

# GeoVLMath: Enhancing Geometry Reasoning in Vision-Language Models via Cross-Modal Reward for Auxiliary Line Creation

Anonymous ACL submission

## Abstract

Auxiliary lines are essential for solving complex geometric problems but remain challenging for large vision-language models (LVLMs). Recent attempts construct auxiliary lines via code-driven rendering, a strategy that relies on accurate and executable code generation to produce visual renderings of the auxiliary lines for subsequent reasoning. However, in complex solid geometry settings, such a strong dependence on precise specifications substantially restricts the robustness of this strategy. Alternatively, we turn to a simpler and more stable solution, representing auxiliary-line constructions as structured textual descriptions. To bridge the gap between textual descriptions and spatial structure, we propose a reinforcement learning framework that enhances diagram-text alignment. The core is a cross-modal reward model that evaluates how well the generated auxiliary-line description matches the ground-truth auxiliary-line diagram. The reward signal drives a GRPO-based RL stage to yield informative auxiliary-line descriptions for the reasoning. To support the training and evaluation, we develop a scalable data pipeline and construct AuxSolidMath, a dataset of 3,018 real-exam geometry problems with paired diagrams and aligned textual fields. Based on this framework, we derive GeoVLMath, an LVLm for solving complex solid geometry.

## 1 Introduction

Geometric problems constitute an important category of mathematical tasks, characterized by intricate spatial structures and multi-step reasoning processes (Ma et al., 2024). They are commonly divided into plane geometry and solid geometry. This study focuses on solid geometry, where reasoning over three-dimensional spatial relations is substantially more complex. Such problems rarely yield to direct application of standard theorems; instead, they often require the deliberate introduction

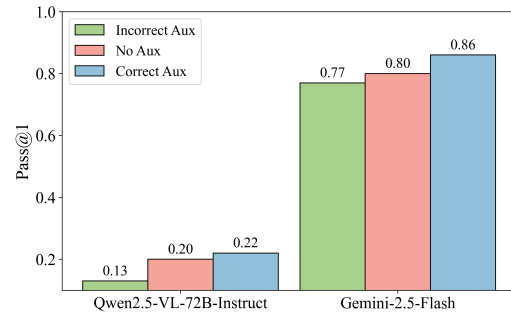


Figure 1: Pass@1 results for Qwen2.5-VL-72B-Instruct and Gemini-2.5-Flash. “Aux” indicates the use of auxiliary-line descriptions.

of auxiliary lines<sup>1</sup> to reveal hidden geometric structure and enable further analysis. These auxiliary lines are essential for anchoring visual diagrams to formal symbolic reasoning and for providing the intermediate steps for rigorous problem-solving.

To validate the above idea, we conduct a pilot study comprising three experimental settings: Incorrect Aux (with incorrect auxiliary lines), No Aux (without auxiliary lines), and Correct Aux (with correct auxiliary lines). As shown in Figure 1, **the use of correct auxiliary lines achieves the highest accuracy, whereas incorrect auxiliary lines lead to the poorest performance.** One possible explanation is that inaccurate auxiliary lines tend to misdirect reasoning and produce errors, while precise auxiliary lines uncover key spatial relationships, thereby enhancing solution accuracy.

Given that accurate auxiliary lines are crucial for the correctness of the solution, the key question is how to obtain them reliably. A seemingly straightforward solution is to explicitly construct them on the diagram. Indeed, several representative methods, such as Visual Sketchpad (Hu et al., 2024), V-Thinker (Qiao et al., 2025), and CodePlot-CoT (Duan et al., 2025), follow a unified paradigm in which the model generates Python-based draw-

<sup>1</sup>In this paper, we use *auxiliary lines* broadly to include both additional lines and coordinate systems.

ing code to render intermediate diagrams that are then fed back into the reasoning process. Despite its conceptual simplicity, this code-driven visual construction paradigm exhibits inherent limitations in solid geometry. It critically relies on precise coordinate information and highly accurate code generation, and becomes particularly fragile when auxiliary lines involve cross-plane relations or skew structures, where minor rendering inaccuracies can distort spatial constraints and mislead reasoning. Our empirical results corroborate this limitation, **showing that explicitly rendered auxiliary-line diagrams underperform text-based auxiliary-line formulations** (See Table 2).

In light of these observations, we adopt a simpler and more stable formulation by representing auxiliary-line constructions as structured textual descriptions. Our main idea is to design a **cross-modal reward model** that measures the consistency between a generated textual auxiliary-line description for the original diagram and a ground-truth auxiliary-line diagram. The reward is computed by jointly encoding the original diagram and the generated auxiliary-line description, and then comparing this pair with the ground-truth auxiliary-line diagram, providing geometry-aware supervision without requiring coordinate assumptions or image manipulation. Building on this reward signal, we train a policy model using Group Relative Policy Optimization (GRPO) to obtain geometry-consistent, generalizable constructions. Training follows a two-stage paradigm inspired by recent progress in reinforcement learning (RL) for reasoning (e.g., DeepSeek-R1 (Guo et al., 2025)): supervised fine-tuning (SFT) for cold start, followed by GRPO-based RL to further elicit structured reasoning and strengthen diagram-text alignment. We instantiate the framework as GeoVLMath, a vision-language model tailored to auxiliary-line-based geometric reasoning. Through the cross-modal supervision, GeoVLMath achieves strong alignment between text and geometric structure, enabling faithful reasoning on complex diagrams.

To effectively train the above model, we require a high-quality dataset that captures both visual and symbolic aspects of real-world geometry problems. However, creating such a dataset is inherently challenging due to the need for automation, scalability, and semantic precision across diverse and noisy educational materials. In response to these challenges, we develop a **robust and scalable data construction pipeline** that transforms raw high

school exam papers into structured multimodal instances suitable for training LVLMs, comprising automated problem identification, automated deduplication and diagram extraction, structured data extraction, and manual verification. While the pipeline is largely automated, this lightweight manual verification step is essential for maintaining data quality, particularly when handling complex symbolic expressions and diagrammatic content in real-world settings. Based on this pipeline, we construct **AuxSolidMath**, a curated dataset of 3,018 solid geometry problems in a rich multimodal format, comprising the problem description, the final answer, the auxiliary-line description, the original diagram, and the auxiliary-line diagram. To our knowledge, AuxSolidMath is the first systematically constructed dataset explicitly tailored to auxiliary-line-based solid geometry reasoning.

We empirically evaluate GeoVLMath against a broad range of LVLMs, including representative approaches that construct auxiliary lines via code-driven rendering. Despite its relatively modest parameter scale, GeoVLMath achieves highly competitive performance, consistently outperforming code-driven rendering methods and surpassing larger models such as Qwen2.5-VL-32B-Instruct (Bai et al., 2025) and GPT-4o (Hurst et al., 2024). These results indicate that supervision grounded in auxiliary-line constructions is more effective for improving geometric reasoning than explicit code-driven diagram rendering or simply scaling model size. Furthermore, our evaluation protocol highlights the value of auxiliary-line-augmented datasets for revealing the limitations of LVLMs in visual reasoning.

**Contributions.** (1) **Cross-modal reward.** We introduce a geometry-aware scalar reward that directly evaluates diagram-text alignment, providing reliable and fine-grained supervision for reinforcement learning in auxiliary-line construction. (2) **AuxSolidMath.** We introduce AuxSolidMath, a curated dataset of 3,018 solid geometry problems from real high-school exams, with aligned diagrams, auxiliary-line annotations, and answers, designed to support training and evaluation of auxiliary-line reasoning. (3) **GeoVLMath.** We introduce GeoVLMath, demonstrating that RL with vision-based rewards effectively optimizes auxiliary-line construction, achieving superior performance to code-driven rendering approaches and competitive results against larger LVLMs.

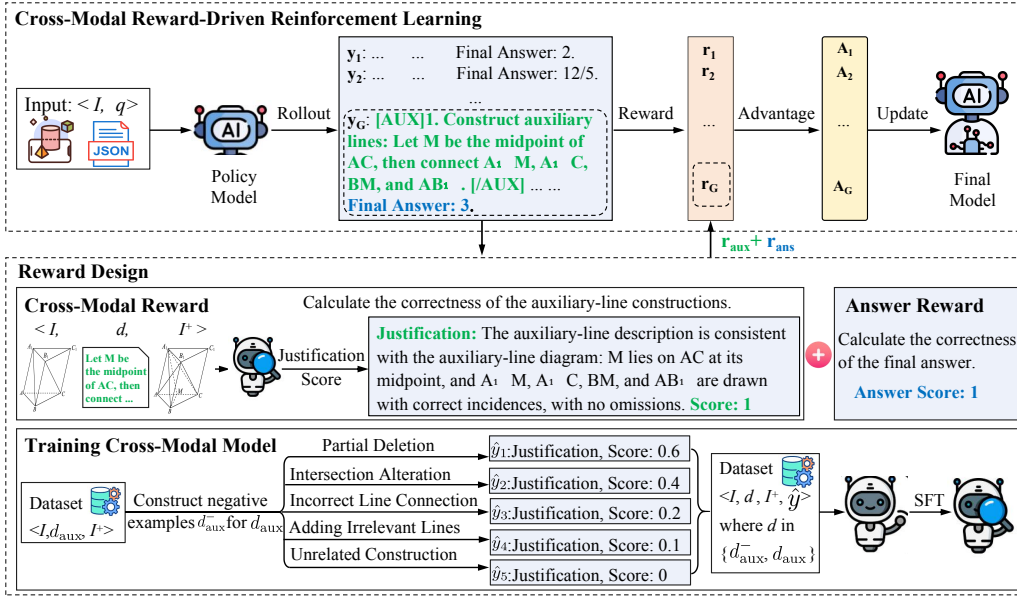


Figure 2: Overview of the cross-modal reward-driven RL. We first fine-tune a cross-modal reward model to evaluate the correctness of auxiliary-line constructions. During the RL phase, the reward model’s consistency score is combined with the answer accuracy reward to produce a composite signal that updates the policy via GRPO.

## 2 Methodology

### 2.1 Problem Definition

We study solid geometry problems whose solutions require the *active construction of auxiliary lines*. Each instance is defined by a pair  $\langle I, q \rangle$ , where  $I$  denotes the original diagram and  $q$  is a question grounded in  $I$ . The goal is to generate a solution  $y$  that includes the auxiliary lines  $aux$ , a sequence of reasoning steps, and the final answer  $ans$ . The  $aux$  is not present in  $I$  and must be introduced during reasoning to expose implicit spatial relations and convert them into explicit geometric constraints.

### 2.2 Framework Overview

We introduce a two-stage training framework for LVLMs that integrates the auxiliary-line construction into the reasoning process. In the first stage, we apply SFT on automatically synthesized chain-of-thought (CoT) data with explicit auxiliary-line steps, enabling the model to actively construct auxiliary lines, thereby establishing a good initialization. In the second stage, we further use RL to encourage the model to construct auxiliary lines that faithfully reflect the geometry of the diagram, boosting the precision of the solution. At the core is a **cross-modal reward model** that provides fine-grained feedback by scoring the agreement between the original diagram plus the generated auxiliary-line description and a reference diagram annotated with the correct auxiliary line. In sum-

mary, our framework combines direct supervision with structured visual feedback, thereby enhancing the reliability of auxiliary-line constructions and overall geometric problem-solving performance.

### 2.3 Supervised Fine-Tuning

We apply SFT on CoT exemplars to initialize the model for RL. Auxiliary-line construction steps are explicitly marked with `[AUX]` and `[/AUX]` to provide structured supervision and support later reward modeling, and the model is trained using a standard next-token prediction objective.

### 2.4 Cross-Modal Reward-Driven RL

As shown in Figure 1, accurate auxiliary-line constructions improve reasoning success. The key challenge is how to integrate such constructions into the reasoning process. Recent methods adopt Python-based drawing code to render auxiliary lines for intermediate reasoning, yet their reliance on precise and executable specifications limits robustness in complex solid geometry. Motivated by this limitation, and inspired by recent advances in textual CoT supervision (Xu et al., 2024; Zhang et al., 2025), we represent auxiliary-line constructions as structured natural language descriptions. To bridge the gap between textual descriptions and spatial structure, we propose a **cross-modal reward model** that scores diagram-text alignment between the original diagram paired with a generated auxiliary-line description and a reference

230 diagram annotated with the ground-truth auxiliary  
 231 lines. This reward provides geometry-aware super-  
 232 vision without relying on executable drawing code  
 233 or precise geometric specifications, enabling robust  
 234 scaling across diverse diagram styles. We integrate  
 235 this cross-modal reward, alongside a final-answer  
 236 reward, into a GRPO-based RL stage to align in-  
 237 termediate constructions with the diagram while  
 238 maintaining final-answer accuracy. An overview of  
 239 the stage is illustrated in Figure 2.

#### 240 2.4.1 Cross-modal Reward Model

241 Given an original diagram  $I$ , a textual description  
 242 of auxiliary lines  $d$  (either the ground-truth de-  
 243 scription  $d_{\text{aux}}$  or a perturbed variant  $d_{\text{aux}}^-$ ), and a  
 244 reference diagram  $I^+$  annotated with the ground-  
 245 truth auxiliary lines, the reward model evaluates  
 246 the geometric relations implied by applying  $d$  to  $I$   
 247 against the additional geometric structures present  
 248 in  $I^+$  but absent from  $I$ . Rather than relying on  
 249 surface-level similarity, it assesses diagram-text  
 250 relational consistency by verifying whether geo-  
 251 metric relations specified in  $d$ , such as parallelism,  
 252 orthogonality, and angle bisection, are satisfied in  
 253 the reference diagram  $I^+$ . Accordingly, the pro-  
 254 posed reward exhibits three desirable properties:  
 255 **(1) Cross-modal relational alignment**, which eval-  
 256 uates diagram-text consistency at the level of geo-  
 257 metric relations rather than surface-level or lexical  
 258 similarity; **(2) Render-free and coordinate-free**,  
 259 as it avoids dependence on executable drawing  
 260 code or pixel-accurate coordinates, relying instead  
 261 on relational correspondence between diagrams  
 262 and textual descriptions; and **(3) Fine-grained su-  
 263 pervision**, assigning intermediate scores to par-  
 264 tially correct yet geometrically meaningful con-  
 265 structions, thereby enabling precise credit assign-  
 266 ment in multi-step reasoning. In summary, the  
 267 cross-modal reward evaluates diagram-text spatial  
 268 consistency at the level of geometric relations, as-  
 269 sessing whether and to what extent the generated  
 270 auxiliary lines align with the intended geometric  
 271 structure, without relying on executable rendering  
 272 code methods. Next, we describe how to automati-  
 273 cally **construct diagram-text supervision** at scale  
 274 and use it to **train the cross-modal reward model**.

275 **Constructing Diagram-Text Supervision.** Each  
 276 training example is represented as  $\langle I, d, I^+, \hat{y} \rangle$ ,  
 277 where  $\hat{y} = (r, s)$  contains a brief justification  $r$  and  
 278 a consistency score  $s \in [0, 1]$  indicating how well  
 279  $d$  aligns with  $I^+$  given  $I$ . We construct this super-  
 280 vision dataset via a fully automated pipeline (see

281 Figure 2). Starting from high-quality supervision  
 282 triplets  $\langle I, d_{\text{aux}}, I^+ \rangle$  constructed as described in  
 283 Section 3, we systematically generate challenging  
 284 negative descriptions to support robust diagram-  
 285 text supervision. Specifically, we design a set  
 286 of rule-based perturbation templates that simulate  
 287 common auxiliary-line construction errors, includ-  
 288 ing *partial deletion*, *intersection alteration*, *incor-  
 289 rect line connections*, *adding irrelevant lines*, *unre-  
 290 lated auxiliary lines*. Building on these templates,  
 291 we further leverage a large language model to syn-  
 292 thesize diverse and linguistically varied negative  
 293 descriptions  $d_{\text{aux}}^-$ . These negatives are lexically  
 294 fluent and semantically plausible, yet geometri-  
 295 cally inconsistent with the intended construction  
 296 encoded in the reference diagram  $I^+$ , thereby pro-  
 297 viding hard counterexamples beyond surface-level  
 298 mismatches. To assess the consistency between  
 299 each description  $d \in \{d_{\text{aux}}, d_{\text{aux}}^-\}$  and the target  
 300 construction  $I^+$ , we adopt an *LVM-as-a-Judge*  
 301 strategy. Concretely, the LVM is prompted to  
 302 evaluate diagram-text alignment by comparing the  
 303 original diagram paired with the description  $\langle I, d \rangle$   
 304 against the reference diagram  $I^+$ , and to produce  
 305 both a natural-language rationale  $r$  and a scalar  
 306 alignment score  $s \in [0, 1]$ . This automated eval-  
 307 uation yields interpretable explanations alongside  
 308 valued supervision signals, enabling scalable and  
 309 fine-grained diagram-text supervision that spans  
 310 faithful auxiliary-line descriptions as well as adver-  
 311 sarily perturbed constructions.

312 **Training Cross-Modal Reward Model.** Given  
 313 the input triplet  $\langle I, d, I^+ \rangle$ , the model outputs a  
 314 structured prediction  $\hat{y} = (r, s)$ . We train the  
 315 model by maximizing the conditional likelihood of  
 316 the serialized output:

$$317 p_\phi(\hat{y} | I, d, I^+) = \prod_{i=1}^T p_\phi(\hat{y}_i | I, d, I^+, \hat{y}_{<i}) \quad (1)$$

318 where  $T$  denotes the length of the generated se-  
 319 quence  $\hat{y}$ ,  $\hat{y}_i$  is the  $i$ -th token in the output, and  
 320  $\hat{y}_{<i}$  represents the sequence of previously gener-  
 321 ated tokens. The consistency score is indicated as  
 322  $r_{\text{aux}} = s = \text{Score}(\hat{y})$ , where higher values corre-  
 323 spond to better consistency.  
 324

325 Through this training, we obtain a reward model  
 326 that provides precise feedback on diagram-text  
 327 alignment between auxiliary-line descriptions and  
 328 the reference diagram. This model serves as a key  
 329 component of our framework, guiding the policy  
 330 toward auxiliary-line constructions that are geomet-  
 331 rically consistent and diagram-grounded.

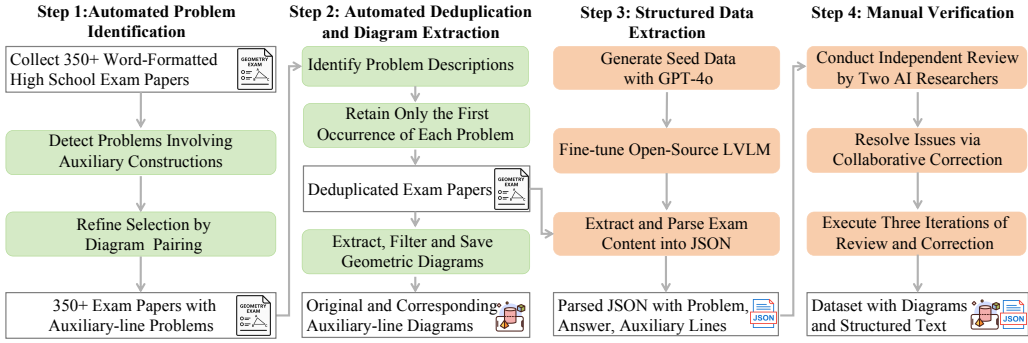


Figure 3: Overview of the proposed data creation pipeline.

## 2.4.2 Optimization

We adopt GRPO as the policy optimization algorithm. The overall reward signal combines the cross-modal reward with a final-answer reward defined as a binary score, yielding 1 if the predicted final answer matches the ground truth and 0 otherwise, *i.e.*,  $r = \alpha r_{\text{aux}} + (1 - \alpha)r_{\text{ans}}$ .

Given a geometric diagram  $I$  and a question  $q$ , GRPO samples a set of response sequences  $\{y_1, y_2, \dots, y_G\}$  from the old policy  $\pi_{\theta_{\text{old}}}$ . The policy  $\pi_{\theta}$  is then optimized by maximizing the following objective:

$$\mathcal{L}_{\text{GRPO}} = \frac{1}{G} \sum_{i=1}^G \left( \min \left( \frac{\pi_{\theta}(y_i | I, q)}{\pi_{\theta_{\text{old}}}(y_i | I, q)} A_i, \text{clip} \left( \frac{\pi_{\theta}(y_i | I, q)}{\pi_{\theta_{\text{old}}}(y_i | I, q)}, 1 - \epsilon, 1 + \epsilon \right) A_i \right) - \beta \mathbb{D}_{\text{KL}}(\pi_{\theta} \| \pi_{\text{ref}}) \right) \quad (2)$$

Here,  $G$  denotes the group size, while  $\epsilon$  and  $\beta$  are hyperparameters for clipping and the KL penalty.

## 3 Data Creation

We curate the AuxSolidMath dataset to support training and evaluation of auxiliary-line reasoning in solid geometry. As illustrated in Figure 3, our data creation pipeline proceeds through four progressive steps: **automated problem identification, automated deduplication and diagram extraction, structured data extraction, and manual verification**. The pipeline standardizes raw exam problems into semantically aligned, high-quality multimodal instances that support training vision-language models for auxiliary-line geometric reasoning (More details in Appendix B). Figure 4 shows an example from the dataset. Each instance is represented as a five-tuple consisting of the *problem description*, the *final answer*, the *auxiliary-line description*, the *original diagram*, and the *auxiliary-line diagram*.

**Automated Problem Identification.** We curate AuxSolidMath from over 350 authentic high-school geometry exam sets using a two-stage filter

that selects problems explicitly requiring auxiliary-line constructions, based on cue-verb retrieval from reference solutions and verification of paired original and auxiliary-line-annotated diagrams.

**Automated Deduplication and Diagram Extraction.** We automatically deduplicate the dataset and extract paired diagrams to ensure unique and high-quality instances. Duplicates are detected by textual matching, and only the first occurrence is retained. For each retained problem, we extract the original and auxiliary-line-annotated diagrams and apply OpenCV-based filtering to discard low-resolution or unclear images.

**Structured Data Extraction.** Building on the high-quality diagram pairs obtained in the previous step, we extract three textual fields for each problem: the problem description, the final answer, and the auxiliary-line description. Because the source Word files embed MathType formulas that standard parsers handle poorly, we render pages as images and parse them with a fine-tuned LVM. All outputs are packaged in a structured JSON format.

**Manual Verification.** Each instance is independently reviewed by two AI researchers to ensure accuracy, completeness, uniqueness, semantic consistency, and visual quality. Instances flagged by either reviewer undergo collaborative re-examination, with up to three review rounds to identify and correct subtle or ambiguous errors. This step resolves complex symbolic expressions and diagrammatic ambiguities that automated tools often misinterpret, and remains indispensable for ensuring high data fidelity in model training and evaluation.

**Dataset Statistics.** AuxSolidMath comprises 3,018 solid geometry problems collected from real high school examination papers. Within this dataset, we curate a new benchmark, GeoAuxBench, designed specifically to evaluate a model’s ability to construct auxiliary lines, a skill

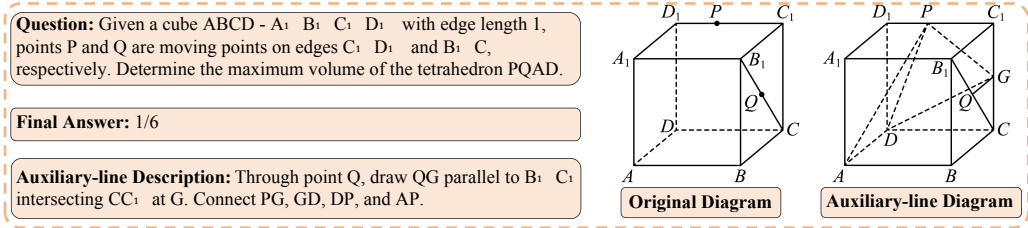


Figure 4: An example from the AuxSolidMath dataset.

essential to solving complex geometry problems. GeoAuxBench contains 302 examples and is divided into two difficulty levels, *Easy* (150) and *Hard* (152), using the original difficulty annotations from the source exam papers rather than post hoc labels. *Hard* problems involve reasoning over implicit spatial relations (e.g., cross-plane or hidden projections), while *Easy* problems rely on explicit relations. By inheriting these exam-defined criteria, GeoAuxBench supports realistic evaluation of auxiliary-line construction in solid geometry.

## 4 Experiments

### 4.1 Experimental Settings

**Benchmark.** We conduct our evaluation on **GeoAuxBench**, a newly introduced benchmark for auxiliary-line construction with two difficulty tiers: **Easy** and **Hard**. We further report results on widely used benchmarks, *i.e.*, **MathVision** (Wang et al., 2024a), **SolidGeo** (Wang et al., 2025a), and the **OlympiadBench** subset of SolidGeo.

**Metrics.** We evaluate model performance using Pass@k (Chen et al., 2021b), a widely adopted metric introduced by OpenAI. Specifically, we report Pass@1 and Pass@5: Pass@1 measures the accuracy of a single generated solution, while Pass@5 denotes the proportion of problems for which at least one of five generated solutions is correct.

**Models.** We assess GeoVLMath at two model scales, 3B and 7B, both built upon the Qwen2.5-VL backbone. We compare GeoVLMath against 20 strong LVLMM baselines, covering both closed-source and open-source models, as well as representative methods following the code-driven visual construction paradigm, including V-Thinker (Qiao et al., 2025) and CodePlot-CoT (Duan et al., 2025).

### 4.2 Main Results

As shown in Table 1 and Table 2, we make three key observations: **(1) GeoVLMath achieves strong and robust performance among models of comparable scale**

**across benchmarks.** On GeoAuxBench, both GeoVLMath-3B and GeoVLMath-7B outperform their corresponding base models, Qwen2.5-VL-3B-Instruct and Qwen2.5-VL-7B-Instruct, on pass@5. GeoVLMath-3B improves by **+3.53%** (11.59%  $\rightarrow$  15.12%), while GeoVLMath-7B yields a larger gain of **+10.16%** (15.89%  $\rightarrow$  26.05%). Moreover, GeoVLMath consistently surpasses the representative code-driven visual construction method V-Thinker on Pass@5, improving by **+12.14%** (13.91%  $\rightarrow$  26.05%), despite not relying on executable drawing code or intermediate diagram rendering. Beyond this task-aligned benchmark, GeoVLMath also achieves the best performance among the compared models on widely used public benchmarks. These results demonstrate that the auxiliary-line-aware training signal generalizes beyond GeoAuxBench and improves geometric reasoning robustness. **(2) GeoAuxBench-Hard is a challenging benchmark that clearly distinguishes LVLMM capabilities in geometric reasoning.** For example, even strong closed-source models such as Gemini-2.5-Flash and gpt-5-mini achieve Pass@1 scores of **63.16%**, whereas Qwen2.5-VL-72B-Instruct attains only **13.16%**, highlighting a substantial performance gap. This is primarily because the benchmark demands deliberate auxiliary-line construction and multi-step spatial reasoning while minimizing shortcut opportunities, and provides reference diagrams for fine-grained error analysis. Together, GeoAuxBench-Hard serves as a concise yet highly discriminative testbed for differentiating LVLMM capabilities. **(3) Model scale alone does not compensate for insufficient auxiliary-line awareness.** On GeoAuxBench-Easy, GeoVLMath-7B achieves higher pass@5 than Qwen2.5-VL-32B-Instruct (**35.56%** vs. **23.33%**). On GeoAuxBench-Hard, GeoVLMath-7B also outperforms Qwen2.5-VL-32B-Instruct (**16.67%** vs. **13.16%**). Error analysis shows that Qwen2.5-VL-32B-Instruct rarely constructs auxiliary lines and thus overlooks latent spatial constraints, whereas GeoVLMath-7B proac-

LVLM	Easy		Hard		Average	
	Pass@1	Pass@5	Pass@1	Pass@5	Pass@1	Pass@5
<b>Closed-source LVLMs</b>						
gpt-5-mini	89.33	92.67	63.16	75.00	76.16	83.78
o4-mini-2025-04-16	84.00	93.33	60.53	74.34	72.19	83.77
GPT-4o	8.67	25.33	6.58	15.13	7.62	20.20
Gemini-2.0-Flash	37.33	62.67	25.00	39.47	31.12	50.99
Gemini-2.5-Flash	84.00	91.33	63.16	78.95	73.51	85.10
Claude 3.7 Sonnet 20250219	15.33	41.33	13.16	28.29	14.24	34.77
Claude Sonnet 4 20250514	56.00	77.33	30.92	44.74	43.38	60.93
<b>Open-source LVLMs (3B-14B)</b>						
V-Thinker (7B)	8.00	17.33	3.29	10.53	5.63	13.91
InternVL3-8B	9.33	25.33	5.92	15.79	7.61	20.53
Llama-3.2-11B-Vision-Instruct	2.00	12.00	3.29	5.92	2.65	8.94
InternVL3-14B	13.33	28.67	5.92	15.13	9.60	21.86
Qwen2.5-VL-3B-Instruct	2.00	14.89	1.97	8.33	1.98	11.59
<b>GeoVLMath-3B (Ours)</b>	12.89	20.44	5.70	9.87	9.27	15.12
Qwen2.5-VL-7B-Instruct	5.14	20.67	3.95	11.18	4.54	15.89
<b>GeoVLMath-7B (Ours)</b>	<b>14.67</b>	<b>35.56</b>	<b>5.92</b>	<b>16.67</b>	<b>10.27</b>	<b>26.05</b>
<b>Open-source LVLMs (17B-78B)</b>						
CodePlot-CoT (32B)	4.00	8.67	2.63	5.26	3.31	6.95
Qwen2-VL-72B-Instruct	6.00	15.33	5.26	8.55	5.63	11.92
Qwen2.5-VL-32B-Instruct	20.67	23.33	11.18	13.16	15.89	18.21
Llama-4-Scout-17B-16E-Instruct	20.67	36.67	7.89	18.42	14.24	27.48
InternVL3-38B	19.33	41.33	10.53	21.71	14.90	31.46
Qwen2.5-VL-72B-Instruct	24.00	40.67	13.16	19.74	18.54	30.14
InternVL3-78B	16.67	36.67	9.21	21.05	12.92	28.81

\* Bold indicates the best results for models of similar sizes.

Table 1: Overall evaluation on GeoAuxBench (%).

LVLM	MathVision		OlympiadBench		SolidGeo	
	Pass@1	Pass@5	Pass@1	Pass@5	Pass@1	Pass@5
V-Thinker (7B)	16.36	31.82	4.60	13.22	10.39	29.38
CodePlot-CoT (32B)	10.00	29.09	2.87	5.17	16.62	30.86
Qwen2.5-VL-3B-Instruct	5.45	16.36	2.30	6.32	5.04	12.17
<b>GeoVLMath-3B (Ours)</b>	5.45	20.91	14.94	25.86	14.54	32.34
Qwen2.5-VL-7B-Instruct	11.82	20.91	2.87	13.22	5.34	14.84
<b>GeoVLMath-7B (Ours)</b>	<b>16.36</b>	<b>36.36</b>	<b>25.86</b>	<b>50.57</b>	<b>17.51</b>	<b>42.34</b>

Table 2: Overall evaluation on widely used benchmarks (%).

tively introduces appropriate auxiliary lines and exploits the induced constraints during reasoning. This comparison highlights that auxiliary-line supervision, rather than model scale alone, is critical for reliable geometric reasoning.

### 4.3 Cross-modal Reward Model

Leveraging AuxSolidMath triplets  $\langle I, d_{\text{aux}}, I^+ \rangle$ , we apply rule-based perturbations to simulate typical auxiliary-line errors and use the resulting data to train a cross-modal reward model on Qwen2.5-VL-7B. The dataset comprises 2,970 training examples and 330 test examples. We train the model for 3 epochs with a batch size of 16, using the AdamW optimizer with a learning rate of  $2e-5$  and a cosine learning rate scheduler with a 0.1 warm-up ratio. During training, the vision tower and projection modules are frozen, while the language model remains trainable. The model achieves a pass@1 accuracy of **98.18%** on the test set, indicating reliable alignment between textual auxiliary-line descriptions and their visually annotated counterparts.

### 4.4 Ablation Studies

**Cross-Modal Reward.** We assess the role of cross-modal supervision with two variants, keeping all other settings unchanged. (a) *w/o cross-modal reward*. This variant removes all supervision related to auxiliary lines and trains the model solely for final-answer accuracy, with no supervision on whether auxiliary lines are introduced. This allows us to assess the effect of training with answer-only supervision, approximating a setting where auxiliary lines are omitted from the training objective. (b) *Textual reward*. In this variant, cross-modal consistency is replaced with a text-only semantic similarity objective that evaluates the similarity between the generated auxiliary-line description and the ground-truth annotation. Concretely, we use *EmbeddingGemma* (DeepMind, 2025c) to encode sentences and compute a similarity score for training. This variant favors fluent textual descriptions but does not enforce grounding to the input diagram. *Findings.* As reported in Table 3, removing the cross-modal reward re-

	Easy		Hard		Average	
	Pass@1	Pass@5	Pass@1	Pass@5	Pass@1	Pass@5
GeoVLM <sub>Math-7B</sub>	14.67	35.56	5.92	16.67	10.27	26.05
w/o Cross-Modal Reward	10.89 <sub>↓3.78</sub>	32.22 <sub>↓3.34</sub>	4.82 <sub>↓1.10</sub>	13.60 <sub>↓3.07</sub>	7.83 <sub>↓2.44</sub>	22.85 <sub>↓3.20</sub>
Textual Reward	10.67 <sub>↓4.00</sub>	28.44 <sub>↓7.12</sub>	4.39 <sub>↓1.53</sub>	12.50 <sub>↓4.17</sub>	7.51 <sub>↓2.76</sub>	20.42 <sub>↓5.63</sub>
w/o RL	3.33 <sub>↓11.34</sub>	20.44 <sub>↓15.12</sub>	3.95 <sub>↓1.97</sub>	11.18 <sub>↓5.49</sub>	3.64 <sub>↓6.63</sub>	15.78 <sub>↓10.27</sub>

Table 3: Results of ablation studies (%).

sults in performance degradation (Average pass@1: **10.27** → **7.83**, pass@5: **26.05** → **22.85**), underscoring the importance of geometry-aware supervision for instances that require introducing auxiliary lines. Substituting it with a purely textual similarity objective performs even worse (Average pass@1: **10.27** → **7.51**, pass@5: **26.05** → **20.42**), consistent with our pilot finding in Section 1 that incorrect auxiliary lines can be worse than none. These declines suggest that lexical alignment introduces spurious signals and conflicts with precise diagram grounding, favoring surface-level paraphrases over geometry-aware reasoning. Error analysis reveals distinct failure modes: (a) often ignores auxiliary-line construction and overfits to answer-only cues; (b) produces fluent but visually inconsistent descriptions (e.g., incorrect lines) that fail to constrain diagram-based reasoning. Overall, text-only alignment fails to faithfully capture geometric structure, motivating the need for visually grounded, structure-preserving diagram-text alignment to support rigorous verification of geometric measurements and structural relations.

**Reinforcement Learning.** To quantify the contribution of reinforcement learning, we remove the RL stage and train an SFT-only variant. As shown in Table 3, GeoVLM<sub>Math-7B</sub> trained with SFT+RL consistently outperforms its SFT-only counterpart, with clear performance degradation when the RL stage is removed. This improvement reflects the role of RL in moving beyond strict imitation: reward-aligned optimization encourages exploration of more effective strategies and enables credit assignment for beneficial intermediate steps, rather than relying on surface-level matching alone. As a result, RL acts as a post-SFT catalyst that consolidates preliminary SFT competence into robust multi-step reasoning, particularly in scenarios requiring auxiliary-line construction.

## 5 Related Work

Recent LVLMs (Anthropic, 2025b; DeepMind, 2025a; OpenAI, 2025) have advanced geometric problem solving, particularly in plane geometry.

Prior work mainly follows two lines: direct generation of answers or reasoning paths from multimodal inputs (Ning et al., 2025; Xia et al., 2025; Gao et al., 2025), which is constrained by intrinsic reasoning capacity, and tool-augmented reasoning that generates executable code for symbolic computation or geometric operations (Zhao et al., 2025; Sharma et al., 2025; Chen et al., 2024). More recently, a code-driven visual construction paradigm has emerged, where models generate code to construct or modify diagrams during reasoning, as exemplified by Visual Sketchpad (Hu et al., 2024), V-Thinker (Qiao et al., 2025), and CodePlot-CoT (Duan et al., 2025). While these methods enable interactive visual feedback, they depend on precise code execution and explicit coordinate annotations, and tightly couple auxiliary-line construction with the LVLM’s intrinsic reasoning, thereby limiting robustness in solid geometry. In contrast, our approach incorporates auxiliary-line construction into an RL framework guided by a cross-modal reward model, decoupling construction quality from intrinsic reasoning, without explicit geometric specifications, enabling robust auxiliary-line reasoning across model scales.

## 6 Conclusion

Auxiliary-line reasoning in solid geometry exposes a key limitation of current LVLMs: the lack of reliable learning signals for structured visual construction. We address this challenge by formulating auxiliary-line construction as a vision-language alignment problem and introducing a RL framework guided by a cross-modal reward that directly measures diagram-text correspondence. This design enables stable optimization without relying on explicit geometric specifications or executable code. To support learning at scale, we automatically construct **AuxSolidMath**, a high-quality dataset of real-exam solid geometry problems with aligned diagrams and auxiliary-line annotations, providing a reusable resource for the community. Leveraging this framework, **GeoVLM<sub>Math</sub>** consistently improves auxiliary-line reasoning.

## 619 Limitations

620 While our approach demonstrates strong effective-  
621 ness for auxiliary-line reasoning in solid geome-  
622 try, our experiments primarily consider settings  
623 where auxiliary-line constructions admit concise  
624 high-level descriptions. Extending the evaluation  
625 to even broader or more diverse geometric con-  
626 structions remains an interesting direction for fu-  
627 ture work. In addition, our cross-modal reward  
628 model relies on annotated auxiliary-line diagrams  
629 for supervision during reinforcement learning. Al-  
630 though we introduce AuxSolidMath to support this  
631 setting, exploring more scalable forms of weak or  
632 self-supervised reward signals is a promising direc-  
633 tion for future research.

## 634 References

635 Anthropic. 2025a. Claude 3.7 sonnet and claude  
636 code. [https://www.anthropic.com/news/  
637 claude-3-7-sonnet](https://www.anthropic.com/news/claude-3-7-sonnet).

638 Anthropic. 2025b. Claude 4. [https://www.  
639 anthropic.com/news/claude-4](https://www.anthropic.com/news/claude-4).

640 Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wen-  
641 bin Ge, Sibao Song, Kai Dang, Peng Wang, Shijie  
642 Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Ming-  
643 Hsuan Yang, Zhaohai Li, Jianqiang Wan, Pengfei  
644 Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8  
645 others. 2025. Qwen2.5-vl technical report. *CoRR*,  
646 abs/2502.13923.

647 Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin,  
648 Chongyu Chen, and Xiaodan Liang. 2022. Unigeo:  
649 Unifying geometry logical reasoning via reformulat-  
650 ing mathematical expression. In *Proceedings of the  
651 2022 Conference on Empirical Methods in Natural  
652 Language Processing, EMNLP 2022*, pages 3313–  
653 3323.

654 Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang,  
655 Lingbo Liu, Eric P. Xing, and Liang Lin. 2021a.  
656 Geoqa: A geometric question answering benchmark  
657 towards multimodal numerical reasoning. In *Find-  
658 ings of the Association for Computational Linguistics:  
659 ACL/IJCNLP 2021*, pages 513–523.

660 Jingchang Chen, Hongxuan Tang, Zheng Chu, Qiang-  
661 long Chen, Zekun Wang, Ming Liu, and Bing Qin.  
662 2024. Divide-and-conquer meets consensus: Un-  
663 leashing the power of functions in code generation.  
664 In *Advances in Neural Information Processing Sys-  
665 tems 38: Annual Conference on Neural Information  
666 Processing Systems 2024, NeurIPS 2024*.

667 Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan,  
668 Henrique Pondé de Oliveira Pinto, Jared Kaplan,  
669 Harri Edwards, Yuri Burda, Nicholas Joseph, Greg  
670 Brockman, Alex Ray, Raul Puri, Gretchen Krueger,

Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela  
Mishkin, Brooke Chan, Scott Gray, and 39 others.  
2021b. Evaluating large language models trained on  
code. *CoRR*, abs/2107.03374.

DeepMind. 2025a. Gemini 2.5 flash. [https://  
675 deepmind.google/models/gemini/flash/](https://deepmind.google/models/gemini/flash/).  
676

DeepMind. 2025b. Gemini 2.5 pro. [https://  
677 deepmind.google/models/gemini/pro/](https://deepmind.google/models/gemini/pro/).  
678

Google DeepMind. 2025c. Embeddinggemma:  
A 308m multilingual text embedding model.  
[https://ai.google.dev/gemma/docs/  
679 embeddinggemma?hl=zh-cn](https://ai.google.dev/gemma/docs/embeddinggemma?hl=zh-cn).  
680  
681  
682

Chengqi Duan, Kaiyue Sun, Rongyao Fang, Manyuan  
Zhang, Yan Feng, Ying Luo, Yufang Liu, Ke Wang,  
Peng Pei, Xunliang Cai, Hongsheng Li, Yi Ma, and  
Xihui Liu. 2025. Codeplot-cot: Mathematical vi-  
sual reasoning by thinking with code-driven images.  
*CoRR*, abs/2510.11718.  
683  
684  
685  
686  
687  
688

Yumeng Fu, Jiayin Zhu, Lingling Zhang, Bo Zhao,  
Shaoxuan Ma, Yushun Zhang, Yanrui Wu, and  
Wenjun Wu. 2025. Geolax: A benchmark for  
evaluating mllms’ geometry performance on long-  
step problems requiring auxiliary lines. *CoRR*,  
arXiv:2508.06226v1.  
689  
690  
691  
692  
693  
694

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wan-  
jun Zhong, Yufei Wang, Lanqing Hong, Jianhua  
Han, Hang Xu, Zhenguo Li, and Lingpeng Kong.  
2025. G-llava: Solving geometric problem with  
multi-modal large language model. In *The Thirteenth  
International Conference on Learning Representa-  
tions, ICLR 2025*.  
695  
696  
697  
698  
699  
700  
701

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao  
Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shi-  
rong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025.  
Deepseek-r1: Incentivizing reasoning capability in  
llms via reinforcement learning. *arXiv preprint  
arXiv:2501.12948*.  
702  
703  
704  
705  
706  
707

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Os-  
tendorf, Luke Zettlemoyer, Noah A. Smith, and Ran-  
jay Krishna. 2024. Visual sketchpad: Sketching as  
a visual chain of thought for multimodal language  
models. In *Advances in Neural Information Pro-  
cessing Systems 38: Annual Conference on Neural  
Information Processing Systems 2024, NeurIPS 2024,  
Vancouver*.  
708  
709  
710  
711  
712  
713  
714  
715

Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam  
Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow,  
Akila Welihinda, Alan Hayes, Alec Radford, Alek-  
sander Madry, Alex Baker-Whitcomb, Alex Beutel,  
Alex Borzunov, Alex Carney, Alex Chow, Alex Kir-  
illov, Alex Nichol, Alex Paino, and 79 others. 2024.  
Gpt-4o system card. *CoRR*, abs/2410.21276.  
716  
717  
718  
719  
720  
721  
722

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chun-  
yuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-  
Wei Chang, Michel Galley, and Jianfeng Gao. 2024.  
Mathvista: Evaluating mathematical reasoning of  
723  
724  
725  
726

727	foundation models in visual contexts. In <i>The Twelfth International Conference on Learning Representations, ICLR 2024</i> .	
728		
729		
730	Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In <i>Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021</i> , pages 6774–6786.	
731		
732		
733		
734		
735		
736		
737		
738	Bin Ma, Pengpeng Jian, Cong Pan, Yanli Wang, and Wei Ma. 2024. A geometric neural solving method based on a diagram text information fusion analysis. <i>Scientific Reports</i> , 14(1):31906.	
739		
740		
741		
742	Meta. 2024. <i>Llama 3.2: Revolutionizing edge ai and vision with open, customizable models</i> .	
743		
744	Meta. 2025. The llama 4 herd: The beginning of a new era of natively multimodal ai innovation. <a href="https://ai.meta.com/blog/llama-4-multimodal-intelligence/">https://ai.meta.com/blog/llama-4-multimodal-intelligence/</a> .	
745		
746		
747		
748	Maizhen Ning, Zihao Zhou, Qiufeng Wang, Xiaowei Huang, and Kaizhu Huang. 2025. GNS: solving plane geometry problems by neural-symbolic reasoning with multi-modal llms. In <i>AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence</i> , pages 24957–24965.	
749		
750		
751		
752		
753		
754	OpenAI. 2025. Gpt-5 system card. <a href="https://cdn.openai.com/gpt-5-system-card.pdf">https://cdn.openai.com/gpt-5-system-card.pdf</a> .	
755		
756	OpenAI. 2025. Openai o3 and o4-mini system card. <a href="https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf">https://cdn.openai.com/pdf/2221c875-02dc-4789-800b-e7758f3722c1/o3-and-o4-mini-system-card.pdf</a> .	
757		
758		
759		
760	Runqi Qiao, Qiuna Tan, Minghan Yang, Guanting Dong, Peiqing Yang, Shiqiang Lang, Enhui Wan, Xiaowan Wang, Yida Xu, Lan Yang, Chong Sun, Chen Li, and Honggang Zhang. 2025. V-thinker: Interactive thinking with images. <i>CoRR</i> , abs/2511.04460.	
761		
762		
763		
764		
765	Aditya Sharma, Aman Dalmia, Mehran Kazemi, Amal Zouaq, and Christopher Pal. 2025. Geocoder: Solving geometry problems by generating modular code through vision-language models. In <i>Findings of the Association for Computational Linguistics: NAACL 2025</i> , pages 7340–7356.	
766		
767		
768		
769		
770		
771	Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. In <i>Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024</i> .	
772		
773		
774		
775		
776		
777		
778	Peijie Wang, Chao Yang, Zhong-Zhi Li, Fei Yin, Dekang Ran, Mi Tian, Zhilong Ji, Jinfeng Bai, and Chenglin Liu. 2025a. SOLIDGEO: measuring multimodal spatial math reasoning in solid geometry. <i>CoRR</i> , abs/2505.21177.	
779		
780		
781		
782		
	Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model’s perception of the world at any resolution. <i>CoRR</i> , abs/2409.12191.	783 784 785 786 787 788 789 790
	Xiaofeng Wang, Yiming Wang, Wenhong Zhu, and Rui Wang. 2025b. Do large language models truly understand geometric structures? In <i>The Thirteenth International Conference on Learning Representations, ICLR 2025</i> .	791 792 793 794 795
	Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. <i>arXiv preprint arXiv:2510.18234</i> .	796 797 798
	Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, Conghui He, Botian Shi, Tao Chen, Junchi Yan, and Bo Zhang. 2025. Geox: Geometric problem solving through unified formalized vision-language pre-training. In <i>The Thirteenth International Conference on Learning Representations, ICLR 2025</i> .	799 800 801 802 803 804 805 806
	Shicheng Xu, Liang Pang, Huawei Shen, Xueqi Cheng, and Tat-Seng Chua. 2024. Search-in-the-chain: Interactively enhancing large language models with search for knowledge-intensive tasks. In <i>Proceedings of the ACM on Web Conference 2024, WWW 2024</i> , pages 1362–1373.	807 808 809 810 811 812
	Bohan Zhang, Xiaokang Zhang, Jing Zhang, Jifan Yu, Sijia Luo, and Jie Tang. 2025. Cot-based synthesizer: Enhancing LLM performance through answer synthesis. In <i>Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025</i> , pages 6286–6303.	813 814 815 816 817 818
	Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2024. MATHVERSE: does your multi-modal LLM truly see the diagrams in visual math problems? In <i>Computer Vision - ECCV 2024 - 18th European Conference</i> , pages 169–186.	819 820 821 822 823 824 825
	Junbo Zhao, Ting Zhang, Jiayu Sun, Mi Tian, and Hua Huang. 2025. Pi-gps: Enhancing geometry problem solving by unleashing the power of diagrammatic information. <i>CoRR</i> , abs/2503.05543.	826 827 828 829
	Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. <i>CoRR</i> , abs/2403.13372.	830 831 832 833
	Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, and 32	834 835 836 837 838

839 others. 2025. Internvl3: Exploring advanced training  
840 and test-time recipes for open-source multimodal  
841 models. *CoRR*, abs/2504.10479.

## 842 A The Use of AI Assistants

843 In this paper, ChatGPT was used exclusively for  
844 language polishing, including grammar correction,  
845 phrasing, and stylistic refinement. It was not used  
846 to generate scientific content such as research ideas,  
847 methods, experiments, or related work. No confi-  
848 dential, personal, or proprietary information was  
849 shared with the model. The authors take full re-  
850 sponsibility for the scientific content, which was  
851 entirely authored and verified by the authors.

## 852 B Data Creation

853 In this section, we detail the four progressive steps  
854 of our data creation pipeline.

### 855 B.1 Automated Problem Identification

856 To construct the AuxSolidMath dataset, we first  
857 collect over 350 sets of high school geometry prob-  
858 lems from publicly available online sources. Given  
859 that the dataset is intended to support constructive  
860 geometric reasoning, we specifically target prob-  
861 lems that necessitate auxiliary line constructions as  
862 integral components of their solutions.

863 To efficiently identify such problems, we de-  
864 sign an automated two-stage filtering pipeline using  
865 Python scripts. In the first stage, we detect prob-  
866 lems whose solutions contain explicit mentions of  
867 auxiliary-line constructions. Specifically, we ap-  
868 ply regular expression patterns to locate question  
869 number markers that are explicitly present in the  
870 exam papers and use these markers to segment the  
871 content into individual problem units. For each  
872 problem, we examine the solution for verbs that  
873 signal the introduction of auxiliary lines (e.g., "con-  
874 nect," "construct," "draw," "establish"). Problems  
875 lacking such terms are discarded, while those con-  
876 taining relevant cues are retained. In the second  
877 stage, we further refine the selection by ensuring  
878 that each retained problem contains both the origi-  
879 nal diagram and an auxiliary-line diagram. To this  
880 end, we quantify the number of diagrams associ-  
881 ated with each problem. Problems with fewer than  
882 two diagrams are excluded, whereas those with at  
883 least two, which usually represent the original and  
884 modified diagrams, are preserved. This automated  
885 pipeline enables scalable and consistent filtering  
886 of auxiliary-line geometry problems, significantly  
887 reducing manual annotation effort.

## 888 B.2 Automated Deduplication and Diagram 889 Extraction

890 Upon identifying geometry problems requiring aux-  
891 iliary lines, we employ an automated pipeline to  
892 deduplicate instances and extract the associated di-  
893 agrams. This step guarantees the uniqueness and  
894 visual quality of data instances for downstream  
895 model training.

896 **Problem Deduplication.** To eliminate duplicate  
897 problems, we retain only the first occurrence of  
898 each unique problem based on its textual content.  
899 Concretely, we initialize a global problem set as  
900 an empty collection. We then sequentially process  
901 all Word-formatted exam papers, examining only  
902 the problem descriptions while ignoring the associ-  
903 ated solutions and diagrams. For each problem, if  
904 its description is not already present in the global  
905 set, we add the problem; otherwise, we discard it  
906 as a duplicate. This procedure ensures that iden-  
907 tical problems, which often recur across different  
908 examinations, are retained only once.

909 **Diagram Extraction.** Following deduplication,  
910 we extract, filter, and store the geometric diagrams  
911 associated with each retained problem. A key chal-  
912 lenge lies in reliably distinguishing true geomet-  
913 ric figures from image-embedded mathematical  
914 expressions (e.g., MathType equations), as both  
915 appear in Word exam papers. Existing Python li-  
916 braries are unable to make this distinction accu-  
917 rately, often misclassifying equations as diagrams  
918 and introducing significant noise into the extrac-  
919 tion process. To overcome this limitation, we innova-  
920 tively integrate the Apache POI library through a  
921 custom Java implementation, enabling fine-grained  
922 control over the parsing of Word documents. This  
923 setup enables reliable identification and extraction  
924 of genuine geometric diagrams while effectively fil-  
925 tering out formula-rendered images. To further en-  
926 sure visual quality, the extracted diagrams are then  
927 processed using OpenCV to discard low-resolution  
928 or unclear diagrams. The remaining diagrams are  
929 subsequently saved using a standardized naming  
930 convention that distinguishes between the original  
931 and the annotated versions of the auxiliary lines.  
932 To be more specific, for each problem indexed by  
933  $i$ , we store two images:  $\{i\}.png$ , which contains  
934 the original diagram, and  $\{i\}_{auxiliary}.png$ , which  
935 includes the corresponding auxiliary-line diagram.  
936 This consistent format facilitates downstream align-  
937 ment between textual and visual modalities within  
938 the multimodal processing pipeline.

939 **B.3 Structured Data Extraction**

940 Building on the high-quality geometric diagrams

941 obtained in the previous step, we proceed to extract

942 the corresponding textual content for each instance,

943 including the problem description, the final answer,

944 and the auxiliary-line description. This extraction

945 process is non-trivial, as the original Word docu-

946 ments frequently embed mathematical expressions

947 in MathType formats that are not reliably supported

948 by standard document parsing tools.

949 To address this challenge, we render the pro-

950 cessed Word documents as images, thereby en-

951 abling LVLMs to leverage their visual reasoning ca-

952 pabilities. Although this approach appears straight-

953 forward, open-source models such as Qwen2.5-

954 VL-7B-Instruct (Bai et al., 2025) often struggle

955 to accurately parse complex geometry problems

956 involving symbolic notation and mathematical ex-

957 pressions. In contrast, closed-source models like

958 GPT-4o (Hurst et al., 2024) exhibit significantly

959 stronger performance, but their reliance on com-

960 mercial APIs introduces substantial costs and limits

961 scalability in large-scale applications. To balance

962 accuracy with scalability, we adopt a hybrid strat-

963 egy<sup>2</sup>. Specifically, we first utilize an advanced

964 closed-source model (*i.e.*, GPT-4o) to generate a

965 small, high-quality seed dataset comprising 300

966 manually verified instances. This curated dataset is

967 then used to fine-tune an LVLm (*i.e.*, Qwen2.5-VL-

968 7B-Instruct), resulting in a lightweight, domain-

969 adapted model capable of accurate and scalable

970 text extraction. The final output consists of the ex-

971 tracted problem description, the final answer, and

972 the auxiliary-line description, all encapsulated in a

973 structured JSON format. This unified representa-

974 tion facilitates consistent data handling and serves

975 as a foundation for training a robust open-source

976 text extraction model. By releasing this model,

977 we aim to contribute a practical and reusable re-

978 source to the broader research community working

979 on geometry-aware vision-language understanding.

### 980 B.4 Manual Verification

981 To ensure the quality and reliability of the final

982 dataset, we perform a manual verification step that

983 assesses each data instance in terms of accuracy,

984 completeness, uniqueness, and semantic consis-

985 tency, alongside visual quality criteria such as im-

<sup>2</sup>DeepSeek-OCR(Wei et al., 2025) was released on Octo-  
ber 20, 2025 and was not available during data construction. A  
Qwen2.5-VL-based extractor fine-tuned on 300 seed samples  
achieved approximately 98% accuracy and was therefore used.

986 age clarity and resolution. Two AI researchers

987 serve as independent checkers. Each instance is in-

988 dependently reviewed by both researchers. If either

989 checker identifies a potential issue, the instance is

990 collaboratively revised. This process is repeated up

991 to three times per instance, ensuring that all errors,

992 including subtle or ambiguous ones, are systemati-

993 cally identified and corrected. Manual verification

994 plays a critical role in resolving complex symbolic

995 expressions and ambiguous diagrammatic content

996 that automated tools may misinterpret. Despite

997 its relatively low cost and effort, this step remains

998 indispensable for ensuring the high data fidelity

999 necessary for a reliable model.

## 1000 C Experimental Setup

### 1001 C.1 Models

1002 On the closed-source models, we include lead-

1003 ing models such as gpt-5-mini (OpenAI, 2025),

1004 o4-mini (OpenAI, 2025) and GPT-4o (Hurst

1005 et al., 2024), Gemini-2.0-Flash and Gemini-

1006 2.5-Flash (DeepMind, 2025a), Claude 3.7 Son-

1007 net (Anthropic, 2025a) and Claude Sonnet 4

1008 20250514 (Anthropic, 2025b). These models

1009 represent the forefront of multimodal reason-

1010 ing among closed-source models, although their

1011 internal architectures remain undisclosed. On

1012 the open-source models, we consider several

1013 publicly available high-performance models, in-

1014 cluding the Qwen2 VL (Wang et al., 2024b)

1015 and Qwen2.5 VL series (Bai et al., 2025), In-

1016 ternVL 3 families (Zhu et al., 2025), LLaMA-

1017 3.2-11B-Vision-Instruct (Meta, 2024) and Llama-

1018 4-Scout-17B-16E-Instruct (Meta, 2025). In addi-

1019 tion, we include representative methods follow-

1020 ing the code-driven visual construction paradigm,

1021 namely V-Thinker (Qiao et al., 2025) and CodePlot-

1022 CoT (Duan et al., 2025), which generate executable

1023 code to construct or modify diagrams as part of the

1024 reasoning process. These approaches provide an

1025 important comparison point for evaluating differ-

1026 ent strategies of integrating visual construction into

1027 multimodal reasoning. Together, these models en-

1028 compass a range of design paradigms, parameter

1029 scales, and instruction tuning strategies, forming a

1030 comprehensive and robust foundation for evaluat-

1031 ing auxiliary-line reasoning in multimodal settings.

1032 Note that models such as Gemini-2.5 Pro (Deep-

1033 Mind, 2025b) and OpenAI o3 (OpenAI, 2025) are

1034 excluded from our study due to limited accessibility

1035 and high inference costs.

## C.2 Training Implementation Details

We adopt a two-stage training paradigm based on the Qwen2.5-VL series, including Qwen2.5-VL-3B and Qwen2.5-VL-7B, consisting of the SFT stage and the RL stage.

**SFT Stage.** The SFT phase is conducted using the LLaMA-Factory framework (Zheng et al., 2024). For Qwen2.5-VL-7B, we train the model for 5 epochs with a per-device batch size of 2 and a gradient accumulation step of 8 (effective batch size of 16). We use the AdamW optimizer with a learning rate of  $2e-5$  and apply a cosine learning rate scheduler with a warmup ratio of 0.1. The model is trained in bf16 precision. Vision and projection modules are frozen during this stage, while the language model remains unfrozen. For Qwen2.5-VL-3B, we adopt the same training configuration as the 7B variant, except learning rate and training epochs. Specifically, Qwen2.5-VL-3B is trained for 5 epochs with a learning rate of  $3e-5$ .

**RL Stage.** For the Qwen2.5-VL-7B model, both training and validation data are loaded from Parquet files containing question-diagram pairs. The maximum response length is set to 8,192 tokens, and both the rollout and validation batch sizes are set to 16. The actor is optimized using AdamW (learning rate  $2e-6$ , weight decay  $1e-2$ , no warmup). KL regularization is applied using the `low_var_kl` penalty with a coefficient of  $1e-2$ . Training runs for 6 epochs using bf16 precision, with gradient checkpointing and partial FSDP offloading enabled for memory efficiency.

**Rewards.** The overall reward is the sum of a cross-modal auxiliary-line consistency reward and a final-answer accuracy reward, where the auxiliary-line component is weighted by  $\alpha = 0.1$ . For Qwen2.5-VL-3B, we adopt the same RL configuration as the 7B model, with adjustments to the batch size and the number of training epochs. Specifically, Qwen2.5-VL-3B is trained for 4 epochs with a batch size of 8.

All training was conducted on a server equipped with two NVIDIA A100 80GB and two NVIDIA A800 80GB GPUs. The SFT stage was performed on two A100 GPUs, whereas the reinforcement learning stage utilized all four GPUs. For response generation during training and evaluation, we enabled stochastic sampling with temperature set to 0.7 and top-p set to 0.95.

## D Related Work

In this section, we also review benchmarks and datasets for geometric reasoning. Most benchmarks and datasets for geometric problem solving focus on plane geometry, where diagrams and problems involve two-dimensional figures. Well-known resources in this area include Geometry3K (Lu et al., 2021), GeoQA (Chen et al., 2021a), UniGeo (Chen et al., 2022), and GeomRel (Wang et al., 2025b), which primarily cover plane geometry problems. A concurrent benchmark, GeoLaux (Fu et al., 2025), explores the use of auxiliary lines in plane geometry, but is limited to simple cases and lacks engagement with the spatial complexity of solid geometry. Nonetheless, there remains a lack of dedicated resources for solid geometry, even though solving such problems often requires interpreting three-dimensional relationships and drawing auxiliary lines to uncover hidden spatial structures. While SolidGeo (Wang et al., 2025a) is a recent benchmark that focuses exclusively on solid geometry, it does not explicitly require auxiliary lines for solving its problems, leaving this important aspect of spatial reasoning underexplored. Similarly, other benchmarks such as MathVision (Wang et al., 2024a), MathVista (Lu et al., 2024), and MathVerse (Zhang et al., 2024) contain only a limited number of solid geometry problems, and these also do not require auxiliary lines to reach the solution. As a result, these resources fall short of evaluating a model’s ability to solve complex solid geometry problems where auxiliary lines are essential for uncovering implicit spatial relationships. To address this gap, we present AuxSolidMath, the first dedicated dataset for solid geometry problems that require auxiliary lines to solve. It offers comprehensive multimodal supervision, including the original diagram, the problem statement, textual descriptions of the required auxiliary lines, the final answer, and a corresponding diagram annotated with those lines, enabling models to learn how the auxiliary lines facilitate solid geometry reasoning.

## E Prompts

### E.1 Prompts for Supervised Fine-tuning

Figure 5 presents the two-part prompt template used in our supervised dataset. The system prompt assigns the solver role and enforces formatting: auxiliary lines must be wrapped in `[AUX] . . . [/AUX]` and the final answer must appear as plain text in `Final Answer: . . .`. The user

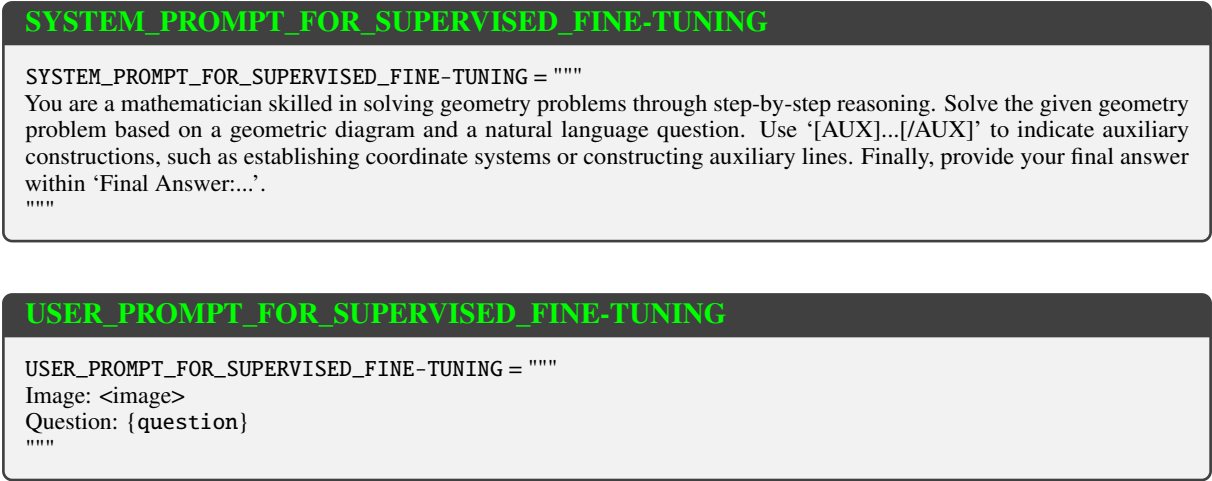


Figure 5: Prompts for supervised fine-tuning.

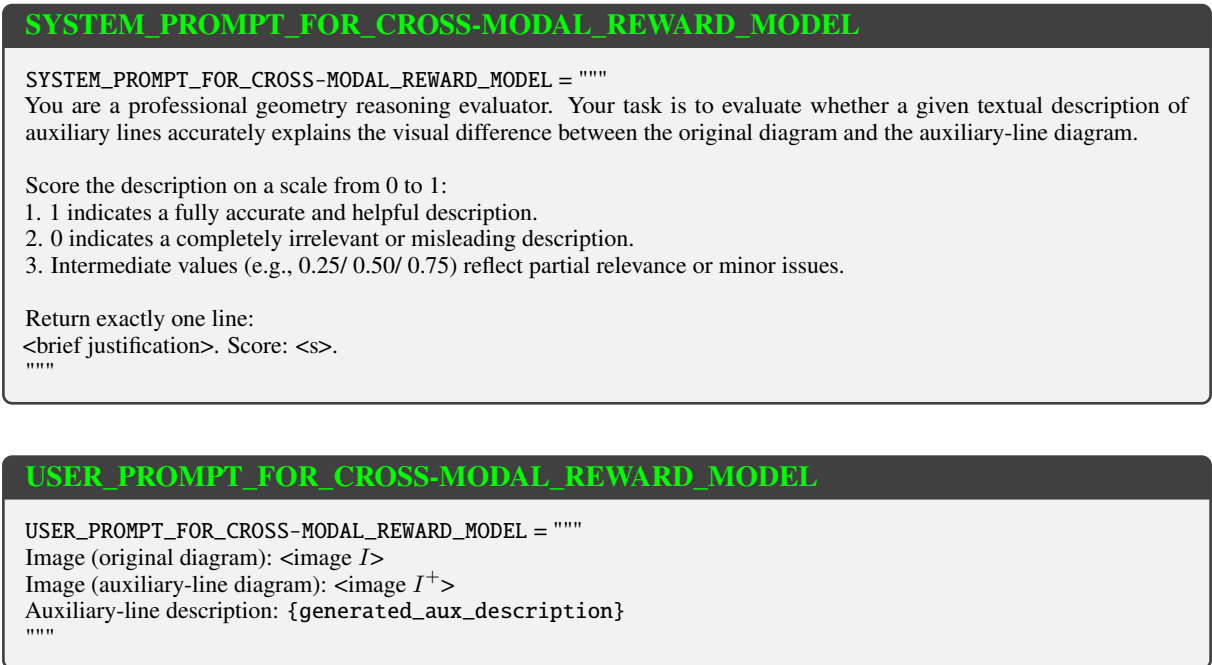


Figure 6: Prompts for cross-modal reward model.

1135 prompt is multimodal, pairing a diagram referenced  
 1136 by the <image> token with the natural language  
 1137 question {question}, which yields explicit reason-  
 1138 ing steps and a final answer.

1139 **E.2 Prompts for Cross-Modal Reward Model**

1140 Using the prompt as shown in Figure 6, the cross-  
 1141 modal reward model compares the description of  
 1142 the auxiliary line generated by the policy model  
 1143 against a pair of diagrams, the original image  $I$   
 1144 and its auxiliary-line counterpart  $I^+$ , and returns  
 1145 a single line justification and a calibrated score in  
 1146  $[0, 1]$  that measures visual–textual agreement. The

1147 instruction emphasizes the correctness of auxiliary-  
 1148 line constructions and adherence to geometric con-  
 1149 straints. Higher scores indicate stronger alignment.

**F Representative Examples** 1150

1151 As illustrad in Figure 7 present qualitative ex-  
 1152 amples from the AuxSolidMath dataset, includ-  
 1153 ing the question, the final answer, the auxiliary-  
 1154 line description, the original diagram, and the  
 1155 auxiliary-line diagram. The examples showcase  
 1156 diverse strategies for constructing auxiliary lines  
 1157 and demonstrate that explicit annotations reveal the  
 1158 key spatial constraints.

## 1159 **G Case Study**

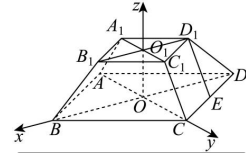
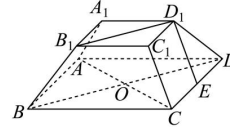
### 1160 **G.1 Successful Cases**

1161 In this section, we present representative cases pro-  
1162 duced by GeoVLMath-7B (see Figure 8). The ex-  
1163 amples illustrate how valid auxiliary-line construc-  
1164 tions capture essential spatial constraints, which  
1165 in turn lead to correct final answers. Collectively,  
1166 these results provide concrete evidence of both the  
1167 effectiveness of our training framework and the  
1168 model’s capacity to tackle complex geometric rea-  
1169 soning requiring robust visual understanding and  
1170 precise symbolic inference.

### 1171 **G.2 Failure Cases**

1172 In this section, we illustrate typical failure cases  
1173 generated by GeoVLMath-7B (see Figure 9). The  
1174 observed failures are exemplified by mis-specified  
1175 coordinate systems and auxiliary-line descriptions  
1176 that are irrelevant to the diagram, both of which  
1177 fail to capture essential spatial constraints and may  
1178 yield incorrect final answers. To mitigate such  
1179 errors, we will explore diffusion-based drawing  
1180 modules that render auxiliary lines directly on the  
1181 original diagram and support iterative correction as  
1182 part of future work.

**Question:** As shown in the diagram, consider the regular truncated quadrilateral pyramid  $ABCD - A_1 B_1 C_1 D_1$ . Given that  $\angle B_1 BA = 60^\circ$ ,  $AB = 2A_1 B_1 = 4$ , and  $E$  is the midpoint of segment  $CD$ , determine the sine of the angle between line  $ED_1$  and plane  $ABB_1 A_1$ .



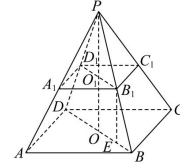
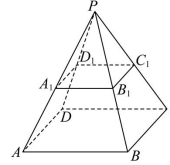
**Final Answer:**  $\sqrt{6}/3$

**Original Diagram**

**Auxiliary-line Diagram**

**Auxiliary-line Description:** Connect  $A_1 C_1$ , which intersects  $B_1 D_1$  at  $O_1$ , then connect  $OO_1$ . Take  $O$  as the origin, and let the lines along  $OB$ ,  $OC$ , and  $OO_1$  be the  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively, then establish the three-dimensional Cartesian coordinate system  $O$ - $xyz$ .

**Question:** As shown in the diagram, consider the regular truncated quadrilateral pyramid  $ABCD - A_1 B_1 C_1 D_1$ . The edge lengths of the upper and lower bases are 2 and 3, respectively, and the lateral edge length is  $\sqrt{2}$ . By extending  $AA_1$ ,  $BB_1$ ,  $CC_1$ ,  $DD_1$ , they intersect at point  $P$ . Determine the volume of the quadrilateral pyramid  $P - A_1 B_1 C_1 D_1$ .



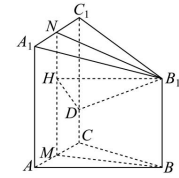
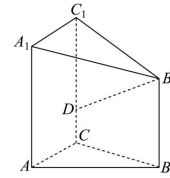
**Final Answer:**  $4\sqrt{6}/3$

**Original Diagram**

**Auxiliary-line Diagram**

**Auxiliary-line Description:** Through point  $P$ , construct  $PO$  perpendicular to the base  $ABCD$  at point  $O$ , intersecting the plane  $A_1 B_1 C_1 D_1$  at point  $O_1$ . Through point  $B_1$ , construct  $B_1 E$  perpendicular to the base  $ABCD$  at point  $E$ .

**Question:** Given the geometric solid shown in the diagram, the base  $ABC$  is an equilateral triangle with a side length of 4. The lateral face  $AA_1 C_1 C$  is a square, and the plane  $AA_1 C_1 C$  is perpendicular to the plane  $ABC$ . Point  $D$  lies on edge  $CC_1$ , with  $CD = (1/4) * CC_1$ , and  $BB_1 = 3CD$ . Determine the sine of the angle between line  $B_1 D$  and plane  $AA_1 C_1 C$ .



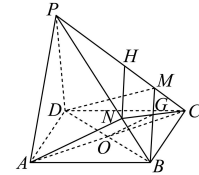
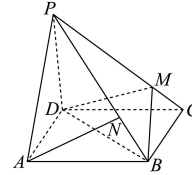
**Final Answer:**  $\sqrt{15}/5$

**Original Diagram**

**Auxiliary-line Diagram**

**Auxiliary-line Description:** Let  $M$  be the midpoint of  $AC$  and  $N$  be the midpoint of  $A_1 C_1$ . Then connect  $MN$ ,  $BM$ , and  $NB_1$ . Through point  $B_1$ , construct  $B_1 H$  perpendicular to  $MN$ , intersecting  $MN$  at point  $H$ , then connect  $DH$ .

**Question:** As shown in the diagram, consider the quadrilateral pyramid  $P - ABCD$ , where the base  $ABCD$  is a parallelogram. Points  $M$  and  $N$  are located on segments  $PC$  and  $PB$ , respectively. Given that  $PM:MC = 3:1$  and  $AN \parallel$  plane  $BDM$ , determine the ratio  $PN:NB$ .



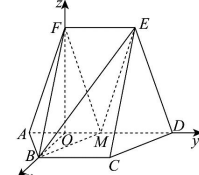
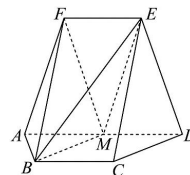
**Final Answer:** 2

**Original Diagram**

**Auxiliary-line Diagram**

**Auxiliary-line Description:** Connect  $AC$ , which intersects  $BD$  at point  $O$ , connect  $CN$ , which intersects  $BM$  at point  $G$ , and then connect  $OG$ . Construct  $NH$  parallel to  $BM$ , intersecting  $PC$  at point  $H$ .

**Question:** As shown in the diagram, in a pentahedron with vertices  $A, B, C, D, E, F$ , quadrilaterals  $ABCD$  and  $ADEF$  are both isosceles trapezoids. It is given that  $EF$  is parallel to  $AD$ ,  $BC$  is parallel to  $AD$ ,  $AD = 4$ ,  $AB = BC = EF = 2$ ,  $ED = \sqrt{10}$ ,  $FB = 2\sqrt{3}$ , and  $M$  is the midpoint of  $AD$ . Find the sine of the dihedral angle  $F$ - $BM$ - $E$ .



**Final Answer:**  $4\sqrt{3}/13$

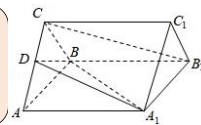
**Original Diagram**

**Auxiliary-line Diagram**

**Auxiliary-line Description:** Draw  $BO$  perpendicular to  $AD$  intersecting  $AD$  at  $O$ , and connect  $OF$ . Take point  $O$  as the origin, and let the lines  $OB$ ,  $OD$ , and  $OF$  be the  $x$ -axis,  $y$ -axis, and  $z$ -axis, respectively, to establish the three-dimensional Cartesian coordinate system  $O$ - $xyz$ .

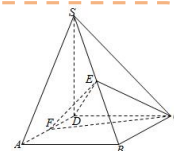
Figure 7: Representative examples from the AuxSolidMath dataset.

**Question:** As shown in the diagram, in the regular triangular prism  $ABC - A_1 B_1 C_1$ , the side length of the base is 2, and the lateral edge length is  $\sqrt{3}$ . D is the midpoint of AC. Find the volume of the triangular pyramid  $D - A_1 C_1 B$ .



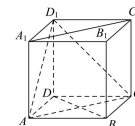
**Solution:** **[AUX]1. Construct auxiliary lines: Take the midpoint E of  $BC_1$ , and connect  $A_1 E$  and  $D_1 E$ .** Determine parallel lines: Since  $ABC - A_1 B_1 C_1$  is a regular triangular prism,  $AB \parallel C_1 D_1$  and  $AB = C_1 D_1$ . So quadrilateral  $ABC_1 D_1$  is a parallelogram. Identify equal lengths: Given the side length of the base is 2 and the lateral edge length is  $\sqrt{3}$ , triangle  $C_1 DC$  is equilateral with side length 2. Calculate the volume: Since  $ABC - A_1 B_1 C_1$  is a regular triangular prism and triangle  $ABC$  is equilateral with side length 2, the height of the pyramid from D to the plane  $A_1 C_1 B$  can be calculated as  $\sqrt{3}$ . The area of the triangular base  $A_1 C_1 B$  is  $(\sqrt{3}/4) * (2)^2 = \sqrt{3}$ . Therefore, the volume of the pyramid is  $(1/3) * \text{base area} * \text{height} = (1/3) * \sqrt{3} * \sqrt{3} = 1$ . **Final Answer: 1.**

**Question:** As shown in the diagram, in the quadrilateral pyramid  $S - ABCD$ ,  $SD$  is perpendicular to the plane  $ABCD$ . The base  $ABCD$  is a rhombus, with E and F being the midpoints of  $SB$  and  $AD$ , respectively. If  $\angle BAD = 60^\circ$ ,  $SD = 4$ , and  $AB = 2$ , find the volume of the triangular pyramid  $C - DEF$ .



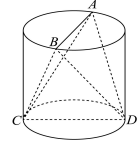
**Solution:** **[AUX]1. Construct auxiliary lines: Take the midpoint H of AB, and connect DH.** Since  $\angle BAD = 60^\circ$  and  $AB = AD = 2$ , triangle  $ABD$  is equilateral. Since H is the midpoint of AB,  $AH = 1$ . By Pythagoras theorem in right triangle  $ADH$ ,  $DH = \sqrt{AD^2 - AH^2} = \sqrt{2^2 - 1^2} = \sqrt{3}$ . Since  $ABCD$  is a rhombus with  $AB = BC = CD = DA = 2$  and  $\angle ADC = 120^\circ$ , we can calculate DC. Using cosine rule in triangle  $ADC$ ,  $DC^2 = AD^2 + AC^2 - 2 * AD * AC * \cos(120^\circ) = 2^2 + (2^2) - 2 * 2 * 2 * \cos(\pi/3) = 4 + 4 - 4 = 1$ . So  $DC = 1$ . Triangle  $CDE$  has vertices: C, D, E. E is the midpoint of SB. Since S is the apex and  $SD \perp$  plane  $ABCD$ , plane  $SAB \perp$  plane  $ABCD$ . Therefore,  $DE \perp$  plane  $ABCD$ . Since CH lies in plane  $ABCD$ , we conclude  $DE \perp CH$ . Since  $AB = BC = CD = DA = 2$ , and H is the midpoint of AB,  $AC \perp BH$ . Similarly, since  $CD \perp DA$  (because  $AD^2 + CD^2 = 2^2 + 1^2 = 5$ ,  $AC^2 = 2^2 - 1^2 = 3$ ) and  $DC = 1$ , we can deduce  $CH \perp CD$ . Thus  $CH \perp$  plane  $CDE$ . The area of triangle  $CDE$  is:  $\text{Area}(CDE) = (1/2) * DC * DE = (1/2) * 1 * 2 = 1$ . The height of the pyramid  $C - DEF$  is CH. We need to find CH. In triangle  $CDB$ ,  $DB = 2\sqrt{2}$  (by Pythagoras theorem). So  $CH = AC \sin(60^\circ) = 2 * (\sqrt{3}/2) = \sqrt{3}$ . The volume of pyramid  $C - DEF$  is:  $\text{Volume}(C - DEF) = (1/3) * \text{Area}(CDE) * CH = (1/3) * 1 * \sqrt{3} = \sqrt{3}/3$ . **Final Answer:  $\sqrt{3}/3$ .**

**Question:** As shown in the diagram, the edge length of the cube  $ABCD - A_1 B_1 C_1 D_1$  is 2. Find the sine of the angle between line  $BD$  and plane  $ACD_1$ .



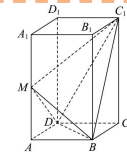
**Solution:** **[AUX]1. Establish coordinate system: Take point D as the origin, and let the lines DA, DC, and  $DD_1$  be the x-axis, y-axis, and z-axis, respectively, to establish the three-dimensional Cartesian coordinate system D-xyz.** Determine coordinates of points: Since the edge length of the cube  $ABCD - A_1 B_1 C_1 D_1$  is 2, we have the coordinates of the vertices as follows:  $D = (0, 0, 0)$ ,  $A = (2, 0, 0)$ ,  $C = (0, 2, 0)$ ,  $B = (2, 2, 0)$ ,  $D_1 = (0, 0, 2)$ ,  $A_1 = (2, 0, 2)$ ,  $C_1 = (0, 2, 2)$ ,  $B_1 = (2, 2, 2)$ . Find the coordinates of point E: E is the midpoint of BD. The coordinates of B are (2, 2, 0) and D are (0, 0, 0). The midpoint E is the average of the coordinates of B and D:  $E = ((2+0)/2, (2+0)/2, (0+0)/2) = (1, 1, 0)$ . Find the vector BD: We have point B = (2, 2, 0) and point D = (0, 0, 0). Vector  $BD = D - B = (0 - 2, 0 - 2, 0 - 0) = (-2, -2, 0)$ . Find a normal vector to the plane  $ACD_1$ : To find the normal vector, we need two vectors in the plane  $ACD_1$ . Vector  $AC = C - A = (0 - 2, 2 - 0, 0 - 0) = (-2, 2, 0)$ . Vector  $AD_1 = D_1 - A = (0 - 2, 0 - 0, 2 - 0) = (-2, 0, 2)$ . Normal vector n to the plane  $ACD_1$  is the cross product of AC and  $AD_1$ :  $n = AC \times AD_1 = (-2, 2, 0) \times (-2, 0, 2) = (2 * 2 - 0 * 0, 0 * (-2) - (-2) * 2, (-2) * 0 - 2 * (-2)) = (4, 4, 4)$ . We can take a simpler normal vector by dividing by 4, so we can take  $n = (1, 1, 1)$ . Calculate the sine of the angle between line BD and plane  $ACD_1$ : The sine of the angle  $\theta$  between line BD and plane  $ACD_1$  is given by the absolute value of the cosine of the angle between vector BD and normal vector n.  $\sin(\theta) = |\cos(\text{angle between } BD \text{ and } n)| = |(BD \cdot n) / (|BD| * |n|)|$ .  $BD \cdot n = (-2, -2, 0) \cdot (1, 1, 1) = (-2)(1) + (-2)(1) + (0)(1) = -2 - 2 + 0 = -4$ .  $|BD| = \sqrt{((-2)^2 + (-2)^2 + 0^2)} = \sqrt{4 + 4} = \sqrt{8} = 2\sqrt{2}$ .  $|n| = \sqrt{(1^2 + 1^2 + 1^2)} = \sqrt{1 + 1 + 1} = \sqrt{3}$ .  $\sin(\theta) = |-4 / (2\sqrt{2} * \sqrt{3})| = |-4 / (2\sqrt{6})| = |-2/\sqrt{6}| = 2/\sqrt{6} = 2\sqrt{6}/6 = \sqrt{6}/3$ . **Final Answer:  $\sqrt{6}/3$ .**

**Question:** As shown in the diagram, the diameter of the base of the cylinder is 3, and the slant height is 4. AB and CD are the diameters of the upper and lower bases of the cylinder, respectively, and  $AB \perp CD$ . Find the volume of the triangular pyramid A-BCD.



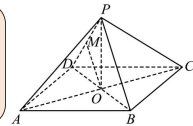
**Solution:** [AUX]1. Construct auxiliary lines: Take the midpoint M of AB, and connect MD and MC. Determine the base area of the triangular pyramid: Since  $AB \perp CD$  and  $AB = CD = 3$  (diameter), the area of the triangular base ABCD is given by:  $\text{Area} = (1/2) * AB * CD = (1/2) * 3 * 3 = 4.5$ . Determine the height of the pyramid: The height of the triangular pyramid A-BCD is the perpendicular distance from A to the plane BDC. Given that  $AB \perp$  plane BDC, the height is AB itself, which is given as 4 (slant height). Calculate the volume of the triangular pyramid A-BCD:  $\text{Volume} = (1/3) * \text{Area of base} * \text{Height} = (1/3) * 4.5 * 4 = 6$ . **Final Answer: 6.**

**Question:** As shown in the diagram, in the regular quadrilateral prism ABCD-A<sub>1</sub>B<sub>1</sub>C<sub>1</sub>D<sub>1</sub>, M is the midpoint of AA<sub>1</sub>, AB = 2, AA<sub>1</sub> = 4. Find the cosine value of the dihedral angle C<sub>1</sub>-BD-M.



**Solution:** [AUX]1. Establish coordinate system: Take point D as the origin, and let the lines DA, DC, and DD<sub>1</sub> be the x-axis, y-axis, and z-axis, respectively, to establish the three-dimensional Cartesian coordinate system D-xyz. Determine coordinates of points: Since ABCD-A<sub>1</sub>B<sub>1</sub>C<sub>1</sub>D<sub>1</sub> is a regular quadrilateral prism with base side length AB = 2 and height AA<sub>1</sub> = 4, we have the coordinates of the vertices as follows: D = (0, 0, 0), A = (2, 0, 0), C = (0, 2, 0), B = (2, 2, 0), D<sub>1</sub> = (0, 0, 4), C<sub>1</sub> = (0, 2, 4), A<sub>1</sub> = (2, 0, 4), M is the midpoint of AA<sub>1</sub>, so M = ((2+2)/2, (0+0)/2, (0+4)/2) = (2, 0, 2). Find normal vectors of planes C<sub>1</sub>BD and MBD: For plane C<sub>1</sub>BD, we need two vectors in the plane: C<sub>1</sub>D and C<sub>1</sub>B. Vector C<sub>1</sub>D = D - C<sub>1</sub> = (0, 0, 0) - (0, 2, 4) = (0, -2, -4). Vector C<sub>1</sub>B = B - C<sub>1</sub> = (2, 2, 0) - (0, 2, 4) = (2, 0, -4). The normal vector n<sub>1</sub> of plane C<sub>1</sub>BD is the cross product of C<sub>1</sub>D and C<sub>1</sub>B: n<sub>1</sub> = C<sub>1</sub>D × C<sub>1</sub>B = (0, -2, -4) × (2, 0, -4) = ((-2)(-4) - (-4)(0), (-4)(2) - 0(-4), 0(0) - (-2)(2)) = (8, -8, 4). We can take a simpler normal vector by dividing by 4, so n<sub>1</sub>' = (2, -2, 1). For plane MBD, we need two vectors in the plane: MB and MD. Vector MB = B - M = (2, 2, 0) - (2, 0, 2) = (0, 2, -2). Vector MD = D - M = (0, 0, 0) - (2, 0, 2) = (-2, 0, -2). The normal vector n<sub>2</sub> of plane MBD is the cross product of MB and MD: n<sub>2</sub> = MB × MD = (0, 2, -2) × (-2, 0, -2) = (2(-2) - (-2)(0), (-2)(-2) - 0(-2), 0(0) - 2(-2)) = (-4, 4, 4). We can take a simpler normal vector by dividing by 4, so n<sub>2</sub>' = (-1, 1, 1). Calculate the cosine of the dihedral angle: The cosine of the dihedral angle θ is the cosine of the angle between the normal vectors n<sub>1</sub>' and n<sub>2</sub>':  $\cos(\theta) = |n_1' \cdot n_2'| / (|n_1'| * |n_2'|)$ . n<sub>1</sub>' · n<sub>2</sub>' = (2, -2, 1) · (-1, 1, 1) = 2(-1) + (-2)(1) + 1(1) = -2 - 2 + 1 = -3. |n<sub>1</sub>'| = √(2<sup>2</sup> + (-2)<sup>2</sup> + 1<sup>2</sup>) = √(4 + 4 + 1) = √9 = 3. |n<sub>2</sub>'| = √((-1)<sup>2</sup> + 1<sup>2</sup> + 1<sup>2</sup>) = √(1 + 1 + 1) = √3.  $\cos(\theta) = |-3| / (3 * \sqrt{3}) = |-3| / (3\sqrt{3}) = |-1/\sqrt{3}| = 1/\sqrt{3} = \sqrt{3}/3$ . **Final Answer: √3/3.**

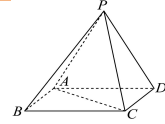
**Question:** In the quadrilateral pyramid P-ABCD, the quadrilateral ABCD is a rhombus, with AC and BD intersecting at point O. PA = PC, PB = PD, ∠BAD = 60°, AB = 2, and M is the midpoint of segment PD. If the angle between line OM and plane ABCD is 60°, find the sine of the dihedral angle between plane PAD and plane PBC.



**Solution:** [AUX]1. Establish coordinate system: Take point O as the origin, and let the lines OA, OB, and OP be the x-axis, y-axis, and z-axis, respectively, to establish the three-dimensional Cartesian coordinate system O-xyz. Determine coordinates of points: Since the rhombus ABCD has side length 2 and ∠BAD = 60°, triangle ABD is equilateral, and triangle BCD is also equilateral. The diagonals AC and BD are perpendicular bisectors of each other. Given AB = 2, we have AO = OC = 1 and BO = OD = √(2<sup>2</sup> - 1<sup>2</sup>) = √3. Thus, the coordinates are: O = (0, 0, 0), A = (1, 0, 0), C = (-1, 0, 0), B = (0, √3, 0). Since PA = PC and PA = √((1-0)<sup>2</sup> + (0-h)<sup>2</sup> + k<sup>2</sup>), PC = √((-1-0)<sup>2</sup> + (0-h)<sup>2</sup> + k<sup>2</sup>). Squaring both PA and PC and equating them: (1-h)<sup>2</sup> + k<sup>2</sup> = (-1-h)<sup>2</sup> + k<sup>2</sup>. h = 0. Substituting h = 0 into PA<sup>2</sup> = (1-0)<sup>2</sup> + (0-0)<sup>2</sup> + k<sup>2</sup> = 1 + k<sup>2</sup> = 1<sup>2</sup> = 1 + (h<sup>2</sup> = 0) = PB<sup>2</sup>. So z coordinate of P = OM = sin(60°) = √3/2. M is the midpoint of PD, M = ((0+0)/2, (0+0)/2, √3/2 + √3/2)/2 = (0, 0, √3/2). B = (0, √3, 0), P = (0, 0, √3), A = (1, 0, 0), C = (-1, 0, 0), D = (0, -√3, 0). Find normal vectors of plane PAD and plane PBC: For plane PAD, we need vectors PA and PD. PA = A - P = (1, 0, -√3), PD = D - P = (0, -√3, -√3). Normal vector n<sub>1</sub> of plane PAD is the cross product of PA and PD: n<sub>1</sub> = PA × PD = (1, 0, -√3) × (0, -√3, -√3) = (-3, √3, -√3). For plane PBC, we need vectors PB and PC. PB = B - P = (0, √3, -√3), PC = C - P = (-1, 0, -√3). Normal vector n<sub>2</sub> of plane PBC is the cross product of PB and PC: n<sub>2</sub> = PB × PC = (0, √3, -√3) × (-1, 0, -√3) = (-3, √3, √3). Calculate the cosine of the dihedral angle:  $\cos(\theta) = |n_1 \cdot n_2| / (|n_1| * |n_2|)$ . n<sub>1</sub> · n<sub>2</sub> = (-3)(-3) + (√3)(√3) + (-√3)(√3) = 9. |n<sub>1</sub>| = √((-3)<sup>2</sup> + (√3)<sup>2</sup> + (-√3)<sup>2</sup>) = √15. |n<sub>2</sub>| = √((-3)<sup>2</sup> + (√3)<sup>2</sup> + (√3)<sup>2</sup>) = √15.  $\cos(\theta) = |9| / (\sqrt{15} * \sqrt{15}) = 9 / 15 = 3/5$ . Calculate the sine of the dihedral angle:  $\sin^2(\theta) = 1 - \cos^2(\theta) = 16/25$ .  $\sin(\theta) = \sqrt{16/25} = 4/5$ . **Final Answer: 4/5.**

Figure 8: Successful cases generated by GeoVLMath-7B.

**Question:** As shown in the diagram, in the quadrilateral pyramid P-ABCD, the base ABCD is a square, PAD is an equilateral triangle, and PC = AC. Find the sine of the angle between line PB and plane PCD.

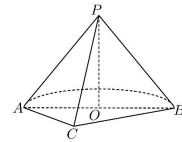


**Ground Truth:**  $\sqrt{6}/4$

**Analysis:** The A-xyz coordinate system is incorrectly established—**AP is not orthogonal to AB or AD**, violating the pairwise orthogonality required for a three-dimensional Cartesian system. Consequently, the coordinates assigned to points and lines are invalid, leading to an incorrect final answer.

**Solution:** [AUX]1. Establish coordinate system: Take point A as the origin, and let the lines AB, AD, and AP be the x-axis, y-axis, and z-axis, respectively, to establish the three-dimensional Cartesian coordinate system A-xyz. [AUX]2. Determine the coordinates of the vertices: Let the side length of the square ABCD be 2. Since A is the origin,  $A = (0, 0, 0)$ . As AB is along the x-axis and B is to the right of A with length 2,  $B = (2, 0, 0)$ . As AD is along the y-axis and D is above A with length 2,  $D = (0, 2, 0)$ . Since ABCD is a square,  $C = (2, 2, 0)$ . As PA is a line from A in the z-direction upwards and triangle PAD is an equilateral triangle with side length 2, the height from P to the base ABCD must be  $\sqrt{2^2 - 1^2} = \sqrt{3}$ . So,  $P = (0, 0, \sqrt{3})$ . Find the vector PB: We have  $P = (0, 0, \sqrt{3})$  and  $B = (2, 0, 0)$ . The vector  $PB = B - P = (2, 0, 0) - (0, 0, \sqrt{3}) = (2, 0, -\sqrt{3})$ . Find the normal vector of the plane PCD: We need two vectors in the plane PCD, for example, vector PC and vector PD. Vector  $PC = C - P = (2, 2, 0) - (0, 0, \sqrt{3}) = (2, 2, -\sqrt{3})$ . Vector  $PD = D - P = (0, 2, 0) - (0, 0, \sqrt{3}) = (0, 2, -\sqrt{3})$ . The normal vector  $n$  of the plane PCD is the cross product of PC and PD:  $n = PC \times PD = (2, 2, -\sqrt{3}) \times (0, 2, -\sqrt{3}) = (2 \cdot (-\sqrt{3}) - (-\sqrt{3}) \cdot 2, (-\sqrt{3}) \cdot 0 - 2 \cdot (-\sqrt{3}), 2 \cdot 2 - 2 \cdot 0) = (-2\sqrt{3} + 2\sqrt{3}, 0 + 2\sqrt{3}, 4 - 0) = (0, 2\sqrt{3}, 4)$ . To take a simpler normal vector, we can use  $n' = (0, \sqrt{3}, 2)$ . Calculate the sine of the angle between line PB and plane PCD: The sine of the angle  $\theta$  between line PB and plane PCD is given by the absolute value of the cosine of the angle between vector PB and normal vector  $n'$ :  $\sin(\theta) = |\cos(\text{angle between PB and } n')| = |(PB \cdot n') / (|PB| \cdot |n'|)|$ .  $PB \cdot n' = (2, 0, -\sqrt{3}) \cdot (0, \sqrt{3}, 2) = 2 \cdot 0 + 0 \cdot \sqrt{3} + (-\sqrt{3}) \cdot 2 = -2\sqrt{3}$ .  $|PB| = \sqrt{2^2 + 0^2 + (-\sqrt{3})^2} = \sqrt{4 + 0 + 3} = \sqrt{7}$ .  $|n'| = \sqrt{0^2 + (\sqrt{3})^2 + 2^2} = \sqrt{0 + 3 + 4} = \sqrt{7}$ .  $\sin(\theta) = |-2\sqrt{3} / (\sqrt{7} \cdot \sqrt{7})| = |-2\sqrt{3} / 7| = 2\sqrt{3} / 7$ . **Final Answer:  $2\sqrt{3}/7$ .**

**Question:** As shown in the diagram, the given solid consists of half of a cone and a triangular pyramid. The base of the cone is a circle with center O and radius 1, and the height of the cone satisfies  $PO = 2$ . The triangular pyramid P-ABC has its base ABC as an isosceles right triangle, where AB is the hypotenuse and coincides with the diameter of the cone's base circle. The base ABC lies on the same plane as the cone's base. Determine the angle between line PC and plane ABC.



**Ground Truth:**  $\arctan 2$

**Analysis:** The generated auxiliary-line description **leaves the geometric configuration unchanged**. Consequently, the subsequent reasoning proceeds from a spurious auxiliary premise rather than from added geometric constraints, leading to an incorrect final answer.

**Solution:** [AUX]1. Construct auxiliary lines: Take the midpoint M of AB, and connect OM. [AUX]2. Identify perpendicularity: Since O is the center of the circle,  $OM \perp AB$ . Also,  $PO \perp OM$  because PO is the height of the cone and O is on the base circle. Since  $AB \cap PO = O$ , and AB and PO are in plane PAB, we conclude  $OM \perp$  plane PAB. Determine angle: The angle between line PC and plane ABC is the angle between line PC and its projection onto plane ABC. Since  $OM \perp$  plane PAB, the projection of point P onto plane ABC lies on line OM. The projection of line PC onto plane ABC is segment AC. Calculate angle: To find the angle between line PC and plane ABC, we need to find the sine of the angle. Given  $PO = 2$  and  $OA = 1$ , we have  $AC = \sqrt{PA^2 - AO^2} = \sqrt{2^2 - 1^2} = \sqrt{3}$ . **Final Answer:  $\arcsin(\sqrt{3}/3)$**

Figure 9: Failure cases generated by GeoVLMath-7B.