Plane Geometry Problem Solving with Multi-modal Reasoning: A Survey

Anonymous ACL submission

Abstract

Plane geometry problem solving (PGPS) has recently gained significant attention as a benchmark to assess the multi-modal reasoning capabilities of large vision-language models. Despite the growing interest in PGPS, the research community still lacks a comprehensive overview that systematically synthesizes recent work in PGPS. To fill this gap, we present a survey of existing PGPS studies. We first categorize PGPS methods into an encoder-decoder framework and summarize the corresponding output formats used by their encoders and decoders. Subsequently, we classify and analyze these encoders and decoders according to their architectural designs. Finally, we outline major challenges and promising directions for future research. In particular, we discuss the hallucination issues arising during the encoding phase within encoder-decoder architectures, as well as the problem of data leakage in current PGPS benchmarks.

1 Introduction

Automated plane geometry problem solving (PGPS) has emerged as an important benchmark in artificial intelligence research due to its unique requirement for multi-modal reasoning with mathematical rigor (Seo et al., 2015; Chen et al., 2021). Typically, geometry problems combine textual descriptions with visual diagrams, each providing essential complementary information. The inherent necessity to integrate linguistic and visual modalities makes plane geometry a compelling testbed for advancing the multi-modal understanding capabilities of AI systems. Furthermore, practical motivations such as developing intelligent tutoring systems (Ritter et al., 2010; Aleven and Koedinger, 2002; Lee et al., 2025) and standardized benchmarks for evaluating AI reasoning (Chen et al., 2021; Cao and Xiao, 2022) highlight the importance of continued research in this area.

Nevertheless, substantial challenges persist in achieving full automation. Foremost among these is the complexity arising from the multi-modal nature of geometry problems, requiring precise alignment between textual statements and corresponding diagram elements (Seo et al., 2014). Resolving ambiguities in textual descriptions through visual references and accurately mapping entities between text and diagrams pose significant hurdles (Sachan et al., 2017; Zhang et al., 2022). Geometric diagrams also introduce unique challenges absent in natural images and other types of diagrams, including precise recognition of abstract symbols, e.g., angle markers and length indicators, accurate detection of geometric primitives, e.g., points, lines, and circles, and interpretation of implicit spatial relationships governed by geometric constraints. Additionally, effective PGPS demands embedding deep geometric domain knowledge, applying geometric axioms and theorems during the reasoning that are often implicitly assumed (Sachan et al., 2017; Sachan and Xing, 2017; Lu et al., 2021). Thus, integrating linguistic comprehension, visual analysis, and geometric reasoning continues to drive the complexity and significance of research in automated PGPS.

Recently, numerous new benchmarks, large-scale datasets, and model architectures have been proposed to tackle the challenges of PGPS. However, despite this rapid progress, most existing surveys on mathematical or multi-modal reasoning address geometry problems only as part of broader domains (Li et al., 2025; Yan et al., 2025; Yuan et al., 2025) and thus fail to examine the unique challenges of PGPS in depth. Consequently, the literature still lacks a dedicated, up-to-date survey centered on PGPS. The goal of this paper is to fill the gap by providing the PGPS research community with a structured overview of the latest benchmarks, datasets, and multi-modal reasoning approaches tailored specifically to PGPS.

The structure of this paper is summarized as follows: We first describe the definition of PGPS and relevant tasks (§2). We then introduce an overall framework for solving PGPS problems as an encoder-decoder architecture with intermediate representations (§3). Next, we review the details of encoder (§4) and decoder (§5) structures. Some additional thoughts are provided from the data collection perspective (Appendix A). Finally, we address the remaining challenges and promising future directions in automated PGPS (§6).

2 Tasks and benchmarks

In this section, we first define the PGPS and then introduce three tasks that are commonly tackled in the PGPS community, along with the benchmarks for each task.

2.1 Definition of PGPS

Euclidean plane geometry studies the properties and relationships among geometric primitives, e.g., points, lines, and circles, in a flat, two-dimensional space (Fitzpatrick and Heiberg, 2007). PGPS involves inferring unknown geometric properties or relationships from a given set of primitives and their known relations, such as determining the length of an unknown side in a triangle given the lengths of two sides and the measure of the included angle.

In real-world scenarios, plane geometry problems usually present as diagram and textual description pairs, as demonstrated in Fig. 1. The diagrams and accompanying textual descriptions typically complement each other in representing geometric primitives and relations. Diagrams usually provide visual information about spatial relationships, whereas textual descriptions explicitly mention properties or relational details. Due to this complementary nature, PGPS methods in real-world applications must not only infer unknown geometric facts but also accurately parse geometric information from these diagrams and text pairs.

2.2 PGPS tasks

We describe the three main tasks, along with the corresponding benchmarks, that are mainly tackled via PGPS research. Fig. 1 illustrates three examples for each task. For further details on the benchmarks from various perspectives, such as reasoning complexity, diagram-text interdependency, and data collection methods, refer to Appendix A.

2.2.1 Direct-answer and multiple-choice tasks

Task description Most PGPS works quantify the capacity of a PGPS method to infer a single, well-defined property of a geometric entity from a unified diagrammatic–textual problem statement. The requested properties fall into two categories: i) numerical targets, e.g., angle magnitude, segment length, or area (Seo et al., 2015; Lu et al., 2021; Chen et al., 2021), and ii) categorical targets, e.g., the perpendicularity or parallelism of two lines (Xu et al., 2025).

PGPS methods are also evaluated through multiple-choice tasks (Lu et al., 2024; Zhang et al., 2025a). While these tasks use the same problems as direct-answer tasks, each multiple-choice problem provides a fixed set of candidate responses. A PGPS method must select the option that correctly identifies the target property, or equivalently, predict a value matching one of the provided choices. For example, in the scenario depicted in Fig. 1, the correct response is the label "c" or its corresponding value, "None."

Evaluation metrics In direct-answer tasks, performance is reported as top-N accuracy: a PGPS method is considered correct when the ground truth answer appears within its N candidate answers. For multiple-choice tasks, the metric depends on the output representation of the method. If the method predicts an option label, evaluation reduces to standard top-1 accuracy. If it produces a value, e.g., scalar, a modified version of top-N accuracy is utilized: the N generated values are scanned in order, and the attempt is scored correct once the first value that coincides with any listed option matches the ground truth.

Benchmarks Most PGPS benchmarks have been proposed to evaluate model performance on direct-answer and multiple-choice tasks. Some benchmarks exclusively consist of plane geometry problems (Alvin et al., 2017; Seo et al., 2015; Lu et al., 2021; Chen et al., 2021; Cao and Xiao, 2022; Zhang et al., 2023, 2024c; Fu et al., 2025; Kazemi et al., 2024; Xu et al., 2025), while others include plane geometry problems as part of broader benchmarks designed for general multi-modal reasoning evaluation (Lu et al., 2024; Zhang et al., 2025a; Yue et al., 2024; Kamoi et al., 2024; Wang et al., 2024a; Zou et al., 2025; Gupta et al., 2024; Wang et al., 2025).

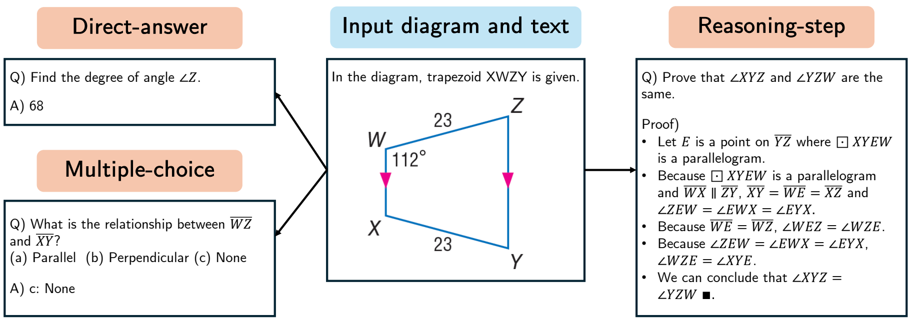


Figure 1: Illustration of three PGPS tasks. The three tasks are commonly used to evaluate PGPS methods in existing benchmarks: i) direct-answer, ii) multiple-choice, and iii) reasoning-step construction.

2.2.2 Reasoning tasks

Task description Some PGPS benchmarks assess methods not only on the correctness of the final answer but also on the soundness of the intermediate reasoning (Chen et al., 2022; Jaiswal et al., 2024). In a widely adopted proving problem setting, a PGPS method must generate a sequence of geometric axioms and theorems that derive the target statement, e.g., two angles are congruent, directly from the given conditions.

Evaluation metrics For reasoning-step construction tasks, top-N accuracy is again adopted, granting success when any of the N predicted reasoning steps exactly reproduces the ground-truth steps.

Benchmarks UniGeo (Chen et al., 2022) is currently the only benchmark designed explicitly to systematically measure reasoning capabilities. Recently, approaches leveraging LLMs have emerged to evaluate individual reasoning steps (Zhang et al., 2025a; Jaiswal et al., 2024). However, these methods inherently rely on LLMs, posing significant limitations. Consequently, proposing diverse and systematic reasoning benchmarks remains an open research challenge.

3 Overall approach

PGPS models typically employ an encoder-decoder architecture, as demonstrated in Fig. 2. The *encoder* jointly processes the diagram and textual description to produce an *intermediate representation* that captures essential geometric information of the problem. The *decoder* then utilizes the extracted intermediate representation to generate a solution, presented as either a theorem sequence, a logic program, or a natural-language description. Finally, the answer is obtained by post-processing the generated solution, e.g., by executing the logic program or extracting the final result from the natural-language description.

Before we discuss the detailed approach to constructing the encoder and decoder, we first review the output formats of the encoder and decoder commonly used across different PGPS tasks.

3.1 Encoder outputs

The output of an encoder forms an intermediate representation that can be further used as an input to a decoder. We categorize the output format of the encoder into i) formal-language description and ii) embedding vectors.

Formal-language description Several studies explicitly extract geometric primitives and relations from given diagram-text pairs, converting them into formal-language descriptions. A formal-language description consists of an *entity* set and a *predicate* set. The entity set contains geometric primitives, e.g., elementary primitives such as points, lines, and circles (Zhang et al., 2022, 2023), or higher-level shapes such as triangles and squares (Seo et al., 2015; Sachan et al., 2017; Sachan and Xing, 2017; Lu et al., 2021), along with non-geometric tokens such as numbers and variable names. The predicates define the relationships among the entities. For instance, an equality predicate binds two entities $\angle ABC$ and 30° to represent the numerical value of the angle, i.e., $\angle ABC = 30°$ or specify geometric relations, such as segments AB and BC being perpendicular, i.e., $AB \perp BC$.

In earlier studies, rule-based approaches (Koo et al., 2008; Bansal et al., 2014) and semantic parsers (Lewis et al., 2020) have been proposed to extract formal-language descriptions from textual descriptions without analyzing the diagram (Seo et al., 2015; Lu et al., 2021). Recent works extend these approaches to extract a formal language description from a diagram-text pair. Consequently, many PGPS studies release datasets consisting of diagrams and formal-language description pairs to train diagram parsers in a supervised way (Seo

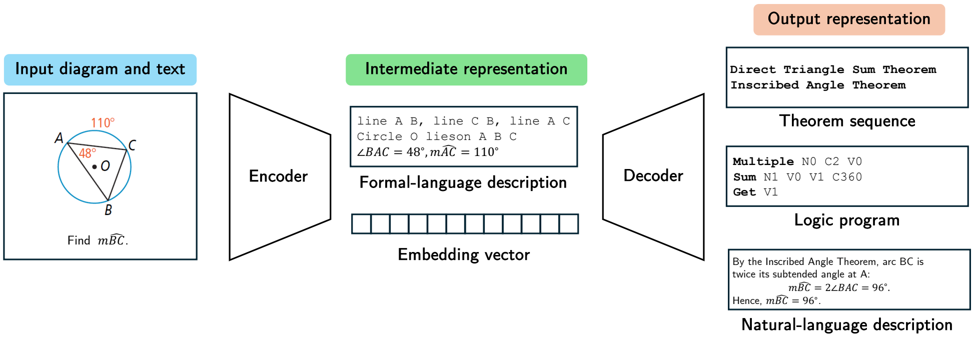


Figure 2: Visualization of the overall structure of PGPS methods. PGPS methods first encode the input diagram and text into an intermediate representation. The encoded representation is then passed to the decoder, which generates the final solution as a theorem sequence, a logic program, or a natural-language description.

et al., 2015; Sachan et al., 2017; Sachan and Xing, 2017; Zhang et al., 2022; Lu et al., 2021; Zhang et al., 2023, 2024c)

Embedding vectors Certain PGPS encoders represent inputs as embedding vectors, typically utilizing one of three strategies: i) embedding diagrams and textual descriptions separately and subsequently merging them (Chen et al., 2021; Cao and Xiao, 2022; Chen et al., 2022; Ning et al., 2023; Liang et al., 2023; Jian et al., 2023b), ii) embedding diagrams exclusively and then combining them with raw textual inputs (Xia et al., 2025; Cho et al., 2025; Shi et al., 2024; Zhang et al., 2025b; Gao et al., 2025; Zhang et al., 2025f; Peng et al., 2025; Xu et al., 2024), or iii) jointly processing diagrams and texts through a unified encoder (Zhang et al., 2023; Li et al., 2024). Although these embeddings are generally less interpretable compared to formal-language descriptions, they enable end-to-end training with the decoder.

3.2 Decoder outputs

Given the output of the encoder, the decoder generates the solution from which the final answer can be derived. We classify decoder output formats into three types: i) theorem sequences, ii) logic programs, and iii) natural-language descriptions.

A sequence of theorems Many PGPS works represent the output of a PGPS problem as a sequence of theorem applications. This approach naturally aligns with a reasoning process, in which theorems are iteratively applied to given entities and predicates to logically derive new geometric facts, including the target predicate specified as a goal (Trinh et al., 2024). Specifically, given geometric entities and predicates extracted from the original description, theorems from a predefined library can be applied to the entities and predicates to derive additional predicates not explicitly stated in the original problem. Recent PGPS datasets provide annotated triples consisting of the formal-language description, the target predicate, and a corresponding reference theorem sequence (Sachan et al., 2017; Sachan and Xing, 2017; Lu et al., 2021; Zhang et al., 2024c).

A logic program A logic program is commonly adopted as an output representation for PGPS. Specifically, inspired by the observation that the reasoning process in PGPS typically involves applying a series of operations to numerical constants and variables provided in the problem (Chen et al., 2021; Amini et al., 2019; Chen et al., 2023), a logic program is defined as a sequence of triples, each consisting of an operation and its operands, such as numerical values and variable names. The operations in these programs fall into two main categories: i) arithmetic functions, ranging from basic operations like addition and multiplication to geometry-specific computations such as the Pythagorean operation (Chen et al., 2021; Cao and Xiao, 2022; Chen et al., 2022), and ii) equality assertions that establish identity between two expressions (Zhang et al., 2023). Several PGPS datasets provide paired examples, each consisting of a diagram-text problem and its corresponding logic program (Chen et al., 2021; Cao and Xiao, 2022; Chen et al., 2022; Zhang et al., 2023).

A natural-language description Recent PGPS methods generate solutions and answers in natural language without relying on a specific template. The inherent flexibility of natural language allows these models to easily provide outputs for a wide range of tasks, e.g., geometric diagram captioning, without being limited to fixed problem-solving formats. To train such methods, various types of PGPS datasets have been proposed. For tasks which focus on problem solving, the output, given a diagram and text, can either be the answer expressed in natural language (Shi et al., 2024) or a reasoning path in the form of a chain-of-thought (Wei et al., 2022) to infer the answer (Zhang et al., 2025b; Gao et al., 2025). In addition to problem solving, datasets have also been proposed for tasks such as geometric diagram captioning (Zhang et al., 2025b; Gao et al., 2025; Cho et al., 2025; Xia et al., 2025) and question answering (Gao et al., 2025).

3.3 Encoder-decoder with desired outputs

Once the intermediate representations and output representations are determined based on target problems or tasks, one can choose an appropriate encoder and decoder that can produce the desired outputs. Fig. 3 summarizes possible combinations of encoder-decoder architectures along with the desired outputs. A combination of encoder, intermediate representation, decoder, and output representation can lead to a specific architecture for PGPS. In the following two sections, we review the possible choices of encoder and decoder structures. We additionally summarize the strengths and weaknesses of each design choice in Table A2 and relate them to the performance results reported in Table A3 and Table A4 in Appendix B.2.

4 Encoders

The encoder extracts the relevant components from the given diagram and text that are necessary for PGPS. We review the neural network-based encoders based on the desired output format: i) formal-language description and ii) embedding vector. Rule-based encoders are reviewed separately in Appendix C.1.

4.1 Formal-language description generation

Recent PGPS approaches adopt neural encoders to generate formal-language descriptions from diverse diagrams and texts, typically training separate encoders for each modality. Neural diagram encoders commonly operate in two stages: primitive detection using object detectors such as RetinaNet (Lin et al., 2017b; Lu et al., 2021) and feature pyramid networks (Lin et al., 2017a; Zhang et al., 2022), followed by relation inference modeled either as a constrained optimization problem (Lu et al., 2021) or as a graph-learning task leveraging graph neural networks (GNNs) (Zhang et al., 2022). For text encoding, subsequent PGPS studies (Sachan et al., 2017; Sachan and Xing, 2017) commonly employ logistic regression models, as originally introduced by GEOS (Seo et al., 2015), to extract primitives and relations from problem statements.

4.2 Embedding vector generation

To enable end-to-end learning, recent PGPS methods employ neural encoders that map both the diagram and text into a unified embedding space, providing a joint vector representation for PGPS. Here, we review the neural encoders based on their training strategy.

Learning from scratch Early PGPS works train joint diagram-text encoders and decoders end-to-end from scratch on target PGPS datasets. Diagram embeddings commonly utilize convolutional neural networks (CNNs), including vanilla CNN (Zhang et al., 2023), ResNet (He et al., 2016; Chen et al., 2021; Cao and Xiao, 2022), DenseNet (Huang et al., 2017; Jian et al., 2023a), and VQ-VAE encoders (van den Oord et al., 2017; Liang et al., 2023), as well as Vision Transformers (ViT) (Dosovitskiy et al., 2021; Ning et al., 2023). Text embeddings are typically produced by sequential models like LSTMs (Hochreiter and Schmidhuber, 1997; Chen et al., 2021; Cao and Xiao, 2022) or Transformer-based encoders (Vaswani et al., 2017), such as vanilla Transformer (Zhang et al., 2023; Ning et al., 2023; Li et al., 2024) and RoBERTa (Liu et al., 2019; Cao and Xiao, 2022). Diagram and text embeddings are fused via co-attention networks (Yu et al., 2019; Chen et al., 2021; Ning et al., 2023), bi-directional GRUs (Chung et al., 2014; Zhang et al., 2023; Li et al., 2024), or Transformers (Chen et al., 2022).

Besides direct optimization on PGPS tasks, joint encoders frequently employ auxiliary objectives for improved performance. Many approaches incorporate self-supervised tasks, including jigsaw-location prediction (Chen et al., 2021; Cao and Xiao, 2022; Jian et al., 2023a), masked-token prediction in text (Devlin et al., 2019; Chen et al., 2022; Zhang et al., 2023; Li et al., 2024) or diagrams (He et al., 2022; Ning et al., 2023), text-conditioned diagram-symbol classification (Ning et al., 2023), and VQ-VAE objective (Liang et al., 2023). Other studies leverage explicit labels, training encoders for geometry-element or knowledge-point classification (Chen et al., 2021; Cao and Xiao, 2022), or contrastive learning between dia-

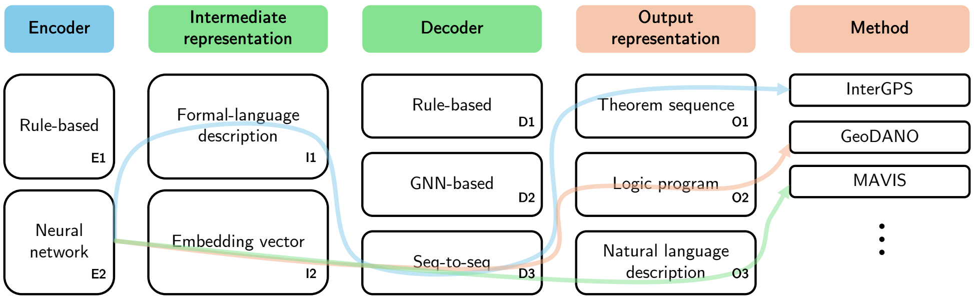


Figure 3: Overview of the PGPS pipeline. PGPS methods can be categorized based on the combination of the encoder, intermediate representation, decoder, and output representation. For example, the InterGPS can be represented as a combination of E2, I1, D3, and O1. We summarize PGPS methods as a combination of these components in Table A1.

gram patches and textual tokens (Li et al., 2024).

Pre-trained encoders To leverage pretrained knowledge and enhance training efficiency, many recent PGPS methods employ neural encoders inspired by the LLaVA architecture (Liu et al., 2023), which integrates a pretrained vision encoder to encode diagrams. Specifically, diagrams are first transformed into visual embeddings using a pretrained vision encoder, followed by a lightweight adapter consisting of a multi-layer perceptron. During training, only the adapter parameters are updated, keeping the vision encoder frozen to preserve general visual knowledge and reduce training cost. While OpenCLIP (Radford et al., 2021) is the most commonly used backbone (Shi et al., 2024; Gao et al., 2025; Xu et al., 2024), other general-purpose models such as SigLIP (Zhai et al., 2023; Zhang et al., 2025f) and InternViT (Chen et al., 2024c; Peng et al., 2025), as well as the math-specific Math-CLIP encoder (Zhang et al., 2025b; Peng et al., 2025), have also been employed.

Fine-tuned encoders Most pretrained vision encoders perform poorly when applied to geometric diagrams (Zhang et al., 2025b; Xia et al., 2025; Cho et al., 2025). To address this limitation, PGPS methods employing the LLaVA-style architecture typically fine-tune the vision encoders before integrating them into downstream pipelines. Two main fine-tuning strategies are common: i) self-supervised methods such as masked auto-encoding (He et al., 2022; Xia et al., 2025), and ii) weakly supervised methods such as CLIP (Zhang et al., 2025b; Cho et al., 2025), direct preference optimization (Rafailov et al., 2023; Huang et al., 2025), or grounding tasks (Li* et al., 2022; Zhang et al., 2025c), which leverage synthetic geometric diagrams and labels pairs. Nevertheless, since synthetic diagrams do not fully capture the characteristics of real-world diagrams, GeoDANO (Cho et al., 2025) further employs few-shot domain adaptation under the same CLIP training objective to minimize the residual domain gap.

5 Decoders

Based on the representations produced by the encoder, the decoder generates the solution to the problem. We survey the PGPS decoders using the following dimensions: i) input representation and ii) architectural design.

5.1 Formal-language description decoder

We begin by outlining three decoder architectures that operate on formal-language inputs: i) rule-based, ii) GNN, and iii) sequence-to-sequence (seq-to-seq) models. Detailed coverage of rule-based decoders is provided in Appendix C.2.

GNN-based decoders A formal-language description, composed of geometric primitives and their relations, naturally corresponds to a graph structure. Exploiting this, several PGPS decoders first encode the formal description as a graph or hypergraph and then generate theorem-application sequences from the resulting graph representation. Such encodings typically follow one of three schemes: i) primitives as nodes and predicates as edges (Peng et al., 2023), ii) primitives and predicates both as nodes connected via edges (Jian et al., 2023a), or iii) predicates as hypernodes and theorems as directed hyperedges forming a hypertree (Zhang et al., 2024b). These encoded structures are subsequently fed into graph-to-sequence decoders, such as Graphormer (Zhang et al., 2024b), graph Transformer (Peng et al., 2023), or graph convolutional network (Kipf and Welling, 2017) followed by LSTM (Jian et al., 2023a), to produce the target theorem sequence.

Seq-to-seq decoders Some approaches treat formal-language descriptions as a flat token sequence and pass it directly to a seq-to-seq model to generate the corresponding theorem sequence. Transformers are predominantly employed for these tasks by encoding the formal description directly (Lu et al., 2021; Wu et al., 2024b; Zou et al., 2024). A few studies instead utilize off-the-shelf LLMs, e.g., o3-mini (OpenAI, 2025b), without additional training (Zhao et al., 2025).

5.2 Seq-to-seq embedding decoders

Several PGPS studies feed either a joint diagram–text embedding or a concatenation of diagram embedding and raw text into a sequence-to-sequence decoder. Early work primarily employs RNN-based decoders such as LSTMs or GRUs (Chen et al., 2021; Cao and Xiao, 2022; Zhang et al., 2023; Li et al., 2024; Ning et al., 2023; Jian et al., 2023b), while later studies commonly adopt encoder–decoder Transformers such as T5 (Raffel et al., 2020; Liang et al., 2023; Chen et al., 2022). The recent proliferation of LLMs has motivated a shift toward fine-tuning encoder-only Transformers, such as LLaMA (Touvron et al., 2023; Cho et al., 2025; Gao et al., 2025; Xu et al., 2024) and Vicuna (Vicuna, 2023; Shi et al., 2024), specifically adapted for PGPS tasks.

6 Challenges and future directions

We examine the remaining challenges in PGPS and propose potential directions for future work. Additionally, the insights appeared from our survey of PGPS studies are available in Appendix D.

6.1 Hallucination in diagram perception

PGPS methods initially extract geometric primitives and relations from diagrams and text, making accurate perception crucial before reasoning. However, studies indicate that PGPS methods frequently misperceive these primitives and relations, especially when generating natural-language descriptions (Huang et al., 2025; Zhang et al., 2025a) as depicted in Fig. A1. For example, Table A5 reveals that GPT-4.1 (OpenAI, 2025a) fails to capture a fundamental geometric relation among the points and lines and produces hallucinations. These hallucinations not only degrade PGPS performance but also diminish dataset quality. Computer vision studies report similar hallucination issues in datasets produced by large VLMs (Zhang et al., 2025e; Sahoo et al., 2024; Li et al., 2023; Chen et al., 2024b), further evidenced in PGPS datasets as shown in Table 1. Consequently, models trained on hallucinated data suffer measurable performance declines (Zhang et al., 2025e; Lai et al., 2025; Yu et al., 2024; Hirota et al., 2024). To mitigate hallucinations in PGPS models, we suggest two different future directions: i) refining the vision encoder architecture and ii) employing visual prompting.

Architectural remedies Recent studies provide architecture-level analyses demonstrating that ViT, while effective on natural images, exhibit limitations on geometric diagrams due to their high-frequency characteristics such as sharp edges and detailed symbolic annotations (Lin et al., 2025; Zhang et al., 2025d). This architecture-level interaction presents a challenge uniquely pertinent to PGPS, thereby offering novel insights beyond widely recognized generic limitations. To preserve the strengths of ViTs in PGPS while exploiting line-level information more effectively, one could i) integrate ViTs with CNN blocks (Wu et al., 2021), ii) adopt a coarse-to-fine ViT architecture (Pu et al., 2022), or iii) attach modules specifically designed to handle high-frequency cues (Lin et al., 2025).

Visual prompting Visual prompting techniques, such as augmenting diagrams with bounding boxes, markers, or segmentation masks, have emerged as promising solutions for mitigating hallucinations (Wu et al., 2024a; Yang et al., 2023; Ma et al., 2025). These methods are especially beneficial for PGPS tasks, as they dynamically highlight relevant primitives and relations during reasoning and facilitate the critical step of drawing auxiliary lines. Augmenting diagrams at test time (Muennighoff et al., 2025) by applying segmentation masks (Ravi et al., 2024) or adding auxiliary constructions aligned with the current reasoning step (Murphy et al., 2024; Hu et al., 2024b) offers a practical approach to enhance multi-modal reasoning performance.

6.2 Evaluation challenges in PGPS

In this section, we introduce two major obstacles to fair and informative evaluation of PGPS models: i) existing PGPS benchmarks remain insufficiently comprehensive, and ii) evaluation is typically confined to the final answer, omitting the reasoning.

Incompleteness of PGPS benchmarks Comprehensive PGPS benchmarks should evaluate perception across diverse, realistic diagrams, ensuring that visual processing is essential for solving

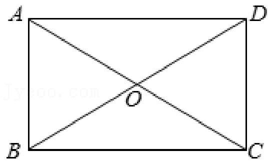
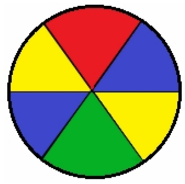
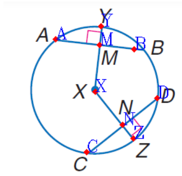
	Example 1	Example 2	Example 3
Diagram			
Question	In the given figure, let's denote the area of triangle AOB as variable x. Find the area of rectangle ABCD in terms of x. **Choices: A: 8 B: 10 C: 12 D: 16**	Based on the image, what is the measure of the interior angle at **vertex A**? Choices: A. 90 degrees B. More than 90 degrees C. Less than 90 degrees D. Cannot be determined	Does the diagram include any line segments that are not perpendicular to each other?
Solution	To determine the area of rectangle ABCD, we can use the fact that triangle AOB is half the area of the rectangle. **Therefore, the area of rectangle ABCD is 2 times the area of triangle AOB, which is 2x.** Hence, the answer is **option B. Answer:D**	Use the properties of the geometric shapes and theorems related to angles to deduce the measure of **the interior angle at vertex A** based on the given image and information. So the **answer is B**	Yes, in the diagram, **line segment YM is not perpendicular to line segment MA**.

Table 1: Examples of hallucinations in the natural-language description datasets annotated with L(V)LM. We visualize the examples from the PGPS datasets, e.g., G-LLaVA and MAVIS, which contain hallucinations in the question or response due to the L(V)LM annotation. We highlight the hallucinations with bold characters.

each problem. However, as shown in Table A6, existing benchmarks do not satisfy these criteria simultaneously. Synthetic diagrams, while scalable, often fail to represent the complexity of real-world scenarios (Zhong et al., 2025; Bates et al., 2025; Wang et al., 2024b), lacking elements such as parallel markers or placeholder objects, as illustrated in Fig. A2. Conversely, manually collected benchmarks better reflect real-world complexity but frequently reuse diagrams from popular PGPS datasets, introducing data leakage and compromising domain generalization evaluations (Hu et al., 2024a; Cao et al., 2024; Chen et al., 2024a).

Even manually curated benchmarks without common PGPS dataset reuse often neglect crucial diagram–text dependencies discussed in Appendix A.2. MathVerse addresses these dependencies explicitly and avoids synthetic diagrams, but still suffers from data leakage, limiting its capability to assess genuine multi-modal reasoning. To overcome these issues, future research should develop synthetic diagram generators that closely replicate real-world complexity or create new datasets that strictly require visual reasoning while rigorously preventing data leakage.

Reasoning-agnostic evaluation Another shortcoming of current PGPS evaluations is their reliance on the final answer alone. Two models that both miss the ground-truth answer can still differ markedly in how sound or informative their intermediate reasoning is, so ignoring the reasoning path obscures meaningful performance gaps. In response, recent studies start to score PGPS methods with an LLM-as-a-judge protocol, asking a LLM to grade the logic and coherence of the generated steps (Zhang et al., 2025a; Sun et al., 2024). Yet this approach suffers from well-known stability issue, i.e., scores fluctuate across runs (Xie et al., 2025; Yamauchi et al., 2025), and raises doubts about the LLM's own factual correctness (Tan et al., 2025). A more rigorous alternative is to convert the reasoning steps generated by the PGPS method into a formal language, e.g., a logic program or script of math assistant (Moura and Ullrich, 2021; Nipkow et al., 2002), and verify each step with a symbolic engine, enabling repeatable, quantitative assessment of the entire reasoning process.

7 Conclusion

In this paper, we examine the tasks, benchmarks, and methods used in existing PGPS research. We summarize the main PGPS approaches as an encoder-decoder architecture, along with the intermediate and output representations utilized across different methods Through the analysis, we outline future research directions addressing current challenges, particularly regarding diagram perception and informative evaluation.

Limitations

In this paper, we primarily survey studies related to PGPS. While our work offers a comprehensive review of the existing PGPS literature, it is limited to two-dimensional geometry. Consequently, we do not address research involving three-dimensional geometry, such as projective and solid geometry, which requires understanding spatial relationships in three-dimensional space.

References

Vincent A.W.M.M. Aleven and Kenneth R. Koedinger. 2002. An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26(2):147–179.

Chris Alvin, Sumit Gulwani, Rupak Majumdar, and Supratik Mukhopadhyay. 2017. Synthesis of problems for shaded area geometry reasoning. In *Artificial Intelligence in Education*, pages 455–458, Cham. Springer International Publishing.

Aida Amini, Saadia Gabriel, Shanchuan Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. 2019. MathQA: Towards interpretable math word problem solving with operation-based formalisms. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2357–2367, Minneapolis, Minnesota. Association for Computational Linguistics.

Mohit Bansal, Kevin Gimpel, and Karen Livescu. 2014. Tailoring continuous word representations for dependency parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 809–815, Baltimore, Maryland. Association for Computational Linguistics.

Averi Bates, Ryan Vavricka, Shane Carleton, Ruosi Shao, and Chongle Pan. 2025. Unified modeling language code generation from diagram images using multimodal large language models. *Machine Learning with Applications*, 20:100660.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Jie Cao and Jing Xiao. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 1511–1520, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Lele Cao, Valentin Buchner, Zineb Senane, and Fangkai Yang. 2024. Introducing GenCeption for multimodal LLM benchmarking: You may bypass annotations. In *Proceedings of the 4th Workshop on Trustworthy Natural Language Processing (TrustNLP 2024)*, pages 196–201, Mexico City, Mexico. Association for Computational Linguistics.

Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. 2022. UniGeo: Unifying geometry logical reasoning via reformulating mathematical expression. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3313–3323, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

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, Online. Association for Computational Linguistics.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024a. Are we on the right way for evaluating large vision-language models? *arXiv preprint arXiv:2403.20330.*

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research.*

Xiaoyu Chen, Dan Song, and Dongming Wang. 2015. Automated generation of geometric theorems from images of diagrams. *Annals of Mathematics and Artificial Intelligence*, 74(3):333–358.

Xuweiyi Chen, Ziqiao Ma, Xuejun Zhang, Sihan Xu, Jianing Yang, David F. Fouhey, Joyce Chai, and Shengyi Qian. 2024b. Multi-object hallucination in vision language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 44393–44418. Curran Associates, Inc.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24185–24198.

Seunghyuk Cho, Zhenyue Qin, Yang Liu, Youngbin Choi, Seungbeom Lee, and Dongwoo Kim. 2025. Geodano: Geometric vlm with domain agnostic vision encoder. *Preprint*, arXiv:2502.11360.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *Preprint*, arXiv:1412.3555.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. 2021. An image is worth 16x16 words: Transformers for image recognition at scale. In *International Conference on Learning Representations*.

R. Fitzpatrick and J.L. Heiberg. 2007. *Euclid's Elements*. University of Texas at Austin, Institute for Fusion Studies Department of Physics.

Daocheng Fu, Zijun Chen, Renqiu Xia, Qi Liu, Yuan Feng, Hongbin Zhou, Renrui Zhang, Shiyang Feng, Peng Gao, Junchi Yan, Botian Shi, Bo Zhang, and Yu Qiao. 2025. Trustgeogen: Scalable and formal-verified data engine for trustworthy multi-modal geometric problem solving. *Preprint*, arXiv:2504.15780.

Wenbin Gan, Xinguo Yu, Ting Zhang, and Mingshu Wang. 2019. Automatically proving plane geometry theorems stated by text and diagram. *International Journal of Pattern Recognition and Artificial Intelligence*, 33(07):1940003.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun 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 Representations*.

Himanshu Gupta, Shreyas Verma, Ujjwala Anantheswaran, Kevin Scaria, Mihir Parmar, Swaroop Mishra, and Chitta Baral. 2024. Polymath: A challenging multi-modal mathematical reasoning benchmark. *Preprint*, arXiv:2410.14702.

Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. 2022. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 16000–16009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yusuke Hirota, Ryo Hachiuma, Chao-Han Huck Yang, and Yuta Nakashima. 2024. From descriptive richness to bias: Unveiling the dark side of generative image caption enrichment. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 17807–17816, Miami, Florida, USA. Association for Computational Linguistics.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Xuhao Hu, Dongrui Liu, Hao Li, Xuanjing Huang, and Jing Shao. 2024a. Vlsbench: Unveiling visual leakage in multimodal safety. *arXiv preprint arXiv:2411.19939*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024b. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kung-Hsiang Huang, Can Qin, Haoyi Qiu, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. 2025. Why vision language models struggle with visual arithmetic? towards enhanced chart and geometry understanding. *Preprint*, arXiv:2502.11492.

Raj Jaiswal, Avinash Anand, and Rajiv Ratn Shah. 2024. Advancing multimodal llms: A focus on geometry problem solving reasoning and sequential scoring. In *Proceedings of the 6th ACM International Conference on Multimedia in Asia*, MMAsia '24, New York, NY, USA. Association for Computing Machinery.

Pengpeng Jian, Fucheng Guo, Cong Pan, Yanli Wang, Yangrui Yang, and Yang Li. 2023a. Interpretable geometry problem solving using improved retinanet and graph convolutional network. *Electronics*, 12(22).

Pengpeng Jian, Fucheng Guo, Yanli Wang, and Yang Li. 2023b. Solving geometry problems via feature learning and contrastive learning of multimodal data. *Computer Modeling in Engineering & Sciences*, 136(2):1707–1728.

Ryo Kamoi, Yusen Zhang, Sarkar Snigdha Sarathi Das, Ranran Haoran Zhang, and Rui Zhang. 2024. Visonlyqa: Large vision language models still struggle with visual perception of geometric information. *arXiv preprint arXiv:2412.00947*.

Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. 2024. Geomverse: A systematic evaluation of large models for geometric reasoning. In *AI for Math Workshop @ ICML 2024*.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. *Preprint*, arXiv:1609.02907.

Terry Koo, Xavier Carreras, and Michael Collins. 2008. Simple semi-supervised dependency parsing. In *Proceedings of ACL-08: HLT*, pages 595–603, Columbus, Ohio. Association for Computational Linguistics.

Zhengfeng Lai, Vasileios Saveris, Chen Chen, Hong-You Chen, Haotian Zhang, Bowen Zhang, Wenze Hu, Juan Lao Tebar, Zhe Gan, Peter Grasch, Meng Cao, and Yinfei Yang. 2025. Revisit large-scale image-caption data in pre-training multimodal foundation models. In *The Thirteenth International Conference on Learning Representations*.

Jimin Lee, Steven-Shine Chen, and Paul Pu Liang. 2025. Interactive sketchpad: A multimodal tutoring system for collaborative, visual problem-solving. In *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, pages 1–14.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Liunian Harold Li*, Pengchuan Zhang*, Haotian Zhang*, Jianwei Yang, Chunyuan Li, Yiwu Zhong, Lijuan Wang, Lu Yuan, Lei Zhang, Jenq-Neng Hwang, Kai-Wei Chang, and Jianfeng Gao. 2022. Grounded language-image pre-training. In *CVPR*.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 292–305, Singapore. Association for Computational Linguistics.

Yunxin Li, Zhenyu Liu, Zitao Li, Xuanyu Zhang, Zhenran Xu, Xinyu Chen, Haoyuan Shi, Shenyuan Jiang, Xintong Wang, Jifang Wang, Shouzheng Huang, Xinping Zhao, Borui Jiang, Lanqing Hong, Longyue Wang, Zhuotao Tian, Baoxing Huai, Wenhan Luo, Weihua Luo, and 3 others. 2025. Perception, reason, think, and plan: A survey on large multimodal reasoning models. *Preprint*, arXiv:2505.04921.

Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2024. LANS: A layout-aware neural solver for plane geometry problem. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2596–2608, Bangkok, Thailand. Association for Computational Linguistics.

Zhenwen Liang, Tianyu Yang, Jipeng Zhang, and Xiangliang Zhang. 2023. UniMath: A foundational and multimodal mathematical reasoner. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7126–7133, Singapore. Association for Computational Linguistics.

Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. 2017a. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. 2017b. Focal loss for dense object detection. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007.

Wang Lin, QingSong Wang, Yueying Feng, Shulei Wang, Tao Jin, Zhou Zhao, Fei Wu, Chang Yao, and Jingyuan Chen. 2025. Non-natural image understanding with advancing frequency-based vision encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 29756–29766.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *Preprint*, arXiv:1907.11692.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. 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, Online. Association for Computational Linguistics.

Huan Ma, Yan Zhu, Changqing Zhang, Peilin Zhao, Baoyuan Wu, Long-Kai Huang, Qinghua Hu, and Bingzhe Wu. 2025. Spurious feature eraser: Stabilizing test-time adaptation for vision-language foundation model. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(18):19296–19304.

Leonardo de Moura and Sebastian Ullrich. 2021. The lean 4 theorem prover and programming language. In *Automated Deduction – CADE 28: 28th International Conference on Automated Deduction, Virtual*

Event, July 12–15, 2021, Proceedings, page 625–635, Berlin, Heidelberg. Springer-Verlag.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. *Preprint*, arXiv:2501.19393.

Logan Murphy, Kaiyu Yang, Jialiang Sun, Zhaoyu Li, Anima Anandkumar, and Xujie Si. 2024. Autoformalizing euclidean geometry. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org.

Maizhen Ning, Qiu-Feng Wang, Kaizhu Huang, and Xiaowei Huang. 2023. A symbolic characters aware model for solving geometry problems. In *Proceedings of the 31st ACM International Conference on Multimedia*, MM '23, page 7767–7775, New York, NY, USA. Association for Computing Machinery.

Tobias Nipkow, Markus Wenzel, and Lawrence C. Paulson. 2002. *Isabelle/HOL: a proof assistant for higher-order logic*. Springer-Verlag, Berlin, Heidelberg.

OpenAI. 2023. Gpt-4v(ision) system card. `https://openai.com/index/gpt-4v-system-card`.

OpenAI. 2025a. Introducing gpt-4.1 in the api. `https://openai.com/index/gpt-4-1/`.

OpenAI. 2025b. Openai o3-mini. `https://openai.com/index/openai-o3-mini`.

Shuai Peng, Di Fu, Yijun Liang, Liangcai Gao, and Zhi Tang. 2023. GeoDRL: A self-learning framework for geometry problem solving using reinforcement learning in deductive reasoning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 13468–13480, Toronto, Canada. Association for Computational Linguistics.

Tianshuo Peng, Mingsheng Li, Hongbin Zhou, Renqiu Xia, Renrui Zhang, Lei Bai, Song Mao, Bin Wang, Conghui He, Aojun Zhou, Botian Shi, Tao Chen, Bo Zhang, and Xiangyu Yue. 2025. Chimera: Improving generalist model with domain-specific experts. *Preprint*, arXiv:2412.05983.

Mengyang Pu, Yaping Huang, Yuming Liu, Qingji Guan, and Haibin Ling. 2022. EDTER: edge detection with transformer. *CoRR*, abs/2203.08566.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. 2021. Learning transferable visual models from natural language supervision. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8748–8763. PMLR.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Nikhila Ravi, Valentin Gabeur, Yuan-Ting Hu, Ronghang Hu, Chaitanya Ryali, Tengyu Ma, Haitham Khedr, Roman Rädle, Chloe Rolland, Laura Gustafson, Eric Mintun, Junting Pan, Kalyan Vasudev Alwala, Nicolas Carion, Chao-Yuan Wu, Ross Girshick, Piotr Dollár, and Christoph Feichtenhofer. 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Steven Ritter, Brendon Towle, R. Charles Murray, Robert G M. Hausmann, and John Connelly. 2010. A cognitive tutor for geometric proof. In *Proceedings of the 10th International Conference on Intelligent Tutoring Systems - Volume Part II*, ITS'10, page 453, Berlin, Heidelberg. Springer-Verlag.

Mrinmaya Sachan, Kumar Dubey, and Eric Xing. 2017. From textbooks to knowledge: A case study in harvesting axiomatic knowledge from textbooks to solve geometry problems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 773–784, Copenhagen, Denmark. Association for Computational Linguistics.

Mrinmaya Sachan and Eric Xing. 2017. Learning to solve geometry problems from natural language demonstrations in textbooks. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 251–261, Vancouver, Canada. Association for Computational Linguistics.

Pranab Sahoo, Prabhash Meharia, Akash Ghosh, Sriparna Saha, Vinija Jain, and Aman Chadha. 2024. A comprehensive survey of hallucination in large language, image, video and audio foundation models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 11709–11724, Miami, Florida, USA. Association for Computational Linguistics.

Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, and Oren Etzioni. 2014. Diagram understanding in geometry questions. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 2831–2838. AAAI Press.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1466–1476, Lisbon, Portugal. Association for Computational Linguistics.

Wenhao Shi, Zhiqiang Hu, Yi Bin, Junhua Liu, Yang Yang, See-Kiong Ng, Lidong Bing, and Roy Ka-Wei Lee. 2024. Math-LLaVA: Bootstrapping mathematical reasoning for multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4663–4680, Miami, Florida, USA. Association for Computational Linguistics.

Kai Sun, Yushi Bai, Ji Qi, Lei Hou, and Juanzi Li. 2024. MM-MATH: Advancing multimodal math evaluation with process evaluation and fine-grained classification. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 1358–1375, Miami, Florida, USA. Association for Computational Linguistics.

Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Yuan Tang, Alejandro Cuadron, Chenguang Wang, Raluca Popa, and Ion Stoica. 2025. Judgebench: A benchmark for evaluating LLM-based judges. In *The Thirteenth International Conference on Learning Representations*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

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.

Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. 2017. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA. Curran Associates Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Vicuna. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90 `https://lmsys.org/blog/2023-03-30-vicuna/`.

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 *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Zhikai Wang, Jiashuo Sun, Wenqi Zhang, Zhiqiang Hu, Xin Li, Fan Wang, and Deli Zhao. 2025. Benchmarking multimodal mathematical reasoning with explicit visual dependency. *Preprint*, arXiv:2504.18589.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. 2024b. Charxiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. 2021. Cvt: Introducing convolutions to vision transformers. *Preprint*, arXiv:2103.15808.

Junda Wu, Zhehao Zhang, Yu Xia, Xintong Li, Zhaoyang Xia, Aaron Chang, Tong Yu, Sungchul Kim, Ryan A. Rossi, Ruiyi Zhang, Subrata Mitra, Dimitris N. Metaxas, Lina Yao, Jingbo Shang, and Julian McAuley. 2024a. Visual prompting in multimodal large language models: A survey. *Preprint*, arXiv:2409.15310.

Wenjun Wu, Lingling Zhang, Jun Liu, Xi Tang, Yaxian Wang, Shaowei Wang, and Qianying Wang. 2024b. E-gps: Explainable geometry problem solving via top-down solver and bottom-up generator. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13828–13837.

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 *The Thirteenth International Conference on Learning Representations*.

Qiujie Xie, Qingqiu Li, Zhuohao Yu, Yuejie Zhang, Yue Zhang, and Linyi Yang. 2025. An empirical analysis of uncertainty in large language model evaluations. 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, Jun Song, and Bo Zheng. 2025. Geosense: Evaluating identification and application of geometric principles in multimodal reasoning. *Preprint*, arXiv:2504.12597.

Shihao Xu, Yiyang Luo, and Wei Shi. 2024. Geo-llava: A large multi-modal model for solving geometry math problems with meta in-context learning. In *Proceedings of the 2nd Workshop on Large Generative Models Meet Multimodal Applications*, LGM3A '24, page 11–15, New York, NY, USA. Association for Computing Machinery.

Yusuke Yamauchi, Taro Yano, and Masafumi Oyamada. 2025. An empirical study of llm-as-a-judge: How design choices impact evaluation reliability. *Preprint*, arXiv:2506.13639.

Yibo Yan, Jiamin Su, Jianxiang He, Fangteng Fu, Xu Zheng, Yuanhuiyi Lyu, Kun Wang, Shen Wang, Qingsong Wen, and Xuming Hu. 2025. A survey of mathematical reasoning in the era of multimodal large language model: Benchmark, method & challenges. *Preprint*, arXiv:2412.11936.

Jianwei Yang, Hao Zhang, Feng Li, Xueyan Zou, Chunyuan Li, and Jianfeng Gao. 2023. Set-of-mark prompting unleashes extraordinary visual grounding in gpt-4v. *Preprint*, arXiv:2310.11441.

Weichen Yu, Ziyan Yang, Shanchuan Lin, Qi Zhao, Jianyi Wang, Liangke Gui, Matt Fredrikson, and Lu Jiang. 2024. Is your text-to-image model robust to caption noise? *Preprint*, arXiv:2412.19531.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*.

Yuan Yuan, Zhaojian Li, and Bin Zhao. 2025. A survey of multimodal learning: Methods, applications, and future. *ACM Comput. Surv.*, 57(7).

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, Cong Wei, Botao Yu, Ruibin Yuan, Renliang Sun, Ming Yin, Boyuan Zheng, Zhenzhu Yang, Yibo Liu, Wenhao Huang, and 3 others. 2024. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9556–9567.

Xiaohua Zhai, Basil Mustafa, Alexander Kolesnikov, and Lucas Beyer. 2023. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 11975–11986.

Jiaxin Zhang and Yashar Moshfeghi. 2024. GOLD: Geometry problem solver with natural language description. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 263–278, Mexico City, Mexico. Association for Computational Linguistics.

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, IJCAI-22*, pages 1636–1643. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Ming-Liang Zhang, Fei yin, and Cheng-Lin Liu. 2023. A multi-modal neural geometric solver with textual clauses parsed from diagram. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI-23*, pages 3374–3382. International Joint Conferences on Artificial Intelligence Organization. Main Track.

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. 2025a. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024*, pages 169–186, Cham. Springer Nature Switzerland.

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Shanghang Zhang, Peng Gao, and Hongsheng Li. 2025b. MAVIS: Mathematical visual instruction tuning with an automatic data engine. In *The Thirteenth International Conference on Learning Representations*.

Shan Zhang, Aotian Chen, Yanpeng Sun, Jindong Gu, Yi-Yu Zheng, Piotr Koniusz, Kai Zou, Anton van den Hengel, and Yuan Xue. 2025c. Open eyes, then reason: Fine-grained visual mathematical understanding in mllms. *Preprint*, arXiv:2501.06430.

Shan Zhang, Aotian Chen, Yanpeng Sun, Jindong Gu, Yi-Yu Zheng, Piotr Koniusz, Kai Zou, Anton van den Hengel, and Yuan Xue. 2025d. Primitive vision: Improving diagram understanding in MLLMs. In *Forty-second International Conference on Machine Learning*.

Xiaokai Zhang, Na Zhu, Yiming He, Jia Zou, Cheng Qin, Yang Li, and Tuo Leng. 2024a. Fgeo-sss: A search-based symbolic solver for human-like automated geometric reasoning. *Symmetry*, 16(4).

Xiaokai Zhang, Na Zhu, Cheng Qin, Yang Li, Zhenbing Zeng, and Tuo Leng. 2024b. Fgeo-hypergnet: Geometric problem solving integrating formal symbolic system and hypergraph neural network. *Preprint*, arXiv:2402.11461.

Xiaokai Zhang, Na Zhu, Cheng Qin, LI Yang, 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*.

Xinsong Zhang, Yarong Zeng, Xinting Huang, Hu Hu, Runquan Xie, Han Hu, and Zhanhui Kang. 2025e. Low-hallucination synthetic captions for large-scale vision-language model pre-training. *Preprint*, arXiv:2504.13123.

Zeren Zhang, Jo-Ku Cheng, Jingyang Deng, Lu Tian, Jinwen Ma, Ziran Qin, Xiaokai Zhang, Na Zhu, and Tuo Leng. 2025f. Diagram formalization enhanced multi-modal geometry problem solver. In *ICASSP 2025 - 2025 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5.

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. *Preprint*, arXiv:2503.05543.

Ling Zhong, Yujing Lu, Jing Yang, Weiming Li, Peng Wei, Yongheng Wang, Manni Duan, and Qing Zhang. 2025. Domaincqa: Crafting expert-level qa from domain-specific charts. *Preprint*, arXiv:2503.19498.

Na Zhu, Xiaokai Zhang, Qike Huang, Fangzhen Zhu, Zhenbing Zeng, and Tuo Leng. 2025. Fgeo-parser: Autoformalization and solution of plane geometric problems. *Symmetry*, 17(1).

Chengke Zou, Xingang Guo, Rui Yang, Junyu Zhang, Bin Hu, and Huan Zhang. 2025. Dynamath: A dynamic visual benchmark for evaluating mathematical reasoning robustness of vision language models. In *The Thirteenth International Conference on Learning Representations*.

Jia Zou, Xiaokai Zhang, Yiming He, Na Zhu, and Tuo Leng. 2024. Fgeo-drl: Deductive reasoning for geometric problems through deep reinforcement learning. *Symmetry*, 16(4).

A Additional axis on benchmark dataset

A.1 Reasoning complexity

We discuss the mathematical concepts and difficulty levels encountered in plane geometry problems used by existing benchmarks and datasets. Typical plane geometry problems involve calculating specific angle measures, arc measures, segment or arc lengths, and areas of designated regions. Computing these numerical values generally requires basic arithmetic and root operations, but may also involve trigonometric functions, such as sine and cosine. Although no standardized quantitative method currently exists to measure problem difficulty, problems can be qualitatively categorized according to their original sources, such as SAT exams (Seo et al., 2015; Sachan et al., 2017; Sachan and Xing, 2017), plane geometry curricula from grades 6–12 American (Lu et al., 2021; Zhang et al., 2023; Sun et al., 2024) or Chinese school (Chen et al., 2021; Cao and Xiao, 2022; Xu et al., 2025), college-level mathematics (Yue et al., 2024), or mathematics competitions, e.g., AMC 8, 10, and 12 (Wang et al., 2024a).

A.2 Diagram-text redundancy

To serve as rigorous benchmarks and datasets for multi-modal reasoning, the collected problems must require simultaneous interpretation of both diagrams and accompanying textual descriptions. By contrast, PGPS problems that can be solved using the text alone cannot effectively evaluate the diagram-text integration capability of PGPS methods. Nevertheless, many existing benchmarks and datasets still contain such problems, thereby inadequately assessing the perception abilities of PGPS methods (Zhang et al., 2025a).

Recent PGPS benchmarks have addressed this limitation by explicitly annotating problems with modality-specific information and subsequently removing redundant textual cues (Lu et al., 2021; Zhang et al., 2023, 2025a). Several benchmarks provide multiple variants of each problem for more fine-grained analysis of diagram-text dependency. For instance, MathVerse (Zhang et al., 2025a) relocates selected information from the text into the diagram, while DynaMath (Zou et al., 2025) generates alternative diagrams and corresponding answers based on a single textual description. Thus, failure to solve certain variants of the same problem indicates that the model is not genuinely utilizing the diagram.

A.3 Data collection methods

We summarize three data collection methods mainly used to construct PGPS datasets.

Human annotation In most cases, datasets are constructed through human annotation based on problems sourced from textbooks, internet sites, or similar resources (Seo et al., 2015; Chen et al., 2021; Lu et al., 2021, 2024; Sun et al., 2024; Yue et al., 2024). This involves manually collecting problems and having human annotators provide the corresponding outputs. Additionally, some studies apply text augmentation techniques, such as back-translation, to diversify the text style and enrich the dataset (Cao and Xiao, 2022).

Synthetic annotation Several PGPS studies create synthetic benchmarks and datasets instead of collecting problems from textbooks or the internet. These studies typically implement synthetic engines to generate diagrams and corresponding structured information. For example, synthetic engines can generate captions containing the geometric information explicitly present in diagrams (Zhang et al., 2025b), or use symbolic reasoning engines to produce reasoning steps that derive the stated goals from diagram-text pairs (Zhang et al., 2025b; Kazemi et al., 2024; Fu et al., 2025). Such synthetic approaches offer clear advantages, including easy scalability and guaranteed completeness of annotations. However, they often struggle to produce sufficiently diverse diagrams that accurately reflect the real-world problems. This limitation is further discussed in §6.2.

L(V)LM-assisted annotation For certain datasets, particularly those with natural-language description as the output representation, LLMs and VLMs such as GPT (Brown et al., 2020) or GPT-4V (OpenAI, 2023) are employed for dataset construction. Specifically, problems and solutions are sourced from datasets like GeoQA+, UniGeo, or PGPS9K, and GPT or GPT-4V are used to augment these by generating multiple problem-solution pairs for a given problem scenario (Gao et al., 2025; Shi et al., 2024; Zhang et al., 2025b). Alternatively, some studies apply the same process to synthetic data, such as diagram-caption pairs generated by a synthetic data engine (Zhang et al., 2025b; Kazemi et al., 2024). However, due to the poor perception ability of GPT-4V, several hallucinations occur in the augmented datasets. We discuss more details about the challenge in §6.1.

B PGPS Methods

B.1 Summary of PGPS methods

We summarize the PGPS methods in terms of the encoder, intermediate representation, decoder, and the output format at Table A1. We also summarize the strengths and weaknesses of each intermediate representation, decoder, and output format in Table A2, providing a practical guide for selecting an appropriate PGPS configuration.

B.2 Comparison of the PGPS methods

We provide a quantitative comparison of existing PGPS methods. Table A3 summarizes the top-10 accuracy of PGPS methods that do not adhere to the E2-I2-D3-O3 pipeline. We report aggregated results across six widely used benchmarks, i.e., GEOS, GeoQA, Geometry3K, UniGeo, PGPS9K, and formalgeo-7K. From these results, we observe that methods employing the E2-I2-D3-O2 configuration, such as LANS and GeoX, match or surpass previously known best performances on most benchmarks.

Table A4 specifically considers methods that adopt the E2-I2-D3-O3 framework, reporting the top-1 accuracy on GeoQA, MathVista, and MathVerse. Among these methods, SVE-Math achieves the highest accuracy on GeoQA across all PGPS methods and sets a new best performance on MathVerse.

Model	GeoQA	MathVista	MathVerse
Math-LLaVA	–	46.60	19.00
MAVIS	66.70	–	27.50
G-LLaVA	64.20	–	–
Chimera-8B	–	**64.90**	–
Geo-LLaVA	65.25	–	–
SVE-Math	**79.60**	50.40	**31.40**

Table A4: Quantitative analysis of the PGPS methods which output natural language description. We compare the top-1 accuracy of the E2-I2-D3-O3 PGPS methods on GeoQA, MathVista, and MathVerse.

C Rule-based Components

C.1 Rule-based encoder

Early PGPS methods relied on classical computer vision and text parsing techniques to independently extract geometric primitives and relations from diagrams and text, merging them into formal-language descriptions. Most studies (Seo et al., 2015; Sachan et al., 2017; Sachan and Xing, 2017; Alvin et al., 2017; Gan et al., 2019) employed rule-based diagram parsers, notably HoughGeo (Chen et al., 2015) or G-Aligner (Seo et al., 2014), which preprocess diagrams to detect geometric primitives using classical computer vision techniques, e.g., Gaussian blur and Hough transforms, and then match detected primitives to literal sets using either handcrafted rules or optimization. For textual extraction, many approaches (Wu et al., 2024b; Zhao et al., 2025; Peng et al., 2023; Zhang et al., 2024b; Jian et al., 2023a; Zou et al., 2024) adopted the InterGPS (Lu et al., 2021) parser, a rule-based method utilizing regular expressions, which is reliable and effective even with limited data.

C.2 Rule-based axiomatic decoder

Several methods that operate on formal-language descriptions determine the required theorem sequence with a rule-based decoder. GEOS++ (Sachan et al., 2017) employs an exhaustive brute-force search to locate a sequence of theorems whose application yields the target predicate. GeoShader (Alvin et al., 2017) specifies a deterministic set of composition rules that directly selects the relevant theorems without search. GEOS-OS (Sachan and Xing, 2017) trains a log-linear model to assign probabilities to candidate theorems and then performs beam search, returning the highest-scoring theorem sequence.

D Takeaways

While we primarily present challenges, our survey indeed reveals several critical insights that can guide future PGPS research. Firstly, we observe that relatively little attention has been paid to optimizing the vision encoder architectures used in PGPS research. Despite recent evidence from multiple studies indicating that ViTs are not suitable for geometric diagrams (Lin et al., 2025; Zhang et al., 2025d), most current approaches continue to prioritize improvements in synthetic data generation rather than revisiting architectural choices. A practical guideline is thus to systematically explore and evaluate alternative or modified encoder architectures tailored specifically for geometric diagrams.

Secondly, our review highlights significant research gaps regarding unexplored combinations within the E-I-D-O framework. For example, the E2-I1-D2-O3 combination has been addressed by

Encoder	Intermediate	Decoder	Output	Methods
E1	I1	–	–	HoughGeo (Chen et al., 2015), G-Aligner (Seo et al., 2014), GEOS (Seo et al., 2015)
E2	I1	–	–	PGDPNet (Zhang et al., 2022), FGeo–Parser (Zhu et al., 2025)
E1	I1	D1	O1	GEOS++ (Sachan et al., 2017), GEOS–OS (Sachan and Xing, 2017), GeoShader (Alvin et al., 2017), S2 (Gan et al., 2019)
E2	I1	D2	O1	FGeo–HyperGNet (Zhang et al., 2024b), GCN–GPS (Jian et al., 2023a), GeoDRL (Peng et al., 2023)
E2	I1	D3	O1	InterGPS (Lu et al., 2021), E–GPS (Wu et al., 2024b), Pi–GPS (Zhao et al., 2025), FGeo–DRL (Zou et al., 2024)
E2	I1	D1	O1	FGeo–SSS (Zhang et al., 2024a)
E2	I1	D2	O3	GOLD (Zhang and Moshfeghi, 2024)
E2	I2	D3	O2	NGS (Chen et al., 2021), DPE–NGS (Cao and Xiao, 2022), Geoformer (Chen et al., 2022), PGPSNet (Zhang et al., 2023), SCA–GPS (Ning et al., 2023), UniMath (Liang et al., 2023), FLCL–GPS (Jian et al., 2023b), LANS (Li et al., 2024), GeoX (Xia et al., 2025), GeoDANO (Cho et al., 2025)
E2	I2	D3	O3	Math–LLaVA (Shi et al., 2024), Visual SKETCH-PAD (Hu et al., 2024b), MAVIS (Zhang et al., 2025b), G–LLaVA (Gao et al., 2025), DFE–GPS (Zhang et al., 2025f), Chimera (Peng et al., 2025), Geo–LLaVA (Xu et al., 2024), SVE-Math (Zhang et al., 2025c)

Table A1: Categorization of existing PGPS methods. We categorize the PGPS methods based on their encoder, intermediate representation, decoder, and output format. The symbols come from Fig. 3.

only a single study, and the combination of GNN-based decoders (D2) with logic program outputs (O2) remains unexplored. A clear recommendation for future work, therefore, is to investigate these neglected combinations, as they may offer substantial performance gains or novel insights.

Category	Variant	Pros / Cons
Encoder	Rule-based	(+) Deterministic, fully transparent inference (–) Sensitive to variations in diagram styles and noises
	Neural network	(+) Robust to variations in diagram styles and noises (–) Lack interpretability in problem processing
Intermediate representation	Formal-language description	(+) Human-readable, semantically explicit (+) Compatible with theorem provers or symbolic engines (–) Introduces a separate parsing stage before reasoning (–) Requires costly expert annotation of formal expression
	Embedding vector	(+) Enables single-stage, end-to-end optimization (+) Enables seamless fusion of image and text features (–) Embeddings lack interpretability (–) Incompatible with rule-based symbolic engines
Decoder	Rule-based	(+) Deterministic, fully transparent inference (–) Coverage confined to handcrafted rule set (–) Combinatorial rule growth hampers scalability
	GNN-based	(+) Naturally exploits edge-level relational structure (+) Invariant to the permutation of predicates (–) Requires detailed graph annotations for each problem
	Seq-to-seq	(+) Flexible, domain-agnostic architecture (+) Fully differentiable, end-to-end optimization (–) Data- and compute-intensive to reach strong performance
Output representation	Theorem sequence	(+) Human-readable, fine-grained proof trace (+) Each step verifiable by a proof assistant (–) Relies on a comprehensive predicate catalogue
	Logic program	(+) Compact and fewer rule types than theorem sequence (+) Directly executable by a logic solver (–) Requires ground numeric constants
	Natural-language description	(+) Easily understandable (+) Expressively flexible (–) Unconstrained form can induce hallucination

Table A2: Strengths and weaknesses of each component in the PGPS pipeline. We summarize the strengths and weaknesses of encoder, intermediate representation, decoder, and output representation, offering practical guidance for assembling a PGPS pipeline.

Pipeline	**Model**	**GEOS**	**GeoQA**	**Geometry3K**	**UniGeo**	**PGPS9K**	**formalgeo7k**
E1-I1-D1-O1	GEOS	32.00	–	–	–	–	–
	GEOS++	44.00	–	–	–	–	–
	GEOS-OS	**58.00**	–	–	–	–	–
E2-I1-D2-O1	FGeo-HyperGNet	–	–	–	–	–	85.53
	GCN-GPS	–	–	56.10	–	–	–
	GeoDRL	–	–	68.40	–	66.70	–
E2-I1-D3-O1	InterGPS	–	–	57.50	–	–	–
	E-GPS	–	–	67.90	–	–	–
	Pi-GPS	–	–	77.80	–	69.80	–
	FGeo-DRL	–	–	–	–	–	**86.40**
E2-I1-D1-O1	FGeo-SSS	–	–	–	–	–	39.71
E2-I2-D3-O2	NGS	–	56.90	58.80	47.40	46.10	–
	DPE-NGS	–	62.70	–	–	–	–
	Geoformer	–	60.30	59.30	56.40	47.30	–
	PGPSNet	–	–	77.90	–	70.40	–
	SCA-GPS	–	64.10	76.70	–	–	–
	FLCL-GPS	–	61.80	–	–	–	–
	LANS	–	–	**82.30**	–	**73.80**	–
	GeoX	–	**69.00**	72.50	**99.50**	63.30	–

Table A3: Quantitative analysis of PGPS methods that output non-natural-language descriptions. We compare the top-10 accuracy of PGPS methods that produce non-natural-language descriptions across the six most widely used PGPS benchmarks.

E Challenges and Future Directions

E.1 Error analysis on wrong responses

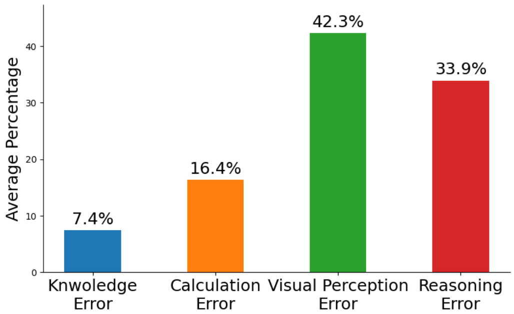


Figure A1: Error analysis on the response of GPT-4V on MathVerse. We analyze the responses of GPT-4V on MathVerse, reporting the average percentage for each type of error across five MathVerse variants, Text Dominant, Text Lite, Vision Intensive, Vision Dominant, and Vision Only, which are reported in MathVerse. Our analysis indicates that incorrect answers predominantly result from visual perception and reasoning errors.

E.2 Examples of perception hallucinations

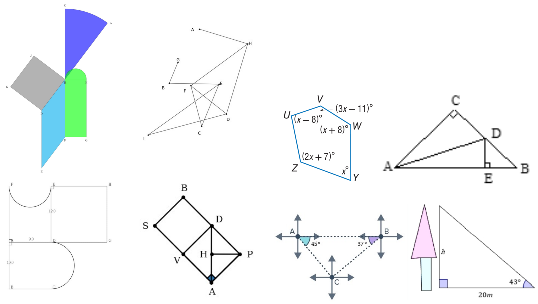
We provide examples of hallucinated responses by GPT-4.1 in Table A5.

E.3 Comprehensivity of current PGPS benchmarks

Methods	Realistic styles of diagrams	No data leakage	Diagram–text interdependence
MMMU	◯	◯	×
Math–V	◯	◯	×
MathVista	◯	×	×
MathVerse	◯	×	◯
GeomVerse	×	◯	×
VisOnlyQA	×	◯	◯
MM–Math	◯	◯	×
GeoEval	×	×	×
DynaMath	◯	×	◯

Table A6: Comprehensivity across existing PGPS benchmarks. The table summarizes benchmark features in terms of realistic diagram styles, absence of data leakage, and consideration of diagram-text interdependence.

E.4 Synthetic and real-world geometric diagrams



(a) Synthetic diagrams (b) Real-world diagrams

Figure A2: Visualization of the synthetic and real-world geometric diagrams. We compare the geometric diagrams, which are synthetically generated or manually collected from existing sources. The synthetic diagrams are from GeomVerse, VisOnlyQA, MAVIS, and GeoDANO. The real-world diagrams are from MathVerse.

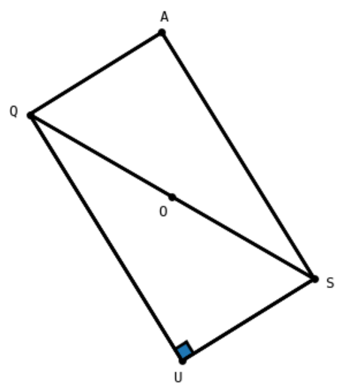
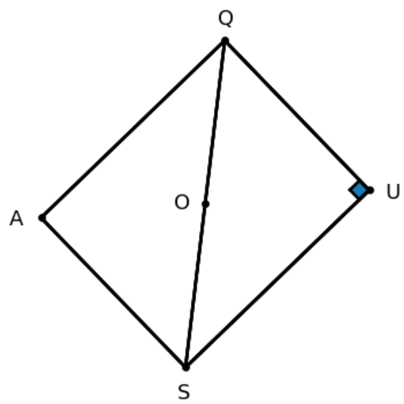
	Example 1	Example 2
Input prompt	In the given diagram, extend the segment SU to form an infinite straight line. This line divides the two-dimensional plane into two half-planes. The points Q and A are also shown in the diagram. Are the points Q and A located on the same side of the line SU?	
Input diagram		
Response	To determine if points Q and A are on the same side of the line SU, observe the diagram: extending SU clearly shows that point Q is on one side of the line, **while point A is on the opposite side**. The quadrilateral QASU suggests **A and Q are separated by the segment SU itself.**	To determine whether points Q and A are on the same side of the line SU, we observe the diagram. The segment SU is extended to form a straight line. Point Q is above SU, **while point A is on the opposite side**, to the left of SU. Therefore, **the points Q and A are on opposite sides of line SU.**

Table A5: Examples of hallucinations occured in GPT-4.1 responses. We analyzed the diagram perception performance of GPT-4.1 (OpenAI, 2025a), specifically determining whether two points are on the same side of a given line. We generated 100 problems using the synthetic data engine from GeoDANO (Cho et al., 2025) and tested them with GPT-4.1, observing a low accuracy of 59%. The examples above illustrate cases where GPT-4.1 hallucinated and provided incorrect answers. Hallucinated parts are highlighted in bold.