Geoint-R1: Formalizing Multimodal Geometric Reasoning with Dynamic Auxiliary Constructions

Jingxuan Wei[1,3], Caijun Jia[1,3], Qi Chen[1,3], Honghao He[1,3], Linzhuang Sun[1,3], Conghui He[2], Lijun Wu[2], Bihui Yu[1,3], Cheng Tan[2*]

[1]Shenyang institute of computing technology, Chinese academy of sciences
[2] Shanghai AI Laboratory [3]University of Chinese Academy of Sciences

weijingxuan20@mails.ucas.edu.cn, chengtan9907@gmail.com

Abstract

Mathematical geometric reasoning is essential for scientific discovery and educational development, requiring precise logic and rigorous formal verification. While recent advances in Multimodal Large Language Models (MLLMs) have improved reasoning tasks, existing models typically struggle with formal geometric reasoning, particularly when dynamically constructing and verifying auxiliary geometric elements. To address these challenges, we introduce Geoint-R1, a multimodal reasoning framework designed to generate formally verifiable geometric solutions from textual descriptions and visual diagrams. Geoint-R1 uniquely integrates auxiliary elements construction, formal reasoning represented via Lean4, and interactive visualization. To systematically evaluate and advance formal geometric reasoning, we propose the Geoint benchmark, comprising 1,885 rigorously annotated geometry problems across diverse topics such as plane, spatial, and solid geometry. Each problem includes structured textual annotations, precise Lean4 code for auxiliary constructions, and detailed solution steps verified by experts. Extensive experiments demonstrate that Geoint-R1 significantly surpasses existing multimodal and math-specific reasoning models, particularly on challenging problems requiring explicit auxiliary element constructions.

Introduction

Mathematical geometric reasoning is fundamental for cognitive, educational, and scientific development, requiring precise logical inference and formal verification. Recent advancements in MLLMs have shown promising results on various reasoning tasks (Yang et al. 2023; Tan et al. 2024; Wei et al. 2024; Wu et al. 2023). However, existing approaches primarily focus on raw textual or visual geometry problems, lacking capabilities in rigorous reasoning and failing to dynamically construct and visualize necessary auxiliary elements in complex multimodal geometric scenarios (Lu et al. 2024; Qiao et al. 2024; Zhang et al. 2024b).

We introduce the formal geometric reasoning task, which involves generating complete and verifiable geometric solutions from multimodal inputs consisting of textual descriptions and geometric diagrams. As illustrated in Figure 1, this task fundamentally differs from existing geometric tasks by requiring not only semantic correctness but also the identifying, formal constructing, and visually demonstrating auxiliary elements. This introduces specific core challenges: **(1) accurately constructing auxiliary lines essential to solving geometric problems, and (2) generating formally verifiable reasoning steps supported by visual feedback that clearly aligns with geometric structures and user intent**.

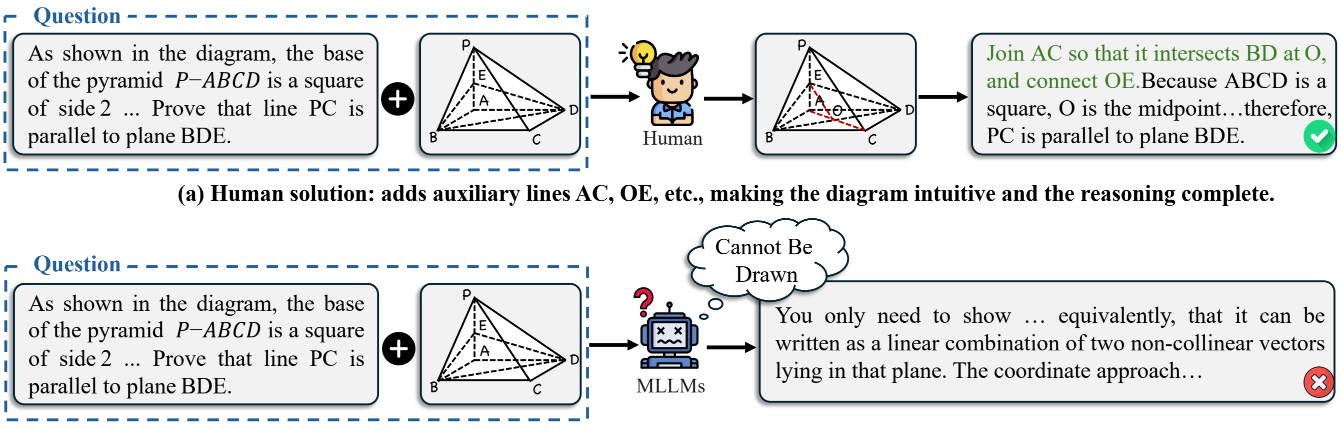
To facilitate progress in formal multimodal geometric reasoning, we introduce **Geoint**, a comprehensive benchmark dataset explicitly designed for formal geometric problem-solving. Geoint encompasses 1,885 carefully curated geometric questions across diverse categories including plane, spatial, and solid geometry problems. Each problem is richly annotated with both structured textual descriptions and accompanying visual diagrams to support multimodal understanding. Furthermore, Geoint leverages the Lean 4 proof assistant to formally represent geometric elements and relationships, enabling rigorous and complete formal reasoning within a verifiable framework.

Building upon the Geoint benchmark, we propose **Geoint-R1**, a geometric reasoning framework characterized by its capability to generate formally verifiable geometric solutions with dynamic auxiliary constructions. Geoint-R1 employs a two-stage training paradigm combining supervised fine-tuning with reinforcement learning guided by a verification reward model that assigns rewards based on correctness, accuracy of auxiliary constructions, and adherence to structured output formats. Additionally, a curriculum learning strategy further enhances the model's ability to handle increasingly complex geometric reasoning tasks.

Our main contributions are summarized as follows:

- We introduce the formal geometric reasoning task along with the Geoint benchmark, a rigorously annotated dataset uniquely integrating structured textual annotations and visual auxiliary line constructions within a complete and verifiable framework.
- We propose the Geoint-R1 framework, an advanced multimodal geometric reasoning model featuring automated identification, construction, and visualization of auxiliary lines within a unified SFT and RL-based pipeline.
- Extensive experiments demonstrate the superior performance of Geoint-R1 compared to state-of-the-art multimodal reasoning models, validating the efficacy of for-

[*] Corresponding author.



As shown in the diagram, the base of the pyramid $P\text{–}ABCD$ is a square of side 2 ... Prove that line PC is parallel to plane BDE.

Join AC so that it intersects BD at O, and connect OE.Because ABCD is a square, O is the midpoint…therefore, PC is parallel to plane BDE.

(a) Human solution: adds auxiliary lines AC, OE, etc., making the diagram intuitive and the reasoning complete.

As shown in the diagram, the base of the pyramid $P\text{–}ABCD$ is a square of side 2 ... Prove that line PC is parallel to plane BDE.

You only need to show … equivalently, that it can be written as a linear combination of two non-collinear vectors lying in that plane. The coordinate approach…

(b) MLLM solution: unable to draw the diagram, and the reasoning is incomplete.

Figure 1: Comparison between human and MLLM solutions to geometric problems requiring auxiliary constructions.

mal auxiliary construction, verification reward model, and training strategies.

Related Work

Mathematical Benchmarks

Recent advances in mathematical reasoning benchmarks have produced a series of multimodal datasets for evaluating large models on mathematical problems with visual content. MathVista (Lu et al. 2024), We-Math (Qiao et al. 2024), MathVerse (Zhang et al. 2024b), and Polymath (Gupta et al. 2024) cover a broad spectrum of topics ranging from plane and solid geometry to pattern recognition, and adopt diverse formats such as multiple-choice, open-ended, and proof-style questions. Many of these benchmarks, emphasize stepwise evaluation and fine-grained error analysis, while MathScape (Zhou et al. 2024), MM-MATH (Sun et al. 2024), and MMSciBench (Ye et al. 2025) further introduce hierarchical or process-based assessments, supporting deeper insight into multi-step reasoning and concept coverage. For spatial and geometric reasoning, SolidGeo (Wang et al. 2025c) provides a collection of annotated three-dimensional geometry problems that enable analysis of spatial understanding. However, these datasets do not provide formali representations that support stepwise visual reasoning or interactive geometric constructions within a complete framework. In contrast, our work constructs a benchmark that links geometry problem to formalized reasoning steps and visualizations of auxiliary constructions, enabling explicit modeling of the reasoning process through the dynamic auxiliary constructions.

Visual Reasoning

Visual reasoning has become a core capability for multimodal models, enabling them to perform complex tasks that require iterative image manipulation. Recent works such as MM-REACT (Yang et al. 2023), VisuoThink (Wang et al. 2025d), CMMCoT (Zhang et al. 2025b), VGR (Wang et al. 2025a), Seg-Zero (Liu et al. 2025a), and VisionReasoner (Liu et al. 2025b) propose diverse frameworks for integrating vision-language models, tool-use, and reinforcement learning to improve visual question answering, multi-image reasoning, and structural scene understanding. Methods like MMFactory (Fan, Rahman, and Sigal 2024), DiagramAgent (Wei et al. 2025), MathCoder-VL (Wang et al. 2025b), and PyVision (Zhao et al. 2025) introduce agent-based or programmatic approaches to visual manipulation, supporting flexible tool selection, diagram editing, and code-based image operations. Other approaches leverage explicit reasoning chains and learning paradigms, as seen in PROGRM (Zhang et al. 2025a), GoT (Fang et al. 2025), Show-o2 (Xie, Yang, and Shou 2025), ControlThinker (Han et al. 2025), and T2I-R1 (Jiang et al. 2025), which further enhance visual generation, multi-step planning, and semantic control through supervised or reinforcement learning strategies. However, they are not designed for precise mathematical problems and do not support stepwise auxiliary construction or visual feedback in formal geometric reasoning.

Methodology

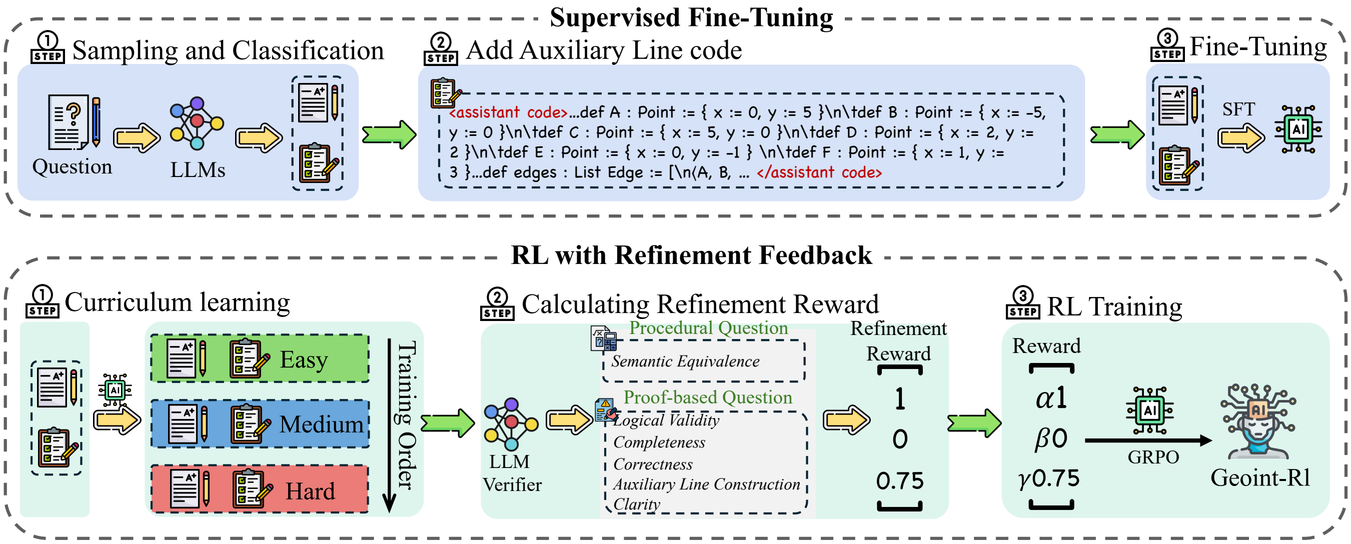
Our goal is to develop a model $\mathcal{F}_\theta$ that, given a textual description T_i and a visual diagram $I_i$of a geometry problem, generates a complete and formally verifiable solution. A solution consists of a sequence of reasoning steps P_i, a final answer A_i, and, when necessary, a set of auxiliary constructions C_i represented in a formal language. Formally, the model predicts the tuple (P_i, C_i, A_i):

$$(P_i, C_i, A_i) = \mathcal{F}_\theta(T_i, I_i). \quad (1)$$

To achieve this, we introduce Geoint-R1, a framework trained in two main stages: Supervised Fine-Tuning (SFT) and Reinforcement Learning with an verification reward model. The overall architecture is illustrated in Figure 2.

Stage 1: Supervised Fine-Tuning

The initial stage aims to teach the base model the fundamental structure of geometric solutions and the syntax of formal constructions. Given the training dataset $\mathcal{D}$ =



```
<assistant code>...def A : Point := { x := 0, y := 5 }\n\tdef B : Point := { x := -5,
y := 0 }\n\tdef C : Point := { x := 5, y := 0 }\n\tdef D : Point := { x := 2, y :=
2 }\n\tdef E : Point := { x := 0, y := -1 } \n\tdef F : Point := { x := 1, y :=
3 }...def edges : List Edge := [\n⟨A, B, ... </assistant code>
```

Figure 2: The Geoint-R1 architecture and training pipeline.

$\{(T_i, I_i, P_i^{ref}, C_i^{ref}, A_i^{ref})\}$, where superscripts denote the reference solution, the objective is to maximize the likelihood of generating the correct solution.

The model $\mathcal{F}_\theta$ is fine-tuned by minimizing the standard negative log-likelihood (NLL) loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{N} \log P(P_i^{ref}, C_i^{ref}, A_i^{ref} | T_i, I_i; \theta), \quad (2)$$

where for problems without auxiliary lines we simply set $C_i^{ref} = \emptyset$. This step uses standard teacher-forcing to ground the model in correct proof structure and Lean4 syntax.

Stage 2: RL with Verification Reward Model

While SFT provides a strong foundation, it may not sufficiently optimize for semantic correctness or logical rigor. The second stage refines the policy of the SFT model, denoted as π_θ, using reinforcement learning guided by a predefined verification reward model. First, we re-apply the same reject-sampling procedure on the SFT outputs to curate a training set where the model still has room for improvement. Each problem in this curated set receives a difficulty score $d_i = 1 - \frac{1}{K}\sum_k \delta_{i,k}$ where $\delta_{i,k}$ is the indicator function for the k-th candidate solution being correct. We sort problems by ascending d_i, implementing a curriculum learning in each sampling that presents easier instances before harder ones.

The RL objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{(T_i, I_i)\sim\mathcal{D}} \left[\mathbb{E}_{(P_i, C_i, A_i)\sim\pi_\theta(\cdot|T_i, I_i)} [R(P_i, C_i, A_i)] \right]. \quad (3)$$

At each step, a single roll-out is sampled for a problem, scored by our verification model into a scalar reward $R(\cdot)$, and used to update θ via GRPO (Shao et al. 2024). The curriculum ordering ensures stable progression from simpler proof patterns to more complex auxiliary constructions.

Verification Reward Model

We design a reward function $R(\cdot)$ that evaluates model outputs with fine-grained feedback on solution quality. Concretely, $R(\cdot)$ is a weighted sum of three components:

Correctness For answer-based questions, the accuracy $\text{acc} \in \{0, 1\}$ is computed by exactly matching with the reference answer. For proof-based questions, the accuracy $\text{acc} \in [0, 1]$ is computed by the GPT-4o's evaluation. We further modulate the correctness score by the accuracy of the auxiliary line construction:

$$F_{corr}(x) = \begin{cases} \min(1, \text{acc} + \lambda), & \text{if auxiliary are correct} \\ \max(0, \text{acc} - \lambda), & \text{otherwise} \end{cases} \quad (4)$$

where the constant λ boosts or penalizes the continuous score depending on auxiliary-line correctness.

Auxiliary Line We award $F_{aux} = 1$ whenever the model exactly matches the reference set of auxiliary constructions (i.e. all needed lines are both identified and precisely encoded in Lean4), and $F_{aux} = 0$ otherwise.

Format To ensure the model's responses can be unambiguously parsed by downstream systems, we require a strict think-answer wrapper. $F_{fmt}(x) = 1$ if the output format matches the required template; 0 otherwise.

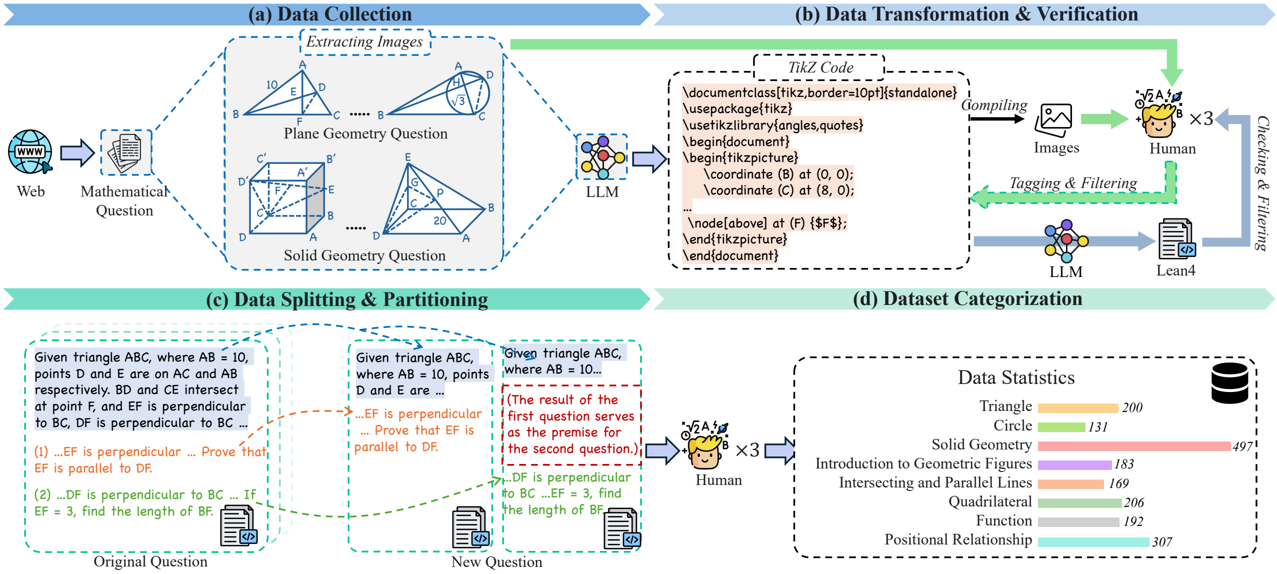
We then combine these sub-rewards into a single scalar:

$$R(x) = \alpha F_{corr}(x) + \beta F_{aux}(x) + \gamma F_{fmt}(x), \quad (5)$$

where α, β, and γ are hyperparameters balancing the different aspects of solution quality.

Geoint Dataset

We present **Geoint**, a multimodal benchmark designed for formal and interactive geometric problem solving. Geoint



```
\documentclass[tikz,border=10pt]{standalone}
\usepackage{tikz}
\usetikzlibrary{angles,quotes}
\begin{document}
\begin{tikzpicture}
    \coordinate (B) at (0, 0);
    \coordinate (C) at (8, 0);
...
 \node[above] at (F) {$F$};
\end{tikzpicture}
\end{document}
```

Given triangle ABC, where AB = 10, points D and E are on AC and AB respectively. BD and CE intersect at point F, and EF is perpendicular to BC, DF is perpendicular to BC ...

(1) ...EF is perpendicular ... Prove that EF is parallel to DF.

(2) ...DF is perpendicular to BC ... If EF = 3, find the length of BF.

Given triangle ABC, where AB = 10, points D and E are ...

...EF is perpendicular ... Prove that EF is parallel to DF.

Given triangle ABC, where AB = 10...

(The result of the first question serves as the premise for the second question.)

...DF is perpendicular to BC ...EF = 3, find the length of BF.

Figure 3: The overview of the Geoint dataset construction pipeline.

comprises a diverse set of geometry questions. A distinguishing feature of Geoint is its detailed annotation of auxiliary constructions—such as dynamically added auxiliary lines. At each geometric reasoning step, enabling explicit modeling and visualization of the problem-solving process. The dataset construction pipeline is illustrated in Figure 3.

Data Collection We begin by sourcing a wide range of geometry problems from publicly available web resources. All problems are manually reviewed and categorized by knowledge type into two main domains: plane and solid geometry.

Data Transformation & Verification For each problem, we extract associated images and convert diagrams into TikZ code. The TikZ code is compiled into diagram and compared against the original, with quality rated by human annotators. For high-quality cases, we employ the DeepSeek-R1 (Guo et al. 2025) to translate TikZ code into Lean4 formal proofs, which are then subject to manual verification for correctness. The resulting Lean4 code and corresponding TikZ diagrams are attached to each problem's textual description to support formalized reasoning and visualization.

Data Splitting & Partitioning Each original geometry problem is decomposed into individual question-answer pairs, splitting multi-step problems into separate entries following a "prompt + question" structure. For multi-part problems where one sub-question depends on the result of the previous question, we explicitly include prior conclusions as context in the prompt for the subsequent sub-question.

Dataset Categorization The final dataset is categorized into nine knowledge types: Triangle, Solid Geometry, Quadrilateral, Function, Positional Relationship, Circle, Geometry Figures, and Intersecting/Parallel Lines. Each entry is double-checked for category accuracy and quality.

Table 1: Sample statistics for Geoint.

Statistic	**Auxiliary line**	**No auxiliary line**
Total samples	429	1456
Training samples	325	1289
Testing samples	104	167
Question type		
Answer-based questions	241	956
Proof-based questions	188	500

Expert Annotation Throughout the pipeline, all manual validation and annotation are conducted by three mathematics professionals with master degree, ensuring the integrity and mathematical soundness of the dataset.

Data Analysis

Geoint contains 1,885 question-answer pairs. Table 1 summarizes the overall sample counts, training/testing splits, and question type distribution for each subset. Table 2 reports the distribution across knowledge categories, as well as token statistics for question, answer, and code. These analyses highlight the diversity of Geoint, including the presence of both answer-based and proof-based questions, and a wide range of diagrammatic and formalized solution lengths.

Geoint exhibits a balanced representation of both auxiliary/non-auxiliary line problems across diverse geometric topics. The average answer and code length for auxiliary line problems are significantly higher, reflecting the complexity of formalized visual reasoning. This fine-grained structure provides a strong foundation for evaluating both visual and formal reasoning in multimodal models.

Table 2: Knowledge type and token statistics for Geoint.

Knowledge type	**Auxiliary line**	**No auxiliary line**
Triangle	50	150
Solid Geometry	118	379
Quadrilateral	54	152
Function	20	172
Positional Relationship	90	217
Circle	38	93
Geometric Figures	26	157
Intersecting/Parallel Lines	33	136
Question length (tokens)		
Minimum	30	29
Maximum	1559	4104
Average	173.92	156.79
Answer length (tokens)		
Minimum	88	5
Maximum	2558	4589
Average	468.38	267.2
Code length (tokens)		
Minimum	41	0
Maximum	1149	0
Average	426.31	0

Evaluation Metrics

We employ tailored automatic evaluation metrics to rigorously assess both answer- and proof-based geometric reasoning within Geoint. For each sample, all evaluations are conducted using the DeepSeek-V3 (Liu et al. 2024a), ensuring consistent and reproducible results.

For **answer-based questions**, we extract the final answer from the model's output and compare it with the reference solution to compute accuracy. The evaluation follows a strict grading prompt, considering mathematical equivalence and only awarding credit for correct final results. The scoring mechanism outputs 1 point for correct or equivalent answers, and 0 otherwise. For **proof-based questions**, we adopt a comprehensive process-based evaluation inspired by DeepTheorem (Zhang et al. 2025c). Each proof is automatically scored along five weighted criteria: logical validity (30%), completeness (20%), correctness (20%), auxiliary line construction (20%), and clarity (10%). Sub-scores for each dimension are aggregated into a total score in $[0, 1]$. The prompts used for automated evaluation are shown in Figure 11 and Figure 12 in the Appendix.

Experiment

Setup. In the SFT, we perform full-parameter updates while freezing the vision encoder and multi-modal projector to preserve visual capacity. The maximum input length is set to 4096 tokens, with a per-device batch size of 4, gradient accumulation of 2, and a learning rate of 1e-5 scheduled by cosine annealing with 10% warmup. Training runs for 4 epochs with bf16 mixed precision and DeepSpeed ZeRO-3 parallelization; checkpoints are saved every 100 steps and loss is logged every 10 steps. Reinforcement learning is applied to further optimize the SFT model using rule-based policy optimization. Training data is curated by rejection sampling, generating 8 candidate outputs per sample and filtering trivial or unsolvable cases. Both training stages are implemented using Qwen2.5-VL-7B (Bai et al. 2025), with RL accelerated by FlashAttention v2 and distributed across 8 GPUs via torchrun and DeepSpeed ZeRO-3.

Table 3: Comparison of overall accuracy (%) on answer- and proof-based geometric reasoning tasks.

Model	Answer	Proof	Average
Phi-3.5-V-4B	2.76	18.89	10.83
LLaVA-v1.6-7B	8.71	36.80	22.76
InternLM-XComposer-2.5-7B	14.02	41.45	27.74
Yi-VL-6B	7.93	29.81	18.87
InternVL3-8B	42.68	71.92	57.30
Qwen-VL-2.5-7B	39.94	69.57	54.76
GPT-4o	42.68	**77.99**	60.34
Gemini-1.5-pro	51.53	73.50	62.52
Gemini-1.5-flash	43.59	66.76	55.18
MMR1-Math-v0-7B	49.08	71.93	60.51
Math-LLaVA-13B	10.87	33.83	22.35
Geoint-R1	**57.01**	72.43	**64.72**

Model For comprehensive benchmarking, we compare our approach against a suite of state-of-the-art multimodal and math-specific models, including both open-source and closed-source ones. The open-source multimodal models include Phi-3.5-V-4B (Abdin et al. 2024), LLaVA-v1.6-7B (Liu et al. 2024b), InternLM-XComposer-2.5-7B (Zhang et al. 2024a), Yi-VL-6B (Young et al. 2024), InternVL3-8B (Zhu et al. 2025), and Qwen-VL-2.5-7B (Bai et al. 2025). Closed-source models consist of GPT-4o (Hurst et al. 2024), Gemini-1.5-pro (Team et al. 2024), and Gemini-1.5-flash (Team et al. 2024). In addition, we include specialized math reasoning models such as MMR1-Math-v0-7B (Leng et al. 2025) and Math-LLaVA-13B (Shi et al. 2024). These models are selected based on their recent advances and strong performance in related multimodal and mathematical reasoning tasks, serving as key baselines.

Main Results

Table 3 presents the accuracy of all evaluated models on answer- and proof-based geometric reasoning tasks. At a macro level, Geoint-R1 achieves the highest overall accuracy (64.72%), while most open-source vision-language models remain below 30%, reflecting the general difficulty of the benchmark. For answer-based questions, Geoint-R1 leads with 57.01%, and Gemini-1.5-pro, GPT-4o, and MMR1-Math-v0-7B also perform well, all exceeding 49%. In contrast, proof-based questions are more challenging: GPT-4o achieves the highest accuracy (77.99%), with Geoint-R1 and Gemini-1.5-pro both above 72%, and most open-source models below 42%. Although Geoint-R1 does not surpass the largest proprietary models on proof-based tasks, it remains highly competitive despite its much smaller parameter size (7B), suggesting that model scale may be a factor in achieving top performance on complex proofs. Overall, Geoint-R1 delivers the best average accuracy across both question types, indicating its balanced effectiveness on this challenging multimodal benchmark.

Results on Auxiliary Line Problems

As shown in Figure 4, most models struggle on problems that require auxiliary line construction, with open-source models generally achieving less than 40% accuracy. For answer-based questions, Geoint-R1 achieves the highest score (68.63%), outperforming all other models by a notable margin. In proof-based questions, Geoint-R1 achieves 68.40%, remaining highly competitive and comparable to Gemini-1.5-pro (69.43%), MMR1-Math-v0-7B (67.21%), and just below GPT-4o (74.81%), despite using far fewer parameters. These results underscore Geoint-R1's robust performance in complex geometric reasoning tasks that require explicit visual reasoning capability for both formal answer- and proof-based problems.

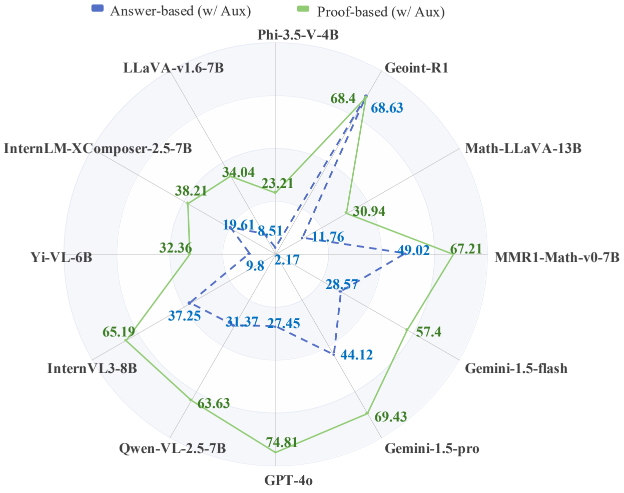


Figure 4: Accuracy (%) on auxiliary line problems.

Results on Non-Auxiliary Line Problems

As shown in Figure 5, all models achieve higher accuracy on non-auxiliary line problems than on auxiliary line tasks, likely because these questions are less complex and do not require explicit geometric constructions. Most closed-source and leading open-source models exceed 45% on answer-based questions, with Geoint-R1 reaching 51.77%, closely matching Gemini-1.5-pro (54.91%), GPT-4o (49.56%), and MMR1-Math-v0-7B (49.11%). For proof-based questions, Geoint-R1 achieves 76.39%, among the top performers together with Gemini-1.5-pro, GPT-4o, and several other math-focused models. While open-source baselines still lag on answer-based questions, the gap is reduced here.

Ablation Study

Table 4 presents the ablation results for Geoint-R1 by removing LLM Verify, reinforcement learning (RL), and curriculum learning (CL) modules. At the macro level, Geoint-R1 achieves the highest average accuracy, and removing any single module leads to a significant drop. The effect is most pronounced when verification reward is removed, especially on auxiliary line problems. On proof-based tasks, Geoint-R1 achieves an average of 72.43%, with all ablated models falling behind, again with the largest drop observed after removing verification reward. Both auxiliary and non-auxiliary line cases show consistent performance degradation across all ablations, demonstrating that these components are each essential. These findings confirm the effectiveness of our modular design for complex geometric reasoning, with each component contributing meaningfully to both answer- and proof-based performance.

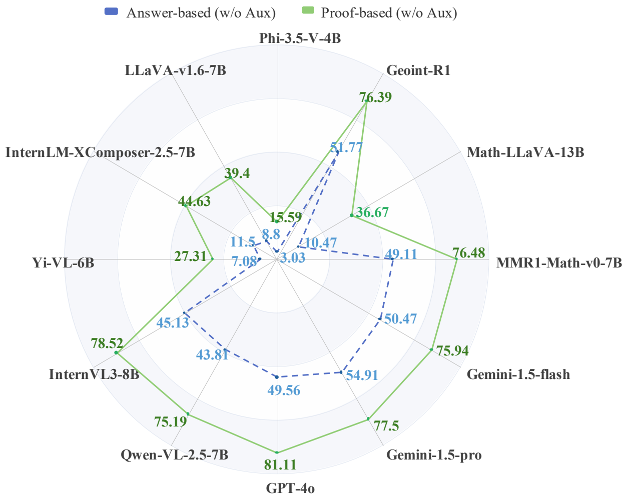


Figure 5: Accuracy (%) on non-auxiliary line problems.

Table 4: Ablation results (%) of Geoint-R1.

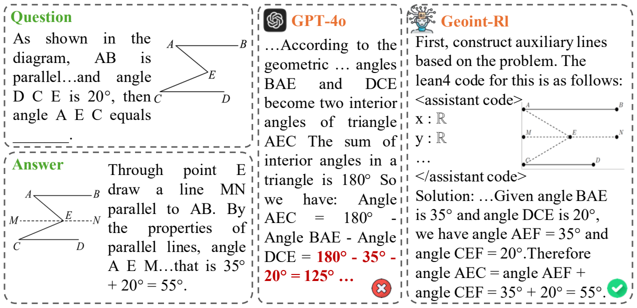
Model	Answer-based			Proof-based			Average
	w/ Aux	w/o Aux	Avg	w/ Aux	w/o Aux	Avg	
Geoint-R1	**68.63**	**51.77**	**57.01**	**68.40**	**76.39**	**72.43**	**64.72**
-w/o Verify	32.35	47.35	42.68	64.15	73.52	68.88	55.78
-w/o RL	37.25	46.46	43.60	62.55	71.48	67.06	55.33
-w/o CL	38.24	49.12	45.73	63.30	72.41	67.90	56.82

Case Studies

We present two representative examples to demonstrate Geoint-R1's advantage in auxiliary line construction and formal reasoning as follows:

Example 1 (Answer-based): As shown in Figure 6, Geoint-R1 constructs a parallel through E and directly applies angle properties to compute $\angle AEC$, while GPT-4o ignores auxiliary lines and arrives at an incorrect result. These results highlight the necessity of explicit auxiliary construction for robust and interpretable geometric reasoning.

Example 2 (Proof-based): Figure 7 considers a spatial proof requiring demonstration that $PC \parallel$ plane BDE in a pyramid. Geoint-R1 solves this by constructing the intersection point O of diagonals, connecting O to E, and proving $OE \parallel PC$, thus delivering a concise and verifiable proof. In contrast, GPT-4o employs vector operations but fails to establish the required geometric relationships.



Question

As shown in the diagram, AB is parallel…and angle D C E is 20°, then angle A E C equals ________.

Answer

Through point E draw a line MN parallel to AB. By the properties of parallel lines, angle A E M…that is 35° + 20° = 55°.

GPT-4o

…According to the geometric … angles BAE and DCE become two interior angles of triangle AEC The sum of interior angles in a triangle is 180° So we have: Angle AEC = 180° - Angle BAE - Angle DCE = **180° - 35° - 20° = 125° …**

Geoint-Rl

First, construct auxiliary lines based on the problem. The lean4 code for this is as follows:

<assistant code>

x : ℝ

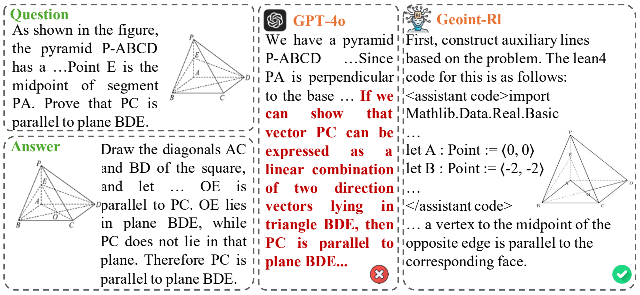
y : ℝ

…

</assistant code>

Solution: …Given angle BAE is 35° and angle DCE is 20°, we have angle AEF = 35° and angle CEF = 20°.Therefore angle AEC = angle AEF + angle CEF = 35° + 20° = 55°.

Figure 6: Auxiliary lines support accurate planar inference.



Question

As shown in the figure, the pyramid P-ABCD has a …Point E is the midpoint of segment PA. Prove that PC is parallel to plane BDE.

Answer

Draw the diagonals AC and BD of the square, and let … OE is parallel to PC. OE lies in plane BDE, while PC does not lie in that plane. Therefore PC is parallel to plane BDE.

GPT-4o

We have a pyramid P-ABCD …Since PA is perpendicular to the base … **If we can show that vector PC can be expressed as a linear combination of two direction vectors lying in triangle BDE, then PC is parallel to plane BDE...**

Geoint-Rl

First, construct auxiliary lines based on the problem. The lean4 code for this is as follows:

<assistant code>import Mathlib.Data.Real.Basic

…

let A : Point := ⟨0, 0⟩

let B : Point := ⟨-2, -2⟩

…

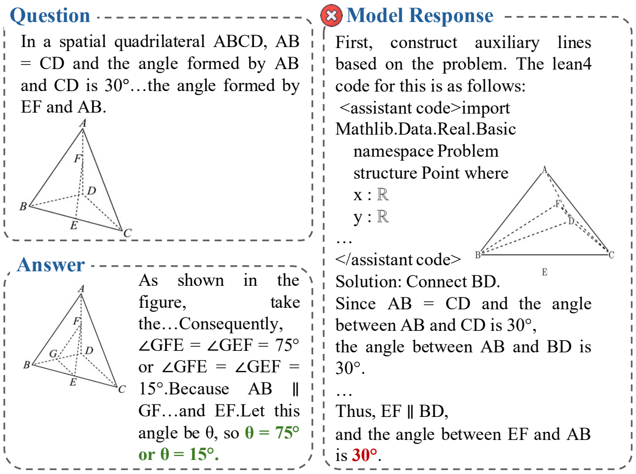
</assistant code>

… a vertex to the midpoint of the opposite edge is parallel to the corresponding face.

Figure 7: Auxiliary construction for spatial proof.

Error Analysis

While Geoint-R1 achieves strong results overall, several typical errors still arise. As shown in Figure 8, the model may misapply geometric theorems. For example, it wrongly infers that $EF \parallel BD$ simply because E and F are midpoints, neglecting the precise conditions required by the midline theorem and failing to construct key auxiliary points.



Question

In a spatial quadrilateral ABCD, AB = CD and the angle formed by AB and CD is 30°…the angle formed by EF and AB.

Answer

As shown in the figure, take the…Consequently, ∠GFE = ∠GEF = 75° or ∠GFE = ∠GEF = 15°.Because AB ∥ GF…and EF.Let this angle be θ, so **θ = 75° or θ = 15°.**

Model Response

First, construct auxiliary lines based on the problem. The lean4 code for this is as follows:

<assistant code>import Mathlib.Data.Real.Basic

namespace Problem

structure Point where

x : ℝ

y : ℝ

…

</assistant code>

Solution: Connect BD.

Since AB = CD and the angle between AB and CD is 30°,

the angle between AB and BD is 30°.

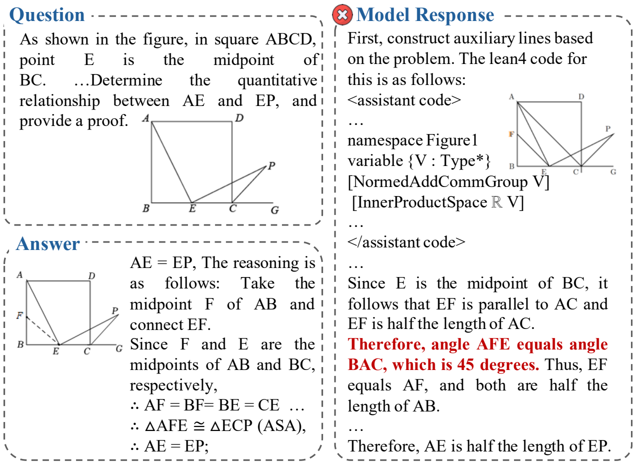
…

Thus, EF ∥ BD,

and the angle between EF and AB is **30°**.

Figure 8: Spatial geometry: mistaken parallelism inference due to midline theorem misapplication.

Figure 9 shows a planar geometry case where the model constructs the auxiliary line correctly but fails to establish the right proportional relationships. The logical chain is incomplete, and the reasoning cannot justify the equality between AE and EP. As illustrated in Figure 10, variable confusion and computational mistakes can propagate throughout a multi-step solution. Here, the model mislabels midpoints and ultimately arrives at an impossible value for $\sin\theta > 1$. These examples demonstrate that Geoint-R1, while powerful, still struggles with strict theorem application, maintaining logical completeness, and tracking symbolic accuracy—especially in complex geometric scenarios.



Question

As shown in the figure, in square ABCD, point E is the midpoint of BC. …Determine the quantitative relationship between AE and EP, and provide a proof.

Answer

AE = EP, The reasoning is as follows: Take the midpoint F of AB and connect EF.

Since F and E are the midpoints of AB and BC, respectively,

∴ AF = BF= BE = CE …

∴ △AFE ≅ △ECP (ASA),

∴ AE = EP;

Model Response

First, construct auxiliary lines based on the problem. The lean4 code for this is as follows:

<assistant code>

…

namespace Figure1

variable {V : Type*}

[NormedAddCommGroup V]

[InnerProductSpace ℝ V]

…

</assistant code>

…

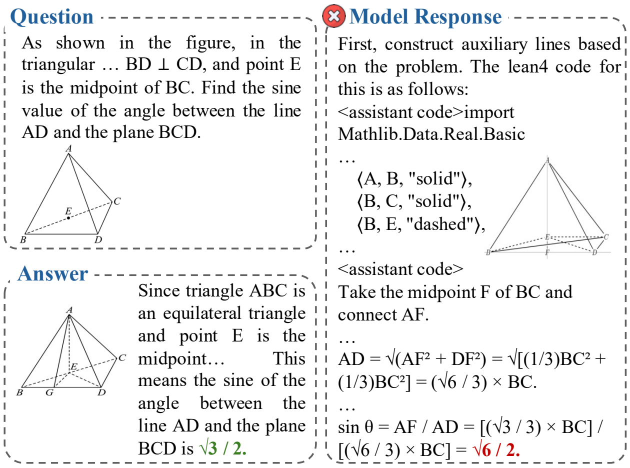
Since E is the midpoint of BC, it follows that EF is parallel to AC and EF is half the length of AC.

Therefore, angle AFE equals angle BAC, which is 45 degrees. Thus, EF equals AF, and both are half the length of AB.

…

Therefore, AE is half the length of EP.

Figure 9: Planar geometry: incomplete reasoning after correct auxiliary construction.



Question

As shown in the figure, in the triangular … BD ⊥ CD, and point E is the midpoint of BC. Find the sine value of the angle between the line AD and the plane BCD.

Answer

Since triangle ABC is an equilateral triangle and point E is the midpoint… This means the sine of the angle between the line AD and the plane BCD is **√3 / 2.**

Model Response

First, construct auxiliary lines based on the problem. The lean4 code for this is as follows:

<assistant code>import Mathlib.Data.Real.Basic

…

⟨A, B, "solid"⟩,

⟨B, C, "solid"⟩,

⟨B, E, "dashed"⟩,

…

<assistant code>

Take the midpoint F of BC and connect AF.

…

AD = √(AF² + DF²) = √[(1/3)BC² + (1/3)BC²] = (√6 / 3) × BC.

…

sin θ = AF / AD = [(√3 / 3) × BC] / [(√6 / 3) × BC] = **√6 / 2.**

Figure 10: Symbolic confusion and formula error in a multi-step planar problem.

Conclusion

In this paper, we introduce Geoint-R1, a multimodal geometric reasoning framework designed to generate formally verifiable solutions with dynamically constructed auxiliary lines. Geoint-R1 integrates supervised fine-tuning, reinforcement learning with verification reward and curriculum learning to effectively handle increasingly complex geometric reasoning tasks. Furthermore, we present the Geoint benchmark, a multimodal geometry dataset incorporating formal Lean4 code annotations, consisting of 1,885 high-quality problems covering diverse geometry categories.

Comprehensive experimental evaluations show that Geoint-R1 achieves superior performance over existing state-of-the-art multimodal and mathematical reasoning models, especially excelling in tasks that necessitate explicit auxiliary constructions. Our framework provides a robust foundation for future research in formalized multimodal reasoning and demonstrates significant potential for advancing applications in geometric problem-solving.

References

Abdin, M.; Aneja, J.; Awadalla, H.; Awadallah, A.; Awan, A. A.; Bach, N.; Bahree, A.; Bakhtiari, A.; Bao, J.; Behl, H.; et al. 2024. Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone. *arXiv preprint arXiv:2404.14219*.

Bai, S.; Chen, K.; Liu, X.; Wang, J.; Ge, W.; Song, S.; Dang, K.; Wang, P.; Wang, S.; Tang, J.; et al. 2025. Qwen2. 5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Fan, W.-C.; Rahman, T.; and Sigal, L. 2024. MMFactory: A Universal Solution Search Engine for Vision-Language Tasks. *arXiv preprint arXiv:2412.18072*.

Fang, R.; Duan, C.; Wang, K.; Huang, L.; Li, H.; Yan, S.; Tian, H.; Zeng, X.; Zhao, R.; Dai, J.; et al. 2025. Got: Unleashing reasoning capability of multimodal large language model for visual generation and editing. *arXiv preprint arXiv:2503.10639*.

Guo, D.; Yang, D.; Zhang, H.; Song, J.; Zhang, R.; Xu, R.; Zhu, Q.; Ma, S.; Wang, P.; Bi, X.; et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Gupta, H.; Verma, S.; Anantheswaran, U.; Scaria, K.; Parmar, M.; Mishra, S.; and Baral, C. 2024. Polymath: A challenging multi-modal mathematical reasoning benchmark. *arXiv preprint arXiv:2410.14702*.

Han, F.; Jiao, Y.; Chen, S.; Xu, J.; Chen, J.; and Jiang, Y.-G. 2025. ControlThinker: Unveiling Latent Semantics for Controllable Image Generation through Visual Reasoning. *arXiv preprint arXiv:2506.03596*.

Hurst, A.; Lerer, A.; Goucher, A. P.; Perelman, A.; Ramesh, A.; Clark, A.; Ostrow, A.; Welihinda, A.; Hayes, A.; Radford, A.; et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Jiang, D.; Guo, Z.; Zhang, R.; Zong, Z.; Li, H.; Zhuo, L.; Yan, S.; Heng, P.-A.; and Li, H. 2025. T2i-r1: Reinforcing image generation with collaborative semantic-level and token-level cot. *arXiv preprint arXiv:2505.00703*.

Leng, S.; Wang, J.; Li, J.; Zhang, H.; et al. 2025. MMR1: Advancing the Frontiers of Multimodal Reasoning. https://github.com/LengSicong/MMR1.

Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; Lu, C.; Zhao, C.; Deng, C.; Zhang, C.; Ruan, C.; et al. 2024a. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

Liu, H.; Li, C.; Li, Y.; and Lee, Y. J. 2024b. Improved baselines with visual instruction tuning. In *CVPR*, 26296–26306.

Liu, Y.; Peng, B.; Zhong, Z.; Yue, Z.; Lu, F.; Yu, B.; and Jia, J. 2025a. Seg-zero: Reasoning-chain guided segmentation via cognitive reinforcement. *arXiv preprint arXiv:2503.06520*.

Liu, Y.; Qu, T.; Zhong, Z.; Peng, B.; Liu, S.; Yu, B.; and Jia, J. 2025b. VisionReasoner: Unified Visual Perception and Reasoning via Reinforcement Learning. *arXiv preprint arXiv:2505.12081*.

Lu, P.; Bansal, H.; Xia, T.; Liu, J.; Li, C.; Hajishirzi, H.; Cheng, H.; Chang, K.-W.; Galley, M.; and Gao, J. 2024. MathVista: Evaluating Mathematical Reasoning of Foundation Models in Visual Contexts. In *ICLR*.

Qiao, R.; Tan, Q.; Dong, G.; Wu, M.; Sun, C.; Song, X.; GongQue, Z.; Lei, S.; Wei, Z.; Zhang, M.; et al. 2024. We-math: Does your large multimodal model achieve human-like mathematical reasoning? *arXiv preprint arXiv:2407.01284*.

Shao, Z.; Wang, P.; Zhu, Q.; Xu, R.; Song, J.; Bi, X.; Zhang, H.; Zhang, M.; Li, Y.; Wu, Y.; et al. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Shi, W.; Hu, Z.; Bin, Y.; Liu, J.; Yang, Y.; Ng, S. K.; Bing, L.; and Lee, R. 2024. Math-LLaVA: Bootstrapping Mathematical Reasoning for Multimodal Large Language Models. In *EMNLP*, 4663–4680.

Sun, K.; Bai, Y.; Qi, J.; Hou, L.; and Li, J. 2024. MM-MATH: Advancing Multimodal Math Evaluation with Process Evaluation and Fine-grained Classification. In *Findings of the EMNLP*, 1358–1375.

Tan, C.; Wei, J.; Gao, Z.; Sun, L.; Li, S.; Guo, R.; Yu, B.; and Li, S. Z. 2024. Boosting the power of small multimodal reasoning models to match larger models with self-consistency training. In *ECCV*, 305–322. Springer.

Team, G.; Georgiev, P.; Lei, V. I.; Burnell, R.; Bai, L.; Gulati, A.; Tanzer, G.; Vincent, D.; Pan, Z.; Wang, S.; et al. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*.

Wang, J.; Kang, Z.; Wang, H.; Jiang, H.; Li, J.; Wu, B.; Wang, Y.; Ran, J.; Liang, X.; Feng, C.; et al. 2025a. VGR: Visual Grounded Reasoning. *arXiv preprint arXiv:2506.11991*.

Wang, K.; Pan, J.; Wei, L.; Zhou, A.; Shi, W.; Lu, Z.; Xiao, H.; Yang, Y.; Ren, H.; Zhan, M.; et al. 2025b. MathCoder-VL: Bridging Vision and Code for Enhanced Multimodal Mathematical Reasoning. *arXiv preprint arXiv:2505.10557*.

Wang, P.; Yang, C.; Li, Z.-Z.; Yin, F.; Ran, D.; Tian, M.; Ji, Z.; Bai, J.; and Liu, C.-L. 2025c. SOLIDGEO: Measuring Multimodal Spatial Math Reasoning in Solid Geometry. *arXiv preprint arXiv:2505.21177*.

Wang, Y.; Wang, S.; Cheng, Q.; Fei, Z.; Ding, L.; Guo, Q.; Tao, D.; and Qiu, X. 2025d. Visuothink: Empowering lvlm reasoning with multimodal tree search. *arXiv preprint arXiv:2504.09130*.

Wei, J.; Tan, C.; Chen, Q.; Wu, G.; Li, S.; Gao, Z.; Sun, L.; Yu, B.; and Guo, R. 2025. From Words to Structured

Visuals: A Benchmark and Framework for Text-to-Diagram Generation and Editing. In *CVPR*, 13315–13325.

Wei, J.; Tan, C.; Gao, Z.; Sun, L.; Li, S.; Yu, B.; Guo, R.; and Li, S. Z. 2024. Enhancing human-like multimodal reasoning: a new challenging dataset and comprehensive framework. *Neural Computing and Applications*, 36(33): 20849–20861.

Wu, J.; Gan, W.; Chen, Z.; Wan, S.; and Yu, P. S. 2023. Multimodal large language models: A survey. In *BigData*, 2247–2256. IEEE.

Xie, J.; Yang, Z.; and Shou, M. Z. 2025. Show-o2: Improved Native Unified Multimodal Models. *arXiv preprint arXiv:2506.15564*.

Yang, Z.; Li, L.; Wang, J.; Lin, K.; Azarnasab, E.; Ahmed, F.; Liu, Z.; Liu, C.; Zeng, M.; and Wang, L. 2023. Mm-react: Prompting chatgpt for multimodal reasoning and action. *arXiv preprint arXiv:2303.11381*.

Ye, X.; Li, C.; Chen, S.; Wei, W.; and Tang, X. 2025. MMSciBench: Benchmarking Language Models on Chinese Multimodal Scientific Problems. *arXiv preprint arXiv:2503.01891*.

Young, A.; Chen, B.; Li, C.; Huang, C.; Zhang, G.; Zhang, G.; Wang, G.; Li, H.; Zhu, J.; Chen, J.; et al. 2024. Yi: Open foundation models by 01. ai. *arXiv preprint arXiv:2403.04652*.

Zhang, D.; Zhang, S.; Yang, Z.; Zhu, Z.; Zhao, Z.; Cao, R.; Chen, L.; and Yu, K. 2025a. ProgRM: Build Better GUI Agents with Progress Rewards. *arXiv preprint arXiv:2505.18121*.

Zhang, G.; Zhong, T.; Xia, Y.; Yu, Z.; Li, H.; He, W.; Shu, F.; Liu, M.; She, D.; Wang, Y.; et al. 2025b. Cmmcot: Enhancing complex multi-image comprehension via multi-modal chain-of-thought and memory augmentation. *arXiv preprint arXiv:2503.05255*.

Zhang, P.; Dong, X.; Zang, Y.; Cao, Y.; Qian, R.; Chen, L.; Guo, Q.; Duan, H.; Wang, B.; Ouyang, L.; et al. 2024a. Internlm-xcomposer-2.5: A versatile large vision language model supporting long-contextual input and output. *arXiv preprint arXiv:2407.03320*.

Zhang, R.; Jiang, D.; Zhang, Y.; Lin, H.; Guo, Z.; Qiu, P.; Zhou, A.; Lu, P.; Chang, K.-W.; Qiao, Y.; et al. 2024b. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *ECCV*, 169–186. Springer.

Zhang, Z.; Xu, J.; He, Z.; Liang, T.; Liu, Q.; Li, Y.; Song, L.; Liang, Z.; Zhang, Z.; Wang, R.; et al. 2025c. Deeptheorem: Advancing llm reasoning for theorem proving through natural language and reinforcement learning. *arXiv preprint arXiv:2505.23754*.

Zhao, S.; Zhang, H.; Lin, S.; Li, M.; Wu, Q.; Zhang, K.; and Wei, C. 2025. PyVision: Agentic Vision with Dynamic Tooling. *arXiv preprint arXiv:2507.07998*.

Zhou, M.; Liang, H.; Li, T.; Wu, Z.; Lin, M.; Sun, L.; Zhou, Y.; Zhang, Y.; Huang, X.; Chen, Y.; et al. 2024. Mathscape: Evaluating mllms in multimodal math scenarios through a hierarchical benchmark. *arXiv preprint arXiv:2408.07543*.

Zhu, J.; Wang, W.; Chen, Z.; Liu, Z.; Ye, S.; Gu, L.; Tian, H.; Duan, Y.; Su, W.; Shao, J.; et al. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Appendix

Evaluation Prompts

Answer-based Question Grading Prompt

Now you are playing the role of a strict grading teacher. Your task is to review and score the students’ answers based on the standard answers. Throughout the grading process, you need to be familiar with the following key points:
- Grading is only based on the final answers given by the students to determine their correctness, and it does not require checking whether the intermediate solution steps are correct.
- First, extract the final answer from the students’ solutions and display it in the analysis results. Then, judge whether the answer is correct.
- Based on your analysis results, give the score. When presenting the scoring basis, you should describe it in segments according to the logic of the analysis. The summary of the scoring basis should be placed at the end, and it can be in the following format: ”In conclusion, the student’s answer should receive x points” (x represents the specific score of the student).
- Based on your analysis, give the score and display it in a code block in ”JSON” format.

Scoring ranges: 1 point if the answer is correct or mathematically equivalent, 0 otherwise.
Output format:
[Grading Basis]: [Total Score]: float [JSON]: ”score”: float

Figure 11: Evaluation prompt for answer-based questions.

Proof-based Question Grading Prompt

Your task is to evaluate the quality of the proof based on five criteria and give a score between 0 and 1: logical validity (30%), completeness (20%), correctness (20%), construction of auxiliary lines (20%), and clarity (10%).
Instructions: 1. Analyze the proof step by step.
2. For each criterion:
- Logical Validity: Check if each step follows logically from the previous one. Flag any logical errors.
- Completeness: Verify if all necessary cases and steps are included to prove the theorem.
- Correctness: Confirm if the final conclusion is correct.
- Construction of auxiliary lines: Determine whether auxiliary lines have been successfully constructed and provide the corresponding Lean4 code for the image.
- Clarity: Assess if the proof is clear, unambiguous, and well-explained.

3. Assign a sub-score (0 to 1) for each criterion and compute the total score using the weights: (0.3 × validity) + (0.2 × completeness) + (0.2 × correctness) + (0.2 × construction) + (0.1 × clarity).
4. Provide a brief explanation (2-3 sentences) summarizing any errors or issues and justifying the score.
Final output format:
[JSON]: ”score”: float, ”validity”: float, ”completeness”: float, ”correctness”: float, ”construction”: float, ”clarity”: float, ”explanation”: str

Figure 12: Evaluation prompt for proof-based questions, following the process-based approach of DeepTheorem (Zhang et al. 2025c).