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

Anonymous ACL submission

Abstract

Geometry theorem proving relies on visual reasoning with text and diagrams. While MLLMs show potential in mathematics, their use in multimodal automated theorem proving remains largely unexplored. In this paper, we introduce the **M**ultimodal **A**utomated **T**heorem **P**roving benchmark (**MATP-BENCH**), a novel multimodal, multi-level, and multi-language benchmark designed to evaluate MLLMs in this role as multimodal automated theorem provers. MATP-BENCH consists of 1,056 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, making the benchmark compatible with a wide range of theorem-proving frameworks. Grounding our analysis in a Structural Causal Model, we identify the distinct challenge of MATP: it requires not only direct mapping of explicit inputs for theorem formalization but, more critically, latent causal planning to discover unobserved auxiliary constructions. Our evaluation reveals that while advanced MLLMs show promise in formalization, they struggle significantly with synthesizing these latent auxiliary constructions, often generating ineffective auxiliary steps or ignoring visual constraints.

1 Introduction

Recent advancements in Large Language Models (LLMs) have significantly driven 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 such as Lean 4 (Moura and Ullrich, 2021), Isabelle (Wenzel et al., 2008), and Coq (Chlipala, 2013). However, 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 such as geometry, where visual elements such as diagrams are often indispensable for understanding and reasoning, and 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 human-written proofs into Lean 4, rather than autonomous generation. Similarly, AlphaGeometry (Trinh et al., 2024) solves Olympiad geometry problems with symbolic text inputs, and its solutions are not verifiable in proof assistants such as Lean 4.

To this end, we present the **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 1,056 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 in natural language. For humans, the reasoning process in such tasks involves not only drawing intuitive insights from visual representa-

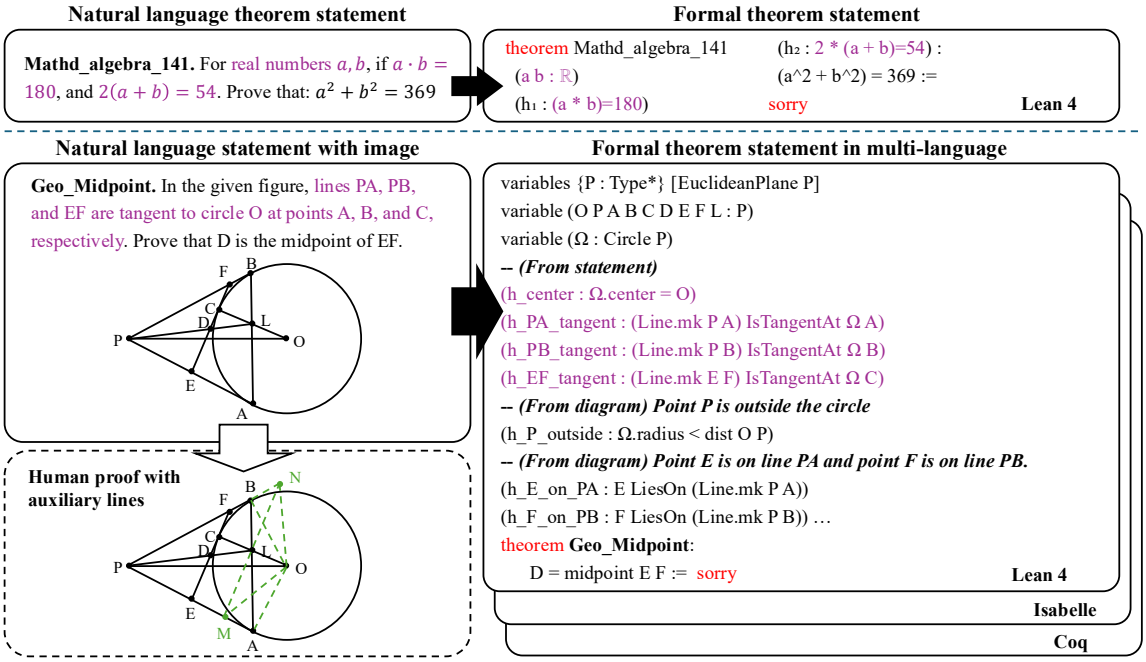


Figure 1: In traditional ATP, such as the miniF2F benchmark (top), theorem formalization relies solely on text statements, whereas our MATP task setting (bottom) requires model to also extract critical premises not explicitly expressed in the text by analyzing accompanying diagrams (see ***From diagram*** 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. The `sorry` keyword denotes omitted proofs.

tions but also constructing auxiliary diagrams and identifying implicit structural relations. We use MATP-BENCH to evaluate various MLLMs. 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.

To investigate the root causes of these failures, we ground our analysis in a **Structural Causal Model** (Peters et al., 2017), a graphical framework commonly used to describe the causal mechanisms within a system (Pearl, 2009). We distinguish that while formalization relies on the ***observational mapping*** of explicit inputs, theorem proving requires ***latent causal planning*** to discover unobserved auxiliary constructions. Our case studies further identify specific failure modes: (1) Visual-Symbolic Misalignment: despite correct textual extraction, models frequently ignore critical visual constraints, leading to missing premises; (2) Ineffective Auxiliary Constructions: models often introduce valid but irrelevant auxiliary lines that fail to advance the causal path of the proof. In summary, our contributions are as follows:

- We introduce MATP-BENCH, which contains 1,056 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 various 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.
- We formulate a Structural Causal Model to capture the fundamental difference between formalization and proving. Our analysis identifies latent causal planning as the primary bottleneck. We further categorize failure modes, such as visual-symbolic misalignment and ineffective auxiliary lines, offering insights for future improvements.

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; Zhang et al., 2024), 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). 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.

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	✓	✓				✓		
LeanEuclid (Murphy et al., 2024)	173	✓		✓	✓	✓	✓		
PutnamBench (Tsoukalas et al., 2024)	640	✓	✓				✓	✓	✓
AlphaGeometry-test (Trinh et al., 2024)	30	✓	✓		✓				
ProverBench (Ren et al., 2025b)	325	✓	✓				✓		
GeoTrust-test (Fu et al., 2025a)	240	✓			✓	✓			
Geoint-R1 (Wei et al., 2025)	1885	✓	✓		✓	✓	✓		
GeoLaux (Fu et al., 2025b)	2186				✓				
LeanGeo (Song et al., 2025)	122	✓	✓						
GeoSketch (Weng et al., 2025)	390				✓				
MATP-BENCH (ours)	1,056	✓	✓	✓	✓	✓	✓	✓	✓

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

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. However, the exploration of ATP in multimodal settings remains limited.

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. 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 advancing multimodal automated theorem proving. 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 formal proof of the theorem 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 statement***.

As shown in the *bottom part* of Figure 1, the natural language statement NT and the information from the multimodal input MI are complementary, forming a complete theorem. Hence, the model must first generate the complete formal theorem FT, and then construct a valid formal proof FP. The entire MATP task can be summarized as:

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

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

4 MATP-BENCH

MATP-BENCH is a novel benchmark designed to evaluate MLLMs as automated theorem provers. We detail the benchmark construction as follows.

	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.

(1) Multimodal Context and Multilanguage Theorem. As shown in Figure 1, unlike traditional text-only datasets, MATP-BENCH introduces multimodal context to jointly evaluate models on visual understanding, mathematical reasoning, and symbolic manipulation. Our benchmark presents the following challenges: (i) ***Visual Understanding***: Accurately extracting key information from theorem-related images, akin to human perception, to construct formal theorem statements; (ii) ***Mathematical Reasoning***: Employing rigorous mathematical reasoning to derive complete proofs based on the provided natural language descriptions and images; (iii) ***Neural-Symbol Proof Generation***: Demonstrating proficiency in these formal languages to strictly translate the mathematical reasoning process into verifiable formal proofs.

(2) Hierarchy and Diversity. As shown in Table 2, MATP-BENCH spans high school, university, and competition levels, with examples provided in Appendix A. 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 analytic geometry, plane geometry, and 3D geometry***. These theorems challenge models to perform complex cross-modal reasoning and multi-step logical deduction, aiming to systematically evaluate the depth of reasoning of models in structured mathematical tasks.

(3) Task Formulation. As mentioned in Section 3, we aim to achieve end-to-end multimodal automated theorem proving (Task 1). To ensure the generated formal theorem aligns with the original problem, we separately establish multimodal theorem formalization (Task 2) for verification, following LeanEuclid (Murphy et al., 2024). Thus, we divide the task into two progressively 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, as shown in Figure 1. The expected output comprises a formal theorem and valid proof. This is significantly challenging as the model must first accurately formalize the theorem, then 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.

(4) 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: (i) ***Multiple-choice questions***: by extracting key information from the image and incorporating the correct answer, the problem is transformed into a concrete theorem; (ii) ***Fill-in-the-blank questions***: the correct answer is directly inserted into the problem statement; (iii) ***Open-ended questions***: interrogative forms are converted into declarative statements based on the given answer. Our data construction process: (i) ***Formalization***: On average, fully formalizing a high school, university, and competition problem takes approximately 25, 30, and 18 minutes respectively (in one language). This variation arises because high school and university problems are often concise with rich visual information requiring preprocessing, whereas competition problems are more detailed and easier to formalize; (ii) ***Peer Review***: Each formalization undergoes a strict review by at least one other team

	Task 1	Lean4				Coq				Isabelle			
		H.	U.	C.	Avg	H.	U.	C.	Avg	H.	U.	C.	Avg
	Human*	100	100	100	100	100	100	100	100	100	100	100	100
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
	Gemini-3.0	**12.50**	**6.15**	**4.02**	**7.56**	26.17	**17.51**	**13.80**	19.12	**14.20**	**10.50**	**4.72**	**9.81**
	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
	Qwen3VL	6.85	4.20	2.45	4.50	18.50	12.10	9.90	13.50	7.85	5.50	2.25	5.20
	DS-Prover	5.85	2.90	0.95	3.23	-	-	-	-	-	-	-	-
	KM-Prover	6.70	3.55	1.45	3.90	-	-	-	-	-	-	-	-
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
	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
	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
	Gemini-3.0	**7.80**	**4.20**	**2.55**	**4.85**	19.50	**12.80**	8.47	**13.59**	**9.65**	**7.20**	1.72	**6.17**
	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
	Qwen3VL	4.50	2.80	1.20	2.83	12.50	8.50	5.50	8.83	5.10	3.60	1.20	3.30
	DS-Prover	4.10	1.85	0.50	2.15	-	-	-	-	-	-	-	-
	KM-Prover	4.65	2.40	0.85	2.63	-	-	-	-	-	-	-	-
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
	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
	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
	Gemini-3.0	**4.95**	1.72	**1.50**	**2.72**	**12.80**	**8.50**	6.15	**9.14**	**6.20**	**4.89**	0.00	**3.67**
	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
	Qwen3VL	2.85	1.65	0.60	1.70	7.20	4.50	3.10	4.93	3.15	2.40	0.50	2.02
	DS-Prover	2.15	0.95	0.20	1.10	-	-	-	-	-	-	-	-
	KM-Prover	2.55	1.25	0.45	1.42	-	-	-	-	-	-	-	-

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

member to ensure accuracy; and (iii) ***Expert Verification***: To verify solvability and establish a human baseline, we randomly sampled 60 problems (20 per difficulty level), on which expert annotators achieved a 100% success rate (Task 1) through meticulous manual proof construction, confirming the rigorousness and correctness of our benchmark. Unlike text-only formalization, a central challenge here is the incompleteness of natural language descriptions, where essential assumptions are often ***implicit in the diagrams***.

(5) Preventing Modality and Data Leakage. To mitigate modality leakage, MATP-BENCH provides raw natural language NT and diagrams as input rather than formal statements. To strictly rule out data leakage from pre-training corpora (e.g., Mathlib4, AFP, and Coq Stdlib), we implement a comprehensive decontamination check. Specifically, we match all 1,056 formal theorem statements against these libraries using N-gram matching. As detailed in Appendix E, we find **0.0% overlap** for long sequences ($N = 30$), confirming that our theorems are novel. This conclusion is further supported by empirical results: even the strongest models achieve less than 10% Pass@5 accuracy on Task 1. This significant performance gap contradicts the hypothesis of memorization-based retrieval, verifying that models must rely on genuine multimodal reasoning.

5 Experiments

5.1 Experimental settings

Methods: We conduct extensive experiments on a wide variety of advanced multimodal LLMs. Specifically, the proprietary models include ***o1***, ***GPT-4.1***, `Claude-3.7-Sonnet-Thinking` (***Claude-3.7***), `Gemini-2.0-Flash-Thinking` (***Gemini-2.0***), and `Gemini-3-Flash-Preview`

	Task 2	Lean4				Coq				Isabelle			
		H.	U.	C.	Avg	H.	U.	C.	Avg	H.	U.	C.	Avg
	Human†	100	100	100	100	100	100	100	100	100	100	100	100
pass@10	o1	53.12	61.28	63.32	59.24	42.65	45.64	63.37	50.50	51.88	**63.50**	**63.08**	**60.14**
	Claude-3.7	55.07	60.42	61.64	59.04	39.07	**51.22**	65.20	51.83	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
	Gemini-3.0	**60.50**	**68.45**	63.41	**64.13**	**48.20**	50.80	62.53	**53.84**	**56.40**	60.15	61.92	59.49
	GPT-4.1	47.58	58.19	**65.27**	57.01	37.03	43.89	**68.35**	49.76	48.10	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
	Qwen3VL	48.50	56.20	58.80	54.50	32.40	38.50	55.20	42.03	40.50	55.80	45.20	47.17
pass@5	o1	49.64	56.14	**60.52**	55.76	38.95	42.83	63.58	48.45	43.16	**59.62**	**57.72**	**52.39**
	Claude-3.7	47.58	56.89	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
	Gemini-3.0	**52.83**	**59.21**	56.50	**56.18**	**40.56**	43.20	**56.41**	**50.05**	42.60	52.50	52.40	50.50
	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
	Qwen3VL	44.20	52.50	50.40	49.03	28.60	34.20	52.50	38.43	36.80	51.20	39.50	42.50
pass@1	o1	33.05	45.61	**43.15**	40.24	**27.66**	**35.49**	57.26	**40.13**	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
	Gemini-3.0	**40.50**	**49.63**	42.80	**44.31**	26.80	32.50	52.41	37.23	25.20	**46.43**	**40.22**	**37.38**
	GPT-4.1	26.19	37.22	33.06	31.82	16.46	28.61	**58.72**	26.36	16.73	43.23	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
	Qwen3VL	28.50	38.60	32.40	33.17	15.50	22.40	45.20	27.70	18.20	35.50	25.60	26.43

Table 4: Experimental results of **Multimodal Theorem Formalization** (Task 2), which only requires model to generate formalized theorem. †Human performance serves as the expert-annotated ground truth for this task. We use GPT-4o as the judge and adopt **pass@n** (n=1,5,10) as the evaluation metric.

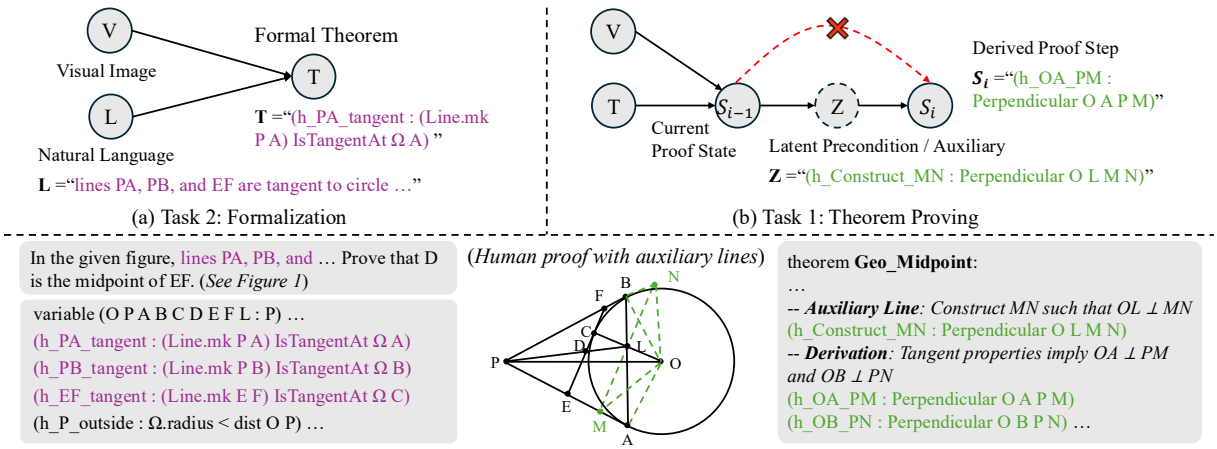
(***Gemini-3.0***) ; while the open-source models include `Qwen2.5-VL-Instruct-70B` (***Qwen2.5VL***) (Team, 2025), `Qwen3-VL-235B-Instruct` (***Qwen3VL***) (Bai et al., 2025), and `Llama3.2-Vision-Instruct-11B` (***Llama3.2V***) (Liu et al., 2023b). To evaluate specialized formal reasoning pipelines, we introduce: ***DS-Prover*** and ***KM-Prover***. These utilize Gemini-3.0 to formalize statements, followed by `DeepSeek-Prover-V2-7B` (Ren et al., 2025a) or `Kimina-Prover-Distill-8B` (Wang et al., 2025a) for proof generation. Since their formalization step is identical to standalone Gemini-3.0 and they lack Coq/Isabelle support, we evaluate them exclusively on Task 1 and the Lean 4 subset.

Metrics: We adopt **pass@n** ($n = 1, 5, 10$) as the primary metric for both tasks. For Task 1, we evaluate success based on formal verification by the proof assistant. To prevent "theorem hacking" (e.g., generating trivially true but irrelevant statements), a successful Task 1 output must strictly pass both the formal verification and the semantic consistency check (defined in Task 2). For ***Task 2***, leveraging the strong discriminative capability of LLMs (Chen et al., 2024; Gu et al., 2025), we employ GPT-4o as a judge to assess semantic consistency against our ground truth. Details of the prompts and evaluation settings are provided in Appendix H. To validate the judge's reliability, we conduct a meta-evaluation on a stratified sample of 300 instances, where the judge achieves ***93.3% agreement*** with human expert annotations (Cohen's $\kappa = 0.88$). See Appendix D for judge reliability analysis.

5.2 Main results

Lean 4. In the Lean 4 experiments, Gemini-3.0 achieves the highest overall success rate of 12.50% on high school problems at pass@10. We observe a consistent performance decline across all models as difficulty increases, with competition problems proving particularly challenging and frequently causing models such as Llama3.2V and Qwen2.5VL to score zero. This highlights a significant capability gap, as the top-performing Gemini-3.0 achieves an average pass@10 score of 7.56%, which is more than three times the 1.61% average of Qwen2.5VL. Notably, specialized text-only provers (DS/KM-Prover) underperform their multimodal backbone Gemini-3.0, confirming that lacking visual context during proof generation severely hinders latent causal planning.

Coq. In the Coq experiments, o1 achieves 28.37%



(c) An example of multimodal theorem formalization and proving with latent auxiliary line construction.

Figure 2: Structural Causal Analysis. **(a)** Task 2: **Observational Mapping** of explicit inputs. **(b)** Task 1: **Latent Causal Planning**, requiring the discovery of unobserved auxiliary constructions (Z). **(c)** An illustrative example showing the formalization process and the necessity of latent auxiliary line construction to complete the proof.

on high school problems (pass@10), marking the highest score recorded in Task 1. We attribute this partially to Coq's powerful built-in decision procedures like `lra`. However, success rates drop steeply on harder tasks; for instance, Llama3.2V scores zero on competition problems. While o1 holds the top average score of 19.43%, more than five times that of Qwen2.5VL (3.59%), it faces strong competition from Gemini-3.0, which actually outperforms o1 on university and competition level problems.

Isabelle. Isabelle proves to be a significant challenge for all models, with generally lower success rates. The maximum score is 14.20%, achieved by Gemini-3.0 on high school problems at pass@10. While o1 and GPT-4.1 constitute the top tier, a clear downward trend is visible as difficulty increases, becoming most pronounced on competition-level problems where most models score zero. The performance gap remains substantial at pass@10, as Gemini-3.0's 9.81% average score roughly triples that of Qwen2.5VL (2.27%). More analysis is provided in Appendix B and C.

5.3 Main Bottleneck in Multimodal Automated Theorem Proving

To rigorously explain the performance disparity between Task 1 and Task 2, we ground our analysis in a Structural Causal Model (SCM) (Peters et al., 2017), as illustrated in Figure 2. We formalize the cognitive mechanisms distinguishing the two tasks:

- **Task 2 (Formalization)** is modeled as an ***Observational Mapping*** problem. Since all geometric entities required for the theorem statement T are explicitly visible in the visual input V and language L, the objective is to estimate $\hat{T} = \text{argmax}_T\, P(T \mid V, L)$. As shown in Figure 2(a), this relies on ***semantic grounding***, a capability where MLLMs excel by matching visual patterns to formal syntax.
- **Task 1 (Proving)**, in contrast, involves Latent Causal Planning. Valid proof steps often depend on geometric preconditions that are not explicitly stated. These preconditions manifest as latent auxiliary constructions (Z), which mediate the causal path. Theoretically, the model must marginalize over this latent space:

$$P(S_i \mid S_{i-1}) = \sum_Z \underbrace{P(S_i \mid S_{i-1}, Z)}_{\text{Reasoning}} \cdot \underbrace{P(Z \mid S_{i-1}, T, V)}_{\text{Planning}} \tag{3}$$

where S_{i-1} is the current proof state.

Empirical Validation Our experimental results confirm this structural divergence. ***(1) Success in Observational Mapping:*** In Task 2, models achieve relatively high accuracy (e.g., GPT-4.1 reaches 56.40% on competition-level formalization in Table 4), confirming that they function as effective "translators" when the causal variables are fully observed. ***(2) Failure in Latent Planning:*** In Task 1, performance drops drastically (e.g., GPT-4.1 falls to 9.32% on high school problems in Table 3). This gap indicates a deficit in the planning term $P(Z \mid S_{i-1}, T, V)$. As shown in Figure 3 (a), model fails to discover the necessary latent variable Z and generates invalid proof steps.

Necessity of Visual Information. We conduct a "Text-Only" ablation on Task 2 using GPT-4.1. As shown in Table 5, removing visual input causes a

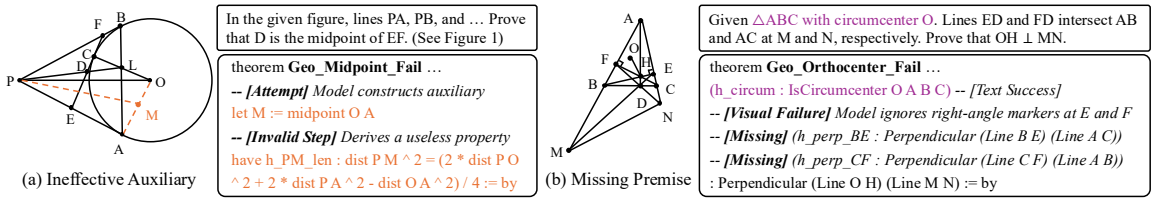


Figure 3: Representative failure traces. **(a) Ineffective Auxiliary:** The model generates a geometrically valid but irrelevant auxiliary construction (orange), failing to advance the proof. **(b) Missing Premise:** While correctly extracting premises from the text (purple), the model ignores visual right-angle markers.

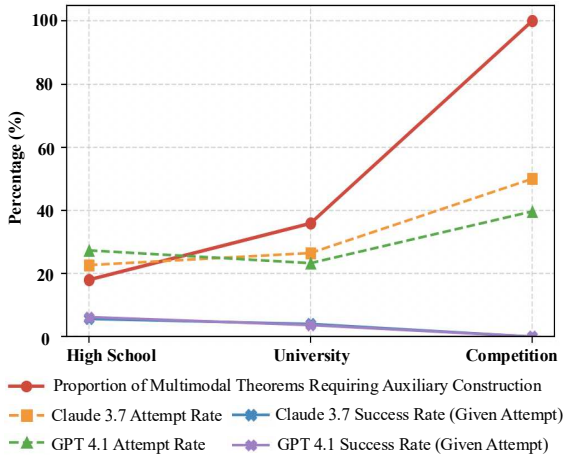


Figure 4: Auxiliary construction analysis by question difficulty level and model, evaluated using *Pass@10*.

drastic accuracy drop from 44.6% to 4.5%. This confirms that visual context is indispensable for extracting geometric conditions, proving the task cannot be solved via text shortcuts alone.

Setting	Lean 4	Isabelle	Coq	Overall Avg
Multimodal	49.2%	45.4%	39.1%	44.6%
Text-Only	4.8%	5.6%	2.9%	4.5%
Δ (↓)	-44.4%	-39.8%	-36.2%	-40.1%

Table 5: Performance comparison (Pass@5) on Task 2 under Multimodal (standard) vs. Text-Only settings.

5.4 Analysis of Auxiliary Constructions

To evaluate the model's capability in generating auxiliary constructions, we operationalize auxiliary use with a deterministic rule-based method: we define an **Attempt** as detecting a new geometric variable declaration (e.g., via `let`) absent in the premises, and **Effective Use** as the explicit reference of this variable in a subsequent verified proof step. We manually verify a random subset of 100 instances, finding 97% alignment between this algorithmic detection and expert judgment, confirming that our algorithm accurately identifies genuine geometric construction attempts (e.g., creating points, lines) while ignoring trivial variable reassignments. Figure 4 shows that the need for auxiliary constructions rises significantly with difficulty. While top models such as Claude-3.7 and GPT-4.1 show a rising "Attempt Rate", their "Effective Success Rate" remains disproportionately low. This gap indicates that while models superficially mimic auxiliary constructions, they fail to generate correct or meaningful instances required for the proof. Recent research suggests that visual prompts (Shtedritski et al., 2023; Yang et al., 2023a; Hu et al., 2024; Wang et al., 2025e) or interactive sketching tools (Hu et al., 2024) offer a promising direction.

5.5 Failure Case Study

Figure 8 categorizes error types, revealing that Claude and GPT-4.1 share a profile dominated by invalid proof steps, missing preconditions, and underutilized multimodal information (72% of total errors). This highlights challenges in complex reasoning and implicit visual constraint extraction. In contrast, open-source models like Qwen2.5VL suffer more from fundamental formalization issues such as incorrect imports. Ultimately, all models struggle with visual-symbolic alignment and maintaining coherent reasoning. Figure 3 illustrates representative failure traces from GPT-4.1: Case (a) demonstrates an ineffective auxiliary construction where the model introduces a geometrically valid but irrelevant line that fails to advance the proof. Case (b) illustrates a missing premise error where visual right-angle markers are ignored despite correct textual extraction.

6 Conclusion

In this paper, we introduce MATP-BENCH, a multimodal, multilevel, and multilanguage benchmark comprising 1,056 theorems from high school to competition levels, each formalized in Lean 4, Coq, and Isabelle. Experiments with mainstream MLLMs reveal significant performance gaps, highlighting current limitations and identifying the construction of correct proofs as the primary bottleneck in multimodal automated theorem proving.

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, Qwen3-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. First, 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 process ensures the formal proofs are novel and have never been exposed to existing models, preventing data leakage and enabling a fair 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*.

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*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

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.

Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Pan Zhou, Yao Wan, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *Preprint*, arXiv:2402.04788.

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*.

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*.

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. 2025a. Trustgeogen: Scalable and formal-verified data engine for

trustworthy multi-modal geometric problem solving. *arXiv preprint arXiv:2504.15780*.

Yumeng Fu, Jiayin Zhu, Lingling Zhang, Bo Zhao, Shaoxuan Ma, Yushun Zhang, Yanrui Wu, and Wenjun Wu. 2025b. Geolaux: A benchmark for evaluating mllms' geometry performance on long-step problems requiring auxiliary lines. *Preprint*, arXiv:2508.06226.

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.

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.

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*.

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*.

Judea Pearl. 2009. *Causality*. Cambridge university press.

Jonas Peters, Dominik Janzing, and Bernhard Schölkopf. 2017. *Elements of causal inference: foundations and learning algorithms*. The MIT press.

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*.

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. 2025a. Deepseek-prover-v2: Advancing formal mathematical reasoning via reinforcement learning for subgoal decomposition. *Preprint*, arXiv:2504.21801.

ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanjia Zhao, Liyue Zhang, Zhe Fu, Qihao Zhu, Dejian Yang, and 1 others. 2025b. 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.

Chendong Song, Zihan Wang, Frederick Pu, Haiming Wang, Xiaohan Lin, Junqi Liu, Jia Li, and Zhengying Liu. 2025. Leangeo: Formalizing competitional geometry problems in lean. *Preprint*, arXiv:2508.14644.

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*.

Jingxuan Wei, Caijun Jia, Qi Chen, Honghao He, Linzhuang Sun, Conghui He, Lijun Wu, Bihui Yu, and Cheng Tan. 2025. Geoint-r1: Formalizing multimodal geometric reasoning with dynamic auxiliary constructions. *Preprint*, arXiv:2508.03173.

Shichao Weng, Zhiqiang Wang, Yuhua Zhou, Rui Lu, Ting Liu, Zhiyang Teng, Xiaozhang Liu, and Hanmeng Liu. 2025. Geosketch: A neural-symbolic approach to geometric multimodal reasoning with auxiliary line construction and affine transformation. *Preprint*, arXiv:2509.22460.

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 5 and Figure 6, 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 $\frac{25}{3}$.

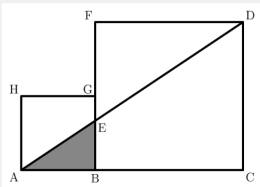


Figure 5: An example of a high school level mathematics problem, requiring the calculation of a triangle's area within a configuration of two adjacent squares (ABGH and BCDF) of differing side lengths.

B Multimodal Automated Theorem Proving

Table 3 presents 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.

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.

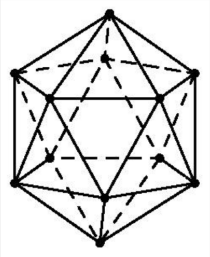


Figure 6: An example of a university level mathematics problem, requiring the determination that the number of non-repeating paths from the top to bottom vertex of a regular icosahedron, under downward or horizontal movement constraints, is 810.

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, such as o1 in Coq and Isabelle, are relatively stronger at generating a correct proof on the first try.

C Multimodal Theorem Formalization

Table 4 presents the experimental results for Multimodal Theorem Formalization (Task 2), which evaluates models on generating formalized theorems

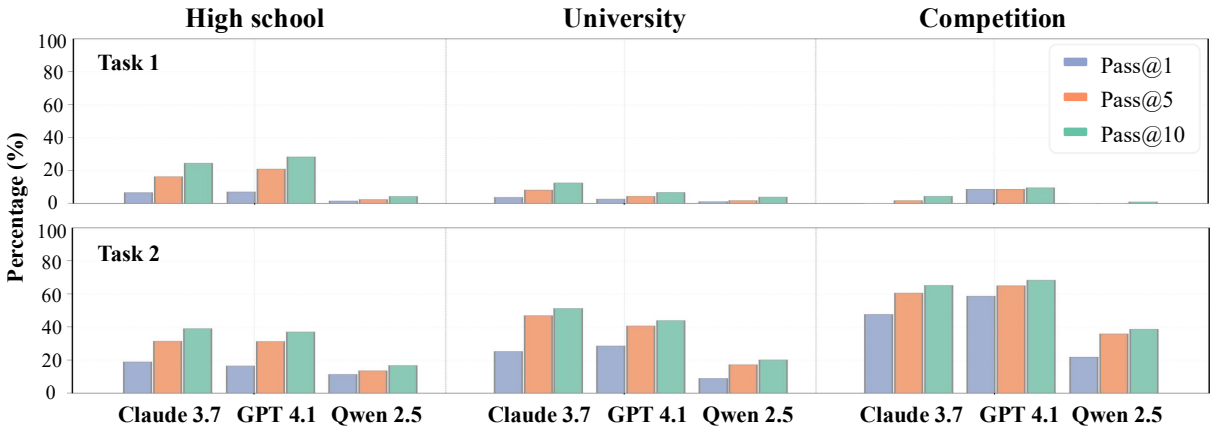


Figure 7: Performance of different MLLMs (Gemini-2.0-flash-thinking, GPT4.1, and Qwen2.5-VL-Instruct-70B) on multimodal theorem automated proving (Task 1) and theorem formalization (Task 2), across varying difficulty levels.

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 particular strength in first-attempt formalization (pass@1), frequently leading in Coq and Isabelle. Models such as 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 Judge Reliability Analysis

To ensure the fairness and accuracy of using GPT-4o as an automated judge for Task 2, and to address concerns regarding the reliability of the LLM-as-a-judge metric, we conducted a rigorous human verification process. We randomly sampled 300 instances, stratified evenly across Lean 4, Coq, and Isabelle (100 instances each). Human experts independently annotated these instances, and we compared their judgments (Correct/Incorrect) with the decisions made by the GPT-4o judge.

Table 6 presents the detailed agreement metrics. We report both Cohen's κ and Accuracy to demonstrate alignment, along with a breakdown of error types (False Positive and False Negative Rates) to quantify reliability. The overall Cohen's κ of 0.88 indicates "almost perfect" agreement according to standard interpretation scales. **GPT-4o demonstrates a high alignment with human experts.** Our detailed error analysis reveals distinct error modes:

- **False Positives (Main Source of Error):** The judge is slightly biased towards leniency (4.3% overall). A typical failure mode involves overlooking missing non-degeneracy assumptions. For example, GPT-4o might accept a formalization that omits a condition such as $x \neq 0$ for a denominator, which technically invalidates the theorem but is often implicit in natural language.

- **False Negatives (Rarer):** False negatives (2.4% overall) typically involve complex syntactic variations, such as using correct but less common library definitions that the model fails to recognize as semantically equivalent to the reference.

Given that the judge's accuracy consistently exceeds 90% and the error rates are balanced (with a slight bias towards leniency), we conclude that GPT-4o is a reliable proxy for evaluating Task-2. To ensure robustness, we explicitly interpret our Task-2 leaderboards with a $\pm 5\%$ uncertainty band.

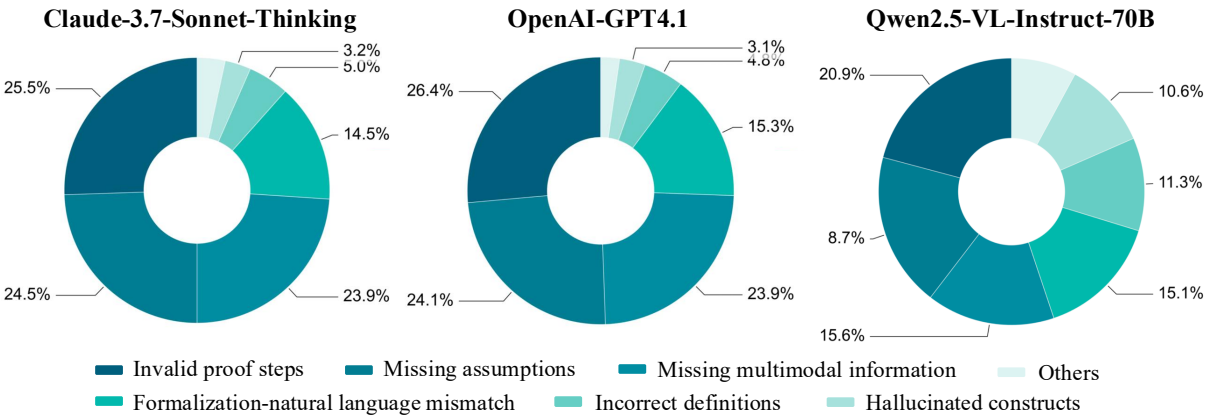


Figure 8: We perform an error analysis on the results of a reasoning model (Claude-3.7-Sonnet-Thinking) and two non-reasoning models (GPT4.1 and Qwen2.5-VL-Instruct-70B), all three being competitive on MATP tasks (Lean 4), with the figure illustrating the seven most frequent error types.

Category	Cohen's κ	Accuracy	False Positive Rate *(Over-lenient)*	False Negative Rate *(Over-strict)*
Overall	**0.88**	**93.3%**	**4.3%**	**2.4%**
Lean 4	0.91	95.0%	3.0%	2.0%
Coq	0.89	94.0%	3.5%	2.5%
Isabelle	0.84	91.0%	6.3%	2.7%

Table 6: Detailed Judge Reliability Statistics (N=300). Comparison between GPT-4o Judge and Human-Majority Labels. The high Cohen's κ (> 0.8) confirms strong agreement. The error analysis breaks down discrepancies into False Positives (judge incorrectly accepts invalid formalization) and False Negatives (judge rejects valid formalization).

Importantly, the performance gaps between the top-performing models (e.g., Claude-3.7/GPT-4.1) and smaller models in our main results significantly exceed this noise margin, ensuring that our core findings and model rankings remain valid.

E Data Decontamination Analysis

To ensure the novelty of our benchmark, we perform an N-gram overlap analysis between MATP-BENCH and the standard libraries of Lean 4 (Mathlib4), Isabelle (AFP), and Coq (Stdlib). We tokenize the formal statements and check for overlapping sequences of length $N = 10$, 20, and 30.

Table 7 presents the results. While we observe minor accidental overlaps for short common phrases ($N = 10$, e.g., variable declarations or imports), the overlap drops to 0.0% for longer sequences ($N = 30$). This confirms that the complete mathematical propositions in our benchmark are unique and not present in the open-source repositories used for model training.

Language	N=10	N=20	N=30
Lean 4	2.4%	0.1%	**0.0%**
Isabelle	1.8%	0.0%	**0.0%**
Coq	2.1%	0.1%	**0.0%**

Table 7: N-gram overlap rates between MATP-BENCH and public formal libraries. N represents the number of consecutive tokens matched. The 0.0% overlap at $N = 30$ verifies that no full statements are leaked.

F Performance Comparison Across Formal Languages

Figure 9 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

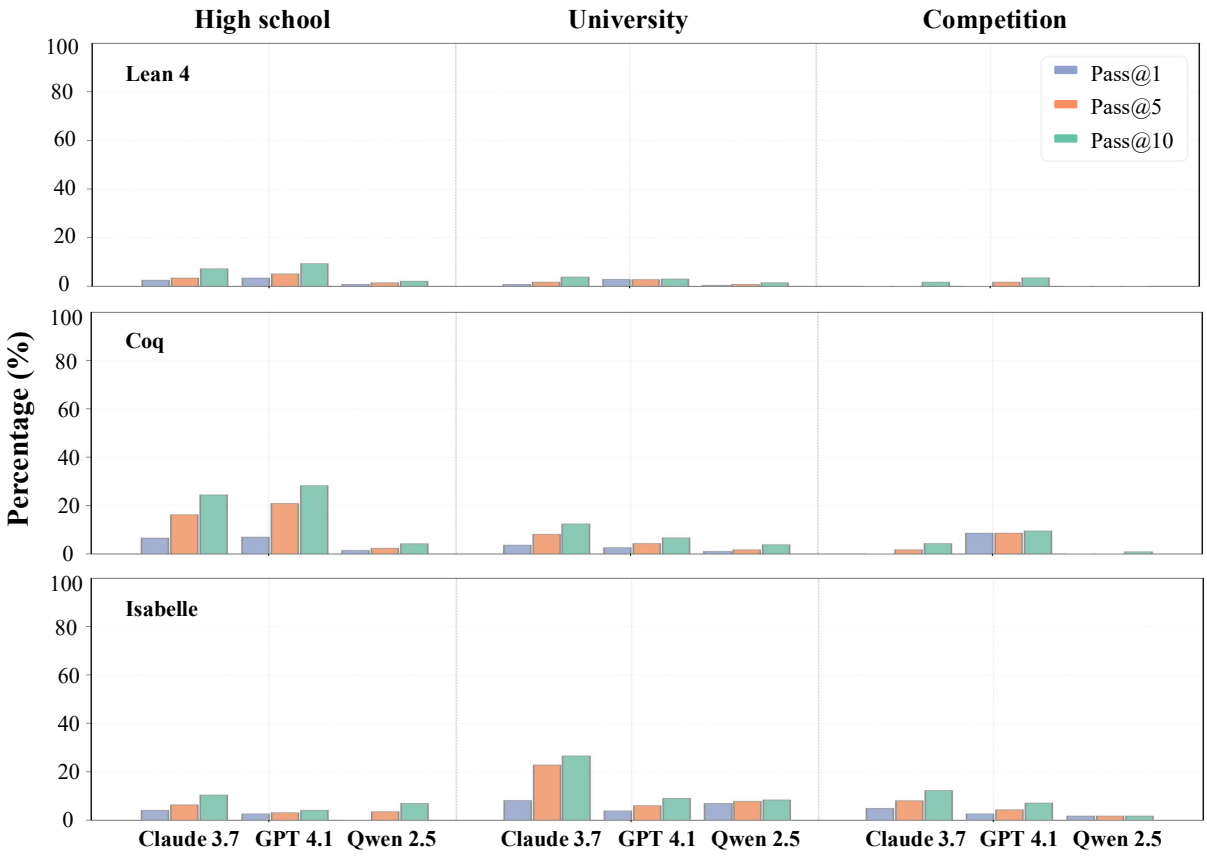


Figure 9: We present the performance of different MLLMs (Gemini-2.0-flash-thinking, OpenAI-GPT4.1, and Qwen2.5-VL-Instruct-70B) on multimodal automated theorem proving task across varying difficulty levels, evaluated using *Pass@1, Pass@5, and Pass@10* metrics.

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.

G Stability Analysis and Ablation Studies

To investigate the sensitivity of our results to hyperparameter settings and prompt strategies, we conduct an ablation study using **GPT-4.1** on the **Lean 4**. We establish a baseline using our default setting (Few-shot, Temperature $T = 0.5$) and compared it against a high-temperature setting ($T = 0.9$) and a Zero-shot prompt variant.

Table 8 summarizes the performance across pass@1, pass@5, and pass@10 metrics. As observed in Table 8, our default configuration ($T = 0.5$) consistently achieves the best performance.

- **Temperature Sensitivity:** Increasing the temperature to $T = 0.9$ causes a performance degradation (e.g., Task 2 Pass@1 drops from 24.14% to 19.50%). While higher temperatures theoretically encourage diversity, in formal reasoning, they often induce syntactic hallucinations or invalid logical leaps that fail rigorous verification. $T = 0.5$ proves to be the optimal sweet spot, balancing exploration with adherence to formal rules.

- **Impact of Few-shot Learning:** The removal of in-context examples (Zero-shot) results in a catastrophic performance drop (e.g., Task 1 Pass@10 falls to 1.10%). This underscores that MLLMs rely heavily on few-shot demonstrations to grasp the strict syntax, import conventions, and structural requirements of Lean 4 formalization.

Task	Setting	Temp (T)	Pass@1	Pass@5	Pass@10
Task 1	**Few-shot (Default)**	**0.5**	**2.08%**	**3.40%**	**5.71%**
	Few-shot	0.9	1.65%	2.95%	4.82%
	Zero-shot	0.5	0.42%	0.85%	1.10%
Task 2	**Few-shot (Default)**	**0.5**	**24.14%**	**42.92%**	**47.58%**
	Few-shot	0.9	19.50%	38.20%	43.10%
	Zero-shot	0.5	8.60%	15.30%	18.25%

Table 8: **Stability Analysis on GPT-4.1 (Lean 4).** We compare our default setting (Few-shot, $T = 0.5$) against a high-temperature setting ($T = 0.9$) and a Zero-shot variant. The results indicate that $T = 0.5$ yields the optimal balance for reasoning accuracy. Increasing temperature to 0.9 degrades performance due to hallucinations, while removing few-shot examples (Zero-shot) leads to significant failure in adhering to formal syntax.

H Prompts

Figures 10 and 11 outline the prompts 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 12 presents the prompt for multimodal theorem formalization task. The prover receives the multimodal question, and is required to formalize it into a precise theorem.

I 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.

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 10: 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 11: 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 12: Prompt for multimodal theorem formalization task.

Combinatorial Geometry on a Regular Icosahedron (Problem Q113)

```
import Mathlib.Data.Fin.Basic
import Mathlib.Data.List.Basic
import Mathlib.Data.List.Nodup
import Mathlib.Data.List.Chain
import Mathlib.Data.Set.Basic
import Mathlib.Data.Fintype.Basic
import Mathlib.Data.Finset.Basic
import Mathlib.Data.Finset.Card
namespace IcosahedronPaths
inductive IcosahedronVertex where
  | top
  | upper (i : Fin 5)
  | lower (i : Fin 5)
  | bottom
deriving DecidableEq, Repr
instance : Fintype IcosahedronVertex :=
  Fintype.ofList
    ([IcosahedronVertex.top] ++
     (List.map IcosahedronVertex.upper (List.finRange 5)) ++
     (List.map IcosahedronVertex.lower (List.finRange 5)) ++
     [IcosahedronVertex.bottom])
    (by intro x; cases x <;> simp [List.mem_append, List.mem_map, List.mem_finRange])
def level (v : IcosahedronVertex) : Nat :=
  match v with
    | .top => 0
    | .upper _ => 1
    | .lower _ => 2
    | .bottom => 3
(truncated)
```

Figure 13: Case study of formalizing complex 3D geometries beyond standard libraries for Problem Q113 shown in Figure 6 (part 1).

Combinatorial Geometry on a Regular Icosahedron (Problem Q113)

```
def isAdjacent (v w : IcosahedronVertex) : Prop :=
    (v ≠ w) ∧
    match v, w with
     | .top, .upper _ => True
     | .upper _, .top => True
     | .upper i, .upper j =>
       (i.val + 1) % 5 = j.val ∨ (j.val + 1) % 5 = i.val
     | .upper i, .lower j =>
        i = j ∨ i.val = (j.val + 1) % 5
     | .lower j, .upper i =>
        i = j ∨ i.val = (j.val + 1) % 5
     | .lower i, .lower j =>
       (i.val + 1) % 5 = j.val ∨ (j.val + 1) % 5 = i.val
     | .lower _, .bottom => True
     | .bottom, .lower _ => True
     | _, _ => False
def isValidStep (v₁ v₂ : IcosahedronVertex) : Prop :=
    isAdjacent v₁ v₂ ∧ (level v₂ = level v₁ ∨ level v₂ = level v₁ + 1)
def isIcosahedronPath (p : List IcosahedronVertex) : Prop :=
    p.head? = some .top ∧
    p.getLast? = some .bottom ∧
    p.Nodup ∧
    List.IsChain isValidStep p
theorem numPathsFromTopToBottom_eq_810 :
    let PathsSet : Set (List IcosahedronVertex) :=
      { p | isIcosahedronPath p }
    ∃ (finitePaths : Finset (List IcosahedronVertex)),
      ↑finitePaths = PathsSet ∧ finitePaths.card = 810 := by sorry
end IcosahedronPaths
```

Figure 14: Case study of formalizing complex 3D geometries beyond standard libraries for Problem Q113 shown in Figure 6 (part 2).

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 15: Prompt for LLM-as-Judge.