

# W4VE: MULTIMODAL REASONING IN GEOMETRY TASKS

**Ruichen Wang\***   **Siqi Wang\***   **Yuanyu Wang\***   **Yan Wang\***  
 {ruichenw, swang4, helenwan, yanwang5}@andrew.cmu.edu

## ABSTRACT

We study whether vision–language models can truly reason about geometry rather than merely label diagrams. On a MathVerse Zhang et al. (2024b) subset of multiple-choice problems, eight baselines—ranging from text-only LLMs to state-of-the-art VLMs—still struggle with fine-grained spatial relations. We introduce an iterative pipeline that alternates *planning questions*, *VQA fact extraction*, and *belief-state reasoning*. Without additional training, the full GPT-4o variant doubles baseline description quality and raises accuracy to 40 %, while an open-weights Qwen version reaches 34 %. Results suggest that targeted visual querying, not larger language models alone, is the critical driver of geometric reasoning performance.

## 1 [2 POINTS] INTRODUCTION AND PROBLEM DEFINITION (1-1.25 PAGES)

Multimodal technologies have demonstrated remarkable potential in tackling complex real-world tasks, leveraging their powerful capability to integrate visual and textual information. Inspired by this emerging trend, our team was intrigued by the possibility of effectively applying these technologies within the traditionally challenging domain of mathematics. Specifically, we asked: could the rapidly evolving vision-language models (VLMs) yield breakthrough results when applied to geometric mathematical problems?

To clarify our research motivation, we first examined recent developments in mathematical reasoning technologies. Over the past few years, large language models such as GPT-3 and GPT-4 have made significant strides in algebra, logical reasoning, and solving complex text-based problems, exemplified by their excellent performance on benchmark datasets like GSM8K( Forootani (2025) ). These advancements provided essential insights and encouragement for further research. However, our curiosity grew regarding model performance when reasoning tasks expand beyond pure textual logic, involving visual and spatial interpretation of images.

Consequently, our team decided to focus specifically on geometric reasoning tasks within mathematics. Unlike typical text-based reasoning problems, geometry requires precise perception and reasoning about detailed structural and spatial relationships within images—such as points, lines, surfaces, geometric shape combinations, segment lengths, angle measurements, parallel or perpendicular relationships, shapes’ containment and partitioning, overall symmetry, and spatial layouts. Recent advancements in vision-language models, notably GPT-4V and LLaVA-7B, have demonstrated robust general visual comprehension abilities, seemingly offering ideal solutions for this task ( Guo et al. (2023) , OpenAI (2023) ). However, upon deeper literature review, we discovered that even these state-of-the-art models exhibit significant limitations and shortcomings in specific geometric image reasoning tasks, particularly in handling precise spatial details Zhang et al. (2024b).

Further investigation into possible reasons for these shortcomings led us to examine popular geometric datasets such as MathVerse Zhang et al. (2024b) and Geometry3K. We found that while current datasets commonly provide geometric images, question texts, choices, and answers, they lack detailed, explicit, and structured geometric image descriptions. This absence of descriptive context may significantly limit the models’ ability to thoroughly understand image information, thus impairing their performance on detailed geometric reasoning tasks Zhang et al. (2024b).

\* Everyone Contributed Equally – Alphabetical order

Based on these findings, we clearly defined our research goal: If we proactively generate and provide explicit, structured descriptions of geometric images, can this effectively enhance the performance of vision-language models on geometric reasoning tasks? To verify this question, we explicitly formulated two core hypotheses:

### 1.1 RESEARCH HYPOTHESES

- **Caption Benefit Hypothesis:** Providing detailed and explicit image descriptions can significantly enhance model accuracy on geometric problems compared to using only raw images or text.
- **Task-Focused Caption Hypothesis:** Descriptions specifically tailored to focus on visual features directly related to geometric problem-solving (e.g., specific angles, segment relations) will more effectively improve model reasoning capabilities and overall problem-solving performance compared to general image descriptions.

Building on this motivation, our final solution extends the classic perception-reasoning split into an *iterative* three-component loop—*planning* → *targeted visual question answering* → *belief-state reasoning*—so that the system can actively request missing geometric facts before committing to a final answer.

### 1.2 CONTRIBUTIONS

To achieve and validate these hypotheses, our project clearly offers four main contributions:

1. **Proposing and validating a novel multimodal geometric reasoning pipeline:** We explicitly divided the geometric reasoning task into two distinct stages: perception and reasoning.
  - Perception Stage (VLM Stage): Utilizing advanced vision-language models (such as GPT-4V and Qwen-VL) to generate structured descriptions of geometric images.
  - Reasoning Stage (LLM Stage): Combining generated detailed descriptions with problem texts to enable final reasoning and answer generation by large language models (e.g., GPT-4o).
2. **Introducing a task-oriented structured image description generation strategy for geometric problems:** Our method addresses the lack of structured image descriptions in existing datasets by clearly instructing the model-generated descriptions to focus specifically on visual features directly relevant to problem-solving, thereby enhancing the model’s precise understanding of geometric details.
3. **Conducting comprehensive and detailed baseline comparative analyses:** We systematically compared eight different model combinations and strategies, ranging from pure text models (GPT-3.5, GPT-4o, Mistral 7B) and simple multimodal fusion models (CLIP+GPT-3.5, BLIP+GPT-3.5) to advanced vision-language model combinations (GPT-4V, Qwen-VL, InternLM). Through our experiments, we clearly showcased differences in description generation capabilities, reasoning capabilities, and overall accuracy among these models.
4. **Providing clear and in-depth experimental insights and analysis:** By analyzing Chain-of-Thought (CoT) reasoning scores, description generation scores, and their combined overall accuracy, we conclusively demonstrated that detailed image descriptions have a significant positive impact on overall geometric reasoning accuracy. Furthermore, our experimental design allows us to identify whether errors originate from the perception (mis-perception) or reasoning (miss reasoning) stage, providing valuable guidance for future model accuracy improvements.

## 2 [5 POINTS] RELATED WORK AND BACKGROUND (5 PAPERS PER PERSON)

**Related Datasets** Several recent works have advanced the field of mathematical and geometric problem-solving using multi-modal models. The MATH dataset Hendrycks et al. (2021) provides a large-scale benchmark focused on text-based mathematical problem-solving, while MathVista Lu et al. (2023) expands this space by evaluating multi-modal models on visual math problems that combine diagrams with textual descriptions. GeoEval Zhang et al. (2024a) introduces a specialized benchmark for geometric reasoning, designed to test the ability of the models to interpret complex figures and solve geometry questions. MathVerse Zhang et al. (2024b) further investigates whether multi-modal LLMs can truly “see” and reason over diagrams, revealing important gaps between image understanding and symbolic reasoning.

**Prior Work** To address these challenges, a variety of models have been proposed. G-LLaVA Gao et al. (2023) adapts multi-modal LLMs to geometric problem-solving by integrating visual and textual features, while Inter-GPS Lu et al. (2021) introduces formal symbolic reasoning pipelines to enhance interpretability and solution accuracy. UniMath Liang et al. (2023) aims to build a foundational multi-modal mathematical reasoner across different modalities and task types. ChatGLM-Math Xu et al. (2024b) focuses on improving text-based math problem-solving through a self-critique pipeline, showing that even unimodal LLMs can benefit from better reasoning structures. Dual-Reasoning Geometry Solver (Dual-GeoSolver) Xiao et al. (2024) explores a dual-reasoning approach inspired by human problem-solving strategies, emphasizing the importance of both visual and symbolic understanding. Reason-and-Execute prompting method Duan et al. (2024) proposes a framework for breaking down complex geometry questions into executable steps to enhance structured reasoning.

More recently, GeoX Xia et al. (2024) presents a unified pretraining framework that jointly formalizes visual diagrams and symbolic descriptions, enabling better alignment between vision and language for geometric problem-solving. Similarly, Geo-LLaVA Xu et al. (2024a) extends the capabilities of multi-modal LLMs by applying meta in-context learning and retrieval-augmented training on solid geometry datasets, achieving state-of-the-art results on benchmarks like GeoQA (from Duan et al. (2024)) and GeoMath (original to Geo-LLaVA Xu et al. (2024a)). GeoGPT4V Cai et al. (2024) explores synthetic generation of geometric figures to augment training and evaluation data, further pushing the boundaries of geometry-focused multimodal learning.

**Unimodal and Multimodal Baselines** Baseline models in this domain often start from unimodal language models such as ChatGLM-Math Xu et al. (2024b) or solvers evaluated on the MATH dataset Hendrycks et al. (2021), providing a reference point for text-only problem-solving capabilities. In parallel, foundational vision-language models like ViLT Kim et al. (2021) and BLIP-2 Li et al. (2023) have laid critical groundwork for multi-modal learning, evaluated through image-text retrieval and captioning tasks with metrics like Recall@1, BLEU, and CIDEr. GPT-4V Yang et al. (2023) builds upon these efforts, expanding vision-language capabilities into more complex reasoning domains, including math and science-related tasks.

**Relevant techniques** Relevant techniques have also evolved alongside model architectures. Multi-modal Chain-of-Thought Zhang et al. (2023b) reasoning introduces structured multi-step prompting to enhance complex problem-solving across visual and textual modalities. Compositional Chain-of-Thought Mitra et al. (2024) further explores how reasoning steps can be broken down and composed dynamically, improving both flexibility and generalization in multi-modal tasks. In their paper “Beyond Lines and Circles” the authors Mouselinos et al. (2024) investigates persistent challenges in LLM geometric reasoning, emphasizing that current models still struggle with deep understanding of spatial relations. Finally, Mavis Zhang et al. (2024c) leverages automated visual instruction tuning to construct high-quality training data at scale, addressing data scarcity issues in mathematical vision-language tasks.

Together, these datasets, models, baselines, and techniques form a rapidly growing research landscape focused on advancing multi-modal reasoning, particularly for challenging domains like mathematical and geometric problem solving.

### 3 [1 POINTS] TASK SETUP AND DATA

The primary task of our project is to solve mathematical geometry reasoning problems by integrating textual and visual information to accurately answer multiple-choice questions (MCQs).

We utilize a subset of the MathVerse dataset Zhang et al. (2024b), which includes 750 training samples and 250 testing samples. These samples are categorized into four versions based on the distribution of information across modalities:

- **Text Dominant:** Provides detailed textual descriptions with supporting images.
- **Vision Dominant:** Images contain the majority of critical information, supplemented by minimal text.
- **Text Lite:** Very brief textual information, with images as the main information carrier.
- **Vision Only:** Solely visual information without any text.

In our task design, the geometric reasoning process is explicitly divided into two stages:

- **Perception Stage:** Vision-language models (e.g., GPT-4V, Qwen-2.5-VL) are employed to generate structured descriptions of geometric images. These descriptions focus on extracting key entities and relationships, such as points, lines, angles, and spatial configurations.
- **Reasoning Stage:** The structured image descriptions are combined with the original question texts and passed into large language models (e.g., GPT-4o) for chain-of-thought (CoT) based reasoning and final MCQ answer generation.

This two-stage separation enables a clearer analysis of bottlenecks and strengths within perception and reasoning individually.

To further reduce residual uncertainty, our experiments also evaluate an iterative variant that loops through “Planner → VQA → Belief-state” cycles (maximum 3 rounds) before entering the final reasoning stage. This setting mirrors the full pipeline detailed in Section 5 and allows us to quantify the benefit of active information acquisition.

## 4 [1 POINTS] BASELINES

To systematically evaluate the impact of different modality input strategies on geometric reasoning tasks, we designed and tested eight baseline models, covering three main categories: unimodal reasoning, simple multimodal fusion, and competitive multimodal understanding. These baselines establish a solid foundation for subsequent model comparisons and performance analysis.

### 4.1 UNIMODAL TEXT-DOMINANT BASELINES

We first assessed the reasoning performance using text-dominant input, evaluating the following large language models:

- **GPT-3.5 (Text Dominant)**
- **GPT-4o (Text Dominant)**
- **Mistral-7B (Text Dominant)**

In this setting, the models receive question text, with any necessary visual information embedded within the text description. This enables us to benchmark the maximal reasoning capabilities achievable without detailed visual input and quantify the added value of multimodal integration.

### 4.2 SIMPLE MULTIMODAL FUSION BASELINES

We then explored basic multimodal strategies involving direct feature fusion, employing two modeling approaches:

- **GPT-3.5 + CLIP**: CLIP is used to separately extract embeddings from images and text, followed by linear mapping fusion, with the resulting fused representation passed into GPT-3.5 for answer generation.
- **GPT-3.5 + BLIP**: BLIP generates textual descriptions from images, which are concatenated with the question text and input into GPT-3.5 for reasoning.

These simple fusion baselines assess the effectiveness of low-cost visual-text integration approaches for geometric reasoning tasks.

### 4.3 COMPETITIVE MULTIMODAL BASELINES

To further explore the potential of vision-language models in complex reasoning, we evaluated three advanced multimodal understanding methods:

- **GPT-3.5 + InternLM-XComposer2**: InternLM-XC2 generates joint visual-textual descriptions, which are combined with the problem text and passed to GPT-3.5 for final reasoning.
- **GPT-4V**: GPT-4V directly processes combined image and text inputs, producing detailed visual understanding and logical reasoning steps.
- **Qwen-2.5-VL**: Qwen-2.5-VL performs end-to-end joint visual-textual reasoning, directly generating final answers.

In all baseline experiments, the final MCQ answer is consistently generated by GPT-3.5 to control for differences in language model reasoning ability and ensure fair comparison across different input strategies.

Through the systematic comparison of these eight baselines, we provide an in-depth analysis of the contributions of visual information at various stages of feature extraction, description generation, and logical reasoning, forming a strong empirical basis for our proposed modeling improvements.

## 5 [3 POINTS] PROPOSED MODEL (&gt;1 PAGE)

Our proposed model consists of three main parts. Essentially, **Planner** asks the initial and follow-up questions, **VQA** reads the image, **Reasoner** finishes the problem.

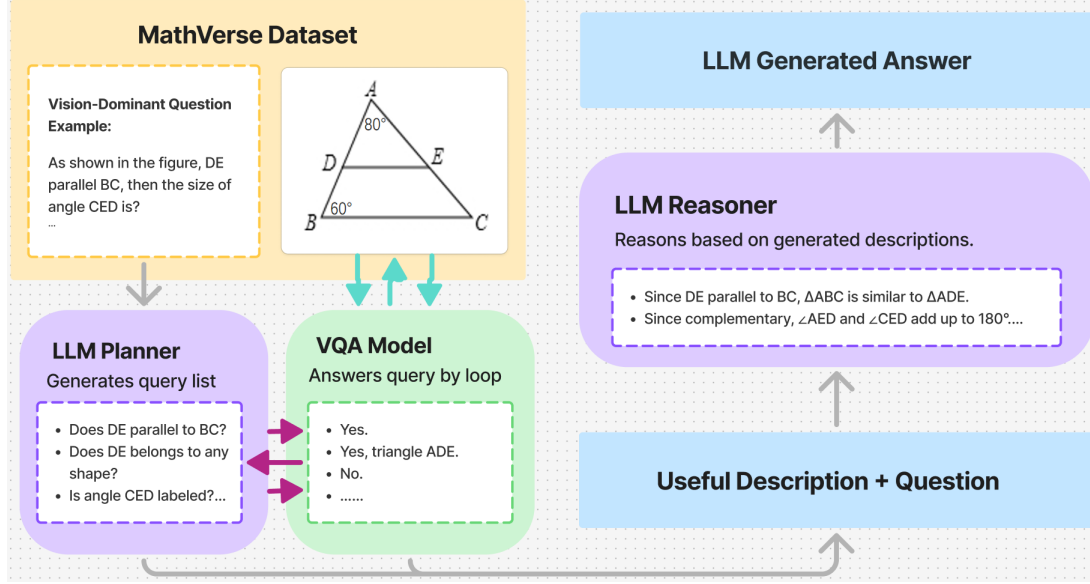


Figure 1: Proposed pipeline structure

A more detailed description of each part is as such:

Stage	Component	Input it receives	Output it produces	How the output is used
1	Planner LM	Plain question text; Previous Q/A log (empty in round 0)	Up to 3 short follow-up queries, one per line. If none are needed, returns "NONE".	Each query is sent, one by one, to Stage 2.
2	VQA model	Query text asking for relevant information in diagram; Original diagram image	A short answer ("Yes, ABC is a triangle.", "60°", "UNKNOWN", etc.).	The pair $Q_n/A_n$ is appended to the belief-state string.
3	Belief-state builder (simple Python code)	All previous Q/A pairs; New Q/A from Stage 2	Belief-state string—a newline-separated transcript (e.g., "Q1: ... A1: ... Q2: ... A2: ...").	The updated string is passed back to Stage 1 for the next round and to Stage 4 once looping stops.
4	Reasoner LM	Final belief-state string; Original question text	A two-part response. <b>Reasoning Step:</b> short chain of logic in prose; <b>Final Answer:</b> one option letter (A/B/C/D).	Final Answer is the pipeline's prediction for accuracy grading.

Table 1: Description of each stage in the pipeline.

**Loop control:**

The planner, VQA model, and belief-state builder (Stages 1  $\rightarrow$  2  $\rightarrow$  3) repeat until either:

- The planner determines the current belief-state string contains enough information and thus outputs **NONE**, or
- The hard cap of rounds (currently set to 3) is reached.

After that, Stage 4 runs once and the pipeline ends.

Here are a table of different choices of models for each part.

Pipeline tag	Planner LM	VQA model	Reasoner LM
P-1	Mathstral-7B-v0.1	InternLM-XC2-VL-7B	Mathstral-7B-v0.1
P-2	Qwen-Chat-7B	Qwen-VL-Chat	Qwen-Chat-7B
P-3	GPT-3.5-Turbo	GPT-4o-VQA	GPT-3.5-Turbo
P-4	GPT-4o (text)	GPT-4o (vision)	GPT-4o (text)

Table 2: Planner, VQA, and Reasoner models used in each version of pipeline.

### 5.1 LOSS FUNCTIONS

None are needed.

All models are frozen; gradients are not computed or updated. We only do forward passes.

Yet there is a logical loss being minimised at run-time:

- Planner’s implicit loss to minimize number of model generations / API calls – ask as few extra questions as possible while still solvable. We let the model decide when to stop, so this loss minimization is implemented by prompt tuning, asking the model to return “NONE” when no new information seems useful.

### 5.2 CHANGES TO TRAINING DATA

Our pipeline didn’t require any training, thus no extra images, labels, or fine-tuning examples are added.

Filtering to `question_type = "Vision Dominant"` is the only data step. This is because in this category much of the geometric detail (right-angle marks, parallel ticks, labeled lengths, etc.) appears only in the diagram, while the text is deliberately minimal. By filtering to this subset, we guarantee that

- Stage 2 (VQA) is essential. Without the extra queries the problem is unsolvable.
- The evaluation can measure the quality of our query-planning and vision reading rather than pure text reasoning.

The **belief-state string** is a temporary text, thus not stored back to the dataset.

### 5.3 HYPERPARAMETERS AND THEIR EFFECTS

Name (in code)	Default	Effect
MAX_ROUNDS	3	Upper bound on LM↔VQA cycles. 2 is faster-but-weaker; 4 adds cost in API calls and wait time with <0.2 pt gain in description quality score.
temperature (Planner)	0.2	Keeps questions short and focused.
max_new_tokens (Planner / Reasoner)	128 / 200	Prevents truncation.
do_sample=False in VQA	–	This forces greedy decoding, ensures no random token picking, so angle values and simple “Yes/No” answers can hopefully stay fixed and repeatable instead of drifting.
float16 / bfloat16	–	Reduces VRAM, helps the model to fit in L4/A100 GPU’s. No measurable accuracy loss.

Table 3: Hyperparameters and their effects in the pipeline.



Methods	Metrics		
	Accuracy	CoT Reasoning	CoT Description
<i>Baseline models – unimodal (text-only LM)</i>			
GPT-3.5-turbo (Text)	0.22	–	–
GPT-4o (Text)	<b>0.58</b>	–	–
Mistral-7B Jiang et al. (2023) (Text)	0.18	–	–
<i>Baseline models – competitive (Reasoner LM + Vision model)</i>			
GPT-3.5 + BLIP Li et al. (2022)	0.13	–	–
GPT-3.5 + CLIP Radford et al. (2021)	0.12	1.28	0.06
GPT-3.5 + InternLM-XC2 Zhang et al. (2023a)	0.14	2.20	1.46
GPT-4V Yang et al. (2023)	0.44	2.60	2.46
Qwen2.5-VL Qwen et al. (2025)	0.52	2.68	2.28
<i>Proposed models: Planner LM + VQA model + Reasoner LM</i>			
Mathstral-7B + InternLM-VL-7B + Mathstral-7B	0.10	1.40	1.18
Qwen 2.5-Chat-7B + Qwen 2.5-VL-Chat + Qwen 2.5-Chat-7B	0.34	3.08	2.46
GPT-3.5-Turbo + GPT-4o-VQA + GPT-3.5-Turbo	0.16	2.66	2.62
GPT-4o text + GPT-4o vision + GPT-4o text	0.40	<b>4.00</b>	<b>2.86</b>

Table 4: Overall Accuracy, Chain-of-Thought (CoT) Reasoning, and CoT Description scores for unimodal text baselines, competitive multimodal baselines, and proposed planner-VQA-reasoner pipelines.

## 6 [1 POINTS] RESULTS (1 PAGE)

Replace columns with the correct metrics for your task (extrinsic). Include multiple versions of your final model. You do not need to run on the test set but are encouraged to try if you have nice results on Dev.

**Brief Discussion.** The full GPT-4o pipeline achieves the best Chain-of-Thought (CoT) scores (4.00 reasoning, 2.86 description) and ranks second in accuracy.

Our Qwen→Qwen setup is the strongest open-weights alternative, trailing GPT-4o by 6 percentage points in accuracy but matching its description quality.

Mathstral → InternLM underperforms, confirming that stronger vision is required on this dataset.

**Clarification on Dev.** *Dev* refers to a 200-problem subset of MathVerse Zhang et al. (2024b) that we created for validation: 100 items are **free-form** questions and 100 are **multiple-choice**. The result table shown in this section is the overall average. Section 7 breaks the metrics down by these two question types.

## 7 [3 POINTS] ANALYSIS (2 PAGES)

## 7.1 INTRINSIC METRICS

## 7.1.1 CHAIN-OF-THOUGHT COHERENCE

**Chain-of-Thought Coherence (reasoning\_score):** For every problem, the model produces a free-form explanation (reasoning\_response). We first distil this narration into **exactly six atomic steps**  $s_{1..6}$  with a GPT-4 instruction following “*Extract 6 key intermediate steps needed to reach the answer.*” The concise list is subsequently graded by GPT-4o-V, which sees the *diagram*, the *question*, the *ground-truth answer*, and the six steps, and returns a binary vector

$$[g_1, g_2, \dots, g_6] \in \{0, 1\}^6,$$

where  $g_i = 1$  indicates that step  $s_i$  is *both logically valid and visually grounded*. The item-level score is

$$\text{reasoning\_score} = \sum_{i=1}^6 g_i \in \{0, \dots, 6\}.$$

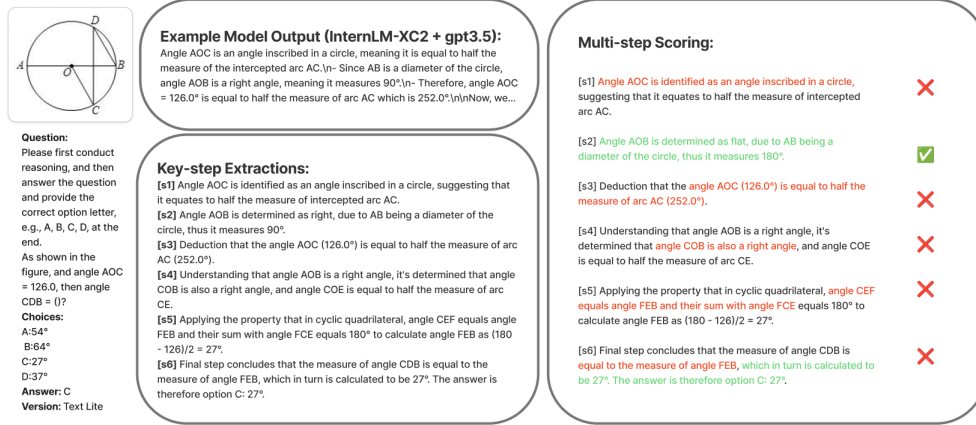


Figure 2: CoT Evaluation on intermediate reasoning steps

Methods	CoT Reasoning Score		
	multi-choice	free-form	overall
<i>Baseline models: Reasoner LM + Vision Model</i>			
GPT-3.5 + CLIP Radford et al. (2021)	1.36	1.20	1.28
GPT-3.5 + InternLM-XC2Zhang et al. (2023a)	2.60	1.80	2.20
GPT-4V Yang et al. (2023)	2.80	2.40	2.60
Qwen2.5-VL Qwen et al. (2025)	2.36	3.00	2.68
<i>Proposed models: Planner LM + VQA model + Reasoner LM</i>			
Mathstral-7B + InternLM-VL-7B + Mathstral-7B	1.52	1.28	1.40
Qwen 2.5-Chat-7B + Qwen 2.5-VL-Chat + Qwen 2.5-Chat-7B	2.80	3.36	3.08
GPT-3.5-Turbo + GPT-4o-VQA + GPT-3.5-Turbo	2.64	2.68	2.66
GPT-4o text + GPT-4o vision + GPT-4o text	4.24	3.76	4.00

Table 5: Chain-of-Thought (CoT) description scores for baseline versus proposed models.

## 7.1.2 CHAIN-OF-THOUGHT PROBLEM DESCRIPTION FAITHFULNESS

**Chain-of-Thought Problem Description Faithfulness (description\_score):** In solving the math problems, the success depends on accurately extracting the facts given in the written description, because the diagram will add many important information. GPT-4o therefore read each original prompt and extracts six important facts (e.g. geometric entities, angle equalities, numeric labels). The same model then compares that reference list with the solver’s `fused_description` and returns a binary presence vector  $[d_1, \dots, d_6]$ , yielding

$$\text{description\_score} = \sum_{i=1}^6 d_i.$$

A perfect score of 6 implies the model preserved *every* given fact; lower totals reveal hallucinations or omissions that can cascade into downstream reasoning errors. The metric’s average over the text-dominant split measures how reliably a solver constructs an accurate internal representation of the problem statement.

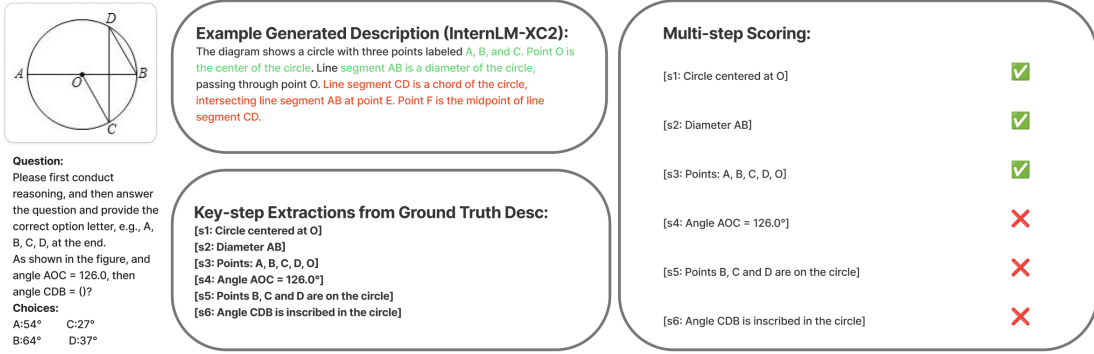


Figure 3: CoT Evaluation on Generated Description

Methods	CoT Description Score		
	multi-choice	free-form	overall
<i>Baseline models: Reasoner LM + Vision Model</i>			
GPT-3.5 + CLIP	0.12	0.00	0.06
GPT-3.5 + InternLM	1.68	1.24	1.46
CPT-4o + GPT-4V	2.92	2.00	2.46
Qwen 2.5+ Qwen 2.5-VL	2.68	1.88	2.28
<i>Proposed models: Planner LM + VQA model + Reasoner LM</i>			
Mathstral-7B + InternLM-VL-7B + Mathstral-7B	0.96	1.40	1.18
Qwen 2.5-Chat-7B + Qwen 2.5-VL-Chat + Qwen 2.5-Chat-7B	2.96	1.96	2.46
GPT-3.5-Turbo + GPT-4o-VQA + GPT-3.5-Turbo	3.36	1.88	2.62
GPT-4o text + GPT-4o vision + GPT-4o text	2.28	3.44	2.86

Table 6: Chain-of-Thought (CoT) description scores for baseline versus proposed models.

## 7.1.3 FINAL-ANSWER ACCURACY

**Final-Answer Accuracy (`acc_score`):** Accuracy provides the conventional “right-or-wrong” benchmark across the *entire* dataset. GPT-4o is shown the question, the official answer, and the model’s answer and must reply with 1 if they match exactly (or 0 otherwise). Thus

$$\text{acc\_score} \in \{0, 1\}, \quad \text{Accuracy} = \mathbb{E}[\text{acc\_score}].$$

Accuracy score gives the most straightforward information to evaluate if the question has been answered correctly or not. We pair it with the two six-point intrinsic metrics above to diagnose whether errors stem from perception, reasoning, or a final arithmetic slip.

Methods	Accuracy		
	multi-choice	free-form	overall
<i>Baseline models – unimodal (text-only LM)</i>			
GPT-3.5-turbo (Text)	0.32	0.12	0.22
GPT-4o (Text)	0.64	0.52	0.58
Mistral-7B Jiang et al. (2023) (Text)	0.28	0.08	0.18
<i>Baseline models – competitive (Reasoner LM + Vision model)</i>			
GPT-3.5 + BLIP Li et al. (2022)	0.20	0.07	0.13
GPT-3.5 + CLIP Radford et al. (2021)	0.16	0.08	0.12
GPT-3.5 + InternLM-XC2 Zhang et al. (2023a)	0.24	0.04	0.14
GPT-4V Yang et al. (2023)	0.52	0.36	0.44
Qwen2.5-VL Qwen et al. (2025)	0.64	0.20	0.52
<i>Proposed models: Planner LM + VQA model + Reasoner LM</i>			
Mathstral-7B + InternLM-VL-7B + Mathstral-7B	0.16	0.04	0.10
Qwen 2.5-Chat-7B + Qwen 2.5-VL-Chat + Qwen 2.5-Chat-7B	0.40	0.28	0.34
GPT-3.5-Turbo + GPT-4o-VQA + GPT-3.5-Turbo	0.24	0.08	0.16
GPT-4o text + GPT-4o vision + GPT-4o text	0.56	0.24	0.40

Table 7: Accuracy scores for unimodal text baselines, competitive multimodal baselines, and proposed planner–VQA–reasoner pipelines.

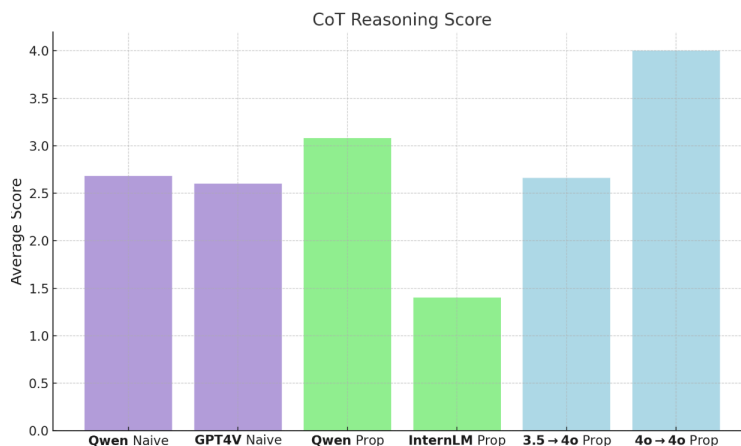


Figure 4: CoT Reasoning Scores

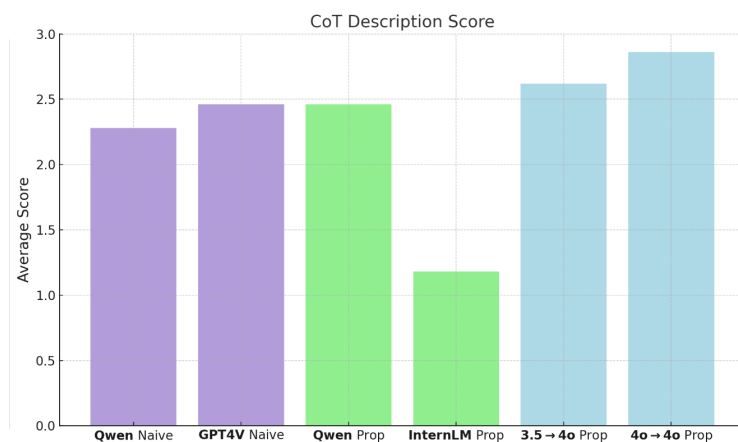


Figure 5: CoT Description Scores

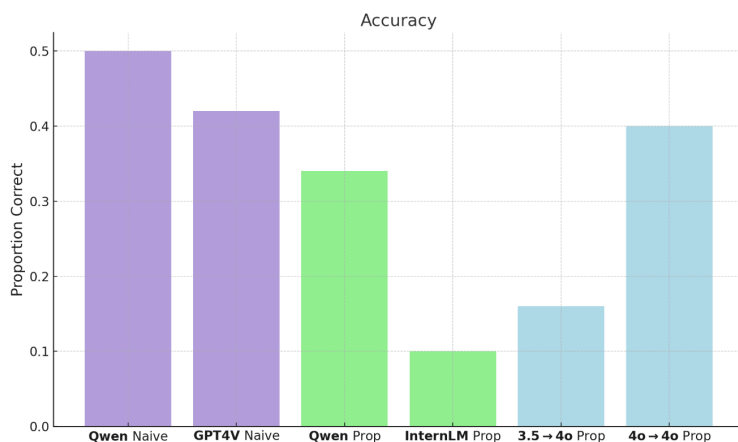


Figure 6: Final Accuracy Scores

**Score Histograms** comparing the best performing baselines with our proposed pipelines. The purple bars represent the best-performing baseline methods. The green bars correspond to open-source models used in our proposed pipelines, while the blue bars indicate closed-source models used in our proposed pipelines.

## 7.2 QUALITATIVE ANALYSIS AND EXAMPLES (FULL PAGE TABLES – MULTIPLE PAGES FOR MOST PROJECTS)

### 7.2.1 ERROR ANALYSIS FOR PIPELINE: GPT-4o TEXT + GPT-4o VISION + GPT-4o TEXT

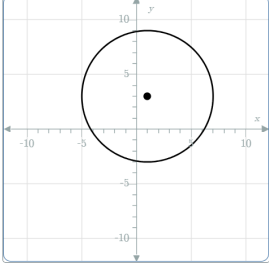
Input Image	Task Question	Key Portion of Model Output
	<b>Prompt:</b> “Consider the circle on the graph. Find the equation of the circle in standard form.”	<b>Caption (vision module):</b> “The circle is <b>centered at the origin (0,0)</b> ; the <b>radius is 5.</b> ”  <b>Reasoning step:</b> Assumes $(h, k) = (0, 0)$ , $r = 5 \implies$ $x^2 + y^2 = 25$ .  <b>Predicted answer:</b> $x^2 + y^2 = 25$ ( <b>incorrect</b> ). <b>Ground truth:</b> $(x - 1)^2 + (y - 3)^2 = 36$ .

Table 8: Example for error analysis: GPT-4o

Stage	What the component <i>should</i> do	What actually happened (pipeline output snippets)	Why that breaks the final answer
1. Image-captioner	Detect geometric primitives (axis location, circle centre, radius).	“The circle is <b>centered at the origin (0, 0)</b> ... <b>The radius is 5.</b> ”	Anchors the pipeline to the wrong $(h, k, r) = (0, 0, 5)$ ; classic object-location hallucination.
2. VQA prompts	Ask clarifying questions to override or confirm the caption and reconcile disagreements.	<i>Q1</i> : centre $\rightarrow$ <b>(0, 5)</b> (contradicts caption). <i>Q2</i> : radius $\rightarrow$ 5.	Pipeline now contains mutually inconsistent facts, but no mechanism flags the conflict between (0, 0) and (0, 5).
3. Reasoning model	Combine trusted facts and output $(x - h)^2 + (y - k)^2 = r^2$ .	Accepts centre = (0, 0), $r = 5 \implies x^2 + y^2 = 25$ .	Symbolically correct given wrong premises—illustrates cascading error: once perception fails, logic can’t recover.

Table 9: End-to-end error analysis for GPT-4o

7.2.2 ERROR ANALYSIS FOR PIPELINE: **GPT-3.5-TURBO + GPT-4O-VQA + GPT-3.5-TURBO**

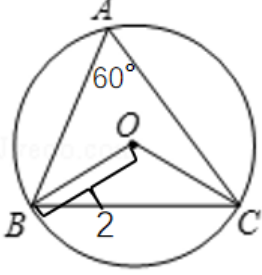
Input Image	Task Question	Key Portion of Model Output
	<p><b>Prompt:</b>  <i>“As shown in the figure, circle O is the circumscribed circle of <math>\triangle ABC</math>. If <math>\angle BAC = 60^\circ</math> and the perpendicular from O to BC is 2 units, find the length of BC.</i></p> <p>Choices: A) <math>\sqrt{3}</math>   B) <math>2\sqrt{3}</math>   C) 4   D) <math>4\sqrt{3}</math>”</p>	<p><b>Caption (vision module):</b>  <i>“The line segment from O to BC is perpendicular and measures <b>2 units, which is the radius of the circle.</b>”</i></p> <p><b>Reasoning step:</b>  Treats <math>\triangle ABC</math> as equilateral, assumes <math>r = 2</math>, and uses <math>BC = 2r \sin(\frac{60^\circ}{2}) = 2</math>.</p> <p><b>Predicted answer:</b> <math>BC = 2</math> (incorrect — choice A).  <b>Ground truth:</b> <math>BC = 4\sqrt{3}</math> (choice D).</p>

Table 10: Example for error analysis: GPT-3.5-turbo

Stage	What the component <i>should</i> do	What actually happened (pipeline output snippets)	Why that breaks the final answer
1. Image-captioner	Parse the geometry: $\angle BAC = 60^\circ$ (inscribed); identify the little right-angle mark as the <i>distance</i> from O to BC ( $d = 2$ ); recognise only $OB, OC$ are radii (unknown).	“The line segment from O to BC is perpendicular and measures <b>2 units, which is the radius of the circle.</b> ”	Mis-labels the 2-unit perpendicular as a radius, so the fact base is anchored to $r = 2$ instead of the correct $r = 4$ .
2. VQA prompts	Ask clarifiers to distinguish “radius” from “distance to chord” and resolve any conflict.	Q: “Is there information about the radius of circle O?” A: “Yes ... the <b>radius is 2.</b> ”	The hallucination is accepted without challenge; pipeline now holds $r = 2$ with high confidence and no contradiction flag.
3. Reasoning model	Use correct relations: central angle $120^\circ$ (since inscribed $60^\circ$ ), chord rule $BC = 2r \sin 60^\circ$ ; combine with $r^2 = (BC/2)^2 + d^2$ .	Assumes “ $\triangle ABC$ is equilateral” and $r = 2$ . Computes $BC = 2r \sin \frac{60^\circ}{2} = 2$ .	Two compounded errors—wrong radius + treating inscribed $60^\circ$ as central—yield $BC = 2$ instead of $4\sqrt{3}$ .

Table 11: End-to-end error analysis for GPT-3.5-turbo

7.2.3 ERROR ANALYSIS FOR PIPELINE: QWEN 2.5-CHAT-7B + QWEN 2.5-VL-CHAT + QWEN 2.5-CHAT-7B

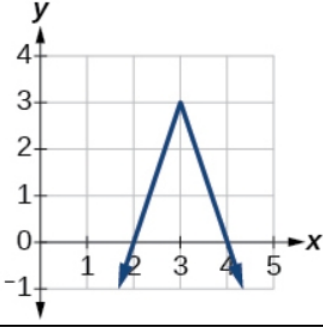
Input Image	Task Question	Key Portion of Model Output
	<p><b>Prompt:</b> “Write an equation for the transformation of <math>f(x) =  x </math> as shown in the figure.”</p>	<p><b>Caption (vision module):</b> “Vertices at (2, -1), (3, 3), (4, -1). Height of the triangle is <b>4 units</b>.”</p> <p><b>Reasoning step:</b> Interprets vertex at (3, 3), claims vertical stretch factor 4 and upward shift 3; concludes <math>g(x) = 4 x - 3  + 3</math>.</p> <p><b>Predicted answer:</b> <math>g(x) = 4 x - 3  + 3</math> (<b>incorrect</b>).</p> <p><b>Ground truth:</b> <math>f(x) = -3 x - 3  + 3</math>.</p>

Table 12: Illustrative instance for error analysis of the absolute-value transformation task.

Stage	What the component <i>should</i> do	What actually happened (pipeline output snippets)	Why that breaks the final answer
1. Image captioner	Read the V-shape precisely: vertex at (3, 3); arms hit the $x$ -axis at (2, 0) and (4, 0); conclude height = 3 and the graph opens <i>downward</i> .	“The vertices of the triangle are at the points (2, -1), (3, 3), and (4, -1) ... The height of the triangle is <b>4 units</b> .”	Mis-places the base 1 unit too low ( $y = -1$ instead of $y = 0$ ), making the height 4 not 3. Orientation (‘opens downwards’) is noted but the sign of the slope is never extracted. Wrong height → wrong stretch factor.
2. VQA prompts	Elicit or confirm shift, stretch and sign; resolve any ambiguity (vertex, slope sign, intercepts).	Some answers return “UNKNOWN”; the extracted fact is that the graph is “a V-shaped graph ... characteristic of an absolute-value function.”	Because the caption already carries a wrong height, the prompt layer fails to retrieve corrective facts (no mention of baseline at $y = 0$ or negative leading coefficient). The erroneous $h = 4$ and unknown sign persist.
3. Reasoning model	Combine vertex (3, 3), downward opening (→ negative sign), and slope magnitude 3 (height 3 over run 1) to obtain $g(x) = -3 x - 3  + 3$ .	Assumes “height is 4 units ... graph is shifted 3 up and 3 right ... <b>stretched vertically by a factor of 4</b> ,” then outputs $g(x) = 4 x - 3  + 3$ .	Two compounded errors: 1. Uses the caption’s $h = 4$ instead of 3. 2. Ignores the downward orientation and keeps a <i>positive</i> stretch, so the arms open upward. The final rule has the wrong slope magnitude and the wrong sign.

Table 13: End-to-end error analysis for Qwen-2.5-7B



### 7.2.4 ERROR ANALYSIS FOR PIPELINE: MATHSTRAL-7B + INTERNLM-VL-7B + MATHSTRAL-7B

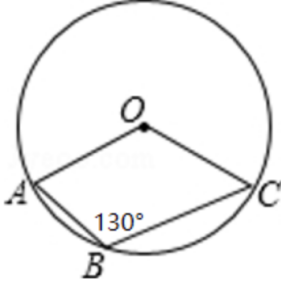
Input Image	Task Question	Key Portion of Model Output
	<p><b>Prompt:</b>  <i>"As shown in the figure, then angle AOC = ()</i>  <b>Choices:</b>  <i>A:100° B:110° C:120° D:130°"</i></p>	<p><b>Caption (vision module):</b>  <i>"Points A, B, C lie on the circle centred at O. Angle ABC = 130°, "</i></p> <p><b>Reasoning step:</b>  The central angle AOC is exactly twice the inscribed angle ABC if they subtend the same arc AC. Thus <math>AOC = 130^\circ \times 2 = 260^\circ</math>. Since <math>260^\circ</math> is larger than <math>180^\circ</math>, we are looking for its complementary angle, <math>360^\circ - 260^\circ = 100^\circ</math>.</p> <p><b>Predicted answer:</b> D.130° (incorrect).  <b>Ground truth:</b> A.100°</p>

Table 14: Example for error analysis: Mathstral-7B + InternLM-VL-XC2

Stage	What the component <i>should</i> do	What actually happened (pipeline output snippets)	Why that breaks the final answer
1. Image-captioner	Detect labeled angle and geometric relationships between points and the circle in the image	"AOC is a triangle. Angle BOC = 180°. Angle BAC is labeled as 130°."	The image module made multiple mistakes, mistakened the angle's names, hallucinates about a non-existent triangle and angles, etc.
2. VQA prompts	Ask clarifying questions to explore the diagram, gather all information it needs.	<i>Q1: Is AOC a triangle? Q2: Are there any labeled scalars in the diagram? Q3: What is the measure of angle BOC?</i>	Based on the current belief-state string it is asking sensible questions. Although it hallucinates about non-existent angle BOC, it is due to false information provided by the VQA model.
3. Reasoning model	Combine trusted facts and use geometric rule of inscribed angles.	Since angle AOC and angle BOC are supplementary angles, we can use the fact that the sum of their measures is 180 degrees to find the measure of angle AOC.	Uses the false fact and arrives on an understanding of the diagram that doesn't make any sense. The output 130° is pure guess.

Table 15: End-to-end error analysis for Mathstral-7B + InternLM-VL-XC2

## 8 [2 POINTS] FUTURE WORK AND LIMITATIONS (1 PAGE)

### 8.1 LIMITATION

Despite the planner–VQA–reasoner architecture delivering promising overall accuracy, our experiments reveal four systemic failure modes that currently cap its performance.

#### 8.1.1 MEANINGFUL QUERIES, POOR VQA ANSWERS

The planner reliably produces semantically relevant follow-up questions, yet the vision–question–answering (VQA) module often supplies incorrect or low-confidence responses. The mismatch is most severe for fine-grained spatial cues—e.g. tick-mark counts or colour-coded labels—where a single pixel error flips the truth value.

#### 8.1.2 HALLUCINATED GEOMETRIC RELATIONSHIPS

The VQA model frequently conflates inscribed with central angles, mistakes perpendicular distances for radii, and misidentifies baselines in composite diagrams. These hallucinations enter the pipeline as *false facts* and propagate into the reasoner, which then delivers logically impeccable—but factually wrong—proofs.

#### 8.1.3 ABSENCE OF MEMORY & CONTRADICTION RESOLUTION

Answered questions are merely appended to the next prompt; no mechanism detects redundancy or inconsistency. Consequently, mutually conflicting facts accumulate, degrading the quality of downstream reasoning.

#### 8.1.4 DATASET GAPS AND INCOMPLETE GROUND TRUTH

Several ground-truth image annotations omit implicit geometric constraints such as “horizontal” or “equal length.” Fine-tuning on these partial labels teaches the model to ignore useful cues and occasionally penalises otherwise correct, more detailed descriptions.

#### 8.1.5 CHAT–API DEPLOYMENT DISCREPANCY

Outputs obtained via the GPT-4o API are noticeably noisier than those observed in the interactive chat window, indicating that our prompting and alignment choices do not yet transfer cleanly across interfaces.

**Implications** Taken together, these limitations show that the pipeline neither ensures long-range factual consistency nor robustly grounds symbolic reasoning in accurate vision features. They motivate future work on memory-aware retrieval, adaptive query stopping, targeted VQA fine-tuning, and systematic dataset re-annotation.

### 8.2 FUTURE WORK

#### 8.2.1 MEMORY-AWARE RETRIEVAL

We will build a lightweight *memory module* that stores every Q–A pair in a structured buffer and exposes read / write APIs to the planner. The planner can then retrieve, update, or discard facts—eliminating redundant inquiries and resolving contradictions before they corrupt downstream reasoning.

#### 8.2.2 ADAPTIVE QUERY STOPPING

By monitoring the entropy of the VQA logit distribution (or another uncertainty proxy), the system will halt question generation once the marginal information gain falls below a preset threshold. This rule reduces inference cost and prevents the pipeline from drowning in low-value, noisy queries.

### 8.2.3 CLOSING THE CHAT-API GAP VIA TARGETED FINE-TUNING

We observe that GPT-4o API calls yield noisier VQA answers than the interactive chat window. To bridge this gap we will fine-tune the VQA backbone on a re-annotated dataset that *zooms into salient regions, crops irrelevant clutter*, and adds explicit tags for angles, lengths, and parallelism—aligning the model’s visual grounding with the planner’s expectations.

### 8.2.4 PROMPT-REFINEMENT LOOP

Before a query reaches the VQA stage, a language model will rewrite it for clarity and geometric specificity, discouraging heuristic shortcuts and encouraging attention to subtle spatial cues. The loop iterates until the refined prompt meets a quality threshold or the adaptive-stopping criterion is triggered.

**Overall Impact** These upgrades directly target the weaknesses identified in our limitation analysis—improving perceptual accuracy, enforcing factual consistency, and enhancing robustness across deployment interfaces.

## 9 [1 POINTS] ETHICAL CONCERNS AND CONSIDERATIONS (UNINTENTIONAL, MALICIOUS, AND DUAL-USE)

Our system raises two primary ethical risks.

### **Misleading Outputs (Educational Fairness).**

Errors in the perception stage—e.g., mis-reading a diagram’s center or angle—can propagate through the chain-of-thought and yield confident yet incorrect answers. If adopted uncritically for homework assistance or automated grading, such mistakes could misinform students and skew assessments. *Mitigation*: integrate automatic contradiction checks within the belief-state; expose a confidence score and key intermediate descriptions so instructors can spot-check results before trusting them.

### **Representation Bias.**

MathVerse Zhang et al. (2024b) mainly contains clean, English-annotated diagrams; hand-drawn sketches and non-Latin labels are under-represented. Consequently, performance may degrade on these styles, disadvantaging certain learner groups. *Mitigation*: expand evaluation with multilingual and hand-drawn diagrams, fine-tune on that data, and publicly document known blind spots to avoid over-promising performance.

By embedding these safeguards and transparently reporting limitations, we aim to reduce harm and enable safer deployment of multimodal geometry-reasoning systems in educational and related contexts.

## REFERENCES

- Shihao Cai, Keqin Bao, Hangyu Guo, Jizhi Zhang, Jun Song, and Bo Zheng. Geogpt4v: Towards geometric multi-modal large language models with geometric image generation. *arXiv preprint arXiv:2406.11503*, 2024.
- Xiuliang Duan, Dating Tan, Liangda Fang, Yuyu Zhou, Chaobo He, Ziliang Chen, Lusheng Wu, Guanliang Chen, Zhiguo Gong, Weiqi Luo, et al. Reason-and-execute prompting: Enhancing multi-modal large language models for solving geometry questions. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pp. 6959–6968, 2024.
- Ali Forootani. A Survey on Mathematical Reasoning and Optimization with Large Language Models. <https://arxiv.org/abs/2503.17726>, 2025. arXiv preprint, arXiv:2503.17726.
- Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023.
- Ziyu Guo, Bowen Deng, Rui Li, Ruonan Liu, Pengcheng Zou, Dayu Shi, Hang Xu, and Zhenguo Li. G-LLaVA: Solving geometric problem with multi-modal large language model. <https://arxiv.org/pdf/2312.11370>, 2023. arXiv preprint, arXiv:2312.11370.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL <https://arxiv.org/abs/2103.03874>.
- Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L  lio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timoth  e Lacroix, and William El Sayed. Mistral 7b, 2023. URL <https://arxiv.org/abs/2310.06825>.
- Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *International conference on machine learning*, pp. 5583–5594. PMLR, 2021.

- Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation, 2022. URL <https://arxiv.org/abs/2201.12086>.
- Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.
- Zhenwen Liang, Tianyu Yang, Jipeng Zhang, and Xiangliang Zhang. Unimath: A foundational and multimodal mathematical reasoner. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 7126–7133, 2023.
- Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021.
- Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023.
- Chancharik Mitra, Brandon Huang, Trevor Darrell, and Roei Herzig. Compositional chain-of-thought prompting for large multimodal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14420–14431, 2024.
- Spyridon Mouselinos, Henryk Michalewski, and Mateusz Malinowski. Beyond lines and circles: Unveiling the geometric reasoning gap in large language models. *arXiv preprint arXiv:2402.03877*, 2024.
- OpenAI. GPT-4V(ision) System Card. [https://cdn.openai.com/papers/GPTV\\_System\\_Card.pdf](https://cdn.openai.com/papers/GPTV_System_Card.pdf), 2023. Published September 25, 2023.
- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. 2021. URL <https://arxiv.org/abs/2103.00020>.
- Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, et al. Geox: Geometric problem solving through unified formalized vision-language pre-training. *arXiv preprint arXiv:2412.11863*, 2024.
- Tong Xiao, Jiayu Liu, Zhenya Huang, Jinze Wu, Jing Sha, Shijin Wang, and Enhong Chen. Learning to solve geometry problems via simulating human dual-reasoning process. *arXiv preprint arXiv:2405.06232*, 2024.
- Shihao Xu, Yiyang Luo, and Wei Shi. Geo-llava: A large multi-modal model for solving geometry math problems with meta in-context learning. In *Proceedings of the 2nd Workshop on Large Generative Models Meet Multimodal Applications*, pp. 11–15, 2024a.
- Yifan Xu, Xiao Liu, Xinghan Liu, Zhenyu Hou, Yueyan Li, Xiaohan Zhang, Zihan Wang, Aohan Zeng, Zhengxiao Du, Wenyi Zhao, et al. Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline. *arXiv preprint arXiv:2404.02893*, 2024b.
- Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. The dawn of lmms: Preliminary explorations with gpt-4v(ision), 2023. URL <https://arxiv.org/abs/2309.17421>.

Jiaxin Zhang, Zhongzhi Li, Mingliang Zhang, Fei Yin, Chenglin Liu, and Yashar Moshfeghi. Geoval: benchmark for evaluating llms and multi-modal models on geometry problem-solving. *arXiv preprint arXiv:2402.10104*, 2024a.

Pan Zhang, Xiaoyi Dong, Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Haodong Duan, Songyang Zhang, Shuangrui Ding, Wenwei Zhang, Hang Yan, Xinyue Zhang, Wei Li, Jingwen Li, Kai Chen, Conghui He, Xingcheng Zhang, Yu Qiao, Dahua Lin, and Jiaqi Wang. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition, 2023a. URL <https://arxiv.org/abs/2309.15112>.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Peng Gao, and Hongsheng Li. MathVerse: Does your multi-modal llm truly see the diagrams in visual math problems? <https://arxiv.org/abs/2403.14624>, 2024b. *arXiv preprint*, arXiv:2403.14624.

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*, 2024c.

Zhuosheng Zhang, Aston Zhang, Mu Li, Hai Zhao, George Karypis, and Alex Smola. Multimodal chain-of-thought reasoning in language models. *arXiv preprint arXiv:2302.00923*, 2023b.

Forootani, A. *A Survey on Mathematical Reasoning and Optimization with Large Language Models*, 2025. URL <https://arxiv.org/abs/2503.17726>.

OpenAI. *GPT-4V(ision) System Card*, September 25 2023. URL [https://cdn.openai.com/papers/GPTV\\_System\\_Card.pdf](https://cdn.openai.com/papers/GPTV_System_Card.pdf).

## A APPENDIX

You may include other additional sections here.