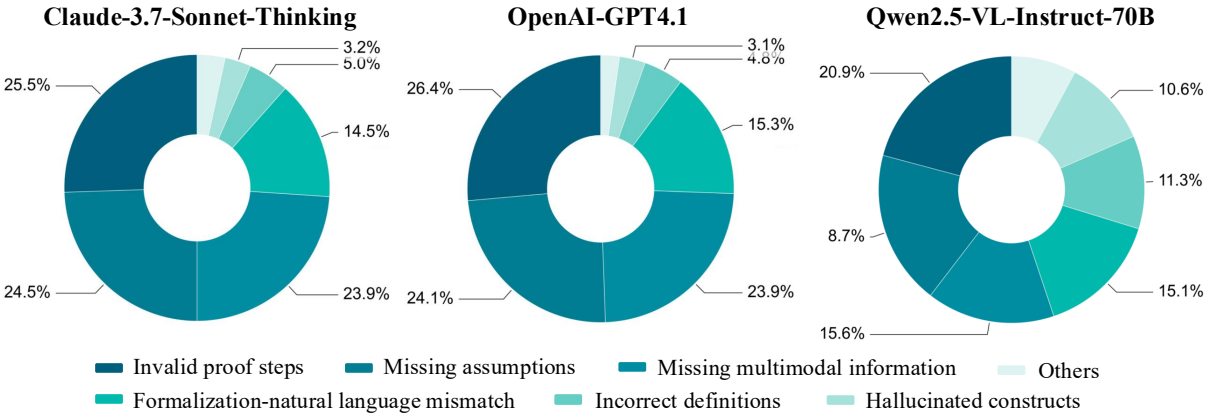

Figure 6: We perform an error analysis on the results of a reasoning model (Claude-3.7-Sonnet-Thinking) and two non-reasoning models (GPT4.1 and Qwen2.5-VL-Instruct-70B), all three being competitive on MATP tasks (Lean 4), with the figure illustrating the seven most frequent error types.

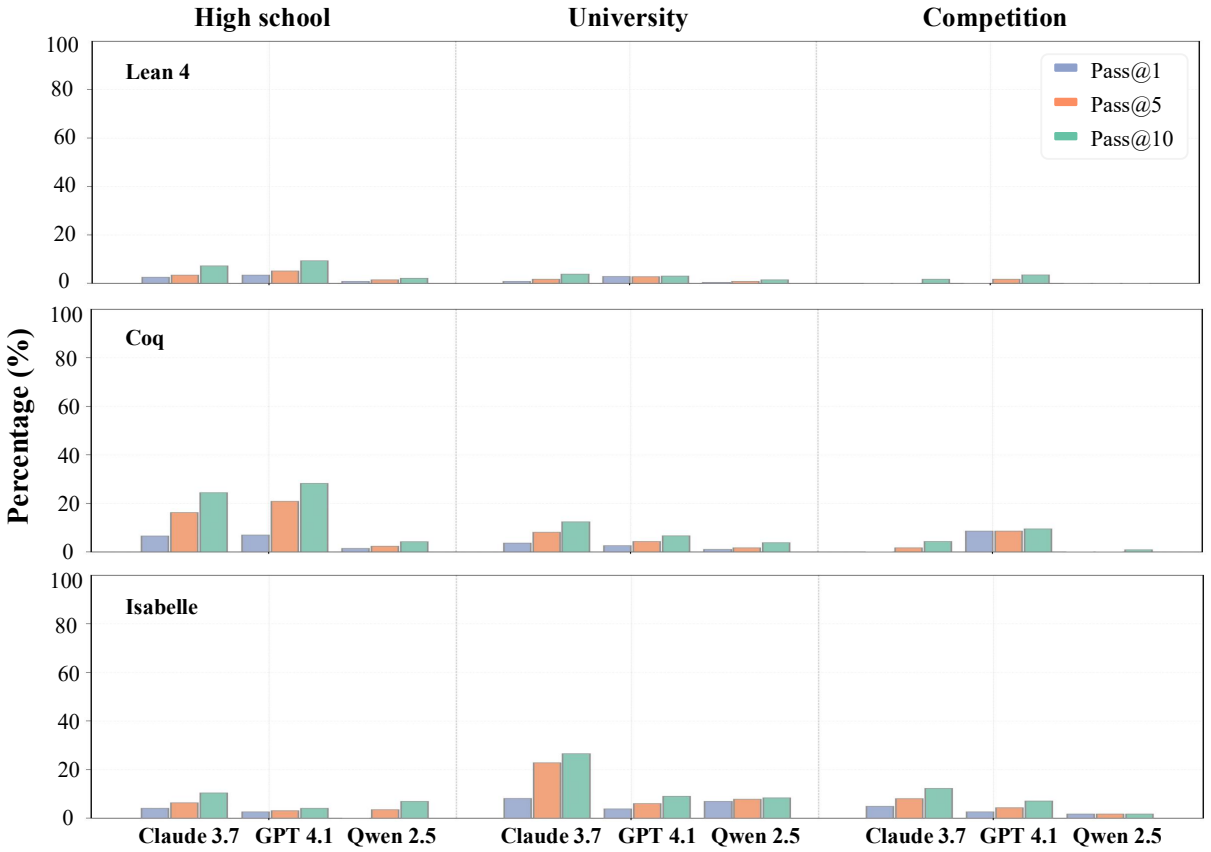

Figure 7: We present the performance of different MLLMs (Gemini-2.0-flash-thinking, OpenAI-GPT4.1, and Qwen2.5-VL-Instruct-70B) on multimodal automated theorem proving task across varying difficulty levels, evaluated using *Pass@1, Pass@5, and Pass@10* metrics.

Prompt for Multimodal Automated Theorem Proving

You are a formal mathematical assistant. Given a natural language description of a theorem and an accompanying diagram, your task is to generate a formal, unambiguous, and complete version of the theorem using Lean 4 formal language, without providing any proof. Note: This is a multimodal theorem formalization task. The natural language description alone may be incomplete or ambiguous, and the diagram contains essential geometric, positional, or structural information required to construct the full formal statement. You must carefully extract and integrate information from both the text and the diagram to ensure the resulting formal theorem is correct, complete, and self-contained. Your output should only include the Lean 4 code block defining the theorem, and follow the style of the following examples (with appropriate imports, variable declarations, hypotheses, and theorem statement):

Here are a few complete form theorem examples:
Example 1:
Image: (image1)
Natural language theorem statement: Let $\triangle ABC$ satisfy $\angle CAB < \angle BCA < \frac{\pi}{2} < \angle ABC$. If the bisector of the external angle at A meets line BC at P, the bisector of the external angle at B meets line CA at Q, and $AP = BQ = AB$, find $\angle CAB$.
Output (Formal theorem using Lean4 language):

```
import Mathlib

open MeasureTheory

theorem example
(A B C X Y : EuclideanSpace ℝ (Fin 2))
(hABC : Collinear ℝ {A, B, C})
(hangles : ∠ C A B < ∠ B C A ∧ ∠ B C A < π/2 ∧ π/2 < ∠ A B C)
(hX : Collinear ℝ {X, B, C} ∧ ∠ X A B = (π - ∠ C A B)/2 ∧ dist
    ↪ A X = dist A B)
(hY : Collinear ℝ {Y, C, A} ∧ ∠ Y B C = (π - ∠ A B C)/2 ∧ dist
    ↪ B Y = dist A B)
: ∠ C A B = example :=
sorry
```

Figure 8: Prompt for multimodal automated theorem proving task (part1).

Prompt for Multimodal Automated Theorem Proving

Example 2:
Image: (image2)
Natural language theorem statement: Let a, b, and c be the side lengths of a triangle with inradius r. If $p = \frac{a+b+c}{2}$, show that

$$\frac{1}{(p-a)^2} + \frac{1}{(p-b)^2} + \frac{1}{(p-c)^2} \geq \frac{1}{r^2}.$$

Output (Formal theorem using Lean4 language):

```
import Mathlib

theorem putnam_a2
(r : ℝ)
(A B C : EuclideanSpace ℝ (Fin 2))
(hABC : Collinear ℝ {A, B, C})
(a b c p : ℝ)
(ha : a = dist B C)
(hb : b = dist C A)
(hc : c = dist A B)
(hp : p = (dist B C + dist C A + dist A B)/2)
(hr : ∃ I : EuclideanSpace ℝ (Fin 2),
(∃! P : EuclideanSpace ℝ (Fin 2), dist I P = r ∧ Collinear ℝ {P,
    ↪ B, C}) ∧
(∃! Q : EuclideanSpace ℝ (Fin 2), dist I Q = r ∧ Collinear ℝ {Q,
    ↪ C, A}) ∧
(∃! R : EuclideanSpace ℝ (Fin 2), dist I R = r ∧ Collinear ℝ {R,
    ↪ A, B}) ∧
(∀ Z : EuclideanSpace ℝ (Fin 2), dist I Z ≤ r → Z ∈ convexHull
    ↪ ℝ {A, B, C}))
: 1/(p - a)^2 + 1/(p - b)^2 + 1/(p - c)^2 ≥ 1/r^2 :=
sorry
```

Strict Instructions:

- Only output the formal theorem in Lean 4, including all necessary imports, variable declarations, hypotheses, and the theorem statement.
- Do NOT include any proof or attempt to prove the theorem.
- Explicitly indicate that this is an unproven theorem by ending the statement with := sorry.
- Do not use by, exact, or any other proof-related keywords.

Figure 9: Prompt for multimodal automated theorem proving task (part2).

Prompt for Multimodal Theorem Formalization

You are a formal mathematical assistant specializing in **multimodal theorem proving**. Given a natural language description of a mathematical theorem and a related diagram, your task is to:

- **Jointly interpret both the text and the image**, extracting all relevant mathematical information, including geometric or algebraic configurations, object relationships, and any labeled points, angles, lines, circles, or symbols present in the diagram.
- When the **natural language description is incomplete or ambiguous**, you must infer and complete the necessary assumptions or details based on the visual content of the image.
- Formulate a **precise, unambiguous, and self-contained formal statement** of the theorem in the **Lean 4 proof assistant language**, including all necessary variable declarations and hypotheses.
- Construct a **complete, rigorous, and correct formal proof** of the theorem in Lean 4, ensuring that it passes verification in the Lean 4 environment.
- The formalization must be **independent and fully self-contained**, requiring no reference to the original natural language or image once generated.

Your output must consist of **Lean 4 code only**, and include the following components:

- All required ‘import‘ statements.
- Declarations of all relevant variables, structures, and assumptions derived from both the text and the image.
- A clear and precise formal statement of the theorem.
- A complete and logically sound proof written in Lean 4, suitable for direct verification.

Follow the conventions and style used in the **Lean 4 mathlib** library to ensure correctness, consistency, and readability.
Now, the output must follow the exact style of the examples:
Image: (image upload)
Natural language theorem statement: row["NL_statement"]
Output (Lean 4 code only):

Figure 10: Prompt for multimodal theorem formalization task.

Prompt for LLM-as-Judge

You are a Coq expert. Compare the *statements* of two formal theorems. Determine if they are logically equivalent by:
1. Checking if they express the *same mathematical proposition*.
2. Ignoring: syntax variations and trivial reordering.

Return *exactly*:
- 'CORRECT' if the theorems are semantically equivalent.
- 'INCORRECT' if they define different propositions.

Reference Coq code:
{reference_coq_code}

Generated Coq code:
{generated_coq_code}

Your judgment (CORRECT/INCORRECT):

Figure 11: Prompt for LLM-as-Judge.