LANS: A Layout-Aware Neural Solver for Plane Geometry Problem

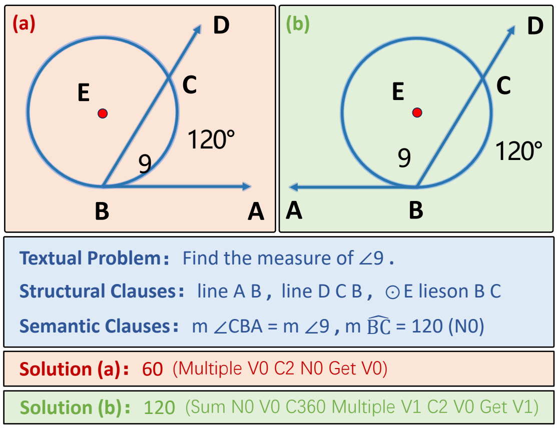
Anonymous ACL submission

Abstract

Geometry problem solving (GPS) is a challenging mathematical reasoning task requiring multi-modal understanding, fusion, and reasoning. Existing neural solvers take GPS as a vision-language task but are short in the representation of geometry diagrams that carry rich and complex layout information. In this paper, we propose a layout-aware neural solver named LANS, integrated with two new modules: multimodal layout-aware pre-trained language module (MLA-PLM) and layout-aware fusion attention (LA-FA). MLA-PLM adopts structural-semantic pre-training (SSP) to implement global relationship modeling, and point-match pre-training (PMP) to achieve alignment between visual points and textual points. LA-FA employs a layout-aware attention mask to realize point-guided cross-modal fusion for further boosting layout awareness of LANS. Extensive experiments on datasets Geometry3K and PGPS9K validate the effectiveness of the layout-aware modules and superior problem-solving performance of our LANS solver, over existing symbolic and neural solvers. The code will be made public available soon.

1 Introduction

Automatic geometry problem solving (GPS) is a long-standing and challenging research topic in both computer vision and natural language processing communities (Bobrow, 1968; Chou et al., 1996; Seo et al., 2015). Each geometry problem consists of a geometry diagram and a textual problem in different modal forms, complementing each other. GPS necessitates comprehensive mathematical reasoning and multi-modal understanding, making it a pivotal testbed for evaluating the high-level multimodal reasoning ability of artificial intelligence. Past research works of GPS were mainly focused on *symbolic solvers* (Seo et al., 2015; Sachan and Xing, 2017; Lu et al., 2021), which are criticized in respect of complex rules and poor adaptability. With the development of deep learning, *neural solvers* (Chen et al., 2021, 2022; Zhang et al., 2023, 2024), treating GPS as a special vision-language reasoning task, have attracted dominant attention recently.



Textual Problem: Find the measure of ∠9 .

Structural Clauses: line A B , line D C B , ⊙E lieson B C

Semantic Clauses: m ∠CBA = m ∠9 , m $\widehat{BC}$ = 120 (N0)

Solution (a): 60 (Multiple V0 C2 N0 Get V0)

Solution (b): 120 (Sum N0 V0 C360 Multiple V1 C2 V0 Get V1)

Figure 1: Examples of plane geometry problems. The geometry diagrams (a) and (b) share the same textual problem, structural clauses, and semantic clauses but have different solutions, where structural clauses and semantic clauses are parsed from diagrams. Layout information plays a crucial role in this situation.

Layout information is typically defined as positional coordinates of elements such as text, paragraphs, tables, and figures within images (Xu et al., 2020; Gupta et al., 2021). Supplying layout details for elements in document images facilitates parsing reading sequences, executing information extraction, and enhancing document comprehension (Appalaraju et al., 2021; Wang et al., 2021; Hong et al., 2015). In the layout of geometric diagram, the coordinate positions of geometric points and symbols play a crucial role in understanding the elements within geometric diagrams. For example, the coordinate positions of geometric symbols "A" as shown in Figure 1, determine which geometric points are named A, while the coordinate position of the non-geometric symbol "120°" determines the numerical assignment of $\angle ABD$ instead of other angle.

Despite considerable efforts devoted to constructing proficiently crafted representations for geometric diagrams, the explicit fusion of positional information into geometric diagrams remains unexplored. Existing neural solvers have adopted different diagram representation schemes, such as *feature maps* (Chen et al., 2021; Cao and Xiao, 2022; Ning et al., 2023), *image patches* (Chen et al., 2022; Ning et al., 2023) and *textual clauses* (Lu et al., 2021; Zhang et al., 2023). For methods based on *image patches* and *feature maps*, several representative geometric problem solvers have employed extensive pre-training strategies, such as jigsaw location prediction (Chen et al., 2021), geometry elements prediction (Chen et al., 2021), masked image modeling (Ning et al., 2023), and character alignment (Ning et al., 2023), to bridge the gap between geometric and natural scene images (Anderson et al., 2018; Yu et al., 2019; Ding et al., 2022). Although rough image pre-training methods have achieved some effectiveness, they often fail to capture finer-grained details. Conversely, methods based on *text clauses* extract the crucial structural and semantic information of geometric problems in the form of clauses. Currently, clause-based approaches yield superior inference results through clause-based deductive reasoning (Lu et al., 2021) or clause pre-training (Zhang et al., 2023). We attribute this to the structured nature of clauses, which makes them more adept at capturing structural information in geometric problems. For example, The structural clause “line B C D" describes a structural relationship that points “B", “C" and “D" lie on one line in order. The semantic clause “m $\widehat{\mathrm{BC}}$ = 120" illustrates a semantic relationship for the degree of arc “$\widehat{\mathrm{BC}}$" and text “120°".

Although the textual clauses are capable of capturing the primary layout relationships within the images, they lose significant spatial information during the conversion process of diagram parsing (Lu et al., 2023; Trinh et al., 2024). They cannot distinguish the geometry diagram (a) and (b) displayed in Figure 1 because of loss of position information. For example, “ CBA" in Figure 1(a) and (b) need the spatial relationship to determine whether it is acute or obtuse. The lack of positional indicators for geometry elements (such as "A," "B," etc.) makes it challenging for neural solvers based on text clauses to distinguish between these ambiguous scenarios.

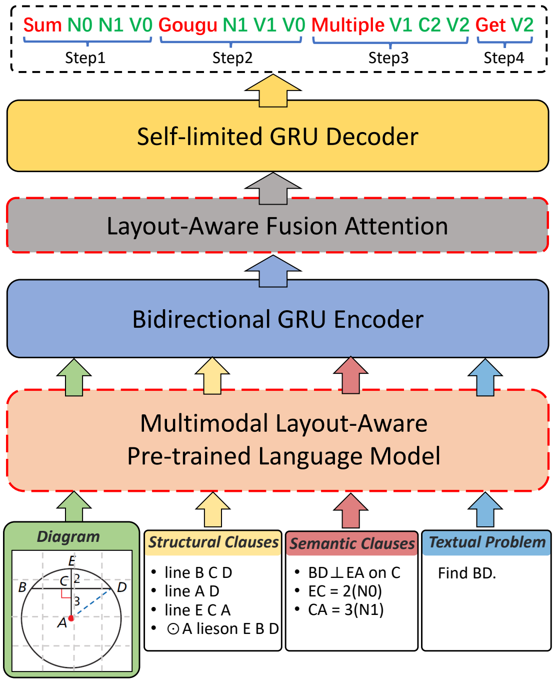


Figure 2: Overview of LANS model. The red dotted boxes are our newly proposed modules in comparison to PGPSNet (Zhang et al., 2023).

Considering the under-representation of geometry diagrams, we propose a layout-aware neural solver called LANS. LANS inputs the diagram image, the textual clauses parsed from a diagram, and the textual problem, and outputs the explainable solution program to solve the geometry problem. As shown in Figure 2, two new modules, multimodal layout-aware pre-trained language model (MLA-PLM) and layout-aware fusion attention (LA-FA), are proposed to endow LANS with layout awareness. We introduce a point-match pre-training (PMP) method within MLA-PLM. This method, based on contrastive learning, aims to model the relationship between text clauses and diagrams using layout information in a data-efficient manner. When integrated with structural-semantic pre-training (SSP) in PGPS solver (Zhang et al., 2023), it shows promising outcomes. Then, to better utilize pre-trained multimodal representations, LA-FA module with the layout-aware attention mask is employed in LANS to fuse the diagram and text clauses representation via point positions. LA-FA further enhances the layout awareness in cross-modal fusion.

The contributions of this work are summarized in four folds: (1) We propose a layout-aware neural solver LANS for GPS, which can represent and fuse geometry diagrams effectively. (2) We introduce the MLA-PLM module with two pre-training strategies SSP and PMP, realizing global relationship modeling and cross-modal alignment of point primitives. (3) We design the LA-FA module, equipped with a layout-aware attention mask directed by point positions, to further strengthen the layout awareness of LANS. (4) Our LANS outperforms existing symbolic solvers, neural solvers, and current multimodal large models significantly on Geometry3K and PGPS9K datasets.

2 Related Work

2.1 Geometry Problem Solving

GPS is a special type of multimodal reasoning that examines geometric spatial structure cognition and mathematical logical reasoning, and also requires the application of geometric theorem knowledge, which make it highly challenging. Existing works of GPS can be classified into two categories: symbolic solvers and neural solvers. The symbolic solvers (Seo et al., 2015; Sachan and Xing, 2017; Lu et al., 2021; Peng et al., 2023) parse the diagram and textual problem into a unified formal language first, and then perform symbolic reasoning by path search and condition matching based on the geometric theorem knowledge. However, symbolic solvers are carefully designed with complex rules and are hard to extend. The neural solvers treat GPS as a visual question answering task and design a special interpretable program to represent the problem-solving process. NGS (Chen et al., 2021) and Geoformer (Chen et al., 2022) use auxiliary self-supervised tasks such as location prediction, elements prediction, and knowledge classification to boost cross-modal semantic representation. PGPSNet (Zhang et al., 2023) expresses the geometry diagram with textual clauses and fuses multi-modal information through structural and semantic pre-training, data augmentation, and self-limited decoding. SCA-GPS (Ning et al., 2023) tries to align characters in text and diagram and enhance the diagram understanding through multi-label classification and masked image modeling pre-training. Although existing neural solvers have achieved impressive performance, they are still coarse-grained at the modal understanding and fusion, especially for geometry diagrams with complex layouts. In this paper, we propose a layout-aware neural solver to improve the understanding and fusion of geometry diagrams and therefore promote GPS.

2.2 Multimodal Pre-training & Layout-Aware Learning

Multimodal pre-training realizes alignment and understanding between different modalities by a series of designed auxiliary tasks and then applies to the specific downstream tasks. Common strategies involve image-text contrastive learning (Radford et al., 2021), image-text matching (Kim et al., 2021), image-grounded text generation (Cho et al., 2021), and masked object classification (Li et al., 2020). With a large amount of pre-training data, these strategies exhibit good performance in multimodal tasks for natural images. However, their alignment methods are coarse-grained and straightforward and do not fit for complex multi-level and fine-grained tasks. Most relevant to our work is the research on document analysis (Liu et al., 2023a). Existing advanced document pre-training methods (Xu et al., 2020, 2021) incorporate textual and visual blocks with fine-grained position embeddings, and adopt masked visual-language modeling and text-image alignment to pretrain document layout, whereas they still do not apply to GPS due to the specificity of geometry objects and small-scale of GPS datasets. DocFormer (Appalaraju et al., 2021) and LayoutReader (Wang et al., 2021) employ meticulously designed attention mechanisms targeting information within text boxes to enhance their perception abilities regarding document content. Our LANS proposes targeted and data-efficient pre-training methods and a geometry layout-aware attention to implement geometry layout awareness.

3 Method

Before presenting the neural solver model, we first describe the formal definition of GPS task here. Given a geometry problem P including a geometry diagram D and a textual problem T_{prob}, the goal is to solve the problem by applying geometric knowledge and obtaining the solution steps S, formulated as $P = \{D, T_{prob}\} \Rightarrow S$. Then solution steps are verified in the form of fill-in-the-blank, multiple-choice, or logical reasons.

3.1 Overall Framework

To fully understand and represent the geometry diagram, we propose a layout-aware neural solver called LANS as displayed in Figure 2. First, the diagram is parsed into the textual clauses using the geometry diagram parser PGDPNet (Zhang et al., 2022), where the structural clauses T_{stru}

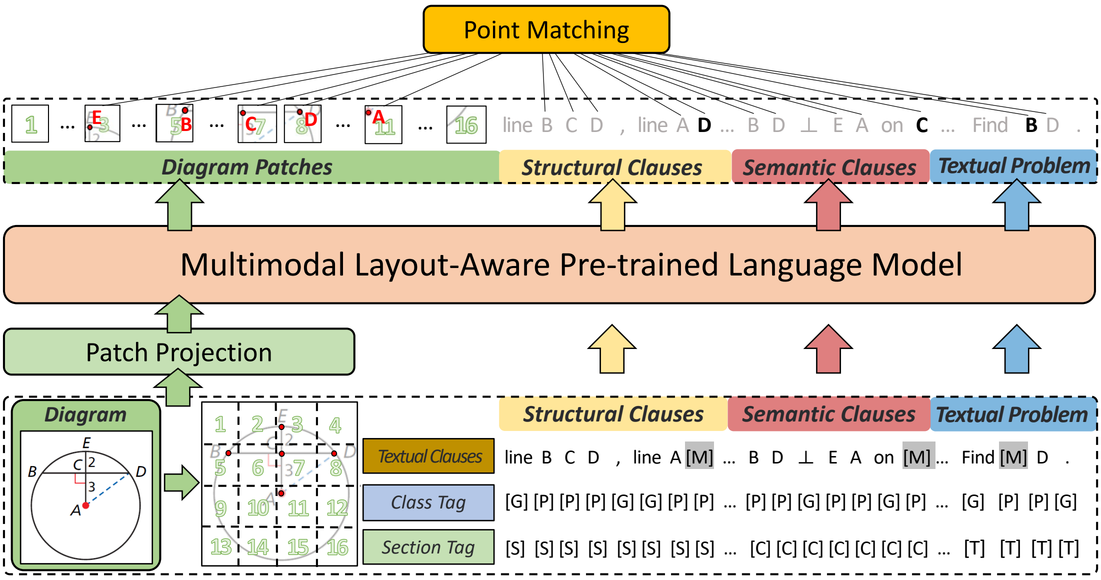

Point Matching
1 ... E 3 ... 5 B ... C 7 8 D ... A 11 ... 16 line B C D , line A D ... B D ⊥ E A on C ... Find B D .
Diagram Patches
Structural Clauses
Semantic Clauses
Textual Problem
Multimodal Layout-Aware Pre-trained Language Model
Patch Projection
Diagram
E
C
B
D
A
2
3
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16
Structural Clauses
Semantic Clauses
Textual Problem
Textual Clauses
line B C D , line A [M] ... B D ⊥ E A on [M] ... Find [M] D .
Class Tag
[G] [P] [P] [P] [G] [G] [P] [P] ... [P] [P] [G] [P] [P] [G] [P] ... [G] [P] [P] [G]
Section Tag
[S] [S] [S] [S] [S] [S] [S] [S] ... [C] [C] [C] [C] [C] [C] [C] ... [T] [T] [T] [T]

Figure 3: Pipeline of multimodal layout-aware pre-training. The geometry problem is the same as that in Figure 2. [M] denotes mask tokens. Class tags and section tags are the same as (Zhang et al., 2023).

describe the connection relations among geometric primitives and the semantic clauses T_{sem} depict the semantic relations between non-geometric primitives and geometric primitives (Zhang et al., 2023). Besides, the visual information of diagram image is represented as patches. Therefore, the input of LANS could be further expressed as $\left\{D=\{d_i\}_{i=1}^{N_D}, T=\{T_{stru}, T_{sem}, T_{prob}\}=\{t_j\}_{j=1}^{N_T}\right\}$ after token concatenation, where N_D is the diagram patch number and N_T is the text token number. Then, these modal tokens are fed into the multimodal layout-aware pre-trained language model (MLA-PLM) and input into the bidirectional GRU encoder to perform fusion encoding. Next, the mixed encoding context $H = \{h_i\}_{i=1}^{N_D+N_T}$ leverages the layout-aware fusion attention (LA-FA) to further boost diagram layout awareness. Finally, the enhanced context is decoded by the self-limited GRU decoder and generates the sequential solution program S in the manner of autoregressive.

3.2 Multimodal Layout-Aware Pre-training

Geometry problems are often solved by humans by depicting the geometric structure in visual form no matter whether it has the geometry diagram or not. Previous neural geometric solvers, such as the NGS (Chen et al., 2021), PGPSNet (Zhang et al., 2023) and SCA-GPS (Ning et al., 2023), do not utilize the diagram layout adequately, thus resulting in unsatisfactory performance of GPS. In this paper, we propose the multimodal layout-aware pre-trained language model (MLA-PLM), with two pre-training strategies: structural-semantic pre-training (SSP) and point matching pre-training (PMP) illustrated in Figure 3, to boost the diagram layout-aware ability during the pre-training stage.

Revisit Structural-Semantic Pre-training To enable the multimodal pre-training module to comprehend text clauses and gain a preliminary understanding of the content and layout of geometry diagrams, we adopted the structural-semantic pre-training (SSP) (Zhang et al., 2023) method used in PGPS. MLA-PLM is trained to recover the masked text in a unified text generation manner, and the training loss denotes as L_{SSP}. Concretely, inputs of MLA-PLM include the diagram patch embeddings e_i^D and textual token embeddings e_j^T, where e_i^D is obtained via patch projection and patch-level positional encoding, and e_j^T fuses not only positional encoding but also embedding of class tag and section tag following (Zhang et al., 2023) as:

$$\begin{aligned} e_i^D &= \text{PatchProj}(d_i) + \text{PosEmb}(i),\ 1 \le i \le N_D \\ e_j^T &= \text{TokenEmb}(t_j) + \text{PosEmb}(j) + \text{ClassEmb}(t_j), \\ &\quad + \text{SectEmb}(t_j),\ 1 \le j \le N_T \end{aligned} \tag{1}$$

where $\text{PosEmb}(*)$ is the sequential position encoding of sequences instead of the spatial position of the diagram layout. The concatenated e_i^D and e_j^T are modeled by MLA-PLM and then output e'^D_i and e'^T_j. For SSP in MLA-PLM, we mask 30% of text tokens t_j with mask token $[M]$ following (Cho et al., 2021) but keep tags unchanged.

Point-Match Pre-training We propose the PMP based on contrastive, learning modeling to achieve cross-modal alignment between visual points (one type of geometric primitives in the diagram) and textual tokens of the points. For PMP, we match image patches and points inside image patches with the cosine contrastive loss (He et al., 2020; Grill et al., 2020) as follows:

$$L_{PMP} = \frac{-1}{|\mathcal{P}|} \sum_{j \in \mathcal{P}} \log \frac{\exp(\cos\langle e'^{T}_{j}, e'^{D}_{+}\rangle / \tau)}{\sum_{i=1}^{N_D} \exp(\cos\langle e'^{T}_{j}, e'^{D}_{i}\rangle / \tau)}, \tag{2}$$

where $\mathcal{P} = \{j \mid \text{Class}(t_j) = \text{[P]},\ 1 \leq j \leq N_T\}$ is the index list of text tokens corresponding to points, e'^{D}_{+} is the embedding of the diagram patch that the point t_j is located in, and τ is the temperature coefficient that empirically set as 0.1. Combining SSP and PMP, our pre-training loss is a multi-task learning loss with the mixed training loss $L_{all} = L_{SSP} + L_{PMP}$.

By combining two pre-training strategies SSP and PMP, the solver strengthens the cognition of complex geometry layout. In SSP, the modeling of local relationships leads to the global relationship understanding, for example, we can infer that the mask token in the semantic clause "BD ⊥ EA on [M]" is "C" according to structural clauses "line B C D" and "line E C A". Via PMP, the textual points become aware of layout position from positional encoded image patches by alignment. We do not adopt the simple and direct way of fine-grained 2D position embedding such as in LayoutLM (Xu et al., 2020, 2021). This is because existing GPS datasets do not support large-scale layout understanding pre-training. It is also akin to human geometric cognition in that accurate positioning is not required to understand geometry layout.

3.3 Layout-Aware Fusion Attention

Although LANS has already acquired a certain level of layout understanding through the pre-training strategies above, this ability can fade to some extent during downstream training because of the different training targets of GPS. To compensate for the loss of layout awareness in the GPS training phase, we propose layout-aware fusion attention (LA-FA) to enhance the intra-modal and cross-modal token fusion. LA-FA is located between the bidirectional GRU encoder and the self-limited GRU decoder.

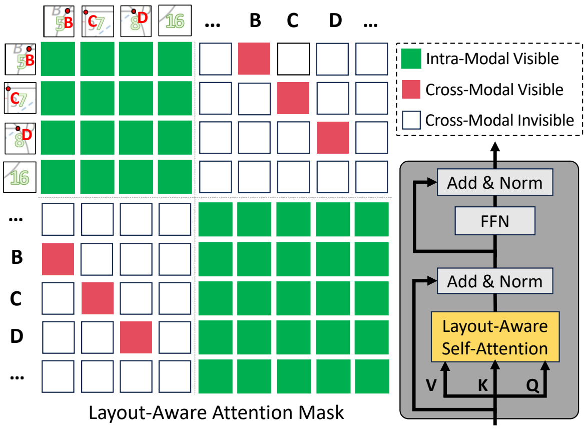


Figure 4: Schematic of Layout-Aware Fusion Attention.

As shown in Figure 4, the LA-FA module is similar to the transformer encoder block (Vaswani et al., 2017) which also contains layer normalization, feed-forward layer, and residual connection except the layout-aware self-attention. Our layout-aware self-attention uses the carefully designed layout-aware attention mask which allows visibility to all intra-modality tokens but restricts cross-modality visibility if the textual point is not inside the image patch in the visual space. Specifically, we construct the mask matrix $M_{i,j}$ $(1 \leq i, j \leq N_D + N_T)$, which consists of value 0 as invisible and value 1 as visible:

$$M_{i,j} = \begin{cases} 1, & \text{if } (i,j) \in VV \\ 1, & \text{if } (i,j) \in TT \\ 1, & \text{if } (i,j) \in VT \& \text{Pos}(t_j) \in \text{Reg}(d_i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $VV = \{(i,j) \mid 1 \leq i, j \leq N_D\}$ is the mask region of visual intra-modality, $TT = \{(i,j) \mid N_D + 1 \leq i, j \leq N_D + N_T\}$ is the mask region of textual intra-modality, $VT = \{(i,j) \mid 1 \leq i, j \leq N_D + N_T\} - VV - TT$ is the mask region of cross-modality, $\text{Pos}(t_j)$ denotes the visual position of point token t_j and $\text{Reg}(d_i)$ refers to the visual region of image patch d_i. Moreover, layout-aware fusion-attention (LA-FA) could be computed by:

$$\text{LA-FA}(Q, K, V, M) = \text{softmax}\left(\frac{QK^T}{\sqrt{m_k}} \cdot M\right) V \tag{4}$$

where Q, K, V are query matrix, key matrix, and value matrix all transformed from encoding context H, and m_k is the dimension of the key vector.

In summary, in the process of cross-modal fusion, LA-FA leverages the point position to guide the attention between diagram and text, strengthening the understanding of diagram layout. For mitigating the optimization burden, we only use one LA-FA block, as adding more blocks does not bring extra improvement according to our experiments.

4 Experiments

4.1 Setup

Model Architecture The patch projection module for diagram chooses the CNN architecture, selecting a light-weight ResNet10 (He et al., 2016) to extract feature map before meshing. Feeding with diagram images resized as 256×256, the patch projection module maps diagram into 8×8=64 image patches. In default, we employ a 6-layer, 8-head, 256-input, and 1024-hidden dimensional transformer (Vaswani et al., 2017) as the architecture of MLA-PLA, and a multi-head attention with the same head number and feature dimension for LA-FA. The bidirectional GRU encoder and self-limited GRU decoder in LANS are adopted following the same architecture as PGPSNet (Zhang et al., 2023). Besides, a dropout layer with the value 0.2 is added behind the patch projection module to prevent overfitting during the training stage.

Training Hyperparameters Details We choose the AdamW optimizer (Loshchilov and Hutter, 2017) with the weight decay 1×10^{-2} and the step decline schedule with the decay rate of 0.5, and the training batch size is set as 128. We provide a more detailed description of the remaining parameters we use during the pre-training and fine-tuning stages in the appendix B.1.

Datasets and Metrics We evaluate the performance of proposed LANS on two plane geometry problem datasets: Geometry3K (Lu et al., 2021) and PGPS9K (Zhang et al., 2023). They all have fine-grained diagram annotation and interpretable solution programs. The textual clauses and point positions used in this paper are converted from the diagram annotation. The solution program consists of several solving steps, each step consists of an operator and associated operands, where the operator corresponds to a geometric theorem and operands are arranged according to the theorem formula. The paired program executor based on Python calculates the numerical results of solution programs. The MLA-PLA module of LANS is pre-trained from scratch on PGPS9K dataset that masks solution programs, because of the shortage of geometric corpus and the great distribution gap in contrast with natural corpus.

Similar to PGPSNet (Zhang et al., 2023), we use three evaluation metrics to assess the numerical performance of our LANS, namely *Completion*, *Choice*, and *Top-3*. In the Completion, the neural solver selects the first executable solution program as the Completion result. The Choice is defined as choosing the correct option from four candidates but selecting one randomly if the outputted answer is not in. In the Top-3, the solution is considered correct if it is among the top three confidence solutions. We set the *Completion* as evaluation metric for ablation study in section 4.3 by default. Given the outstanding capabilities of multimodal large models in addressing multimodal reasoning problems, we compared popular existing open-source multimodal large models in Table 1 with the currently most powerful multimodal model, GPT-4V. Evaluation was conducted in both Completion and Choice modes, where in Completion mode, the large model was required to directly provide answers, and in Choice mode, reference options were added to the prompt for the large model.

4.2 Comparison with State-of-the-art Solvers

We compare LANS with state-of-the-art models, including neural solvers, symbolic solvers, and multimodal large models in Table 1, in terms of both performance and parameter quantity. The results indicate that our LANS achieves excellent model performance by incorporating efficient parameters.

As to symbolic solvers, InterGPS (Lu et al., 2021) solved geometry problems by searching and matching with unified formal language. According to the input source of formal language, InterGPS presents three types of results, e.g., “Predict" means that all formal language is predicted by its parsers, “Digram GT" denotes that formal clauses of diagram use ground truth, and “All GT" indicates that formal clauses of diagram and textual problem are all ground truth. GeoDRL (Peng et al., 2023) improved the search strategy of Inter-GPS with logical graph deduction and deep reinforcement learning. Experimental results show that our LANS outperforms symbolic solvers on all datasets and in all evaluation metrics. Even compared with InterGPS (All GT) which uses annotated formal clauses designed carefully, LANS gains a 3.1% improvement in Completion and a 6.4% improvement in Choice mode on Geometry3K Dataset.

Method	Geometry3K			PGPS9K			Parameters
	Completion	Choice	Top-3	Completion	Choice	Top-3	
Human Expert (Lu et al., 2021)	-	*90.9*	-	-	-	-	-
InterGPS (Predict)* (Lu et al., 2021)	44.6	56.9	-	-	-	-	-
InterGPS (Diagram GT)* (Lu et al., 2021)	64.2	71.7	-	59.8	68.0	-	-
InterGPS (All GT)* (Lu et al., 2021)	69.0	75.9	-	-	-	-	-
GeoDRL (Predict) (Peng et al., 2023)	-	68.4	-	-	-	-	-
Baseline (Neural Solver) (Lu et al., 2021)	-	35.9	-	-	-	-	-
NGS$^{\&}$ (Chen et al., 2021)	35.3	58.8	62.0	34.1	46.1	60.9	80M
Geoformer$^{\&}$ (Chen et al., 2022)	36.8	59.3	62.5	35.6	47.3	62.3	267M
SCA-GPS (Ning et al., 2023)	-	76.7	-	-	-	-	> 310M
PGPSNet (Zhang et al., 2023)	65.0	77.9	80.7	62.7	70.4	79.5	23M
LLaVA-v1.5 (Liu et al., 2023b)	7.6	11.2	-	6.3	9.1	-	7B
mPLUG-Owl2 (Ye et al., 2023)	12.1	17.4	-	10.1	13.1	-	7B
Qwen-VL (Bai et al., 2023)	22.1	26.7	-	20.1	23.2	-	7B
GPT-4V (Achiam et al., 2023a)	38.7	41.4	-	30.2	35.7	-	-
LANS (ours)	**72.1**	**82.3**	**82.8**	**66.7**	**74.0**	**82.2**	26M

Table 1: Performance comparison among state-of-the-art GPS solvers. * denotes results re-produced with the open source code. & denotes methods re-implemented by us.

As to neural solvers, NGS (Chen et al., 2021) and Geoformer (Chen et al., 2022) relied primarily on textual problems to solve problems. Even though re-implementing them with the textual clauses parsed from the diagram and the same augmentation strategies, performance gaps between these two solvers and our LANS are still significant, 32.6% and 31.1% lower in Completion on PGPS9K, respectively. SCA-GPS (Ning et al., 2023) shows similar performance as InterGPS (All GT) because diagram understanding methods, character alignments, and masked image modeling, are coarse-grained and ineffective. PGPSNet (Zhang et al., 2023) employed textual clauses to model diagram layout but lost lots of visual information. Our LANS is enhanced at modal alignment and fusion for better layout awareness and surpasses PGPSNet by 7.1% and 4.0% in Completion on Geometry3K and PGPS9K. The improvements in Top-3 are less than in Completion because most of the correct solutions are concentrated among highly confident candidates.

Our approach far surpasses the performance of current multimodal large models. This may be attributed to the presence of complex symblic OCR information, layout details, and abstract elements in geometric images, where the perception capabilities of the Gemini (Team et al., 2023) and GPT-4V (Achiam et al., 2023b) models are insufficient. Similar phenomena have also been observed in the MathVista (Lu et al., 2024) benchmark. For evaluation results and detailed information on the Completion and Choice multimodal large-scale models, please refer to section C.

4.3 Ablation Study

Effect of Modules To examine the effect of our proposed modules in LANS, we conducted ablation experiments on the Geometry3K dataset, taking PGPSNet solver (Zhang et al., 2023) who owns the SS-PLM module but without the LA-FA module as the baseline. Experimental results presented in Table 2 illustrate that MLA-PLM module with multimodal pre-training is superior to SS-PLM module with only text-modal pre-training and obtains a 5.4% improvement. LA-FA module further boosts GPS via multi-modal feature fusion in the training phase and achieves a 72.1% accuracy, over baseline 7.1%.

Module	Accuracy
Baseline	65.0
+ MLA-PLM	70.4 (+5.4)
+ MLA-PLM + LA-FA	**72.1** (+7.1)

Table 2: Ablation study of modules on Geometry3K.

Role of Pre-training Strategies To validate the role of pre-training strategies within MLA-PLM, we did ablation experiments on both SSP and PMP pre-training strategies. Ablation experiments involved two processes: first pre-training with various strategies and then fine-tuning on Geometry3K. Table 3 verifies that SSP and PMP pre-training strategies all improve GPS, where SSP promotes

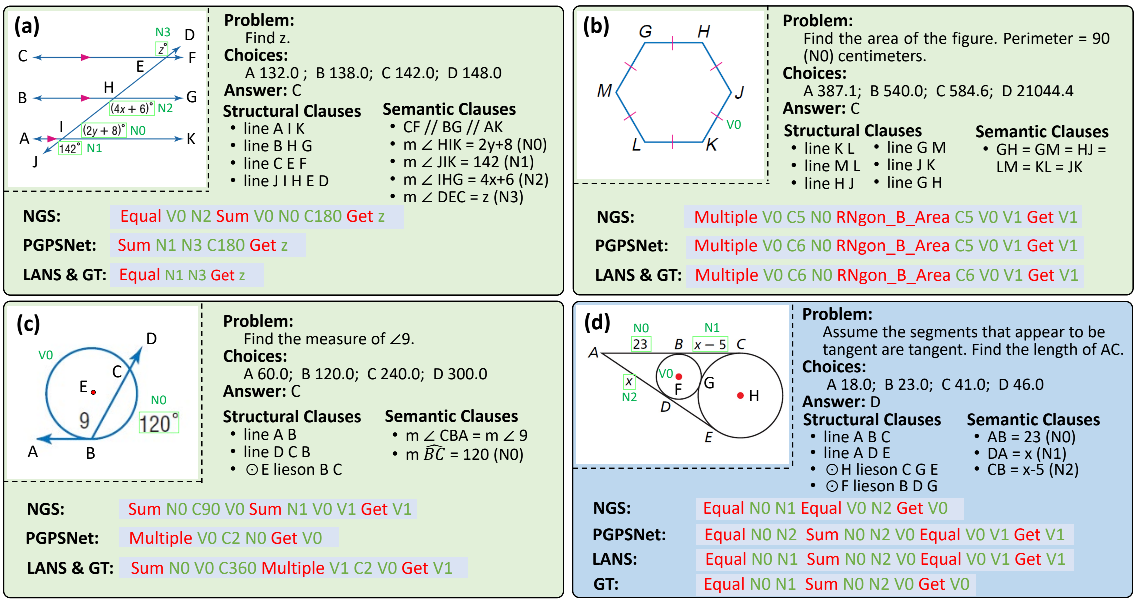

(a)
C
D
F
E
N3
z°
B
H
G
(4x + 6)° N2
A
I
(2y + 8)° N0
K
142° N1
J
Problem:
Find z.
Choices:
A 132.0 ; B 138.0; C 142.0; D 148.0
Answer: C
Structural Clauses
• line A I K
• line B H G
• line C E F
• line J I H E D
Semantic Clauses
• CF // BG // AK
• m ∠ HIK = 2y+8 (N0)
• m ∠ JIK = 142 (N1)
• m ∠ IHG = 4x+6 (N2)
• m ∠ DEC = z (N3)
NGS: Equal V0 N2 Sum V0 N0 C180 Get z
PGPSNet: Sum N1 N3 C180 Get z
LANS & GT: Equal N1 N3 Get z
(b)
G
H
M
J
V0
L
K
Problem:
Find the area of the figure. Perimeter = 90 (N0) centimeters.
Choices:
A 387.1; B 540.0; C 584.6; D 21044.4
Answer: C
Structural Clauses
• line K L • line G M
• line M L • line J K
• line H J • line G H
Semantic Clauses
• GH = GM = HJ = LM = KL = JK
NGS: Multiple V0 C5 N0 RNgon_B_Area C5 V0 V1 Get V1
PGPSNet: Multiple V0 C6 N0 RNgon_B_Area C5 V0 V1 Get V1
LANS & GT: Multiple V0 C6 N0 RNgon_B_Area C6 V0 V1 Get V1
(c)
V0
D
E
C
N0
9
120°
A
B
Problem:
Find the measure of ∠9.
Choices:
A 60.0; B 120.0; C 240.0; D 300.0
Answer: C
Structural Clauses
• line A B
• line D C B
• ⊙E lieson B C
Semantic Clauses
• m ∠ CBA = m ∠ 9
• m $\widehat{BC}$ = 120 (N0)
NGS: Sum N0 C90 V0 Sum N1 V0 V1 Get V1
PGPSNet: Multiple V0 C2 N0 Get V0
LANS & GT: Sum N0 V0 C360 Multiple V1 C2 V0 Get V1
(d)
N0
N1
A
23
B
x − 5
C
x
N2
V0
F
G
H
D
E
Problem:
Assume the segments that appear to be tangent are tangent. Find the length of AC.
Choices:
A 18.0; B 23.0; C 41.0; D 46.0
Answer: D
Structural Clauses
• line A B C
• line A D E
• ⊙H lieson C G E
• ⊙F lieson B D G
Semantic Clauses
• AB = 23 (N0)
• DA = x (N1)
• CB = x-5 (N2)
NGS: Equal N0 N1 Equal V0 N2 Get V0
PGPSNet: Equal N0 N2 Sum N0 N2 V0 Equal V0 V1 Get V1
LANS: Equal N0 N1 Sum N0 N2 V0 Equal V0 V1 Get V1
GT: Equal N0 N1 Sum N0 N2 V0 Get V0

Figure 5: Case analysis on PGPS9K. Solving above problems requires layout awareness of geometry diagram. (a), (b) and (c) are the problems LANS answered correctly, (d) is the problem LANS answered incorrectly.

global relationship recognition and PMP aligns visual points and textual points. The comparison between row 2, row 3, and row 4 demonstrates that the combination of SSP and PMP realizes complex layout understanding synthetically, thus promoting problem-solving together.

Pre-training Strategy	Accuracy
None	38.2
+ SSP	55.4 (+17.2)
+ PMP	66.9 (+28.7)
+ SSP + PMP	**72.1** (+33.9)

Table 3: Ablation study of pre-training strategies on Geometry3K dataset.

Role of Attention Mask To validate the role of attention mask within the LA-FA module, we compare three types of attention masks: w/o LA-FA, vanilla attention mask (Vaswani et al., 2017), and layout-aware attention mask. Compared with the vanilla attention mask with global visibility, layout-aware attention mask guided by point positions promotes modal fusion and strengthens diagram understanding. The results in Table 4 also indicate the significance of layout-aware attention.

Mask Type	Accuracy
w/o LA-FA	70.4
w Vanilla Attention Mask	70.6
w Layout-Aware Attention Mask	**72.1**

Table 4: Ablation study of attention mask on Geometry3K dataset.

4.4 Case Analysis and Fail cases

We also conducted a case analysis to discuss the strengths and weaknesses of solvers. Figure 5 displays four plane geometry problems (a)-(d) involving various geometric layouts, and they rely on good layout awareness to solve them. In case (a), the position of C relative to F determines if ∠JIK and ∠DEC are corresponding or alternate angles. Results show LANS identifies corresponding angles accurately, unlike other solvers. In case (b), the perception of polygon edge number is the key to solving this problem. Contrary to LANS, other solvers cannot count edge numbers correctly through the diagram or textual clauses, resulting in a wrong solution. Case (c) is the same problem as shown in Figure 1 in which textual clauses cannot identify diagram uniquely. In contrast with PGPSNet, LANS can judge the orientation and type of ∠ABC and get the right solution.

5 Conclusion

We propose a layout-aware neural solver LANS to understand complex layouts of plane geometry diagrams. Benefiting from the multimodal layout-aware pre-training, LANS is endowed with abilities of global relationship cognition and cross-modal point alignment. Thanks to layout-aware fusion attention, LANS further improves cross-modal fusion directed by point positions. The experimental results demonstrate the superiority of LANS enhanced with layout awareness.

Limitations

LANS is still limited to point primitives to carry out layout understanding. In the future, we will try to align higher-level geometric primitives to obtain better layout understanding and modal fusion. Besides, LANS may generate redundant solution sequences. Case (d) in Figure 5 is a complex layout scenario that none of the solvers can solve correctly. In conclusion, the case analyses above fully indicate that LANS promotes GPS with enhanced layout awareness. Integrating richer layout information and symbolic cues of elements through multimodal pretraining is a direction worthy of further exploration.

Ethical Impact

As a neural solver addressing multimodal mathematical problems, LANS has the potential for application in educational settings, specifically for the automatic resolution of mathematical problems. This utilization can contribute to promoting educational equity.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023a. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023b. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Peter Anderson, Xiaodong He, Chris Buehler, Damien Teney, Mark Johnson, Stephen Gould, and Lei Zhang. 2018. Bottom-up and top-down attention for image captioning and visual question answering. In *CVPR*.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *CVPR*.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A frontier large vision-language model with versatile abilities. *arXiv preprint arXiv:2308.12966*.

Daniel G. Bobrow. 1968. Natural language input for a computer problem solving system. *Semantic Information Processing*.

Jie Cao and Jing Xiao. 2022. An augmented benchmark dataset for geometric question answering through dual parallel text encoding. *COLING*, 29.

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 *EMNLP*.

Jiaqi Chen, Jianheng Tang, Jinghui Qin, Xiaodan Liang, Lingbo Liu, Eric P. Xing, and Liang Lin. 2021. GeoQA: A geometric question answering benchmark towards multimodal numerical reasoning. In *Findings of ACL*.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *ICML*.

Shang Ching Chou, Xiao Shan Gao, and Jing Zhong Zhang. 1996. Automated generation of readable proofs with geometric invariants: II. Theorem proving with full-Angles. *Journal of Automated Reasoning*, 17(3).

Yang Ding, Jing Yu, Bang Liu, Yue Hu, Mingxin Cui, and Qi Wu. 2022. Mukea: Multimodal knowledge extraction and accumulation for knowledge-based visual question answering. In *CVPR*.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, et al. 2020. Bootstrap your own latent-a new approach to self-supervised learning. In *NeurIPS*.

Kamal Gupta, Justin Lazarow, Alessandro Achille, Larry S Davis, Vijay Mahadevan, and Abhinav Shrivastava. 2021. LayoutTransformer: Layout generation and completion with self-attention. In *CVPR*.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *CVPR*.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *CVPR*.

Teakgyu Hong, Donghyun Kim, Mingi Ji, Wonseok Hwang, Daehyun Nam, and Sungrae Park. 2015. BROS: A pre-trained language model focusing on text and layout for better key information extraction from documents. In *AAAI*.

Wonjae Kim, Bokyung Son, and Ildoo Kim. 2021. Vilt: Vision-and-language transformer without convolution or region supervision. In *ICML*.

Gen Li, Nan Duan, Yuejian Fang, Ming Gong, and Daxin Jiang. 2020. Unicoder-VL: A universal encoder for vision and language by cross-modal pretraining. In *AAAI*.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. 2022. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *ICML*.

Cheng-Lin Liu, Lianwen Jin, Xiang Bai, Xiaohui Li, and Fei Yin. 2023a. Frontiers of intelligent document analysis and recognition: review and prospects. *Journal of Image and Graphics*, 28(08):2223–2252.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023b. Visual instruction tuning. In *NeurIPS*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. In *ICLR*.

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 *ICLR*.

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 *ACL*.

Pan Lu, Liang Qiu, Wenhao Yu, Sean Welleck, and Kai-Wei Chang. 2023. A survey of deep learning for mathematical reasoning. In *ACL*.

Maizhen Ning, Qiu-Feng Wang, Kaizhu Huang, and Xiaowei Huang. 2023. A symbolic characters aware model for solving geometry problems. In *ACM MM*.

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 ACL*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *ICML*.

Mrinmaya Sachan and Eric Xing. 2017. Learning to solve geometry problems from natural language demonstrations in textbooks. In *SEM*.

Minjoon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. 2015. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

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.

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 *NeurIPS*.

Zilong Wang, Yiheng Xu, Lei Cui, Jingbo Shang, and Furu Wei. 2021. LayoutReader: Pre-training of text and layout for reading order detection. In *ACL*.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei Florencio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *ACL*.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *SIGKDD*.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, Fei Huang, and Jingren Zhou. 2023. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration.

Zhou Yu, Jun Yu, Yuhao Cui, Dacheng Tao, and Qi Tian. 2019. Deep modular co-attention networks for visual question answering. In *CVPR*.

Jiaxin Zhang, Yinghui Jiang, and Yashar Moshfeghi. 2024. GAPS: geometry-aware problem solver. *CoRR*, abs/2401.16287.

Ming-Liang Zhang, Fei Yin, Yihan Hao, and Cheng-Lin Liu. 2022. Plane geometry diagram parsing. In *IJCAI*.

Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2023. A multi-modal neural geometric solver with textual clauses parsed from diagram. In *IJCAI*.

A Dataset Details

We evaluated our method on two datasets, Geometry3K and PGPS9K, each containing high-quality diagram images. Table 5 provides a comparison of key information between the two datasets, with PGPS9K featuring a wider variety of question types and higher reasoning complexity.

Dataset	#QA	#Type	#Avg OP	#Avg PL
Geometry3K	3,002	4	1.98	5.35
PGPS9K	9,022	30	2.43	7.45

Table 5: Type, OP and PL represent problem type, operator number and program length, respectively

B Training Details

To ensure the reproducibility of the paper, we provide here the key hyperparameters used during training, as well as the data augmentation methods employed. Additionally, within our method framework, how Patch Projection is relied upon and the granularity of Patch division are crucial for achieving the effectiveness of our approach as described in the paper. We discuss here the impact of these parameters on the replicability of the model.

B.1 Optimzation Parameters Details

During the pre-training phase, the learning rate is initialized to 5×10^{-4} and the learning rate decay is applied at 1,000, 1,800, 2,400, and 3,000 epochs with a total of 3,500 epochs. During the training stage, all modules of LANS train together with an initial learning rate as $1e^{-4}$ for language model MLA-PLM and $1e^{-3}$ for other modules, decaying at 160, 280, 360, 440 and 500 epochs uniformly with a total 520 epochs.

All experiments were conducted on an 8-GPU Titan XP server. Training of the MLA-PLM module took approximately 20 hours on a 4-GPU machine, while fine-tuning of LANS on 4 GPUs took 8 hours.

B.2 Data Augmentation Details

We scale the image to 256 on the longest side and place it in the center of 256×256 blank screen. The diagram is flipped randomly and changes the point positions accordingly. For text, following the work (Zhang et al., 2023), we apply four augmentation strategies: token replacement, connection rotation, representation transposition, and clauses shuffle. These augmentation strategies not only improve the diversity of geometry problems but also provide geometric solvers with basic geometric representation knowledge.

B.3 Impact of hyperparameters

Discussion on the Granularity of Patch Division. To assess the influence of image patches, we adopted four configurations of patch numbers: 1×1, 4×4, 8×8, and 16×16. In Table 6, we observe that LANS benefits from fine-grained partitions of the diagram, based on the comparison of row 1 with rows 2, 3, and 4. However, according to the comparison of row 3 with row 4, problem-solving performance declines if the diagram is over-segmented. The possible explanation is that redundant and blank image grids, which are generated from patch partition, interfere with model attention while increasing the burden of model computation. Therefore, considering overall performance and speed, we choose the 8×8 configuration as our model setup.

Image Patch Num.	Geometry3K	PGPS9K
1 × 1	65.0	62.7
4 × 4	70.5	**66.8**
8 × 8	**72.1**	66.7
16 × 16	69.1	65.4

Table 6: Comparison of Different Image Patch Numbers.

Discussion on the Projection Method of Image Patches. To validate the impact of patch projection schemes, in Table 7, we tested three types of patch projection modules: None, linear layer, and CNN model. None refers to not using the patch projection module and also not inputting image patches. In our experiments, we find that a redundant placeholder in None does harm to GPS due to additional meaningless optimizations. The linear-based patch projection maps image grids linearly and produces corresponding image patches, which is also commonly adopted in recent transformer architectures (Kim et al., 2021; Li et al., 2022). However, this module does not fit to geometry diagram because it may damage the geometric structure. CNN-based patch projection first extracts global features and then mesh feature maps. That module could better understand the overall layout, bringing with higher solving performance and more stable training, and it is also set as the default patch projection module.

Projection Type	Geometry3K	PGPS9K
None	64.2	61.3
Linear	69.4	65.5
CNN	**72.1**	**66.7**

Table 7: Comparison of different patch projections

C Mulitmodal LLM Eval Details

C.1 Eval Prompt Details

We illustrate in Table 8 with examples of how the prompts used for evaluating the multimodal large model vary across different Eval Mode. Our Prompt consists of several components, including

Eval Mode	**Prompt**
Choice	**Role Prompt:** You are a geometric problem-solving robot. Please solve the following geometry problems based on the contents of the diagram and the problem description. **Diagram**: The Diagram is <img>images/img_3755.png</img> **Question:** If RL = 5, RT = 9, and WS = 6, find RW. **Choices:** (A) 5.4 (B) 6.6 (C) 6.0 (D) 7.5 **Format Prompt:** Please give reason process and provide the correct option, such as: the answer is A/B/C/D:.
Completion	**Role Prompt:** You are a geometric problem-solving robot. Please solve the following geometry problems based on the contents of the diagram and the problem description. **Diagram**: The Diagram is <img>images/img_2056.png</img> **Question:** Polygon $ABCD \sim AEFG$, $\angle AGF = 108^\circ$, $GF = 14$, $AD = 12$, $DG = 4.5$, $EF = 8$, and $AB = 26$. Find $\angle ADC$. **Format Prompt:** Please give reason process and provide the correct option, such as: the answer is 15.0:.

Table 8: The prompts used for Choice and Completion Modes in two specific questions.

Model Name	**Model Repository Name/API Version**	**Sampling Parameters**
Qwen-VL	Qwen/Qwen-VL-Chat	do_sample = True, top-k = 5, max_length = 512
LLaVA-1.5	liuhaotian/llava-v1.5-13b	do sample = True, temperature = 0.2, max new tokens = 1024
mPLUG-Owl2	MAGAer13/mplug-owl2-llama2-7b	do sample = True, top-k = 5, max length = 512
GPT4V	gpt-4-1106-vision-preview	Chatbot URL: `https://chat.openai.com`

Table 9: Generating parameters and Huggingface model repository names for multimodal large models

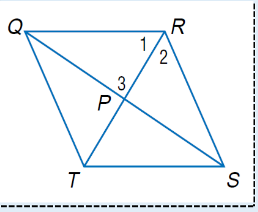
Role Prompt, *Diagram*, *Question*, *Choices*, and *Format Prompt*. The *Role Prompt* specifies the type of problem the large model is tasked with solving and the actions it needs to perform. *Diagram* depicting the form's content, textual description of the *Question*, and *Choices*. To ensure the large model generates standardized output for easy scoring, we have set a *Format Prompt*. The main difference in evaluation between *Choice* and *Completion* modes lies in that, in *completion* mode, the prompt does not provide reference options.

C.2 Generation Parameter Details

We list the relevant important parameters used for evaluation across different multimodal large models in Table 9.

C.3 Output detail and discussion of MLLM

Table 1 demonstrates that multimodal large models perform poorly in solving geometric problems. Some open-source models exhibit performance lower than random guessing for select questions. We attribute this to the fact that the visual comprehension component of current multimodal large models is primarily tailored to natural scene images and struggles with abstract forms. For instance, in Figure 6, both GPT4V and LLaVA-v1.5 fail to grasp the relational elements within geometric diagrams, such as understanding the reference to Angle 1. Additionally, models like LLaVA-v1.5 may also generate severe model hallucinations, even output in an incorrect format like "So, the answer is [28.0, 56.0, 62.0, 90.0]," which prevent the extraction of correct results and result in low accuracy.



Question: Use rhombus QRST. Find ∠ 1 if ∠ RST = 56.

Choice: A. 28.0 B. 56.0 C. 26.0 D. 62.0

Structure Clauses		**Semantic Clause**
line Q P S	line R P T	∠ PRQ = ∠ 1
line Q T	line R S	∠ PRS = ∠ 2
line Q R	line T S	∠ QPR = ∠ 3

GT Answer: D.

LLaVA-v1.5: Diagram consists of four lines: QP, RT, QT, and RS. We are given that ∠PRQ = ∠1, ∠PRS = ∠2, and ∠QPR = ∠3. Additionally, we are given that ∠RST = 56. We can start by using the properties of the rhombus QRST. A rhombus has all four sides equal in length. Let's assume that the length of QR = RS = ST = TQ = QR = 1. Now, we can use the information about the angles: ∠PRQ = ∠1 = ∠RST / 2 = 56 / 2 = 28. ∠PRS = ∠2 = ∠RST / 2 = 56 / 2 = 28 ∠QPR = ∠3 = ∠RST / 2 = 56 / 2 = 28. Since ∠RST = 56, we can find the measure of m∠1 by dividing 56 by 2: ∠1 = 56 / 2 = 28. So, the answer is [28.0, 56.0, 62.0, 90.0].

GPT4V: Since QRST is a rhombus, we know the following properties: 1. All sides of QRST are equal. 2. Opposite angles are equal (i.e., ∠QTS = ∠QRS and ∠QRT = ∠QST). The diagonals (Q to S and R to T) bisect each other at 90 degrees. Given that ∠RST = 56 degrees and RST is part of the rhombus, this means that ∠QRT = 56 degrees as well, because they are opposite angles.
Since the diagonals of a rhombus are perpendicular, ∠RPT is 90 degrees. We have the measures of two angles in triangle PRT: ∠RPT = 90 degrees and ∠RST (or ∠QRT) = 56 degrees.
The sum of angles in a triangle is 180 degrees, so we can find ∠1 (or ∠PRQ) as follows: ∠1 = 180 degrees - ∠RPT - ∠RST. ∠1 = 180 degrees - 90 degrees - 56 degrees. ∠1 = 34 degrees.

Figure 6: Case of the multimodal large model LLaVA-v1.5 and GPT4V. The red ones are marked as generated inference hallucinations (Zhang et al., 2023).