# Diagram Parsing for Plane and Solid Geometry with a Unified Formal Language

**Anonymous ACL submission**


## Abstract

Multimodal Large Language Models (MLLMs) have achieved remarkable progress but continue to struggle with geometric reasoning, primarily due to the perception bottleneck regarding fine-grained visual elements. While formal languages have aided plane geometry understanding, solid geometry which requires spatial understanding remains largely unexplored. In this paper, we address this challenge by designing a unified formal language that integrates plane and solid geometry, comprehensively covering geometric structures and semantic relations. We construct GDP-29K, a large-scale dataset comprising 20k plane and 9k solid geometry samples collected from diverse real-world sources, each paired with its ground-truth formal description. To ensure syntactic correctness and geometric consistency, we propose a training paradigm that combines Supervised Fine-Tuning with Reinforcement Learning via Verifiable Rewards. Experiments show that our approach achieves state-of-the-art parsing performance. Furthermore, we demonstrate that our parsed formal descriptions serve as a critical cognitive scaffold, significantly boosting MLLMs' capabilities for downstream geometry reasoning tasks. Our data and code will be released.


## 1 Introduction

Geometry plays a crucial role in mathematics and is widely considered its core (Petersen, 2006); it has always been a subject of great interest in the field of artificial intelligence (Trinh et al., 2024; Zhang et al., 2024a). The challenge of geometric problem solving lies in the integration of complex visual information and symbolic reasoning. Based on the structural properties of geometry diagram, geometry can be categorized into plane geometry and solid geometry (Arana and Mancosu, 2012). Compared to plane geometry, solid geometry demands understanding 3D structures and spatial relationships, making it more complex and a highly challenging problem for artificial intelligence systems (Chou et al., 1996; Wang et al., 2024).

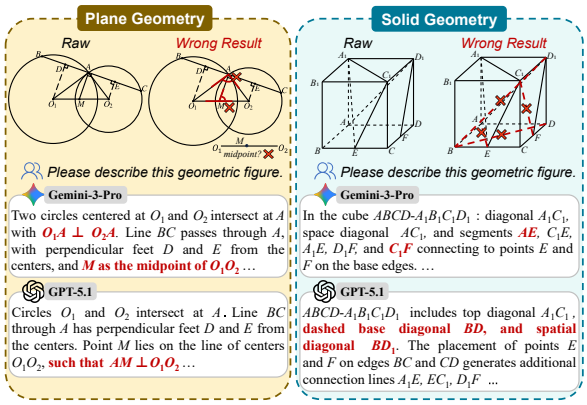



Figure 1: Hallucinations in geometric parsing by SOTA MLLMs. Gemini-3-Pro and GPT-5.1 struggle to correctly parse slightly complex plane geometry and simple solid geometry. Red text indicates parsing errors.

Recent advancements in Multimodal Large Language Models (MLLMs) have demonstrated remarkable capabilities across various vision reasoning tasks (Li et al.; Wang et al., 2025d; Bai et al., 2025). However, Geometry Problem Solving (GPS) remains a challenge (Ma et al., 2025; Zhao et al., 2025). The core difficulty stems from the strict demand for precise geometric perception: MLLMs must accurately identify basic geometric primitives (e.g., points, lines, and planes) and comprehend their relations. Yet, even state-of-the-art (SOTA) models frequently misinterpret geometric figures (Wang et al., 2025b; Li et al., 2024). As shown in Figure 1, the most advanced models, Gemini-3-Pro (Google, 2025) and GPT-5.1 (OpenAI, 2025), still struggle to correctly parse complex plane and solid geometries. This deficiency in fine-grained visual perception is a critical bottleneck, constraining the subsequent reasoning process.

To address the perception challenge, recent research has explored geometry diagram parsing (GDP), aiming to convert geometric diagrams into symbolic representations (Seo et al., 2014; Lu et al.,

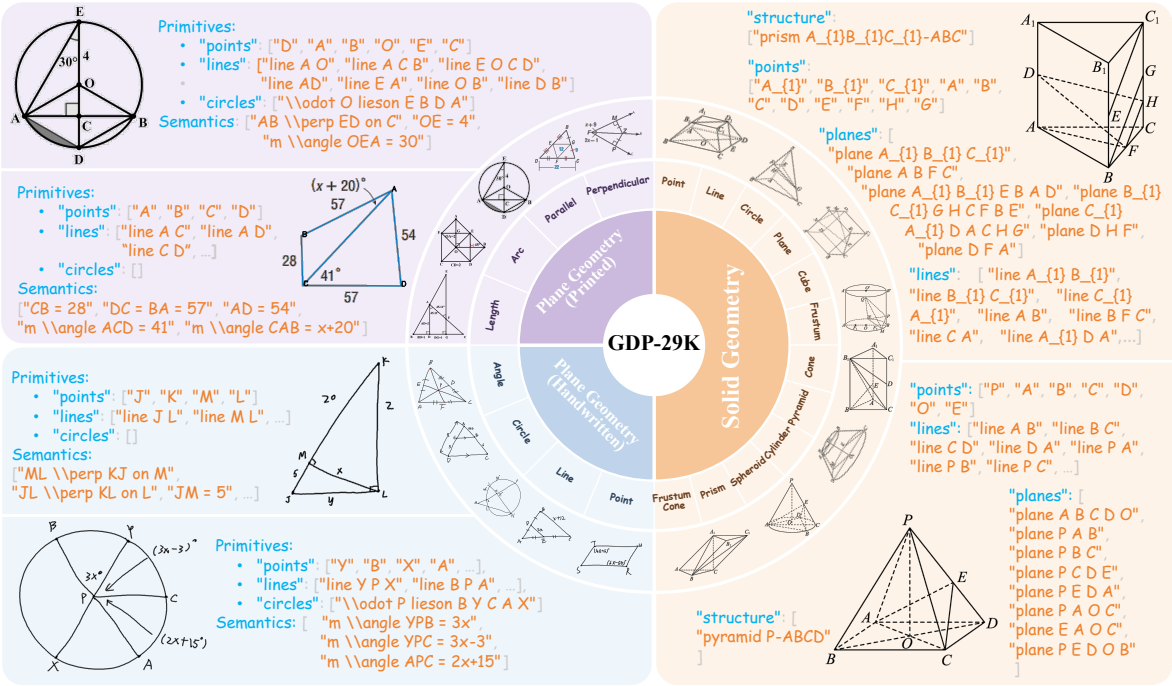



Figure 2: Overview of the GDP-29K dataset for geometry diagram parsing. The dataset spans plane geometry (printed and handwritten) and solid geometry, with each diagram annotated by formal language that captures primitives and semantic constraints.

2021). However, existing works predominantly focus on plane geometry (PGDP), introducing formal languages and datasets like PGDP5K (Zhang et al., 2022) and FormalGeo7K (Zhang et al., 2023b), while solid geometry remains underexplored. Unlike PGDP, solid geometry diagram parsing (SGDP) necessitates understanding of 3D spatial structures, making this task more complex (Wang et al., 2025b). To bridge this gap, we propose a unified formal language that extends established plane geometry formal representations to solid geometry. The formal language covers elements ranging from basic points and lines to high-order structures like planes and solids. Leveraging this language, we construct **GDP-29K**, a large-scale dataset sourced from diverse real-world scenarios. It comprises a plane geometry subset **PGDP-20K** and a solid geometry subset **SGDP-9K**, with each image paired with its ground-truth formal description. Notably, the dataset incorporates varied visual styles, including handwritten diagrams, significantly enriching data diversity. GDP-29K not only expands the scale of plane geometry resources but also fills the void in formal definitions and benchmarks for solid geometry.

Leveraging GDP-29K, we employ a two-stage training paradigm that integrates Supervised Fine-Tuning (SFT) with Reinforcement Learning via Verifiable Rewards (RLVR). To ensure the rigor of the generated formal descriptions, we design a rule-based verifier that guides the policy based on syntactic correctness and geometric consistency. Consequently, our model demonstrates superior parsing capabilities, with scores of 96.4 on PGDP and 94.9 on SGDP benchmark, even surpassing GPT-5.2 and Gemini-3-Flash. Furthermore, we demonstrate the practical utility of our parser in downstream geometry reasoning. Experimental results show that augmenting Qwen3-VL-8B with our parsing outputs drives significant performance boosts, yielding improvements of +10.1% on Geometry3K (Lu et al., 2021), +9.0% on PGPS9K (Zhang et al., 2023a), and +3.1% on SolidGeo (Wang et al., 2025b), with gains also verified across other representative models. In summary, Our contributions are as follows:

- We propose a unified formal language for GDP task, which extends existing plane geometry representations to cover solid geometry structure.
- We construct **GDP-29K**, a large-scale dataset comprising 20K plane and 9K solid geometry diagrams paired with formal descriptions across both printed and handwritten styles, effectively filling the critical data gap for the GDP task.
- We introduce a robust training paradigm combining SFT and RLVR, which ensures syntactic and geometric validity while achieving SOTA

performance on GDP benchmarks.

- Experimental results demonstrate that our parsing outputs significantly enhance downstream multimodal geometric reasoning.

## 2 Related Work

**Geometry Perception Limitations in MLLMs.** Recent MLLMs have demonstrated strong capabilities on several mathematical reasoning benchmarks, such as MathVista (Lu et al.), MathVision (Wang et al., 2024), and We-Math (Qiao et al., 2025). However, performance remains unsatisfactory in GPS (Xu et al., 2025; Wang et al., 2025b). Solving geometry problems requires the model to accurately identify fundamental primitives such as points, lines, circles, and planes; failure to perceive these elements correctly inevitably leads to reasoning errors. Most existing works on GPS rely on end-to-end benchmarks like GeoEval (Zhang et al., 2024a) and MathVerse (Zhang et al., 2024b). However, this approach tends to conflate perception errors with reasoning failures, obscuring the true source of model limitations. In fact, several studies have identified that perception errors remain the primary source of failure in geometric reasoning tasks (Wang et al., 2025a,b). Thus, explicitly decoupling perception from reasoning is imperative.

**GDP Datasets and Formalization.** GDP translates geometric diagrams into formal languages to decouple perception from reasoning. While PGDP benchmarks like Geometry3K (Lu et al., 2021), PGDP5K (Zhang et al., 2022), and FormalGeo7k (Zhang et al., 2023b) have established 2D formalisms, their reliance on limited sources (e.g., Geometry3K and GeoQA (Chen et al., 2021)) restricts visual and structural diversity. This lack of variety potentially hinders the generalization of parsing models across complex, real-world scenarios. Critically, a significant gap remains in solid geometry, which involves complex 3D structures and spatial relationships (Wang et al., 2025b) unaddressed by current formalisms and datasets. To bridge this, we design a formal language for solid geometry that is fully compatible with 2D representations, and introduce GDP-29K—a large-scale dataset comprising 9K solid and 20K plane geometry samples. This resource fills the long-standing void in solid geometry parsing while significantly enhancing the diversity and scale of plane geometry benchmarks.

**Approaches for Geometry Understanding and Reasoning.** Early approaches relied on rule-based heuristics (Lu et al., 2021) or detection-based pipelines (Zhang et al., 2022) to identify geometric primitives. Recently, works like G-LLaVA (Gao et al.), AutoGeo (Huang et al., 2025), and MAVIS (Zhang et al.) have shifted the focus toward geometry QA, utilizing natural language supervision to enhance reasoning. While GeoX (Xia et al.) validates the feasibility of formal language pre-training, many current methods still struggle with the structural rigor required for precise parsing. Inspired by the success of Reinforcement Learning (RL) in mathematical domains (Shao et al., 2024; Guo et al., 2025a), we introduce RLVR to the GDP task—marking the first application of RL to ensure both syntactic correctness and geometric precision in diagram parsing.

## 3 Geometry Formal Representation

In this section, we introduce our unified formal language representation. Designed for conciseness and compatibility, this framework extends existing definitions to address the critical lack of formalisms for solid geometry.

**Inheritance from Plane Geometry.** For plane geometry, we adopt the formal language established in PGPS9K (Zhang et al., 2023a). This representation is concise and close to natural language, describing geometric diagrams as sequences of predicates. It covers fundamental primitives (e.g., points, lines, circles) and semantic relations, including geometric constraints (e.g., parallelism, perpendicularity) and metric attributes (e.g., lengths, angle measures), effectively capturing the topological structure of plane geometry.

**Extension to Solid Geometry.** To address the lack of formal definitions for three-dimensional structures, we extend the formal language to solid geometry. While preserving the syntactic consistency of the plane geometry language, we introduce high-order primitives such as *planes* and *solids*. To achieve comprehensive coverage, we explicitly categorize solid structures into two classes:

- **Polyhedra:** We design specific descriptors for a wide array of multifaceted bodies, ranging from basic forms like *cubes*, *prisms*, and *pyramids* to more complex structures such as *frustums* and *composite polyhedra*.

| Statistic | Number |
|---|---|
| ***Dataset Scale & Style*** | |
| Total Images | 28,977 |
| - Plane Geometry (PG) | 19,965 |
| - Solid Geometry (SG) | 9,012 |
| Style | |
| - Printed | 23,461 |
| - Handwritten | 5,516 |
| ***PG Statistics (Avg. per Image)*** | |
| Points | 5.9 |
| Lines | 5.0 |
| Circles | 0.3 |
| Semantic Relations | 2.4 |
| ***SG Statistics (Avg. per Image)*** | |
| Points | 7.4 |
| Lines | 12.2 |
| Circles | 0.2 |
| Planes | 6.4 |

Table 1: Detailed statistics of GDP-29K.

- **Solids of Revolution:** We strictly define curved geometric bodies formed by rotating a plane curve around an axis, including *spheres*, *cylinders*, *cones*, and *truncated cones*.

For each category, we establish standardized formal templates to ensure structural consistency across diverse solid configurations. This language is characterized by its modularity and high expressiveness, allowing intricate geometric structures to be decomposed into interpretable primitives. Such a design ensures full compatibility with existing plane geometry datasets while enabling the precise description of complex solid structures.

# 4 GDP-29K Dataset

## 4.1 Overview

Based on the formal language defined in Section 3, we construct **GDP-29K**, a large-scale dataset designed to advance geometric diagram parsing tasks. GDP-29K comprises a total of 28,977 samples collected from diverse real-world scenarios, with each image paired with its ground-truth formal description. The dataset contains two subsets:

- **PGDP-20K:** Containing 19,965 plane geometry diagrams. PGDP-20K incorporates a wide spectrum of visual styles, covering both printed diagrams and handwritten sketches. This variety significantly enriches data diversity, enhancing the robustness of model training.

- **SGDP-9K:** Containing 9,012 solid geometry diagrams. To the best of our knowledge, this constitutes the first large-scale dataset tailored for solid geometry parsing, effectively filling the gap in data resources for 3D geometry perception.

Figure 2 illustrates representative samples from both subsets, highlighting the complexity and diversity of the geometric structures. Detailed statistics of GDP-29K are presented in Table 1.

## 4.2 Dataset Construction

The construction of GDP-29K follows a rigorous pipeline comprising data collection, filtering, and labeling, ensuring both diversity and high quality.

**Data Collection.** We aggregated raw geometric images from a wide range of real-world sources, including open-access textbooks, exam papers, and educational websites[1]. To further enhance diversity, we also curated samples from three existing open-source datasets (Zhang et al., 2022; Duan et al., 2025; Guo et al., 2025b). In this initial phase, we accumulated a raw pool of 68,642 plane geometry images and 28,878 solid geometry images.

**Data Filtering.** To ensure the quality of the collected data, we use a three-stage filtering pipeline:

- Stage 1: Image Quality Filtering. Using OpenCV, we computed sharpness metrics to eliminate samples with low-resolution or blurry images, ensuring the retained diagrams were visually clear.

- Stage 2: Semantic Quality Filtering. Leveraging GPT-5.1's visual understanding, we filtered out images with semantic ambiguity, unsuitable for parsing, or unrecognizable text labels.

- Stage 3: Human Verification. Finally, we conducted a comprehensive manual review of retained images, strictly excluding poorly rendered diagrams or severe artifacts hindering parsing.

After this rigorous filtering process, we obtained a refined set of 22,459 plane geometry samples and 9,541 solid geometry samples.

**Data Labeling.** We adopted different annotation strategies for plane and solid geometry. For the former, we utilized a model-assisted pipeline where GPT-5.1 generated initial formal descriptions, which were subsequently refined by expert annotators to accelerate the process. In contrast, solid geometry requires a purely manual approach from scratch to ensure structural rigor, as current MLLMs still struggle with 3D spatial perception.

[1] https://www.jiaoyanyun.com/

| Benchmarks | Language | PG Size | SG Size | Task | Style | Source | SG category | PGFL | SGFL |
|---|---|---|---|---|---|---|---|---|---|
| GeoQA (Chen et al., 2021) | CN | 4849 | 115 | GQA | P | S | 4 | ✗ | ✗ |
| Geometry3K (Lu et al., 2021) | EN | 3002 | 0 | GQA | P | S | ✗ | ✓ | ✗ |
| PGDP5K (Zhang et al., 2022) | EN | 5000 | 0 | PGDP | P | S | ✗ | ✓ | ✗ |
| Formalgeo7k (Zhang et al., 2024c) | EN | 7000 | 0 | MQA | P | O | ✗ | ✓ | ✗ |
| GeoEval (Zhang et al., 2024a) | EN | 2000 | 272 | GQA | P | O | 3 | ✗ | ✗ |
| MATH-Vision (Wang et al., 2024) | EN | 1122 | 244 | MQA | P | S | 4 | ✗ | ✗ |
| OlympiadBench (He et al., 2024) | EN/CN | 1325 | 1322 | MQA | P | S | 6 | ✗ | ✗ |
| MathVerse (Zhang et al., 2024b) | EN | 1746 | 332 | MQA | P | S O | 4 | ✗ | ✗ |
| MV-MATH (Wang et al., 2025a) | EN | 1175 | 372 | MQA | P | S | 6 | ✗ | ✗ |
| GeoSense (Xu et al., 2025) | EN/CN | 1558 | 231 | GQA | P | S O | 6 | ✗ | ✗ |
| SolidGeo (Wang et al., 2025b) | EN/CN | 0 | 3113 | GQA | P | S O | 9 | ✗ | ✗ |
| GDP-29K(Ours) | EN | 19965 | 9012 | GDP | P H | S O | 9 | ✓ | ✓ |

Table 2: Comparison with existing multimodal math benchmarks. SG: Solid Geometry, PG: Plane Geometry. GQA: Geometry QA, MQA: Math QA.**Style:** P=Printed, H=Handwritten. **Source:** S=Self-sourced, O=Collected from Open-source Dataset. PGFL/SGFL: Plane/Solid Geometry Formal Language.

To guarantee the highest label quality, we implemented a strict three-tier quality control protocol—consisting of *Annotation*, *Verification*, and *Final Acceptance*—ensuring that only samples passing all stages were included in the final dataset. Following the annotation, we performed a final redundancy filtering step by identifying and removing samples with identical formal descriptions. This ensured the structural uniqueness of each instance, resulting in the final 28,977 high-quality samples.

### 4.3 Comparison with Existing Benchmarks

As shown in Table 2, GDP-29K advances geometry diagram parsing in two key dimensions. First, in plane geometry, its 19,965 real-world samples surpass the cumulative scale of major existing benchmarks, such as Geometry3K (Lu et al., 2021) and PGDP5K (Zhang et al., 2022), enabling more robust 2D geometric perception. Second, GDP-29K introduces the first formal representation and dataset for solid geometry, featuring 9,012 samples. This fills a critical void in a domain essential for geometric reasoning that previous works have entirely neglected.

## 5 Methodology

Our goal is to develop a multimodal model $\mathcal{M}_\theta$ that, given a visual geometry diagram $I$ and a parsing instruction $Q$, generates a formally rigorous description sequence $Y$. $Y$ consists of a set of geometry primitives and relations defined in our formal language. Formally, the model predicts:

$$Y = \mathcal{M}_\theta(I, Q) \tag{1}$$

To achieve this, we introduce a training framework comprising two stages: Supervised Fine-Tuning (SFT) for syntax alignment and Reinforcement Learning via Verifiable Rewards (RLVR) for enforcing syntactic rigor and geometric consistency.

### 5.1 Stage 1: Supervised Fine-Tuning

The initial stage aims to teach the base model the fundamental syntax of our formal language and the mapping between visual features and geometry primitives. Given the training dataset $\mathcal{D} = \{(I_i, Q_i, Y_i^{ref})\}_{i=1}^N$, where $Y^{ref}$ denotes the ground-truth formal description, the objective is to maximize the likelihood of generating the correct sequence. The model $\mathcal{M}_\theta$ is fine-tuned by minimizing the cross-entropy loss:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T}\log P_\theta(y_{i,t} \mid y_{i,<t}, I_i, Q_i) \tag{2}$$

where $y_{i,t}$ is the $t$-th token of the sequence $Y_i$. This step uses standard teacher-forcing to ground the model in the correct formal syntax.

### 5.2 Stage 2: RL with Verifiable Rewards

While SFT provides a strong foundation, it optimizes token-level likelihood rather than global structural integrity. Consequently, the model may generate outputs that are syntactically plausible but geometrically invalid. To address this, the second stage refines the policy $\pi_\theta$ using RLVR. The objective is to maximize the expected reward:

$$\mathcal{J}(\theta) = \mathbb{E}_{(I,Q)\sim\mathcal{D}}\left[\mathbb{E}_{Y\sim\pi_\theta(\cdot|I,Q)}\left[R(Y)\right]\right] \tag{3}$$

where $R(Y)$ is a scalar reward provided by our rule-based verifier. We optimize this objective using GRPO (Shao et al., 2024), which stabilizes training by normalizing rewards within sampled groups.

| Model | Points | | | Lines | | | Circles | | | Semantics | | | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | Score |
| ***Traditional Methods*** | | | | | | | | | | | | | |
| InterGPS | 47.3 | 90.0 | 62.2 | 3.6 | 52.8 | 6.7 | 1.0 | 7.7 | 1.8 | 11.4 | 11.8 | 11.6 | 20.6 |
| PGDPNet | 88.8 | 94.1 | 91.4 | 66.8 | 73.4 | 70.1 | 61.6 | 61.2 | 61.4 | 64.1 | 51.5 | 57.1 | 70.6 |
| ***Open-source MLLMs*** | | | | | | | | | | | | | |
| LLaVA-OneVision-1.5-7B | 94.6 | 95.2 | 94.3 | 51.9 | 52.1 | 52.0 | 54.4 | 57.1 | 55.7 | 57.5 | 58.4 | 57.9 | 65.0 |
| InternVL3.5-8B-Instruct | 96.2 | 94.8 | 95.6 | 61.9 | 58.7 | 60.2 | 62.3 | 87.5 | 72.8 | 63.1 | 59.4 | 61.2 | 72.5 |
| Qwen3-VL-4B-Instruct | 95.8 | 97.3 | 96.6 | 53.5 | 71.7 | 61.2 | 75.5 | 92.7 | 83.3 | 60.5 | 57.5 | 58.9 | 75.0 |
| Qwen3-VL-8B-Instruct | 97.1 | 95.5 | 96.3 | 56.4 | 69.8 | 62.4 | 76.8 | 94.3 | 84.6 | 66.2 | 55.4 | 60.1 | 75.9 |
| Qwen3-VL-32B-Instruct | 98.4 | 96.5 | 97.4 | 72.9 | 75.2 | 74.0 | 86.2 | 93.5 | 89.6 | 61.8 | 60.7 | 61.3 | 80.5 |
| Qwen3-VL-235B-Instruct | 99.0 | 96.6 | 97.8 | 84.5 | 80.3 | 82.3 | 85.7 | 93.8 | 89.5 | 65.2 | 69.7 | 67.4 | 84.2 |
| Qwen3-VL-235B-Thinking | 99.0 | 96.6 | 97.8 | 91.7 | 89.1 | 90.4 | 89.7 | 92.3 | 91.0 | 76.4 | 68.8 | 72.4 | 87.9 |
| ***Closed-source MLLMs*** | | | | | | | | | | | | | |
| Claude-4.5-Sonnet | 95.9 | 92.7 | 94.3 | 72.7 | 73.4 | 73.0 | 89.6 | 87.5 | 88.5 | 68.3 | 70.1 | 69.1 | 81.2 |
| GPT-5.2-1211 | 99.0 | 95.3 | 97.1 | 89.5 | 81.5 | 85.3 | 94.5 | 88.7 | 91.5 | 78.6 | 73.8 | 76.1 | 87.5 |
| Gemini-3-Flash | 99.6 | 98.1 | 98.9 | **98.2** | 96.8 | 97.5 | 97.4 | 95.1 | 96.2 | 83.5 | 81.8 | 82.7 | 93.8 |
| ***Ours*** | | | | | | | | | | | | | |
| **GDP-4B-SFT** | **99.7** | **99.6** | **99.6** | 96.8 | 97.7 | 97.3 | 97.4 | 97.3 | 97.4 | 87.8 | 87.3 | 87.5 | 95.1 |
| **GDP-4B-RL** | 99.6 | 99.6 | 99.6 | 98.1 | **97.9** | **98.0** | **98.3** | **98.4** | **98.3** | **91.1** | **90.4** | **90.7** | **96.4** |

Table 3: Performance comparison on the PGDP-2K test benchmark. Models are categorized by type. **P**: Precision, **R**: Recall, **F1**: F1-Score. **Semantics**: Semantic Relations. Overall Score represents the aggregate parsing accuracy. **Bold** denotes the best performance.

**Verification Reward.** We design a rule-based reward function $R(Y)$ to enforce both format compliance and semantic accuracy. The total reward is a weighted sum of two components:

$$R(Y) = \lambda_1 R_{fmt}(Y) + \lambda_2 R_{geo}(Y) \qquad (4)$$

where $\lambda_1$ and $\lambda_2$ are hyperparameters balancing structural completeness and accuracy.

**Format Reward ($R_{fmt}$).** To ensure the output adheres to the required structure, $R_{fmt}$ verifies the presence and correctness of tags (e.g., `<points>`, ..., `<solids>`). It returns a binary signal $\mathbb{I}(Y \in \mathcal{F})$, where $\mathcal{F}$ denotes the set of sequences conforming to the predefined structural format.

**Geometric Validity Reward ($R_{geo}$).** This component evaluates the alignment between the parsed primitives $\mathcal{P}$ and the ground-truth $\mathcal{P}^{\text{ref}}$. Recognizing that the $K$ types of primitives defined in our formal language (e.g., points, lines, planes, and semantic relations) present varying levels of perceptual difficulty, we implement a type-aware weighting strategy. For each primitive type $k \in \{1, \ldots, K\}$, we assign a specific weight $\omega_k$ to reflect its complexity. The reward is computed as a weighted sum of the precision within each type:

$$R_{geo}(Y) = \sum_{k=1}^{K} \omega_k \cdot \frac{|\mathcal{P}_k \cap \mathcal{P}_k^{\text{ref}}|}{|\mathcal{P}_k|} \qquad (5)$$

where $\mathcal{P}_k$ and $\mathcal{P}_k^{\text{ref}}$ denote the predicted and ground-truth subsets belonging to the $k$-th type, respectively. This granular reward structure encourages the model to maintain high fidelity across all geometric elements, especially for challenging high-order relations.

## 6 Experiments

### 6.1 Experimental Setup

**Datasets and Metrics.** We conduct experiments on the GDP-29K dataset. We split it into a training set GDP-26K and a test benchmark GDP-3K. The GDP-3K test set is further divided into a plane-geometry subset PGDP-2K and a solid-geometry subset SGDP-1K. We report Precision (P), Recall (R), and F1-score (F1) for each primitive category by comparing the predicted and ground-truth sets.

**Implementation Details.** We utilize Qwen3-VL-4B-Instruct (Bai et al., 2025) as our base model. For SFT, we perform full-parameter fine-tuning on the GDP-26K training set with a maximum sequence length of 4096. For RLVR, we use the ROLL framework (Wang et al., 2025c) with GRPO (Shao et al., 2024) on a curated subset of 2,000 training samples, using a learning rate of $1 \times 10^{-6}$, group size 8, and global batch size 128.

| Model | Points | | | Lines | | | Circles | | | Planes | | | Solids | Overall |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | Acc | Score |
| ***Open-source MLLMs*** | | | | | | | | | | | | | | |
| LLaVA-OneVision-1.5-7B | 91.6 | 96.9 | 94.2 | 54.8 | 66.5 | 60.8 | 34.5 | 57.4 | 43.1 | 31.2 | 62.3 | 41.5 | 69.6 | 61.9 |
| Qwen3-VL-4B-Instruct | 99.0 | 98.5 | 98.8 | 48.7 | 69.5 | 57.3 | 44.7 | 47.2 | 46.0 | 35.0 | 73.9 | 47.5 | 70.8 | 64.0 |
| InternVL3.5-8B-Instruct | 98.5 | 94.8 | 96.6 | 57.8 | 50.4 | 53.8 | 23.3 | 86.1 | 36.7 | 62.4 | 72.6 | 67.1 | 69.7 | 64.8 |
| Qwen3-VL-8B-Instruct | 98.9 | 99.1 | 99.0 | 42.3 | 65.4 | 51.4 | 48.9 | 57.6 | 52.9 | 32.0 | 75.1 | 44.8 | 77.5 | 65.1 |
| Qwen3-VL-32B-Instruct | 99.0 | 99.1 | 99.0 | 63.6 | 77.9 | 70.0 | 42.9 | 91.1 | 58.4 | 27.5 | 82.1 | 41.2 | 83.8 | 70.4 |
| Qwen3-VL-235B-Thinking | 99.2 | 95.2 | 97.1 | 86.7 | 75.3 | 80.6 | 33.3 | 77.2 | 46.6 | 94.5 | 45.1 | 61.0 | 83.2 | 73.7 |
| Qwen3-VL-235B-Instruct | 98.7 | 93.9 | 96.2 | 75.3 | 73.2 | 74.2 | 51.5 | 83.1 | 63.6 | 72.6 | 75.5 | 74.0 | 83.0 | 78.2 |
| ***Closed-source MLLMs*** | | | | | | | | | | | | | | |
| Claude-4.5-Sonnet | 95.7 | 96.1 | 95.9 | 70.1 | 72.6 | 71.8 | 28.5 | 85.2 | 42.7 | 64.8 | 73.4 | 68.8 | 78.3 | 71.5 |
| GPT-5.2-1211 | 98.9 | 98.5 | 98.7 | 78.2 | 68.1 | 72.8 | 57.0 | 76.2 | 65.3 | 81.2 | 71.3 | 75.9 | 72.7 | 77.1 |
| Gemini-3-Flash | **99.4** | **99.7** | **99.6** | **96.3** | 96.0 | 96.1 | **88.7** | 70.3 | 78.5 | 91.9 | 92.8 | 92.4 | 88.9 | 91.1 |
| ***Ours*** | | | | | | | | | | | | | | |
| **GDP-4B-SFT** | 98.9 | 99.2 | 99.0 | 95.7 | 95.9 | 95.8 | 83.3 | 84.2 | 83.7 | 93.5 | 94.5 | 94.0 | 96.6 | 93.8 |
| **GDP-4B-RL** | 99.2 | 99.3 | 99.2 | 96.1 | **96.8** | **96.4** | 86.7 | **86.2** | **86.5** | **95.8** | **95.2** | **95.5** | **97.0** | **94.9** |

Table 4: Performance comparison on the SGDP-1K test benchmark. We report Precision (**P**), Recall (**R**), and F1-Score (**F1**) for multi-component primitives, and Accuracy (**Acc**) for the overall solid type. **Overall Score** represents the aggregate parsing accuracy. **Bold** denotes the best performance.

### 6.2 Main Results

Table 3 and Table 4 summarize the parsing performance on PGDP-2K and SGDP-1K, respectively. Overall, our GDP-4B models achieve the best performance across both benchmarks, demonstrating that the proposed geometry formal language and training pipeline substantially enhance geometric perception beyond general MLLMs.

**Results on PGDP.** On the plane geometry benchmark, our GDP-4B-RL achieves a SOTA score of 96.4, significantly surpassing large-scale MLLMs. While models like GPT and Gemini-3-Flash perform well on basic primitives (e.g., *Points*), they exhibit noticeable performance drops on *Lines* and *Semantic relations*. For instance, despite its massive scale, Qwen3-VL-235B-Thinking achieves only 72.4 F1 on Semantics, whereas our model attains 90.7. This substantial gap underscores that general visual pre-training is insufficient for capturing explicit geometric logic, a capability effectively unlocked by our specialized formal training.

**Results on SGDP.** The challenge of geometry perception is more evident in solid geometry, where most baselines struggle significantly with *Lines*, *Circles*, and *Planes* compared to their PGDP performance. Due to the strong requirement for spatial understanding, even strong models like GPT-5.2 achieve only 72.8 on *Lines*, 65.3 on *Circles* and 75.9 on *Planes*. In contrast, GDP-4B-RL demonstrates robust spatial understanding, maintaining high precision across all primitives and achieving an overall score of 94.9. These results confirm that our framework successfully bridges the gap in solid geometry parsing, enabling the precise perception where general MLLMs fail.

**Effect of RLVR.** The comparison between GDP-4B-SFT and GDP-4B-RL highlights the critical role of verifiable reinforcement learning. We observe that for fundamental primitives such as points and lines, the performance gains from RLVR are relatively marginal, as the SFT model already achieves near-saturated accuracy in these basic perception tasks. In contrast, RLVR demonstrates its primary strength in refining higher-order structures: it boosts the semantics score by 3.2% on PGDP and the plane F1-score by 1.5% on SGDP. This suggests that the reward signal specifically incentivizes the model to transcend simple visual recognition, effectively resolving complex ambiguities and ensuring overall geometric consistency.

### 6.3 Downstream Geometry Reasoning

Having established the superior accuracy of our parsing model, we investigate its practical utility by using the parsed formal descriptions for downstream geometry reasoning task. Table 5 reports the performance of various MLLMs augmented with our parsing results across both plane and solid geometry benchmarks.

As observed, augmenting MLLMs with our formal parsing yields consistent improvements, par-

| Model | Plane Geometry | | | | Solid Geometry | |
|---|---|---|---|---|---|---|
| | GeoQA | PGPS9K | Geometry3K | MathVerse | SolidGeo | MathVerse |
| Ministral-3-8B | 39.6 | 41.2 | 44.8 | 51.2 | 9.6 | 26.0 |
| **+ Ours** | **41.5** (+1.9) | **47.3** (+6.1) | **53.3** (+8.5) | **52.4** (+1.2) | 8.8 (-0.6) | **26.8** (+0.8) |
| Qwen3-VL-8B | 48.9 | 44.9 | 50.1 | 66.8 | 59.0 | 42.0 |
| **+ Ours** | 48.7 (-0.2) | **53.9** (+9.0) | **60.2** (+10.1) | **68.5** (+1.7) | **62.1** (+3.1) | **44.5** (+2.5) |
| Qwen2.5-VL-32B | 59.7 | 38.1 | 46.3 | 54.9 | 52.5 | 36.1 |
| **+ Ours** | **61.7** (+2.0) | **46.8** (+8.7) | **55.8** (+9.5) | 54.9 (+0.0) | **53.8** (+1.3) | 34.4 (-1.7) |
| Qwen3-VL-32B | 67.8 | 69.4 | 73.0 | 73.8 | 73.7 | 45.3 |
| **+ Ours** | **70.6** (+2.8) | **78.0** (+8.6) | **82.6** (+9.6) | **75.9** (+2.1) | **73.9** (+0.2) | **47.0** (+1.7) |
| GPT-5.2-1211 | 55.3 | 78.1 | 84.5 | 76.3 | 60.5 | 64.7 |
| **+ Ours** | **58.8** (+3.5) | **82.2** (+4.1) | **86.4** (+1.9) | **77.8** (+1.5) | **61.3** (+0.8) | **66.3** (+1.6) |

Table 5: Downstream reasoning accuracy (%) on Plane and Solid geometry benchmarks. We compare vanilla MLLMs with those augmented by our formal parsing (**+ Ours**). Δ (in red) denotes the absolute improvement. MathVerse results are reported on its plane and solid geometry subsets, respectively.

ticularly in plane geometry. On visually complex benchmarks like Geometry3K and PGPS9K, Qwen3-VL-8B achieves substantial gains of +10.1% and +9.0%, respectively, and even the advanced GPT-5.2 sees a solid +4.1% boost. We attribute this to the high visual semantic density of these diagrams, where explicit parsing captures subtle constraints (e.g., parallelism, angles) essential for reasoning. In solid geometry, incorporating parsed primitives yields moderate yet positive gains. This narrower margin likely stems from two primary factors: (i) **Textual Explicitness**, where current solid geometry benchmarks often feature problem statements that already explicitly describe the geometric structure, leaving less "new" information for the parser to provide; and (ii) **Intrinsic Semantic Sparsity**, as solid geometry diagrams tend to contain fewer implicit symbolic constraints compared to their planar counterparts.

### 6.4 Impact of Representation Form

To isolate the impact of the parsed geometry description format on geometry reasoning, we compare our Formal Language (FL) against Natural Language (NL) on the PGPS9K benchmark. To ensure strict semantic equivalence, we employ Gemini-3-Pro to translate our parsed formal sequences into coherent NL descriptions, ensuring the two forms differ only in representation. As illustrated in Figure 3, while both augmentation strategies improve over the vanilla baseline, FL consistently outperforms NL in assisting geometric reasoning across all five evaluated models. This superiority suggests that compact, symbolic representations provide higher information density and a stronger inductive bias for geometric reasoning compared to verbose textual descriptions.

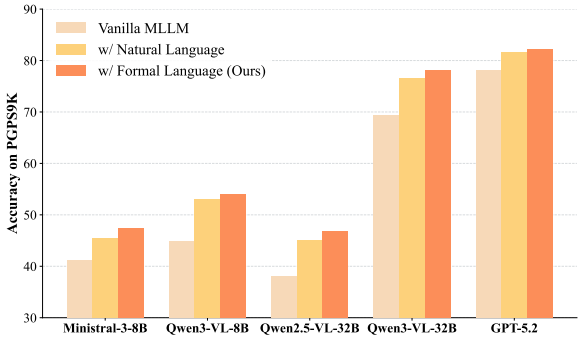



Figure 3: Effect of Representation Forms on PGPS9K Reasoning Accuracy.

## 7 Conclusion

In this work, we address the perception bottleneck in multimodal geometric reasoning by establishing a unified formal language and a parsing framework for both plane and solid geometry. We introduce the GDP-29K dataset, which effectively fills the critical data void in the solid geometry domain and significantly expands image diversity by incorporating both printed and hand-drawn styles. By employing a training paradigm that combines SFT with Reinforcement Learning via Verifiable Rewards, we ensure the syntactic rigor and geometric consistency of the generated formal descriptions. Experimental results demonstrate that our method achieves SOTA parsing performance, and the parsed formal descriptions serve as a vital cognitive scaffold, significantly boosting downstream geometry reasoning capabilities on benchmarks such as Geometry3K, PGPS9K, and MathVerse.

## Limitations

While the GDP-29K dataset and our parsing framework establish a strong baseline, we acknowledge several limitations that guide future research. First, the current formal definitions within GDP-29K do not explicitly distinguish between visible and invisible (e.g., dashed) elements in solid geometry; incorporating explicit visibility attributes could further enhance the depth of solid geometry comprehension and spatial understanding. Second, the visual semantics of our current solid geometry samples are relatively sparse, primarily focusing on basic primitives. Future work aims to construct datasets with richer semantic diversity and more intricate spatial scenarios to further push the boundaries of fine-grained spatial understanding in multimodal models.

## References


Andrew Arana and Paolo Mancosu. 2012. On the relationship between plane and solid geometry. *The Review of Symbolic Logic*, 5(2):294–353.

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

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric Xing, and Liang Lin. 2021. Geoqa: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 513–523.

Shang-Ching Chou, Xiao-Shan Gao, and Jing-Zhong Zhang. 1996. Automated generation of readable proofs with geometric invariants: I. multiple and shortest proof generation. *Journal of Automated Reasoning*, 17(3):325–347.

Chengqi Duan, Kaiyue Sun, Rongyao Fang, Manyuan Zhang, Yan Feng, Ying Luo, Yufang Liu, Ke Wang, Peng Pei, Xunliang Cai, and 1 others. 2025. Codeplot-cot: Mathematical visual reasoning by thinking with code-driven images. *arXiv preprint arXiv:2510.11718*.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing HONG, Jianhua Han, Hang Xu, Zhenguo Li, and 1 others. G-llava: Solving geometric problem with multi-modal large language model. In *The Thirteenth International Conference on Learning Representations*.

Google. 2025. Gemini 3 pro. https://deepmind.google/models/gemini/pro/.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025a. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638.

Shasha Guo, Liang Pang, Xi Wang, Yanling Wang, Huawei Shen, and Jing Zhang. 2025b. Geovlmath: Enhancing geometry reasoning in vision-language models via cross-modal reward for auxiliary line creation. *arXiv preprint arXiv:2510.11020*.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, and 1 others. 2024. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3828–3850.

Zihan Huang, Tao Wu, Wang Lin, Shengyu Zhang, Jingyuan Chen, and Fei Wu. 2025. Autogeo: Automating geometric image dataset creation for enhanced geometry understanding. *IEEE Transactions on Multimedia*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. Llava-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*.

Zhihao Li, Yao Du, Yang Liu, Yan Zhang, Yufang Liu, Mengdi Zhang, and Xunliang Cai. 2024. Eagle: Elevating geometric reasoning through llm-empowered visual instruction tuning. *arXiv preprint arXiv:2408.11397*.

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. In *The Twelfth International Conference on Learning Representations*.

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 *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6774–6786.

Jianzhe Ma, Wenxuan Wang, and Qin Jin. 2025. A survey of deep learning for geometry problem solving. *arXiv preprint arXiv:2507.11936*.

OpenAI. 2025. GPT-5 System Card. Technical report, OpenAI. Version published August 7 2025. Available at: https://openai.com/index/gpt-5-system-card/.

Peter Petersen. 2006. *Riemannian geometry*, volume 171. Springer.

Runqi Qiao, Qiuna Tan, Guanting Dong, MinhuiWu MinhuiWu, Chong Sun, Xiaoshuai Song, Jiapeng Wang, Zhuoma GongQue, Shanglin Lei, Yifan Zhang, and 1 others. 2025. We-math: Does your large multimodal model achieve human-like mathematical reasoning? In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 20023–20070.

Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram understanding in geometry questions. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 28.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

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

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

Peijie Wang, Zhong-Zhi Li, Fei Yin, Dekang Ran, and Cheng-Lin Liu. 2025a. Mv-math: Evaluating multimodal math reasoning in multi-visual contexts. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19541–19551.

Peijie Wang, Chao Yang, Zhong-Zhi Li, Fei Yin, Dekang Ran, Mi Tian, Zhilong Ji, Jinfeng Bai, and Cheng-Lin Liu. 2025b. Solidgeo: Measuring multimodal spatial math reasoning in solid geometry. *arXiv preprint arXiv:2505.21177*.

Weixun Wang, Shaopan Xiong, Gengru Chen, Wei Gao, Sheng Guo, Yancheng He, Ju Huang, Jiaheng Liu, Zhendong Li, Xiaoyang Li, and 1 others. 2025c. Reinforcement learning optimization for large-scale learning: An efficient and user-friendly scaling library. *arXiv preprint arXiv:2506.06122*.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, and 1 others. 2025d. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. 2024. Slow perception: Let's perceive geometric figures step-by-step. *arXiv preprint arXiv:2412.20631*.

Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, and 1 others. Geox: Geometric problem solving through unified formalized vision-language pre-training. In *The Thirteenth International Conference on Learning Representations*.

Liangyu Xu, Yingxiu Zhao, Jingyun Wang, Yingyao Wang, Bu Pi, Chen Wang, Mingliang Zhang, Jihao Gu, Xiang Li, Xiaoyong Zhu, and 1 others. 2025. Geosense: Evaluating identification and application of geometric principles in multimodal reasoning. *arXiv preprint arXiv:2504.12597*.

Jiaxin Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024a. Geoeval: Benchmark for evaluating llms and multi-modal models on geometry problem-solving. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 1258–1276.

Ming-Liang Zhang, Fei Yin, Yi-Han Hao, and Cheng-Lin Liu. 2022. Plane geometry diagram parsing. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 1636–1643. International Joint Conferences on Artificial Intelligence Organization.

Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2023a. A multi-modal neural geometric solver with textual clauses parsed from diagram. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3374–3382.

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

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Shanghang Zhang, Peng Gao, and 1 others. Mavis: Mathematical visual instruction tuning with an automatic data engine. In *The Thirteenth International Conference on Learning Representations*.

Xiaokai Zhang, Na Zhu, Yiming He, Jia Zou, Qike Huang, Xiaoxiao Jin, Yanjun Guo, Chenyang Mao, Yang Li, Zhe Zhu, and 1 others. 2023b. Formalgeo: An extensible formalized framework for olympiad geometric problem solving. *arXiv preprint arXiv:2310.18021*.

Xiaokai Zhang, Na Zhu, Cheng Qin, Yang Li, Zhenbing Zeng, and Tuo Leng. 2024c. Formal representation and solution of plane geometric problems. In *The 4th Workshop on Mathematical Reasoning and AI at NeurIPS'24*.

Yurui Zhao, Xiang Wang, Jiahong Liu, Irwin King, and Zhitao Huang. 2025. Towards geometry problem solving in the large model era: A survey. *arXiv preprint arXiv:2506.02690*.

# A More Details of GDP-29K

In this section, we provide extended details regarding the GDP-29K dataset, including its manual collection process, formal language syntax, and comprehensive statistical analysis.

## A.1 Data Collection

GDP-29K is specifically designed to address the lack of diversity and 3D coverage in existing geometry parsing benchmarks.

**Handwritten Subset.** Our handwritten plane geometry subset is entirely manually drawn. We recruited 10 annotators with diverse handwriting styles to recreate 5,516 geometric diagrams using digital tablets and styluses. This process captures authentic stroke dynamics, varying line thicknesses, and realistic distortions (e.g., imperfect circles and non-straight lines). This high-fidelity data ensures that models trained on GDP-29K possess robust generalization capabilities for real-world educational scenarios, such as grading student sketches.

**Solid Geometry Collection.** The solid geometry samples cover a wide spectrum of 3D structures, including prisms, pyramids, cones, cylinders, and frustums. These diagrams were curated from high-quality geometry textbooks and competitive math examinations. Each diagram was then meticulously annotated with our unified formal language to capture both its topological structure and spatial semantics.

## A.2 Detailed Statistical Analysis of GDP-29K

We performed a comprehensive statistical analysis of the structures and semantic constraints within the GDP-29K dataset to verify its diversity and coverage.

**Structural Diversity.** As illustrated in Figure 4, the SGDP subset (comprising 7,960 analyzed 3D samples) exhibits a rich variety of geometric structures. **Pyramids** constitute the largest portion of the dataset with 3,937 instances (49.46%), reflecting their high frequency in 3D geometry problems. **Cubes** (1,618, 20.33%) and **Prisms** (1,473, 18.51%) follow as the next most prevalent categories. To ensure the model generalizes to complex and curved surfaces, the dataset incorporates **Frustums** (248, 3.12%), **Cones** (156, 1.96%), and **Cylinders** (82, 1.03%), as well as rarer structures like **Spheres** (25, 0.31%) and **Conic Frustums** (11, 0.14%). A small percentage of **Others** (410, 5.15%) includes hybrid or irregular solids. This distribution ensures that our model is exposed to both common polyhedral forms and more challenging rotational solids.

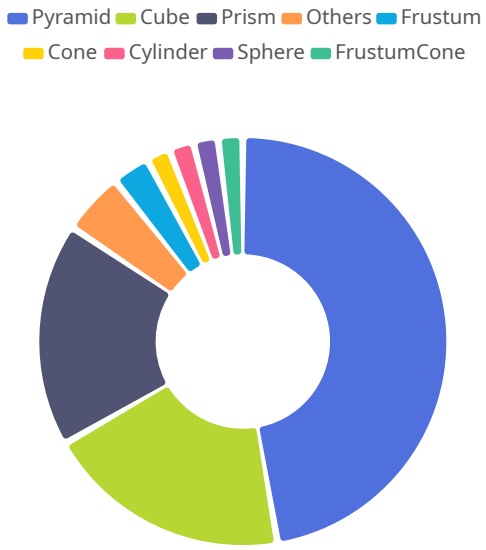



Figure 4: Distribution of 3D structures in the SGDP subset ($N = 9,012$). The dataset covers a wide range of polyhedral forms and rotational solids to facilitate robust spatial perception.

**Semantic Richness.** The distribution of semantic constraints in the PGDP subset (Figure 5) highlights the dataset's focus on rigorous logical relations. Out of 48,613 identified constraints, **Length** measurements (18,247, 37.54%) and **Angle** specifications (16,067, 33.05%) are the most prevalent, providing the metric foundation for geometric reasoning. Notably, **Perpendicularity** (`\perp`) accounts for a significant 12,181 instances (25.06%), emphasizing the importance of topological connectivity and orthogonal relations in theorem proving. Furthermore, **Arc** measures (1,074, 2.21%) and **Parallelism** (1,037, 2.13%) enrich the dataset by ensuring holistic coverage of plane geometry properties.

# B Details of Data Annotation

In alignment with the labeling strategy described in the main text, this section provides further specifics regarding our annotation workforce, the three-tier quality control protocol, and the redundancy filtering process.

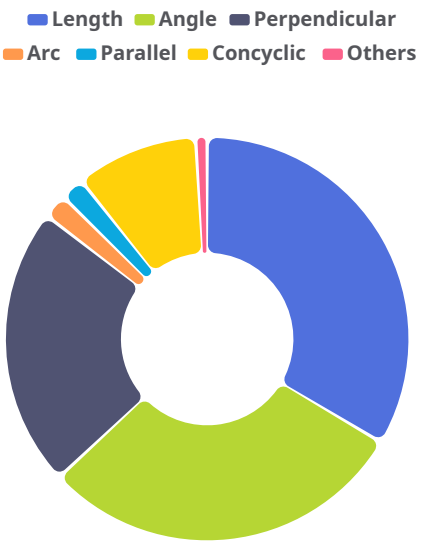



Figure 5: Distribution of semantic predicates across the PGDP subset ($N = 48,613$). Metric constraints and orthogonal relations form the core of the geometric reasoning tasks.

### B.1 Annotation Workforce and Training

We recruited 30 undergraduate students majoring in STEM fields (Science, Technology, Engineering, and Mathematics) to perform the annotation tasks. All participants underwent a standardized training session to familiarize themselves with our formal language's syntax and the 3D spatial relationship definitions. To ensure consistency, each annotator was required to pass a preliminary test consisting of 50 samples before contributing to the final dataset.

### B.2 Three-tier Quality Control Protocol

To maintain a high standard of structural rigor, we implemented a rigorous three-tier workflow:

1. **Annotation Stage:**
   - *Plane Geometry:* Annotators reviewed and corrected initial drafts provided by GPT-5. The primary focus was on fixing vertex ordering and ensuring all geometric constraints (e.g., parallelism) were captured.
   - *Solid Geometry:* Since MLLMs often fail to perceive 3D depth, annotators manually identified all faces, edges, and spatial relations from scratch, following the hierarchical structure of our formal language.
2. **Verification Stage:** A different student from the team acted as a peer reviewer for each annotated sample. They cross-checked the formal description against the original diagram to identify any missing primitives or incorrect semantic tags. Any discrepancies were returned to the original annotator for revision.
3. **Final Acceptance Stage:** Our expert leads (authors of this study) performed a final audit on the verified samples. This stage focused on ensuring the logical consistency of the formal language and the accuracy of complex 3D structures (e.g., non-trivial frustums and spheroids). Only samples with 100% consensus were moved to the final pool.

### B.3 Redundancy Filtering and Final Statistics

After the manual annotation, we performed a structural de-duplication step to enhance dataset diversity. We identified samples with **identical formal descriptions**—defined as instances where all primitives, semantic values, and topological relations were isomorphic—and retained only one representative image per structure.

Following this filtering process, the dataset was finalized at **28,977** high-quality samples. The distribution of these samples ensures that the model learns to generalize across diverse geometric layouts without being biased by repetitive structural patterns.

## C Formal Language Specification

As shown in Table 6, our formal language is characterized by its structural conciseness and a quasi-natural language style, intentionally designed to facilitate more effective understanding and generation by MLLMs. By adopting a syntax that mirrors both standard mathematical notation and intuitive linguistic phrasing (e.g., `AB \perp to CD on X`), we reduce the mapping complexity from visual features to symbolic representations. This alignment leverages the model's pre-trained linguistic knowledge, ensuring that the formalization is not only mathematically rigorous but also highly accessible for model learning and reasoning.

**Key Design Principles.**

- **Topological Precision:** Beyond simple detection, our language explicitly denotes intersection points (e.g., `on X` in perpendicular relations). This provides the model with clear topological anchors, which is crucial for building a consistent geometric graph.

| Category | Formal Syntax (Example) | Geometric Description |
|---|---|---|
| **Primitives** | `point A` | Defines a point vertex named A. |
| | `line A B C` | A line segment passing through points A, B, and C. |
| | `line k lineson A B C` | A line lieson point A B C |
| | `plane A B C D` | A plane defined by vertices A, B, C, and D. |
| | `\odot O lieson A B C` | A circle with center O passing through points A, B, and C. |
| **Semantics** | `AB = 57` | The length of segment $AB$ is 57. |
| | `m \angle ABC = 41` | The measure of $\angle ABC$ is $41^\circ$. |
| | `m \widehat AB = 90` | The angular measure of arc $AB$ is $90^\circ$. |
| **Relations** | `AB \perp to CD on X` | Line $AB$ is perpendicular to $CD$, intersecting at point $X$. |
| | `AB \parallel CD` | Line $AB$ is parallel to line $CD$. |
| **3D Solids** | `solid Cube ABCD-A_{1}B_{1}C_{1}D_{1}` | A cube defined by its bottom and top faces. |
| | `solid Prism ABC-A_{1}B_{1}C_{1}` | A prism defined by its bottom and top faces. |
| | `solid Frustum ABC-A_{1}B_{1}C_{1}` | A frustum defined by its bottom and top bases. |
| | `solid Pyramid O-ABC` | A pyramid defined by apex $O$ and base $ABC$. |
| | `solid Spheriod O-ABCD` | A spheroid defined by center $O$ and surface points $A, B, C, D$. |
| | `solid Cylinder AD-BC` | A cylinder defined by two lateral side segments $AD$ and $BC$. |
| | `solid Cone P-OA` | A cone defined by apex $P$, base center $O$, and base point $A$. |
| | `solid FrustumCone AD-BC` | A conical frustum defined by its lateral side segments. |

Table 6: Detailed syntax and examples of the formal language in GDP-29K, covering 2D primitives and 3D solid structures.

- **Semantic Intuition:** By adopting a quasi-natural language style (e.g., using keywords like `lieson`, `perp to`, and `parallel`), we align the formal syntax with the model's pre-trained linguistic priors. This reduces the cognitive load on the MLLM during the translation from pixels to symbols.
- **Hierarchical Composition:** 3D solids are not treated as isolated entities but are composed of 2D primitives (points, lines, and planes). This design ensures a unified representational space, allowing the model to leverage its 2D parsing experience when tackling complex 3D structures.

## D Hierarchical Prompting Strategy

To accurately bridge the gap between raw geometric images and rigorous formal symbolic language, we propose a hierarchical prompting strategy. We decompose the formalization task into five specialized, decoupled modules: `point_line`, `circle`, `plane`, `solid`, and `semantic`. The detailed design of these prompts is illustrated in Figure 6, 7, 8, 9, and 10.

**Structural Layer: Primitive Extraction.** The first four prompts focus on extracting the "topological skeleton" of the diagram. (1) **`point_line`**: This template identifies all labeled points and their collinearity, enforcing strict ordering to maintain the physical continuity of lines. (2) **`circle`**: It guides the model to locate centers and discrete points on circumferences, ensuring a clear distinction between the boundary and the interior. (3) **`plane`** and (4) **`solid`**: These prompts provide spatial context, where the former handles 2D regional layouts and the latter focuses on 3D volumetric structures, such as identifying hidden edges and face-to-face connectivity in polyhedra.

**Logical Layer: Semantic Constraint Mapping.** (5) **`semantic`**: Building upon the structural skeleton, this template extracts logical relationships. It instructs the model to parse explicit visual markers (e.g., right-angle squares, parallel arrows) into formal clauses (e.g., $\perp$, $\|$). By isolating semantic reasoning from primitive detection, we prevent the model from making unfounded visual assumptions and ensure that every generated clause is grounded in explicit symbolic evidence.

**Capability Elicitation and Fair Comparison.** The core rationale behind this hierarchical decomposition is to maximize the potential of various MLLMs. Geometric formalization is a high-cognitive-load task; by adopting a "divide-and-conquer" approach, we alleviate the instruction-following burden on the models, allowing them to focus on granular sub-tasks. Furthermore, this standardized prompting framework ensures a fair comparison across different model architectures. It eliminates the confounding factor of models' vary-

ing abilities to handle multi-step formatting in a single pass, instead providing a uniform interface to evaluate their true underlying geometric perception capabilities.

**Prompt: Points & Lines**

You are an expert in geometry diagram structure analysis.
Your SOLE task is to identify the Points and Lines from the image.
1. **Points**:
- Identify all labeled points in the diagram.
- Format: $[A, B, C, A_1 \ldots]$
2. **Lines**:
- Identify all lines (Including solid lines and dashed lines).
- A "line" must include **ALL** labeled points lying on that straight segment.
- Do NOT split a line into smaller segments. If $A, B, C$ are collinear, output ONE line line A B C, NOT line A B and line B C.
- Do NOT skip intermediate points. If $B$ is between $A$ and $C$, you MUST write line A B C, NOT line A C.
- Points must be listed in the strict order they appear visually (from one end to the other).
- Format: line P1 P2 P3 P4 P5

**### Output Format Template**
**Points:**
$[list_of_points]$
**Lines:**
line P1 P2 P3 P4 P5
line P6 P7

Figure 6: Prompts for geometric structural analysis. These prompts guide the model to extract the topological skeleton of the diagram.

## E Comparison with Slow Perception

To further verify the superiority of our parsing method, we compare the reasoning performance using our GDP-4B generated results against the Slow Perception method (Wei et al., 2024). We use Qwen3-VL-8B as the base reasoning model to evaluate the accuracy on the Geometry3K and PGPS9K benchmarks.

Table 7: Reasoning accuracy (%) comparison on Qwen3-VL-8B. Our GDP-4B parsing results consistently outperform Slow Perception by providing more concise and structurally coherent geometric descriptions.

| Method | Geometry3K | PGPS9K |
|---|---|---|
| Direct Inference | 50.1 | 44.9 |
| + Slow Perception | 55.2 | 48.5 |
| **+ Ours (GDP-4B)** | **60.2** | **53.9** |

**Prompt: Circles**

You are an expert in geometry diagram structure analysis.
Your SOLE task is to identify **Circles** from the image.

**### Definitions and Rules**
1. **Circles**:
- Identify all circles (or major arcs acting as circles) in the diagram.
- **Structure**: For each circle, you must identify:
a. The **Center** point.
b. **Points on Circumference**: List **ALL** labeled points that lie strictly **ON** the curve.
- **Critical Constraints**:
a. Do not miss any point that on the circle's boundary.
b. Do NOT include points that are inside or outside the circle (except the Center). Only list points on the rim.
- **Format**: \odot Center lieson P1 P2 P3
2. **No Circles**:
- If there are no circles in the diagram, leave the section under "**Circles:**" empty or write "None".

**### Output Format Template**
**Circles:**
\odot O lieson A B C M $A_1$

Figure 7: Prompts for geometric structural analysis. These prompts guide the model to extract the topological skeleton of the diagram.

**Key Observations.** The results in Table 7 show that our **GDP-4B** parsing results achieve significantly higher gains, particularly on PGPS9K ($+9.0\%$). Compared to the more verbose descriptions in Slow Perception, our concise, natural-language-style formalization effectively reduces the cognitive load on the reasoning model and provides a more rigorous symbolic foundation for multi-step deduction.

## F Data Examples

To provide a concrete illustration of the GDP-29K dataset, we present representative examples from both the planar and solid geometry subsets in Figures 11, 12, and 13. These examples demonstrate the capability of our unified formal language to bridge the gap between visual diagrams and symbolic logic.

As shown in the examples, for plane geometry, our formalization accurately captures fundamental primitives such as points, lines, and circles, while simultaneously encoding complex semantic constraints like perpendicularity markers and angle

**Prompt: Semantics**

You are an expert in geometry semantic analysis.
Your task is to extract geometric relationships, equations, and constraints from the image text and symbols.

**### Semantic Clauses Templates**
You must use the following templates.
1. **Perpendicular**:
- Template: AB \perp CD on P
- Note: You MUST include "on P".
2. **Parallel**:
- Template: AB \parallel CD
3. **Angle Measure & Equations**:
- Template: m \angle ABC = 30 or m \angle ABC = 2x + 5
4. **Segment Lengths & Congruence**:
- Template: AB = 5 or AB = CD
5. **Arc Measure**:
- Template: m \widehat AB = 60

**### Constraints & Anti-Redundancy Rules**
1. **Collinear Points Rule**:
- If A-B-C are collinear and perpendicular to D-E, output ONE representative clause (e.g., AC \perp DE on B).
- Do NOT list AB \perp DE, BC \perp DE separately.
2. **Right Angles**:
- If a square symbol is present, use AB \perp CD on B.
- Do NOT output m \angle ABC = 90 if you output the perpendicular clause.
3. **Strict Symbolism (No Visual Assumptions)**:
- **Do NOT** infer relationships based on visual appearance (e.g., lines that "look" parallel or perpendicular).
- **Perpendicularity**: ONLY output \perp if there is an explicit right-angle symbol (square marker) or text declaration.
- **Parallelism**: ONLY output \parallel if there are explicit arrow markers on the lines or text declaration.
- Only output relationships that are **explicitly displayed** in the diagram.

**Output Format Template**
**Semantic Clauses:**
Clause 1
Clause 2
. . .

Figure 8: Prompts for geometric structural analysis. These prompts guide the model to extract the topological skeleton of the diagram.

**Prompt: Solid Structure**

You are an expert in solid geometry diagram parsing.
Your SOLE task is to classify the **3D Structure** of the geometric figure.

**### Allowed Categories (Strictly Choose One)**
You must classify the solid into exactly ONE of the following categories:
1. **Pyramid** (General pyramids, e.g., triangular/rectangular)
2. **Prism** (General prisms, e.g., triangular/hexagonal)
3. **Cube** (Regular hexahedron, all faces are squares)
4. **Frustum** (Truncated pyramid)
5. **Cylinder** (Circular cylinder)
6. **Cone** (Circular cone)
7. **FrustumCone** (Truncated cone)
8. **Spheroid** (Ball shape)

**### Naming Rules**
- After the category keyword, you MUST append the labeled vertices.
- **Format**: Category Vertices
- Examples:
Pyramid P-ABCD
Cube ABCD-$A_1B_1C_1D_1$
Cylinder $O_1$-$O_2$
Spheroid O

**Output Format Template**
**Structure:**
Pyramid P-ABC

Figure 9: Prompts for 3D geometric structural analysis. These prompts guide the model to classify the solid type and identify its defining vertices.

**Prompt: Planes**

```
You are an expert in solid geometry diagram
parsing.
Your ONLY task: output ALL planes visible or
implied in the structure.
Do NOT output any explanation.

### What counts as a "Plane"
1. Boundary Faces: The external flat surfaces
(top, bottom, sides).
2. Internal Sections: Explicitly drawn planes
cutting through the solid.

### Rules & Constraints (CRITICAL)
1. Maximal Point Set Principle:
- List EVERY labeled point on the plane
(vertices, edge points, etc.).
- Do NOT output a plane if it is a subset of
another output.
- Example: Use plane A B E C instead of plane
A B C if E is on BC.
2. Handle Hidden Faces:
- Infer hidden faces based on structure and
dashed lines.
- Only list faces that have at least 3 labeled
points.
3. Output Format:
- Begin with header **Planes:** on the first
line.
- Format: plane P1 P2 P3 ...

Output Format Template
Planes:
plane A B C D
plane A B E
plane C D F G
```

Figure 10: Prompts for solid geometric plane extraction. These prompts enforce the Maximal Point Set Principle to ensure all coplanar labeled points are grouped together.

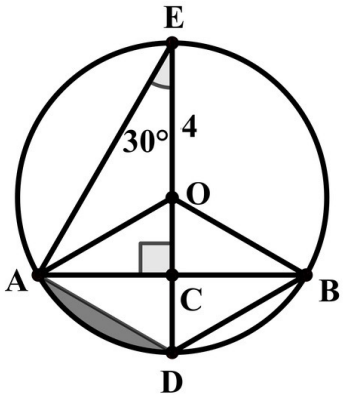
measures. For solid geometry, the parsed outputs successfully represent 3D structural skeletons, including the identification of hidden edges and the connectivity between polyhedral planes and vertices. Notably, our unified formal language is designed to be highly concise and follows a style that closely resembles natural language. This human-readable syntax ensures that the symbolic descriptions remain intuitive and interpretable, while effectively eliciting the logical reasoning capabilities of large multimodal models.

## G Case Studies

To illustrate the effectiveness of the formalized descriptions generated by our parsing method, we provide three qualitative examples in Figure 18, 19, and 20. These cases compare the performance of GPT-5.2-1211 on PGPS9K under two settings: **Direct Inference** and **+ Ours** (reasoning augmented by our GDP-4B formal parsing).

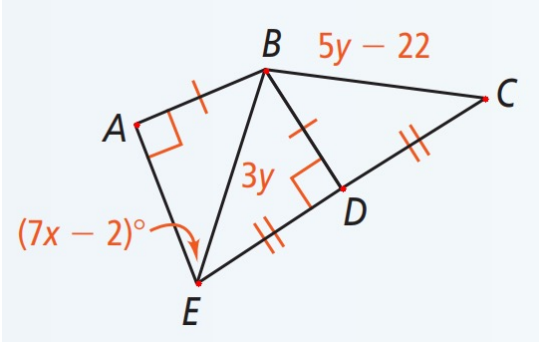
As shown in the examples, our parsing results provide a precise symbolic foundation that corrects the model's reasoning trajectory, leading to the accurate final answer.

## Plane Geometry




```
{
"points" : [ "D", "A", "B",
 "O", "E", "C" ],
"lines" : [
"line A O" ,
"line A C B" ,
"line E O C D" ,
"l ine A D" ,
"line E A",
"line O B" ,
"line D B"
 ],
"circles" : [
"\\odot O lieson E B D A"
 ],
"semantics" : [
"AB \\perp ED on C" ,
"OE = 4" ,
"m \\angle OEA = 30"
 ]
}
```

## Plane Geometry




```
{
"points" : [ "B", "C", "E", "A", "D"],
"lines" : [
"line B C" ,
"line E D C" ,
"line A E" ,
"line A B" ,
"line B E" ,
"line B D"
 ],
"semantics" : [
"AE \\perp AB on A" ,
"EC \\perp BD on D" ,
"BC = 5y - 22",
"BE = 3y" ,
"BD = AB" ,
"DC = ED" ,
"m \\angle AEB = 7x - 2"
 ]
}
```

Figure 11: Representative plane geometry samples from the GDP-29K dataset.

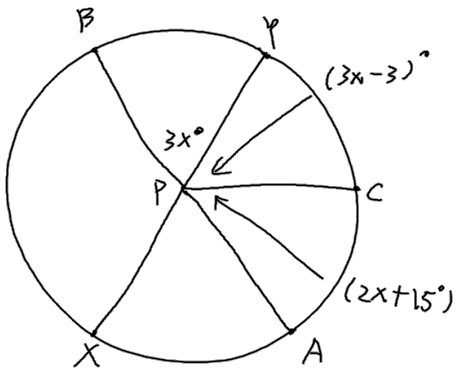
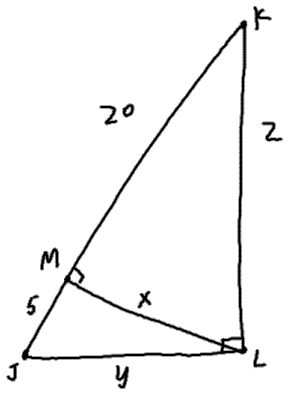
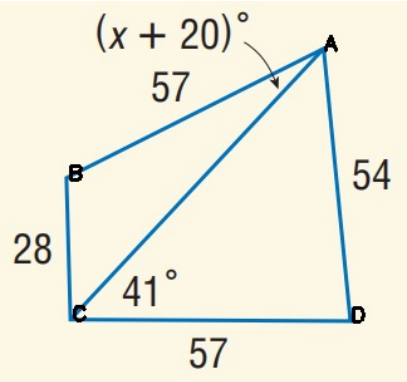
| Plane Geometry | Plane Geometry |
| --- | --- |
|   |   |
| { "points" : [ "Y" , "B" , "X" , "A" , "P" , "C" ], "lines" : [ "line Y P X" , "line B P A" , "line P C" ], "circles" : [ "\\odot P lieson B Y C A X" ], "semantics" : [ "m \\angle YPB = 3x" , "m \\angle YPC = 3x - 3" , "m \\angle APC = 2x+15" ] } | { "points" : [ "J" , "K" , "M", "L" ], "lines" : [ "line J L" , "line M L" , "line K M J" , "line K L" ], "semantics" : [ "ML \\perp KJ on M" , "JL \\perp KL on L" , "JM = 5" , "JL = y" , "KL = z" , "KM = 20" , "LM = x" ] } |

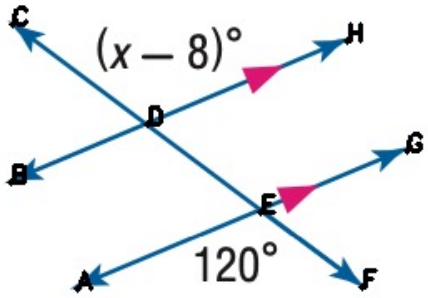
Figure 12: Representative plane geometry samples from the GDP-29K dataset.

## Plane Geometry




```
{
"points" : [ "A" , "B" , "C" , "D"],
"lines" : [
"line A C" ,
"line A D" ,
"line C D"
],
"semantics" : [
"CB = 28" ,
"DC = BA = 57" ,
"AD = 54" ,
"m \\angle ACD = 41" ,
"m \\angle CAB = x+20"
]
}
```

## Plane Geometry




```
{
"points" : [ "A" , "B" , "C" , "D" , "E" ,
"F" , "G" , "H"],
"lines" : [
"line C D E F" ,
"line A E G" ,
"line B D H"
],
"semantics" : [
"BH \\parallel AG" ,
"m \\angle HDC = x - 8" ,
"m \\angle AEF = 120"
]
}
```

Figure 13: Representative plane geometry samples from the GDP-29K dataset.

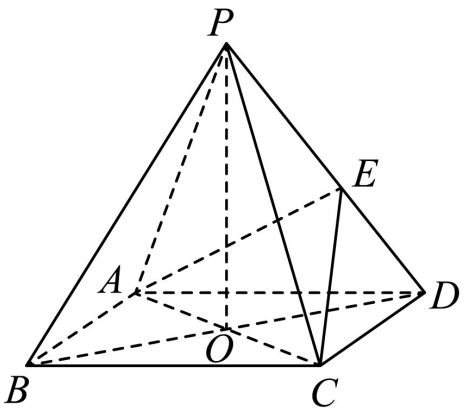
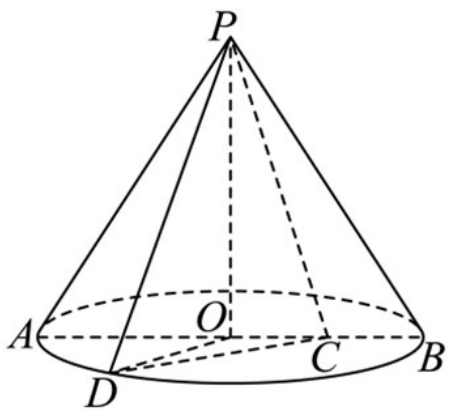
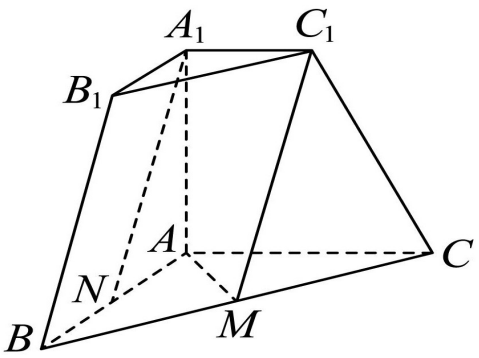
| Pyramid | Cone |
| --- | --- |
|  |  |
| "structure": [<br>"pyramid P-ABCD"<br>],<br>"points": [<br>"P","A","B","C","D","O","E"],<br>"lines": [<br>"line A B", line B C",<br>"line C D", "line D A",<br>"line P A", "line P B",<br>"line P C", "line P E D",<br>"line P O", "line A O C",<br>"line E A", "line E C",<br>"line B O D"<br>],<br>"planes": [<br>"plane A B C D O",<br>"plane P A B",<br>"plane P B C",<br>"plane P C D E",<br>"plane P E D A",<br>"plane P A O C",<br>"plane E A O C",<br>"plane P E D O B"<br>] | "structure": [<br>"cone P-OA"<br>],<br>"points": [<br>"P", "A", "B", "C", "D", "O"<br>],<br>"lines": [<br>"line P A",<br>"line P B",<br>"line P O",<br>"line P D",<br>"line P C",<br>"line O D",<br>"line A O C B",<br>"line C D"<br>],<br>"planes": [<br>"plane O D C",<br>"plane P A O B C",<br>"plane P O D",<br>"plane P D C"<br>],<br>"circles": [<br>"\\odot O lieson A D B"<br>] |

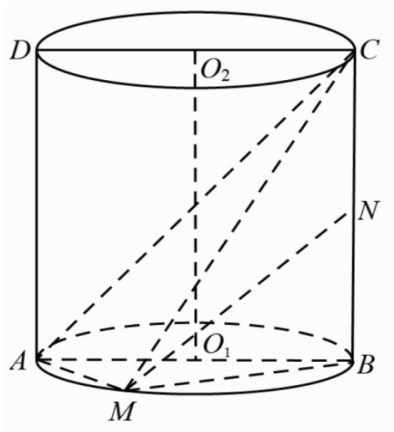
Figure 14: Representative solid geometry samples from the GDP-29K dataset.

## Frustum




```
"structure": [
    "Frustum ABC-A_{1}B_{1}C_{1}"
],
"points": [
    "A", "B", "C",
    "A_{1}", "B_{1}", "C_{1}",
    "M", "N"
],
"lines": [
    "line A N B", "line B M C",
    "line A C", "line A_{1} B_{1}",
    "line B_{1} C_{1}",
    "line A_{1} C_{1}",
    "line A A_{1}", "line B B_{1}",
    "line C C_{1}", "line A_{1} N",
    "line A M", "line C_{1} M"
],
"planes": [
    "plane A N B M C",
    "plane A_{1} B_{1} C_{1}",
    "plane A N B B_{1} A_{1}",
    "plane B M C C_{1} B_{1}",
    "plane A C C_{1} A_{1}"
]
```

## Cylinder




```
"structure": [
    "Cylinder AD-BC"
],
"points": [
    "A", "B", "C", "D",
    "O_{1}", "O_{2}", "M", "N"
],
"lines": [
    "line O_{2} O_{1}",
    "line D O_{2} C",
    "line A O_{1} B", "line A M",
    "line B M", "line D A",
    "line C N B", "line A C",
    "line M C", "line M N"
],
"circles": [
    "\\odot O_{1} lieson A B M",
    "\\odot O_{2} lieson D C"
],
"planes": [
    "plane O_{2} C D",
    "plane O_{1} A B M",
    "plane A B C D",
    "plane A M C",
    "plane M B N C"
]
```

Figure 15: Representative solid geometry samples from the GDP-29K dataset.

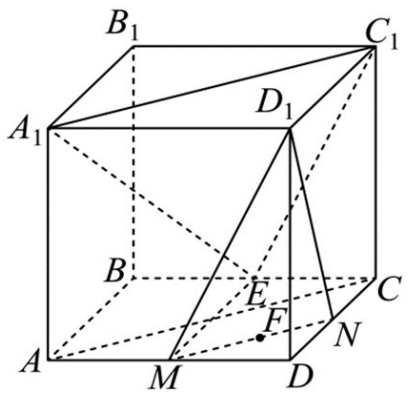
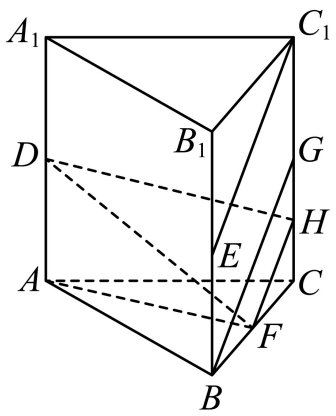
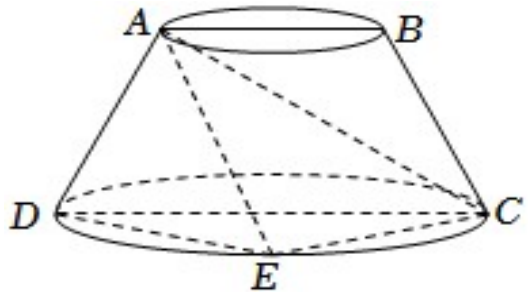
<table>
<tr><th>Cube</th><th>Prism</th></tr>
<tr><td> 
</td><td> 
</td></tr>
<tr><td>"structure": ["Cube ABCD-A_{1}B_{1}C_{1}D_{1}"],<br>"points": [<br>"A", "B", "C", "D",<br>"A_{1}", "B_{1}", "C_{1}", "D_{1}",<br>"M", "N", "E", "F"],<br>"lines": [<br>"line A M D", "line D N C",<br>"line A B", "line B C",<br>"line A_{1} B_{1}",<br>"line B_{1} C_{1}",<br>"line C_{1} D_{1}", "line D_{1} A_{1}",<br>"line A A_{1}", "line B B_{1}",<br>"line C C_{1}", "line D D_{1}",<br>"line A_{1} C_{1}", "line A C",<br>"line A_{1} E", "line C_{1} E",<br>"line M E", "line D_{1} M",<br>"line D_{1} N", "line M F N"],<br>"planes": [<br>"plane A B E C N D M F",<br>"plane A_{1} B_{1} C_{1} D_{1}",<br>"plane A B B_{1} A_{1}",<br>"plane B C C_{1} B_{1}",<br>"plane C N D D_{1} C_{1}",<br>"plane D M A A_{1} D_{1}",<br>"plane D_{1} M N",<br>"plane A C C_{1} A_{1}",<br>"plane A_{1} C_{1} E",<br>"plane C_{1} D_{1} M E"]</td><td>"structure": [<br>"Prism A_{1}B_{1}C_{1}-ABC“<br>],<br>"points": [<br>"A_{1}", "B_{1}", "C_{1}",<br>"A", "B", "C", "D", "E", "F", "H", "G"],<br>"lines": [<br>"line A_{1} B_{1}", "line B_{1} C_{1}",<br>"line C_{1} A_{1}", "line A B",<br>"line B F C", "line C A",<br>"line A_{1} D A", "line B_{1} E B",<br>"line C_{1} G H C", "line A F",<br>"line D F", "line D H",<br>"line H F", "line C_{1} E",<br>"line G B"<br>],<br>"planes": [<br>"plane A_{1} B_{1} C_{1}",<br>"plane A B F C",<br>"plane A_{1} B_{1} E B A D",<br>"plane B_{1} C_{1} G H C F B E",<br>"plane C_{1} A_{1} D A C H G",<br>"plane D H F",<br>"plane D F A"<br>]</td></tr>
</table>

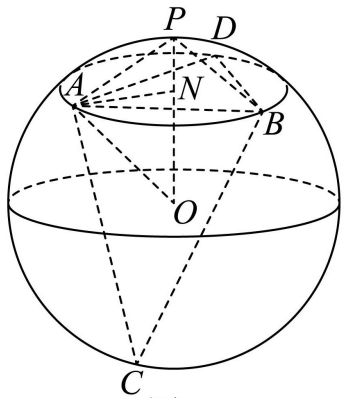
Figure 16: Representative solid geometry samples from the GDP-29K dataset.

## FrustumCone




```
"structure": [
    "FrustumCone AD-BC"
],
"points": [
    "A", "B", "C", "D", "E"
],
"lines": [
    "line A B",
    "line D C",
    "line D E",
    "line E C",
    "line A D",
    "line B C",
    "line A E",
    "line A C"
],
"circles": [
    "\\odot # lieson A B",
    "\\odot # lieson D E C"
],
"planes": [
    "plane D E C",
    "plane A B C D",
    "plane A E C",
    "plane A D E"
]
```

## Spheriod




```
"structure": [
    "Spheriod O-ABCP"
],
"points": [
    "C", "O", "A", "B", "N", "P", "D"
],
"lines": [
    "line C A",
    "line C B",
    "line A B",
    "line O A",
    "line A N",
    "line A P",
    "line A D",
    "line B D",
    "line P B",
    "line O N P"
],
"planes": [
    "plane C A B",
    "plane A B D N",
    "plane P A B",
    "plane P A O N"
],
"circles": [
    "\\odot N lieson A B D",
    "\\odot O"
]
```

Figure 17: Representative solid geometry samples from the GDP-29K dataset.

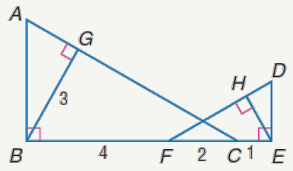
## Result of GPT-5.2 on PGPS9K sample

**Question:**

**Find FH if △ABC ~ △DEF, BG is an altitude of △ABC , EH is an altitude of △DEF, BG = 3, BF = 4, FC = 2, and CE = 1.**

**Choices:**

**A. 1.5**
**B. 2.0**
**C. 2.5**
**D. 2.598**




**Additional Parsed Information:**

″ points ″:[…],
″ lines ″:[…],
″ semantics ″:[
″ AC ⊥ GB on G ″
″ BE ⊥ AB on B ″
″ FD ⊥ HE on H ″
″ BE ⊥ DE on E ″
″ FC = 2 ″
″ EC = 1 ″
″ BF = 4 ″
″ BG = 3 ″]

**Ground Truth: D**

### Original Answer

**Step 1:** Since △ABC ~ △DEF with correspondence A⇔D; B⇔E; C⇔F, the side BC corresponds to EF, and altitudes to corresponding sides scale the same way.
**Step 2:** From the base line: BC = BF + FC = 4 + 2 = 6, EF = EC + CF = 1 + 2 = 3; So the similarity scale factor from △ABC to △DEF is

$$k = \frac{EF}{BC} = \frac{3}{6} = 0.5$$

**Step 3:** Given the altitude BG = 3 in △ABC, the corresponding altitude in △DEF is

$$EH = k \cdot BG = \frac{1}{2} \cdot 3 = 1.5$$

**Step 4:** Thus **FH (the altitude segment to DE , labeled H)** has length 1.5. ✗
Answer: A

### Parsed_inference_output(Included Additional Parsed Information)

**Step 1:** Since △ABC ~ △DEF with correspondence A⇔D; B⇔E; C⇔F, we have

$$\frac{EF}{BC} = \frac{EH}{BG}$$

**Step 2:** From the diagram/data: BF = 4; FC = 2; CE = 1. Thus BC = BF + FC = 4 + 2 = 6, EF = EC + CF = 1 + 2 = 3; So the similarity scale factor from △ABC to △DEF is

$$\frac{EF}{BC} = \frac{3}{6} = 0.5$$

**Step 3:** Altitude scale the same way, hence

$$EH = \frac{1}{2} \cdot BG = \frac{1}{2} \cdot 3 = 1.5$$

**Step 4:** In the right △EHF, **HF is the altitude to hypotenuse DE, so HF ⊥ DE**, and EF is a leg. Thus

$$HF = \sqrt{EF^2 - EH^2} = \sqrt{3^2 - \left(\frac{3}{2}\right)^2} = \sqrt{\frac{27}{4}} = \frac{3\sqrt{3}}{2}$$ ✓

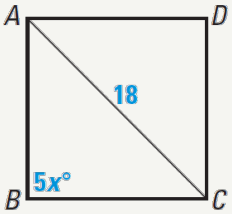
Answer: D

Figure 18: Qualitative comparison between Direct Inference and our method (+ Ours) on PGPS9K. Our formal parsing provides precise symbolic grounding that rectifies reasoning errors.

## Result of GPT-5.2 on PGPS9K sample

**Question:**
ABCD is a square. Find the value of x.

**Choices:**
A. 3.6
B. 9.0
C. 18.0
D. 90.0




**Additional Parsed Information:**

"points ":["A", "B", "C", "D"],
"lines ":["line A B",
"line A D",
"line D C",
"line B C",
"line A C"],
"semantics ":[
"AC = 18 "
"m \\angle ABC = 5x "]

**Ground Truth: C**

**Original Answer**
**Step 1:** In a square, each corner angle is 90°, The diagonal AC bisects the right angle at A, so the angle between the diagonal and a side is 45°.
**Step 2:** Thus **5x° = 45°** ⟹ x = 9.0
Answer: B

*Perceptual error*

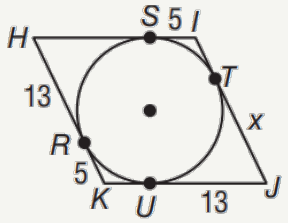
**Parsed_inference_output (Included Additional Parsed Information)**
**Step 1:** Since ABCD is a square, every interior angle is 90°.
**Step 2:** Thus ∠ABC = 90° = 5x° ⟹ x = 18.0
Answer: C

Figure 19: Qualitative comparison between Direct Inference and our method (+ Ours) on PGPS9K. Our formal parsing provides precise symbolic grounding that rectifies reasoning errors.

## Result of GPT-5.2 on PGPS9K sample

**Question:**
The HIJK is circumscribed about the circle.
Find x

**Choices:**
A. 1.923
B. 5.0
C. 13.0
D. 21.0




**Additional Parsed Information:**
"points ":[...],
"lines ":["line H S I",
"line I T J",
"line K U J",
"line H R K"],
"semantics ":[
" IS = KR = 5"
" HR = JU = 13"
" JT = x"]

**Ground Truth: C**

**Original Answer**
**Step 1:** Because HIJK is circumscribed about a circle, tangent segments from the same vertex are equal. Label the tangency points so that: From I: IS = IT = 5; From K: KR = KU = 5; From H: HS = HR; From J: JT = JU
**Step 2:** Given HK = HR + RK = HR + 5 = 13, so HR = 8. Hence HS = 8.
**Step 3:** Now the whole top side: HI = HS + SI = 8 + 5 = 13. For the bottom side: KJ = KU + UJ = 5 + 13 = 18 ⟹ UJ = 13
**Step 4:** Since JU = JT, we get JT = 13. Right side: IJ = IT + TJ = 5 + 13 = 18. **So x = 18.**
Among the choices, the only matching option(by intended setup) is D. 21.0? But the computed value is x = 18 (not listed)

*Perceptual error*

**Parsed_inference_output (Included Additional Parsed Information)**
**Step 1:** Because HIJK is circumscribed about a circle, tangent segments from the same vertex are equal. From the diagram: From I: IS = IT.
**Step 2:** Given IS = 5, so IT = 5.
**Step 3:** From J: JU = JT. Given JU = 13, so JT = x = 13.
Answer: C

Figure 20: Qualitative comparison between Direct Inference and our method (+ Ours) on PGPS9K. Our formal parsing provides precise symbolic grounding that rectifies reasoning errors.