Pi-GPS: Enhancing Geometry Problem Solving by Unleashing the Power of Diagrammatic Information

Supplementary Material

1. Computational Cost Analysis

We conduct a computational cost analysis in Table 1, comparing our method with GeoDRL, whose implementation is publicly available. Both experiments are conducted under the same conditions. The results show that our method incurs only a slightly higher time cost overall. While the introduction of the two additional modules does increase some resource usage, it also leads to a noticeable reduction in the runtime of the symbolic solver. We believe this trade-off is beneficial, especially considering the accuracy gain.

	Time / s					Accuracy
	Parsing	Post Parsing		Solver	Total	
GeoDRL	0.67	Struct GLG: 1.11		1.71	3.49	68.4
Pi-GPS	0.67	Text dis.: 1.52	Thm pred.: 1.57	0.85	4.16	77.8

Table 1. Computational cost analysis.

2. Implementation Details

Prompt for Unspecified Points

According to the question:{problem_text}, what is the {unknown_form} in {text_formal} refer to in the diagram {diagram}? Answer me only in capital letters(eg. ABCD) no more information should be answer. If there is no answer, please answer me $ .If there are multiple {unknown_form} in my ask, please only answer me the first one! Represent a circle only by its center point,eg. Circle(O) Represent a Line by its end point,eg. Line(AB) Represent a Angle by three letters,eg. Angle(ABC)

Figure 1. Prompt for points

Parsers. For text parsing, we adopt the rule-based text parser from Inter-GPS, which accurately converts the problem text into formal languages using rule templates. For diagram parsing, we trained PGDP-Net on the PGDP-5K dataset for 40,000 steps, with a learning rate of 5e-4 and a batch size of 12, following the settings established by the original authors. Training 10,000 steps requires approximately 3 hours on a single Nvidia H800 GPU. In this work, we independently trained PGDP for two primary reasons. First, the pretrained weights for PGDP are no longer publicly accessible. Second, during our reproduction of GeoDRL, we identified inaccuracies in the PGDP outputs that resulted in several unresolved issues, a finding that deviates from the outcomes reported in the original PGDP publication.

Rectifier. For the Rectifier in Text Disambiguation module, we have designed three distinct prompts for the MLLM to address various cases.

Prompt for Unspecified Shapes

Your task is to provide the correct formal language that describes the problem statement based on the text and diagram of a geometry math problem. For example, I will ask what is the Shape($) in Find(AreaOf(Shape($))) refers to, and according to the text and diagram in the problem, Shape($) refers to triangle ABC, so you need to output Triangle(A,B,C) No additional information should be output. If there are multiple Shape($) in my ask, please only answer me the first one.If you can not answer, please answer me Shape($)
the forms of the shapes are as follows:
- **Triangle Triangle(A,B,C)**
- **Parallelogram Parallelogram(A,B,C,D)**
- **Square Square(A,B,C,D)**
- **Rectangle Rectangle(A,B,C,D)**
- **Rhombus Rhombus(A,B,C,D)**
- **Trapezoid Trapezoid(A,B,C,D)**
- **Kite Kite(A,B,C,D)**
- **Pentagon Pentagon(A,B,C,D,E)**
- **Hexagon Hexagon(A,B,C,D,E,F)**
- **Heptagon Heptagon(A,B,C,D,E,F,G)**
- **Octagon Octagon(A,B,C,D,E,F,G,H)**
- **Circle Circle(A)**

According to the question: {problem_text}, what is the first Shape($) in {text_formal} refers to in the diagram {diagram}

Figure 2. Prompt for Unspecified shapes

In the case of dealing with unspecified points, we provide the MLLM with the **problem text**, the parsed **unknown form**, the original **text formal**, and the corresponding **diagram**.Since in this process the MLLM is only required to output the vertices of the figure rather than a full formal description, there is no need to provide a prompt with formal formatting rules. For the vast majority of closed figures, the geometric representation of vertices is unequivocal. However, in instances where potential ambiguities may arise—such as with circles, line segments, or angles—illustrative examples are provided to ensure that the outputs of the MLLM remain consistent with our intended specifications.The supplied prompt is illustrated in Figure 1.

Prompt for Unspecified Areas

Your task is to provide the correct formal language that describes the problem statement based on the text and diagram of a geometry math problem. Such as Shaped($) area refers to triangle ABC, so you need to output Triangle(A,B,C) if the shaded area is not a complete shape and it is two shape combined, you can use Minus(Shape(),Shape()) or Add(Shape(),Shape()) to discribe the area.such as the shaded area is obtained by subtracting the area of the circle D from the area of the triangle AEG, you should answer me Minus(Triangle(A,E,G), Circle(D)) you can Combine the two expressions together, such as Minus(Rectangle(A,B,C,D), Add(Circle(D),Circle(E))) No additional information should be output.if there are multiple Shaped($) in my ask, please only answer me the first one. if you can not answer, please just answer me Shaded($)
the forms of the shapes are as follows:
Triangle Triangle(A,B,C)
Parallelogram Parallelogram(A,B,C,D)
Square Square(A,B,C,D)
Rectangle Rectangle(A,B,C,D)
Rhombus Rhombus(A,B,C,D)
Trapezoid Trapezoid(A,B,C,D)
Kite Kite(A,B,C,D)
Pentagon Pentagon(A,B,C,D,E)
Hexagon Hexagon(A,B,C,D,E,F)
Heptagon Heptagon(A,B,C,D,E,F,G)
Octagon Octagon(A,B,C,D,E,F,G,H)
Sector Sector(O,A,B)
Circle Circle(A)
According to the question:{problem_text}, what is the first Shaded($) in {text_formal} refers to in the diagram{diagram}

Figure 3. Prompt for unspecified areas

In the case of dealing with unspecified shapes, we provide the MLLM with the **problem text**, the original **text formal**, and the corresponding **diagram**.To ensure that the MLLM comprehends the task with greater precision, we have provided a complete process of a simple example within the prompt, thereby establishing a one-shot task paradigm.Since the MLLM is required to independently generate the formal language that represents the shape at this stage, a comprehensive set of rules for describing the shape in formal language is included in the prompt. The supplied prompt is illustrated in Figure 2.

Prompt for LLM Theorem Prediction

You will try to solve a geometry problem. Instead of giving me the answer directly, you should tell me the sequence of the theorems in numbers that should be used to solve this plane geometry problem.For example, if you use 2: thales theorem, 4:parallel lines theorem and 5: triangle anglesum theorem, you just need to answer [2,4,5].You can not answer any other information.Solve the problem using as few theorems as possible!But ensure that the problem can definitely be solved after applying these theorems.
here are the theorems list you can use:
1:circle_definition
2: thales_theorem
3: inscribed_angle_theorem
4:parallel_lines_theorem
5: triangle_anglesum_theorem
6:isosceles_triangle_theorem_side
7: isosceles_triangle_theorem_angle
8:equilateral_triangle_theorem
9: pythagoras_theorem
10: triangle_center_of_gravity_theorem
11: congruent_triangles_proving_theorem
12: congruent_triangles_theorem
13: law_of_sines
14: tangent_secant_theorem
15: chord_theorem
16: angle_bisector_theorem
17: similar_triangle_proving_theorem
18: similar_triangle_theorem
19: similar_polygon_theorem
20: median_line_theorem
21: area_equation_theorem
22: polygon_anglesum_theorem
23: law_of_cosines
Here is the question and the formal language to discribe its diagram.
Question:{problem_text},{Disambiguated Formal}

Figure 4. Prompt for LLM Theorem prediction

Apart from the introduction of the new designed Minus(Shape($),(Shape($)) and Add(Shape($),(Shape($)) formulas to represent shaded geometric areas, the approach for handling unspecified areas is fundamentally identical to that for addressing unspecified shapes.We provide the MLLM with the **problem text**, the original **text formal**, and the corresponding **diagram**. To enhance the MLLM's understanding of the newly introduced Minus and Add formulas, we employ two examples to illustrate their usage in both a simple one-time combination and in multiple complex combinations, thereby establishing a few-shot learning paradigm for the entire process.The supplied prompt is illustrated in Figure 3.

Verifier. Incorporating diagram heuristics, the verifier first identifies when the MLLM output violates geometric constraints and prompts the MLLM to correct the error, allowing up to three attempts. If, after three attempts, the MLLM still fails to produce an answer consistent with the geometric constraints, the process is terminated.Furthermore, if the verifier detects a vertex-ordering error (e.g. the MLLM erroneously outputs Pentagon(A, B, D, E, C) instead of Pentagon(A, B, C, D, E)), it directly corrects the ordering without requesting a new answer from the MLLM.

Theorem Predictor. During the theorem prediction stage, the LLM is provided with both the original problem text and its disambiguated formal representation to ensure accurate problem formulation. In parallel, a complete theorem list with numbering and definitions is supplied for the LLM's selection. To ensure accurate task comprehension and correct output formatting, a comprehensive example is included in the prompt. The specific prompt content is illustrated in Figure 4.

3. Experiment Details

Figure 5 illustrates the standardized test prompt used to elicit responses from various LLMs and MLLMs evaluated on the Geometry3K and PGPS9K datasets. This approach enabled a robust comparison of the models' geometric reasoning and spatial problem-solving capabilities. The uniform testing framework facilitated the identification of each model's strengths and limitations and provided insights for future research.

Prompt for LLM/MLLM Direct Solve

You will try to solve a plane geometry problem with the formal language to discribe it. Directly answer with the numerical answer. Do not output any process.
question:{problem_text},{Disambiguated Formal/Raw Formal} ,diagram{ with/without diagram}

Figure 5. Prompt for LLM/MLLM Direct Solve