GeoDANO: Geometric VLM with Domain Agnostic Vision Encoder

Anonymous ACL submission

Abstract

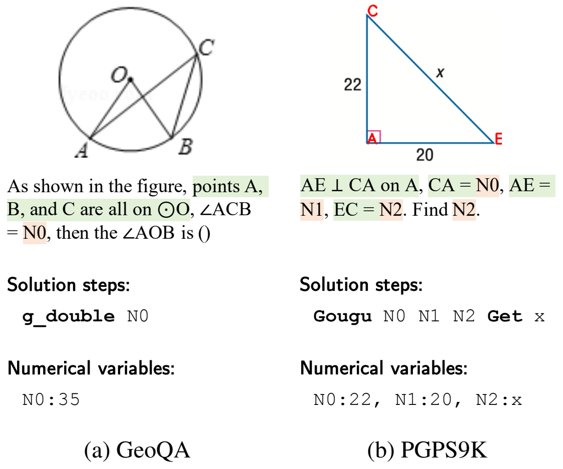
We introduce GeoDANO, a geometric vision-language model (VLM) with a domain-agnostic vision encoder, for solving plane geometry problems. Although VLMs have been employed for solving geometry problems, their ability to recognize geometric features remains insufficiently analyzed. To address this gap, we propose a benchmark that evaluates the recognition of visual geometric features, including primitives such as dots and lines, and relations such as orthogonality. Our preliminary study shows that vision encoders often used in general-purpose VLMs, e.g., OpenCLIP, fail to detect these features and struggle to generalize across domains. To overcome the limitation, we develop GeoCLIP, a CLIP-based model trained on synthetic geometric diagram–caption pairs. Benchmark results show that GeoCLIP outperforms existing vision encoders in recognizing geometric features. We then propose our VLM, GeoDANO, which augments GeoCLIP with a domain adaptation strategy for unseen diagram styles. GeoDANO outperforms specialized methods for plane geometry problems and GPT-4o on MathVerse.

1 Introduction

Large language models (LLMs) have achieved remarkable success in automated math problem solving, particularly through code-generation capabilities integrated with proof assistants (Moura and Ullrich, 2021; Nipkow et al., 2002; Chen et al., 2023; Wu et al., 2022; Hendrycks et al., 2021). Although LLMs excel at generating solution steps and correct answers in algebra and calculus (Zhou et al., 2024), their unimodal nature limits performance in plane geometry, where the solution depends on both diagram and text (Zhou et al., 2024).

Specialized vision-language models (VLMs) have accordingly been developed for plane geometry problem solving (PGPS) (Chen et al., 2021, 2022; Lu et al., 2021; Zhang et al., 2023; Zhang and Moshfeghi, 2024; Li et al., 2024b; Xia et al., 2024). Yet, whether these models genuinely leverage diagrams or rely almost exclusively on textual features remains unclear. This ambiguity arises because existing PGPS datasets typically embed sufficient geometric details within problem statements, potentially making the vision encoder unnecessary (Zhang and Moshfeghi, 2024). Fig. 1 illustrates example questions from GeoQA and PGPS9K, where solutions can be derived without referencing the diagrams.

We propose a new benchmark created via a synthetic data engine, which systematically evaluates the ability of VLM vision encoders to recognize geometric premises. Our empirical findings reveal that previously suggested self-supervised learning (SSL) approaches, e.g., vector quantized variational auto-encoder (VQ-VAE) (Liang et al., 2023) and masked auto-encoder (MAE) (Ning et al., 2023;



As shown in the figure, points A, B, and C are all on ⨀O, ∠ACB = N0, then the ∠AOB is ()

Solution steps:

`g_double N0`

Numerical variables:

`N0:35`

(a) GeoQA

AE ⊥ CA on A, CA = N0, AE = N1, EC = N2. Find N2.

Solution steps:

`Gougu N0 N1 N2 Get x`

Numerical variables:

`N0:22, N1:20, N2:x`

(b) PGPS9K

Figure 1: Examples of diagram-caption pairs and their solution steps written in formal languages from the GeoQA and PGPS9k datasets. The problem description highlights the visual geometric premises and numerical variables in green and red, respectively. A significant difference in the style of the diagram and formal language can be observed.

Xia et al., 2024), and widely adopted encoders, e.g., OpenCLIP (Radford et al., 2021) and DinoV2 (Oquab et al., 2024), struggle to detect geometric features such as perpendicularity and degrees.

To this end, we propose GeoCLIP, a model pre-trained on a large corpus of synthetic diagram–caption pairs. By varying diagram styles (e.g., color, font size, resolution, line width), GeoCLIP learns robust geometric representations and outperforms prior SSL-based methods on our benchmark. Building on GeoCLIP, we introduce a few-shot domain adaptation technique that efficiently transfers the recognition ability to real-world diagrams. We finally propose GeoDANO by combining this domain-adapted GeoCLIP with an LLM, forming a domain-agnostic VLM for solving PGPS tasks.

In our experiments on MathVerse (Zhang et al., 2024a), which encompasses diverse plane geometry tasks and diagram styles, GeoDANO consistently outperforms both task-specific PGPS models and generalist VLMs. Ablation studies confirm the effectiveness of our domain adaptation strategy, showing improvements in optical character recognition (OCR)-based tasks and robust diagram embeddings across different styles.

We summarize the contributions as follows: We propose a novel benchmark for systematically assessing how well vision encoders recognize geometric premises in plane geometry diagrams (§3); We introduce GeoCLIP, a vision encoder capable of accurately detecting visual geometric premises (§4.1), and a few-shot domain adaptation technique that efficiently transfers this capability across different diagram styles (§4.2); We show that our VLM, named GeoDANO, incorporating domain-adapted GeoCLIP, surpasses existing specialized PGPS VLMs and generalist VLMs on the MathVerse benchmark (§5.2) and effectively interprets diverse diagram styles (§5.3).

2 Related Work

In this section, we summarize the known PGPS benchmarks and models. Detailed comparison with previous work and work on contrastive learning in PGPS is reported in Appendix A.

2.1 PGPS benchmarks

Several studies have introduced benchmarks for PGPS, including a set of diagrams and corresponding problem and solution descriptions (Chen et al., 2021; Lu et al., 2021; Zhang et al., 2023; Chen et al., 2022). The problem and solution descriptions are provided in natural languages or formal languages. Often, the solution steps are provided in the form of formal language. Given the dataset, the goal of PGPS is to train a model that produces a valid solution as an executable program.

Recently, MathVerse (Zhang et al., 2024a) provides an alternative view to existing PGPS benchmarks by directly encoding the geometric properties and relations into the diagrams rather than text description. Therefore, it is impossible to produce a valid solution without recognizing the necessary information from diagrams. CogAlign (Huang et al., 2025) introduces a benchmark that evaluates the spatial relationship understanding of pretrained vision encoders via linear probing. However, the questions in this benchmark focus purely on spatial relationships and do not involve symbols representing geometric relations.

2.2 Program generation based PGPS

A core challenge in program generation-based PGPS is processing both diagrams and text to interpret geometric premises. One approach tackles the challenge by converting a diagram into alternative representations such as lists of geometric primitives and relations that can be represented as text (Seo et al., 2015; Sachan et al., 2017; Lu et al., 2021; Zhang and Moshfeghi, 2024; Zhang et al., 2022; Peng et al., 2023). Although reducing the problem to a single modality can be effective, building such converters typically requires labeled diagrams, which are expensive to collect and eventually limit generalization across diverse diagram styles.

Another line of research typically employs vision-language models (VLMs), where a VLM comprises a vision encoder and a language model (Zhang et al., 2023; Chen et al., 2021; Cao and Xiao, 2022; Ning et al., 2023; Chen et al., 2022; Liang et al., 2023; Xia et al., 2024; Li et al., 2024b). The vision encoder produces a visual embedding from the diagram, and the language model then generates solution steps in an autoregressive manner, conditioned on the textual description and the visual embedding. While the VLMs apply to various diagram formats, the visual geometric premises perception of the VLMs remains underexplored due to the abundance of textual information in existing benchmarks.

3 Benchmark for Geometric Premises

In this section, we first develop a benchmark for evaluating a vision encoder's performance in recognizing geometric features from a diagram. We then report the performance of well-known vision encoders on this benchmark.

3.1 Benchmark preparation

We design our benchmark as simple classification tasks. By investigating PGPS datasets, we identify that recognizing *geometric primitives*, such as points and lines, and geometric properties representing *relations between primitives*, such as perpendicularity, is important for solving plane geometry problems. Recognized information forms *geometric premises* to solve the problem successfully. To this end, we carefully curate five classification tasks as follows:

- **Concyclic**: A circle and four points are given. The task is identifying how many of those points lie on the circle.
- **TwoLines**: Two lines, AB and BC, are given alongside other geometric objects. The task is determining whether AB and BC are perpendicular, collinear, or neither.
- **ObjectShape**: A given diagram includes one of the following geometric objects: a segment, triangle, square, or pentagon. The task is to classify which object is present.
- **SquareShape**: A diagram including a square ABCD and other geometric objects is given. The task is to classify whether the square is a trapezoid, parallelogram, or rectangle.
- **AngleDetection**: A diagram is given with at least three points: A, B, and C. The task is to classify the correct angle of ABC from $\{15^\circ, 20^\circ, \dots, 75^\circ\}$.

An example of each task is provided in Fig. 2.

Our benchmark is built on top of AlphaGeometry (Trinh et al., 2024), which is designed to solve IMO-style plane geometry problems. The program provides useful functions such as formal language describing plane diagrams. The language predefines a set of geometric premises listed in Table A1, including all necessary properties to define our benchmark tasks. In addition, once a diagram description is given in formal language, the program renders a corresponding diagram with varying fonts, colors, widths, orientations, and resolutions, allowing us to have diagrams with diverse styles.

	Models	Object Shape	Con cyclic	Two Lines	Square Shape	Angle Detection
Baseline	OpenCLIP	**100.00**	99.13	86.57	85.20	64.81
	SigLIP	**100.00**	**99.71**	89.26	89.31	76.86
	DinoV2	**100.00**	98.01	85.30	91.24	22.43
	ConvNeXT	**100.00**	99.20	89.39	88.13	61.84
	MAVIS-CLIP	91.64	71.78	58.84	60.48	16.12
	GeoGLIP	98.40	91.97	60.22	63.10	11.58
SSL	Jigsaw	86.11	63.85	49.98	61.88	11.44
	MAE	93.99	72.25	71.73	82.70	13.08
	VQ-VAE	63.05	60.97	48.10	57.35	9.22
GeoCLIP	GeoCLIP (F ×)	99.52	98.61	88.33	86.76	65.68
	GeoCLIP (2K)	99.32	98.73	94.73	89.22	74.95
	GeoCLIP	99.21	99.24	**96.05**	**95.95**	**78.56**

Table 1: Results on the proposed visual feature benchmark. We report the test accuracy of the models with the best validation performance.

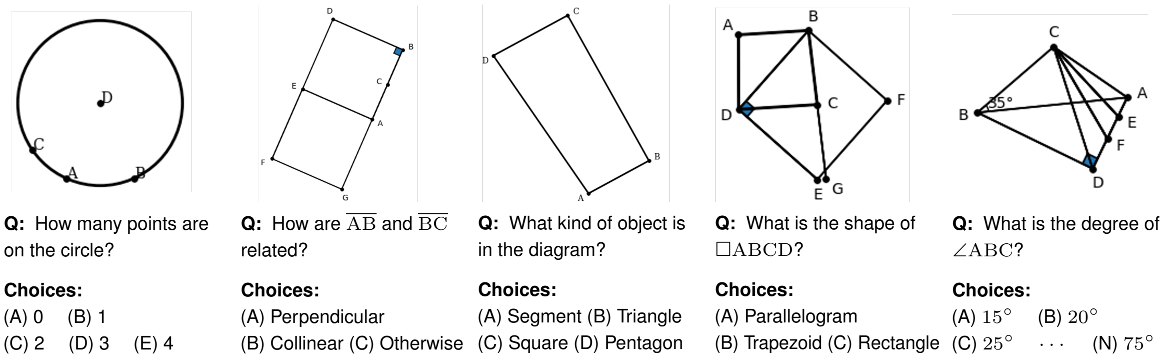
We create question-and-answer pairs based on AlphaGeometry. To sample a diverse set of questions and answers, we first establish a foundational geometric structure corresponding to the key problem of the task and then repeatedly add new points or lines with randomly selected geometric relationships to the existing diagram with the help of the formal language. The pseudo-code for the random question generation is presented in Algorithm 1. For each task, we generate 50,000, 10,000, and 10,000 question-and-answer pairs for training, validation, and testing, respectively. Additional details on the benchmark generation process are available at Appendices C.1 and C.2.

3.2 Results

We evaluate four widely adopted vision encoders for the open-sourced VLMs: OpenCLIP (Radford et al., 2021), SigLIP (Zhai et al., 2023), DinoV2 (Oquab et al., 2024), ConvNeXT (Liu et al., 2022), GeoGLIP (Zhang et al., 2025), and MAVIS-CLIP (Zhang et al., 2024b). For MAVIS-CLIP, we train OpenCLIP using synthetic diagram-caption pairs generated by MAVIS, following the original MAVIS training configuration.

To evaluate the vision encoder, we use linear probing, i.e., adding a linear layer on top of each encoder as a prediction head and training the linear layer from scratch while freezing the parameters of the vision encoder. We use a training set to train the prediction head and report the test accuracy with the best validation performance. The details for the hyper-parameters are described in Appendix C.3.

As shown in Table 1, many existing vision en-



Q: How many points are on the circle?

Choices:
(A) 0 (B) 1
(C) 2 (D) 3 (E) 4

(a) Concyclic

Q: How are $\overline{\mathrm{AB}}$ and $\overline{\mathrm{BC}}$ related?

Choices:
(A) Perpendicular
(B) Collinear (C) Otherwise

(b) TwoLines

Q: What kind of object is in the diagram?

Choices:
(A) Segment (B) Triangle
(C) Square (D) Pentagon

(c) ObjectShape

Q: What is the shape of $\square \mathrm{ABCD}$?

Choices:
(A) Parallelogram
(B) Trapezoid (C) Rectangle

(d) SquareShape

Q: What is the degree of $\angle \mathrm{ABC}$?

Choices:
(A) $15°$ (B) $20°$
(C) $25°$ $\cdots$ (N) $75°$

(e) AngleDetection

Figure 2: Illustration of the proposed visual feature perception benchmark. We introduce five different diagram classification tasks that require visual feature perception to answer geometry-related questions.

coders relatively well recognize the shape of objects but fail at the correct angle between two lines. The encoders also show some difficulties in recognizing the shape of a square and the relationship between two lines. Although the result may seem satisfactory at a glance, these errors will propagate to the downstream tasks when combined with LLMs.

4 GeoCLIP: Enhanced Vision Encoder

In this section, we first propose GeoCLIP, a new vision encoder designed to recognize geometric premises from diverse styles of diagrams. To transfer the recognition to real-world PGPS benchmarks, we then propose a domain adaptation technique for GeoCLIP that leverages a small set of diagram–caption pairs from target domains.

4.1 Training GeoCLIP

We propose a GeoCLIP, a vision encoder trained with the CLIP objective with a newly developed 200,000 diagram-caption examples. From the random diagram generator developed in §3.1, we additionally sample 200,000 diagrams written in the formal language. Directly rendering these samples can result in a diagram that may not preserve the geometric properties. For example, the perpendicularity between two lines cannot be observed from the diagram without having the right angle sign, i.e., ∟. Therefore, we ensure to render the images containing all necessary geometric premises from their visual illustration.

For the caption of a diagram, we filter out some geometric properties from the original description of a diagram used to render the image. Specifically, we only keep the following four properties, concyclic, perpendicularity, angle measures, and length measures, from the visual premises shown in Table A1. After that, we convert the remaining descriptions written in the formal language into natural language. We filter out some properties for two reasons. First, some properties are not recognizable from the rendered diagram without additional information, e.g., congruency. These properties are listed as non-visual premises in Table A1. Second, collinearity and parallelity occur so frequently that they can marginalize others. Some examples of generated captions after filtering and translating are provided in the right-most column of Fig. A1. We call the filtered caption as *GeoCLIP-style caption*.

With this dataset, we fine-tune OpenCLIP (Radford et al., 2021) via the CLIP objective, formulated as:

$$\mathcal{L}_{\text{CLIP}}(\mathcal{D}, g, h) := \mathbb{E}_{\mathcal{D}}\left[-\log \frac{\exp\big(g(D_i)^T\, h(X_i)/\tau\big)}{\sum_{X \in \{X_i\}_i} \exp\big(g(D_i)^T\, h(X)/\tau\big)}\right], \tag{1}$$

where $\mathcal{D} := \{(D_i, X_i)\}_{i=1}^{N}$ is the diagram-caption pairs, g is the vision encoder, h is the text encoder, and τ is a temperature parameter. We named the resulting vision encoder as GeoCLIP. Appendix C.3 provides the details, including hyper-parameters.

We compare the performance of GeoCLIP to other self-supervised approaches trained with the same dataset. We test three self-supervised approaches: Jigsaw (Chen et al., 2021; Cao and Xiao, 2022), MAE (Ning et al., 2023; Xia et al., 2024), and VQ-VAE (Liang et al., 2023), used in previous work to improve the recognition performance of plane diagrams. We use the same architecture used for GeoCLIP for Jigsaw and MAE with the hyper-parameters used in the previous works. For VQ-VAE, we follow the architecture of Liang et al. (2023). All model performances are measured through the linear probing used in §3.2.

As shown in Table 1, GeoCLIP recognizes geometric features better than existing baselines and self-supervised methods. The self-supervised approaches generally perform poorly for the benchmark, justifying the choice of the objective. We also compare the performance of GeoCLIP against other encoders such as OpenCLIP. Note that although we outperform the other encoders in tasks such as SquareShape and AngleDetection, these results might be *unfair* compared to the existing pretrained vision encoders, since the training set of GeoCLIP is similar to the diagrams in the benchmark. The t-SNE plots of the embeddings from the vision encoders are illustrated at Fig. A3.

We further ablate the filtering process in GeoCLIP. To this end, we compare GeoCLIP with its two variants: *GeoCLIP (F ×)*, which uses the captions generated without filtering. We also test *GeoCLIP (2K)*, which is trained on only 2,000 pairs, to see the effectiveness of the large-scale dataset. The results in Table 1 imply that both the filtering and the training set size matter in enhancing geometric properties recognition.

4.2 Domain adaptation of GeoCLIP

Although GeoCLIP enhances the geometric premises recognition on the benchmark set, the diagram styles in existing PGPS benchmarks differ, necessitating further adaptation. To overcome this challenge, we propose a few-shot domain adaptation method utilizing a few labeled diagrams.

A domain-agnostic vision encoder must match the same diagrams drawn in different styles. To do so, we need a target domain diagram translated into the source domain style or the source diagrams translated into the target domain style. With these translated images, we can guide the model to focus on key geometric information instead of irrelevant attributes, such as color and font family. However, in practice, it is difficult to obtain the same diagrams with different styles.

We develop a way to translate the target diagrams into the source style. Thankfully, since well-known PGPS datasets come with diagram captions written in formal languages (Lu et al., 2021), we can easily convert them to the AlphaGeometry-style descriptions. Given the translated descriptions, we utilize the rendering engine of AlphaGeometry to translate the target domain images into the source domain. With the translation, we can generate the same diagram in the source domain style. Fig. A4 provides examples of the diagram pairs with different styles. However, in some cases, the original description contains geometric premises that are unrecognizable from the diagram, such as $\angle ACB = 35.0$ in Fig. 1a. Therefore, we apply the same filtering process used in GeoCLIP to translate the AlphaGeometry-style descriptions into natural languages. Additional details in the translation process is described in Appendix D.1 and Fig. A5.

Formally, let $\mathcal{D}_S := \{(D_S^{(i)}, X_S^{(i)})\}_{i=1}^{N_S}$ be the diagram-caption pairs from source domain S, e.g., the synthetic diagrams, and let $\mathcal{D}_{T_j} := \{(D_{T_j}^{(i)}, X_{T_j}^{(i)})\}_{i=1}^{N_{T_j}}$ be the set of diagram-caption pairs of target domain T_j, e.g., the PGPS benchmarks. With the translation process described above, we can synthesize a style-transferred diagram-caption pair $(\hat{D}_{T_j}^{(i)}, \hat{X}_{T_j}^{(i)})$ for each diagram $D_{T_j}^{(i)}$ and caption $X_{T_j}^{(i)}$ in target domain T_j.

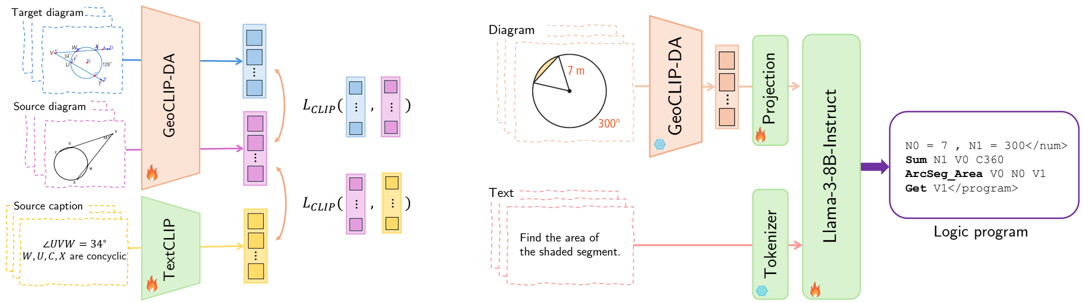
We adapt the domain by fine-tuning the vision encoder through the style-transferred diagram-caption pairs. Let $\hat{\mathcal{D}}_{T_j}$ be a collection of the original diagram and style-transferred captions, i.e., $\hat{\mathcal{D}}_{T_j} = \{(D_{T_j}^{(i)}, \hat{X}_{T_j}^{(i)})\}_{i=1}^{N_{T_j}}$, and let $\hat{\mathcal{D}}_{T_j S}$ be a collection of the original and style transferred diagram pairs, i.e., $\hat{\mathcal{D}}_{T_j S} = \{(D_{T_j}^{(i)}, \hat{D}_{T_j}^{(i)})\}_{i=1}^{N_{T_j}}$. The cross-domain adaptation objective is written as

$$\mathcal{L}_{\text{CLIP-DA}}(\mathcal{D}_S, \{\mathcal{D}_{T_j}\}_j, g, h) := \mathcal{L}_{\text{CLIP}}(\mathcal{D}_S, g, h) + \Sigma_j \mathcal{L}_{\text{CLIP}}(\hat{\mathcal{D}}_{T_j}, g, h) + \mathcal{L}_{\text{CLIP}}(\hat{\mathcal{D}}_{T_j S}, g, g), \quad (2)$$

where g and h are the vision and text encoders of GeoCLIP, respectively. Note that we do not use the original captions from the target domain, since our goal is to adapt the vision encoder to the target domain, not the text encoder.

5 Experiments

In this section, we evaluate the PGPS performance of our VLM equipped with the domain-adapted GeoCLIP on MathVerse (Zhang et al., 2024a). We compare its performance against established PGPS baselines. We also present ablation studies highlighting our VLM's strong visual feature recognition and resilience to domain shifts, both of which are facilitated by the adapted vision encoder.



```
N0 = 7 , N1 = 300</num>
Sum N1 V0 C360
ArcSeg_Area V0 N0 V1
Get V1</program>
```

(a) GeoCLIP-DA

(b) GeoDANO

Figure 3: The overall training process of GeoDANO. The GeoDANO training consists of two stages. (a) Using the CLIP objective, we first train GeoCLIP-DA by optimizing OpenCLIP's vision and text encoders on synthetic diagram-caption pairs and apply domain adaptation. (b) We then construct GeoDANO by combining GeoCLIP-DA with a projection layer and a language model. We train the projection layer and language model jointly in an end-to-end manner using diagram-text pairs annotated with logic programs.

5.1 Experimental settings and training details

Datasets. We use MathVerse (Zhang et al., 2024a) to measure the performance of VLMs. MathVerse is a benchmark designed to evaluate both the reasoning and visual-feature recognition capabilities of VLMs, covering plane geometry, solid geometry, and function problems. It is constructed by compiling problems from various sources, including Geometry3K (Lu et al., 2021), GeoQA (Chen et al., 2021), and GEOS (Seo et al., 2015). Each problem is presented in five variants: *text-dominant*, which provides all essential textual information for solving the problem; *text-lite*, which omits descriptive details, e.g., object shapes, from the text; *vision-intensive*, which removes certain textual conditions that can be inferred from remaining information; *vision-dominant*, which relocates numerical measurements, such as angles and lengths, from the text to the diagram; and *vision-only*, which offers only the diagram as input, embedding all text within the diagram. In the following experiments, we focus on plane geometry problems and exclude the vision-only task.

Training details. We describe the construction of our **geo**metric VLM with **d**omain-**a**gnostic visio**n** enc**o**der, named GeoDANO. Based on GeoCLIP developed in §4.1, we apply the domain adaptation to GeoQA and Geometry3K datasets. For the domain adaptation, we randomly sample 50 diagrams and translate the diagram and caption styles following the procedure described in §4.2. Finally, GeoCLIP is fine-tuned via Eq. (2). We name the GeoQA and Geometry3K adapted GeoCLIP as GeoCLIP-DA.

We combine LLama-3-8b-Instruct (Dubey et al., 2024) and GeoCLIP-DA to construct a VLM. The combined model is then fine-tuned again with the training set of GeoQA and PGPS9K to predict the solution program. For PGPS9K, we use the Geometry3K split. While previous works focusing on PGPS do not consider optical character recognition (OCR) from diagrams since the benchmark datasets, GeoQA and PGPS9K, provide necessary details in problem descriptions, numerical values can appear within diagrams in real-world settings. Therefore, we fine-tune GeoDANO with additional OCR capability by modifying the problem statements. Fig. 3 illustrates the whole pipeline of training GeoDANO, including the training of GeoCLIP-DA. Additional details about the modification process with hyper-parameter configurations can be found in Appendix E.

In addition, we unify the programming languages used in the solution programs of GeoQA and PGPS9K by converting the GeoQA language into the PGPS9K format. The unification makes the output of VLM consistent since both datasets use different types of formal languages.

Baselines. We use two different types of baseline models for the experiments: PGPS *specialist VLMs* and *generalist VLMs*. Specialist VLMs produce a solution program as an output of a given problem, and generalist VLMs produce a natural language solution as an output.

For the specialist VLMs, we test PGPSNet (Zhang et al., 2023), NGS (Chen et al., 2021), SCA-GPS (Ning et al., 2023), GeoFormer (Chen et al., 2022), UniMath-Flan-T5 (Liang et al., 2023), GeoX (Xia et al., 2024), and MAVIS (Zhang et al., 2024b). For GeoX, we use the two variants GeoX-Geo3K and GeoX-GeoQA, which are fine-tuned on Geometry3K and GeoQA, respectively. We mimic MAVIS by replacing the vision encoder of Geo-

Models	Text Dominant		Text Lite		Vision Intensive		Vision Dominant	
	Completion ↑	Top-10 ↑	Completion ↑	Top-10 ↑	Completion ↑	Top-10 ↑	Completion ↑	Top-10 ↑
PGPSNet	4.37	14.55	2.08	12.06	2.08	11.02	-	-
NGS	6.45	34.57	6.64	28.52	5.86	26.37	-	-
SCA-GPS	6.84	18.16	5.66	16.80	3.52	15.23	-	-
GeoFormer	16.22	32.85	16.84	30.77	13.10	29.11	-	-
UniMath-Flan-T5	17.88	32.43	16.42	30.56	13.93	28.27	-	-
GeoX-Geo3K	5.41	9.98	4.16	6.86	3.53	5.61	-	-
GeoX-GeoQA	**24.32**	37.42	17.26	32.43	13.51	16.25	-	-
MAVIS	21.83	42.62	15.80	39.50	12.68	35.55	3.54	10.83
GeoDANO (OC)	19.13	40.12	16.63	34.72	13.31	31.81	1.25	8.12
GeoDANO (GC)	20.37	41.79	18.09	38.25	15.80	35.34	5.62	19.38
GeoDANO (GC-D)	22.66	43.45	21.00	38.46	**18.30**	35.76	6.67	20.42
GeoDANO	23.70	**47.82**	**21.21**	**45.11**	18.09	**42.20**	**12.08**	**36.04**

Table 2: PGPS accuracy on MathVerse benchmark. We compare the performance of GeoDANO against PGPS specialist models, which generate a solution program as an output. GeoDANO-OC, -GC, and -GCD are three variants of our model with different encoders. Further details about these variants can be found in §5.3.

DANO with MAVIS-CLIP, while keeping other components, e.g., the projection layer architecture, language model, and training process, unchanged[1].

For the generalist VLMs, we test two GPT-4o variants (Hurst et al., 2024): gpt-4o-2024-11-20 and gpt-4o-mini-2024-07-18, the InternVL2.5 variants: 8B and 26B models (Chen et al., 2024), SPHINX-MoE (Liu et al., 2024), and Math-PUMA-DeepSeek-Math-VL-7B (Zhuang et al., 2025).

Evaluation metric. For each plane geometry problem, both the specialist VLMs and GeoDANO generate 10 outputs via beam search. Following Zhang et al. (2023), we then use completion accuracy and top-10 accuracy as our primary evaluation metrics. The completion accuracy assesses whether the first successfully executed solution from the beam is correct; the solutions are reviewed in beam order, and success is recorded if the first executable solution produces the correct answer. Top-10 accuracy examines all ten beam outputs, counting a success if any of these solutions yield the correct result upon execution. Note that, as described before, the specialist VLMs do not have OCR capability. For the evaluation, we feed the correct values to the outputs of these models by using the parser developed in Zhang et al. (2023). For the models that are trained in Chinese, i.e., NGS and SCA-GPS, we use problem descriptions translated by GPT-4o (Hurst et al., 2024).

To measure the performance of the generalist VLMs, we use multiple-choice questions instead of open-ended questions due to the difficulty in parsing the final answer from free-form text. We use the multiple-choice question provided in MathVerse as an additional input to each problem. We ask VLMs to produce the answer in a pre-specified form. We report the top-1 accuracy of these models. To compare GeoDANO against the generalist models, we use the same protocol used in Zhang et al. (2023) to measure the accuracy.

5.2 Results

Performance against specialist VLMs. In Table 2, GeoDANO shows the best performance in almost all the problem variants and metrics except the completion accuracy in the text-dominant task. Note that the specialist models cannot solve the vision-dominant problems since these problems do not contain variables representing numerical values, such as a length, in the problem description. When comparing the performance between text and vision-dominant tasks, the top-10 accuracy of GeoDANO on the vision-dominant task is higher than the top-10 accuracy of the specialist models on the text-dominant task, except for GeoX-GeoQA. Given that the two tasks use the same problem set, the result implies that GeoDANO performs better than the specialist models without having the geometric premises in the problem description. In other words, our vision encoder can extract geometric premises accurately from the visual information.

Performance against generalist VLMs. Table 3 reports the performance of generalist VLMs and GeoDANO on multiple choice questions. GeoDANO outperforms proprietary closed models, i.e., GPT-4o variants, and open-sourced models, i.e., the InternVL2.5 variants. Especially, the performance gap between GeoDANO and InternVL2.5-

[1] Reproducing MAVIS is not feasible due to the unavailability of both the trained model checkpoint and the complete dataset used for training.

	Text Dominant	Text Lite	Vision Intensive	Vision Dominant
GPT-4o	40.35	39.18	38.01	36.95
GPT-4o-mini	41.12	39.53	35.59	30.50
InternVL2.5-8B	38.30	36.26	35.09	21.99
InternVL2.5-26B	42.40	40.06	38.01	38.71
SPHINX-MoE	27.49	25.15	26.61	22.58
Math-PUMA	33.04	29.53	28.36	21.11
GeoX-GeoQA	**52.05**	45.91	37.43	-
GeoDANO	48.54	**49.71**	**41.81**	**39.30**

Table 3: Comparison between GeoDANO and generalist VLMs on multiple choice questions.

26B reflects the parameter efficiency of our VLM. While GeoDANO shows impressive results among the variants, the performance of GeoX-GeoQA degrades dramatically as the visual information moves from the text to the diagram. Our work is the first to show that the specialist can compete with the generalist in MathVerse.

5.3 Ablation studies

Variation of GeoCLIP. We perform a detailed empirical analysis to evaluate how effectively the GeoCLIP-style captions and the proposed domain adaptation technique improve GeoDANO's performance. Specifically, we compare GeoDANO against other VLMs trained on the GeoCLIP variants, including OpenCLIP (Radford et al., 2021) and the GeoCLIP without domain adaptation. We also test a variant of GeoCLIP trained with additional diagram-caption pairs from the target domains without having any filtering process. In this case, we utilize all the data in the training sets.

We show the experimental result in Table 2. GeoDANO-OC and GeoDANO-GC represent the VLM with OpenCLIP and GeoCLIP without domain adaptation, respectively. GeoDANO-GCD represents the GeoCLIP with additional unfiltered domain captions. GeoDANO outperforms other variants on most tasks, except the completion accuracy on the vision-intensive task.

OCR performance. We assess the accuracy of GeoDANO and its variants in OCR on the MathVerse diagrams, focusing on the vision-dominant task. We evaluate the OCR performance of the first executable solution program in top-10 VLM predictions. GeoDANO-OC, GeoDANO-GC, GeoDANO-GCD, and GeoDANO achieve 1.84%, 20.26%, 13.95%, and 46.58% accuracy, respectively. The result explains the accuracy improvement of GeoDANO in the vision-dominant task against other variants.

Models	PGPS9K		GeoQA	
	MR ↓	mAP ↑	MR ↓	mAP ↑
OpenCLIP	50.50	27.87	111.70	1.29
GeoCLIP	88.99	17.61	128.73	1.05
GeoCLIP-D	58.83	13.35	107.25	2.86
GeoCLIP-DA	**12.88**	**41.13**	**35.60**	**33.25**

Table 4: Domain adaptation analysis. We report the mean rank (MR) and mean average precision (mAP) of the test diagrams.

Domain adaptation analysis. We examine how effectively GeoCLIP-DA generalizes to new domains with different diagram styles. For this experiment, we compare the embedding similarity between two diagrams representing the same structure in different styles. To create the paired dataset, we use a similar process described in §4.2. Specifically, a total of 100 diagrams are sampled from the test sets of GeoQA and PGPS9K, and these samples are rendered in AlphaGeometry style through the diagram description.

For evaluation, we sample 100 diagrams from each of the target domain's training sets and compare the similarity against the original diagram via cosine similarity. We also compute the similarity between the style-transferred diagram and the original diagram. We report two metrics for test diagrams: the mean rank (MR) and the mean average precision (mAP) of the style-transferred diagram.

As reported in Table 4, GeoCLIP-DA produces similar embeddings for structurally equivalent diagrams, regardless of their stylistic differences. Fig. A6 visualizes the diagram embeddings of OpenCLIP and GeoCLIP-DA. As one can observe, the OpenCLIP embeddings are largely separated by the domain of the diagrams, whereas those of GeoCLIP-DA appear to capture and align with underlying visual features more effectively.

6 Conclusion

In this work, we propose a domain-agnostic PGPS method, GeoDANO, by implementing a synthetic data engine and proposing a contrastive learning framework with domain adaptation. We demonstrate the effectiveness of GeoDANO in visual feature perception at both VLM and vision encoder levels by evaluating on the MathVerse and through a newly proposed geometric feature recognition benchmark for vision encoders. Eventually, the reasoning ability in plane geometry problems is enhanced with the improved perceptual capabilities.

Limitations

In this work, we present a domain-agnostic VLM for PGPS by refining the vision encoder. Although our VLM performs strongly in recognizing visual features, its coverage remains limited to geometric premises. Building on the success of the synthetic data engine and contrastive learning, extending this combination to different kinds of visual features, e.g., sub-structures in molecular graphs (Kamoi et al., 2024), statistics from charts (Masry et al., 2022), and solid geometry, promises further improvements in recognition of VLM. Due to the limitations in the experimental environment, we are unable to test LLMs with more than 30B parameters.

References

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.

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.

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*.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. 2024. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. SimCSE: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

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.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

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*.

Diederik P Kingma. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, et al. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2024b. 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.

Dongyang Liu, Renrui Zhang, Longtian Qiu, Siyuan Huang, Weifeng Lin, Shitian Zhao, Shijie Geng, Ziyi

Lin, Peng Jin, Kaipeng Zhang, Wenqi Shao, Chao Xu, Conghui He, Junjun He, Hao Shao, Pan Lu, Yu Qiao, Hongsheng Li, and Peng Gao. 2024. SPHINX-x: Scaling data and parameters for a family of multi-modal large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 32400–32420. PMLR.

Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. 2022. A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11976–11986.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization. In *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.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*.

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.

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.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel HAZIZA, Francisco Massa, Alaaeldin El-Nouby, Mido Assran, Nicolas Ballas, Wojciech Galuba, Russell Howes, Po-Yao Huang, Shang-Wen Li, Ishan Misra, Michael Rabbat, Vasu Sharma, Gabriel Synnaeve, Hu Xu, Herve Jegou, Julien Mairal, Patrick Labatut, Armand Joulin, and Piotr Bojanowski. 2024. DINOv2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. Featured Certification.

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.

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.

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.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 815–823.

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.

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.

Yuhuai Wu, Albert Qiaochu Jiang, Wenda Li, Markus Norman Rabe, Charles E Staats, Mateja Jamnik, and Christian Szegedy. 2022. Autoformalization with large language models. In *Advances in Neural Information Processing Systems*.

Renqiu Xia, Mingsheng Li, Hancheng Ye, Wenjie Wu, Hongbin Zhou, Jiakang Yuan, Tianshuo Peng, Xinyu Cai, Xiangchao Yan, Bin Wang, et al. 2024. Geox: Geometric problem solving through unified formalized vision-language pre-training. *arXiv preprint arXiv:2412.11863*.

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*, 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.

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. 2024a. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, page 169–186, Berlin, Heidelberg. Springer-Verlag.

Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, et al. 2024b. Mavis: Mathematical visual instruction tuning with an automatic data engine. *arXiv preprint arXiv:2407.08739*.

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

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2024. Solving challenging math word problems using GPT-4 code interpreter with code-based self-verification. In *The Twelfth International Conference on Learning Representations*.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2025. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26183–26191.

Appendix

A Details of Related Work

A.1 Contrastive learning in PGPS

Contrastive learning is applied in diverse domains such as computer vision (Schroff et al., 2015) and natural language processing (Gao et al., 2021). In the context of PGPS, contrastive learning is employed to address domain-specific challenges. GeoX (Xia et al., 2024) applies contrastive learning to the adapter layer of the VLM to enhance formal language comprehension. GeoGLIP (Zhang et al., 2025) utilizes grounded language-image pre-training with synthetic diagram and junction, boundary triples. Other approaches train the vision encoder itself using the contrastive language-image pre-training (CLIP) (Radford et al., 2021) objective: LANS (Li et al., 2024b) aligns patch embeddings from a vision Transformer (ViT) with text token embeddings if they describe the same point. MAVIS (Zhang et al., 2024b) employs diagram–caption pairs generated by a synthetic engine for CLIP, where the captions contain all the information in the diagram.

A.2 Comparison between MAVIS and ours

Our contributions differ from MAVIS primarily in three aspects. Firstly, we introduce a systematic benchmark specifically designed for the quantitative analysis of vision encoders' capabilities in understanding geometric diagrams. This benchmark enables a fine-grained recognition evaluation across distinct geometric features, which is not addressed in MAVIS.

Secondly, we investigate the influence of caption style on vision encoder training by explicitly comparing GeoCLIP-style captions with MAVIS-style captions. MAVIS similarly employs extensive geometric diagram-caption pairs, namely MAVIS-Caption, where the captions include detailed geometric attributes such as object shape and connectivity. However, our empirical results comparing GeoCLIP and GeoCLIP (F ×), where GeoCLIP (F ×) uses captions incorporating all possible geometric premises, demonstrate that redundant geometric information within captions negatively impacts the vision encoder's recognition performance. Moreover, Table 1 shows that fine-tuning OpenCLIP with MAVIS-Caption with the setting as MAVIS yields significantly poorer performance on our visual geometric premise recognition benchmark compared to GeoCLIP and OpenCLIP.

Visual premises	Non-visual premises
• Perpendicularity	• Middle point
• Collinearity	• Congruency in degree
• Concyclicity	• Congruency in length
• Parallelity	• Congruency in ratio
• Angle measure	• Triangle similarity
• Length measure	• Triangle congruency
	• Circumcenter
	• Foot

Table A1: Geometric premises used in AlphaGeometry. *Visual premises* denotes the geometric premises which can be directly perceived from the diagram. *Non-visual premises* requires reasoning to be recognized.

Algorithm 1 Sampling process of the synthetic data engine

Input Geometric relations R, geometric objects O, number of clauses n_c

Output AlphaGeometry program c

```
Initialize points and clauses with the sampled object: $P, C \sim O$
for $i \leftarrow 1$ to $n_c$ do
    Generate points: $P_{\text{new}}$
    Sample relation and points: $r, P_{\text{old}} \sim R, P$
    Construct clause: $C_{\text{new}} = r(P_{\text{new}}, P_{\text{old}})$
    Update points and clauses: $P, C \leftarrow P \cup P_{\text{new}}, C \cup C_{\text{new}}$
Generate program with points and clauses: $c \leftarrow \text{Clauses2Program}(P, C)$
return $c$
```

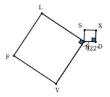
Finally, our work explicitly addresses the issue of domain shift across different diagram styles by proposing a few-shot domain adaptation technique, a critical problem not considered in MAVIS.

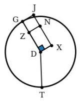
B Synthetic Data Engine

In this section, we provide the details of our synthetic data engine. Based on AlphaGeometry (Trinh et al., 2024), we generate synthetic diagram and caption pairs by randomly sampling an AlphaGeometry program with Algorithm 1. We visualize the AlphaGeometry program and diagram-caption pairs generation process in Fig. A2.

Examples for randomly sampled AlphaGeometry problems and their corresponding diagrams and lists of geometric premises are described in Fig. A1. The types of geometric premises that appear in our

AlphaGeometry problem:	Geometric premises:	Diagram:	GeoCLIP-style caption:

`x0 x1 = segment;`
`x2 = s_angle x0 x1 x2 122;`
`x3 x4 = square x1 x2;`
`x5 x6 = square x0 x1`
	∠DHV = 122°
	DH ⊥ DX . HV ⊥ HL
	HD = DX . VH = HL
	DH ∥ SX . HS ∥ DX . HV ∥ FL . FV ∥ HL
			∠DHV = 122° . DH ⊥ DX . HV ⊥ HL

`x0 x1 x2 x3 = rectangle;`
`x4 x5 = square x2 x3;`
`x6 = on_circle x1 x5`
	DX ⊥ DZ
	DX ∥ NZ . NX ∥ DZ . JN ∥ DZ. NZ ∥ GJ . GZ ∥ JN
	DT = DG . ZN = NJ
	T, J are concyclic
			DX ⊥ DZ . T, G are concyclic

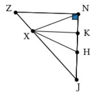
`x0 x1 x2 = iso_triangle;`
`x3 x4 = trisect x0 x1 x2;`
`x5 = intersection_lt x0 x1 x2 x3 x4`
	NZ ⊥ HJ
	∠JXN = ∠XNJ . ∠NXJ = ∠JNX . ∠X = ∠HXK .
	∠HXK = ∠KXN . ∠KXH = ∠NXK
	JX = JN .
	J, H, N are collinear . J, H, K are collinear . J, Z, K are collinear
			NZ ⊥ HJ

Figure A1: Example of randomly sampled AlphaGeometry problems. For each row, the first element describes the randomly sampled AlphaGeometry problem and the others are the geometric premises, diagram, and GeoCLIP-style caption that can be obtained from the AlphaGeometry problem. Note that the GeoCLIP-style caption can be obtained by filtering certain geometric properties, e.g., angle measure, perpendicularity, and concyclicity, from the geometric premises.

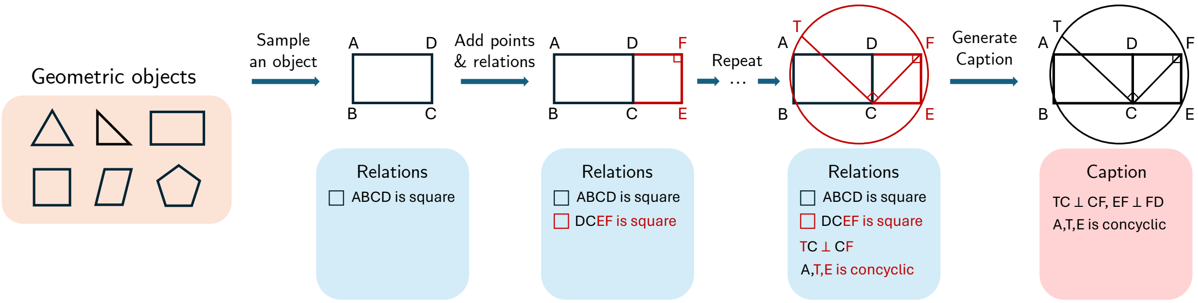


Figure A2: Illustration of the synthetic diagram-caption pairs generation. We first sample an object from a predefined object set. We then iteratively add points and relations to the existing primitives and relations. Finally, we generate a GeoCLIP-style caption based on the resulting primitives and relations.

synthetic data engine are listed in Table A1.

C Details of Benchmark

C.1 Generation details

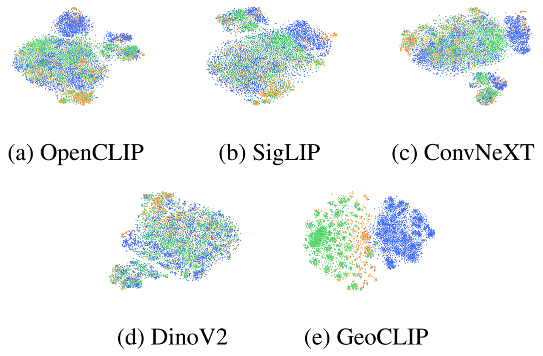
We describe the details of our benchmark generation process in detail. First, for each benchmark, we explicitly define key geometric premises using the formal language provided by AlphaGeometry. For instance, in the AngleDetection benchmark, we specify AlphaGeometry problems that guarantee the presence of angle ABC with degree between 15° and 75° in the diagram.

We then execute Algorithm 1 to generate diagrams and corresponding answers from these predefined geometric specifications. Specifically, Algorithm 1 first samples an initial AlphaGeometry problem containing the essential geometric premise, then incrementally adds random geometric primitives and relations to diversify the diagrams. Finally, the answer to the generated diagram is determined from the initial sampled AlphaGeometry problem. Importantly, in the TwoLines, SquareShape, and AngleDetection benchmarks, the answer obtained from the initial AlphaGeometry problem remains unchanged and visually present despite adding other geometric elements. In contrast, for ObjectShape and Concyclic benchmarks, no additional geometric elements are introduced, further ensuring the accuracy of the answer. Consequently, the answers derived from these formal specifications consistently match the generated diagrams.

C.2 Role of the textual description

During the evaluation of the vision encoders, only the visual diagram serves as input to the model, and no textual information is provided during training or inference. Specifically, the evaluation is conducted using a linear probing approach, wherein the parameters of the vision encoder remain frozen, and only a linear classifier, initialized randomly, is trained atop the visual embeddings produced by the encoder.

Here, the textual questions from our benchmarks are implicitly represented in the classification la-



(a) OpenCLIP (b) SigLIP (c) ConvNeXT

(d) DinoV2 (e) GeoCLIP

Figure A3: The embeddings of the vision encoders on the diagrams of TwoLines task. We visualize the embeddings of the vision encoders on the diagrams of TwoLines task. The blue, orange, and green dots are the diagrams where the two lines AB and BC are collinear, perpendicular, and otherwise, respectively.

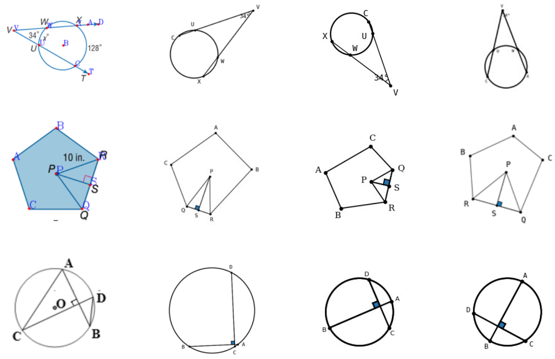


Figure A4: Examples of diagram pairs curated for domain adaptation. For each row, the first diagram is from the target domain, and the remaining diagrams are from the source domain. To generate source domain diagrams, we translate the target diagram by our diagram generator with the textual description of the target image.

bels assigned to each diagram. For instance, the textual question "How are lines AB and BC related?" corresponds directly to classification labels such as "Perpendicular," "Collinear," or "Neither." These labels are used as supervision signals to train the linear classifier. Thus, while the vision encoder receives no explicit textual input, the questions' semantics are reflected indirectly through the classification labels.

C.3 Training details

To evaluate the visual feature perception of the vision encoder, we utilize a linear probing approach, which involves freezing the vision encoder parameters and training a simple linear classifier on top of its features.

We train the linear classifier on the training set of each task for 50 epochs with batch size 128 and learning rate 1e-4. We use Adam optimizer for optimization.

C.4 Visualization of the vision encoders

We visualize the embeddings of the vision encoders used in §3.2 at Fig. A3.

D GeoCLIP-DA

D.1 Details of the translation process

Note that the formal language is mentioned solely as an optional tool to accelerate the translation process when available, but it is not essential. The primary purpose of diagram translation in our method is to generate synthetic diagrams that visually resemble real-world diagrams. Since the objective is visual feature alignment rather than formal semantics, diagrams can effectively be translated manually by simply recreating the visual structure.

While manual translation might appear time-consuming, the effort is minimal and feasible in practice. Specifically, we manually translated only around 50 diagrams per domain, which required less than 3 hours in total. This modest effort substantially improved our model's cross-domain generalization performance. Thus, manual translation without formal annotations is not only practical but also highly beneficial for domain adaptation. The process of the translation is illustrated in Fig. A5

D.2 Domain adaptation data

We adopt GeoCLIP to the two PGPS benchmarks: GeoQA (Chen et al., 2021) and PGPS9K (Zhang et al., 2023). For PGPS9K, we use the Geometry3K split. Fig. A4 shows the pairs used to adapt the domain of GeoCLIP.

D.3 Training details

We start from OpenCLIP (Radford et al., 2021), a pre-trained model where the architecture is ViT-L/14 with image resolution 336 × 336. To train OpenCLIP, we use total of 200,000 diagram-caption pairs generated with our synthetic data engine. For the domain adaptation to GeoQA and Geometry3K datasets, we randomly sample 50 diagrams and translate the diagram and caption styles following the procedure described in §4.2. Finally, GeoCLIP is fine-tuned via Eq. (2). We name the GeoQA and Geometry3K adopted GeoCLIP as

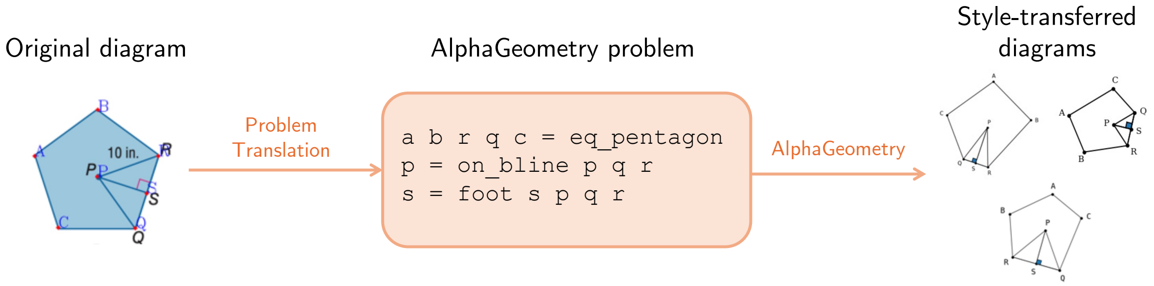


Figure A5: Illustration of the translation process for domain adaptation. We first translate the given geometric diagram from the target domain into an AlphaGeometry problem. Once we obtain an AlphaGeometry problem that accurately captures the visual premises of the target diagram, we can generate multiple diagrams sharing the same visual geometric premises in the AlphaGeometry style.

GeoCLIP-DA.

We set the batch size for the source domain diagram-caption pairs to 256. For the domain adaptation parts, i.e., applying CLIP on the diagram-caption pairs and the diagram pairs of target domains, we vary the batch size to 32. We set weight decay to 0.2. We optimize for 50 epochs using Adam optimizer (Kingma, 2014) and a cosine annealing scheduler with 2,000 warmup steps, and the maximum learning rate is set to be 1e-4. We train the model with eight RTX3090 GPUs for approximately 24 hours.

D.4 Visualization of GeoCLIP-DA embeddings

We compare the embeddings between GeoCLIP-DA and OpenCLIP in Fig. A6.

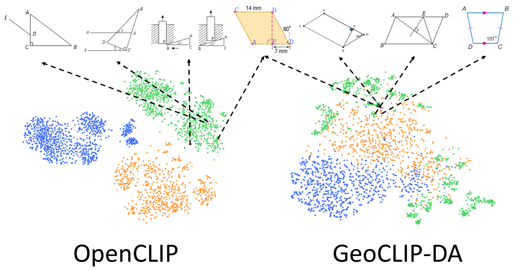


Figure A6: Visualization of OpenCLIP and GeoCLIP-DA embeddings. The orange, green, and blue dots represent PGPS9K, GeoQA, and synthetic diagrams, respectively. In the top row, the three diagrams on the left and right are those with the highest cosine similarities to the center under OpenCLIP and GeoCLIP-DA, respectively.

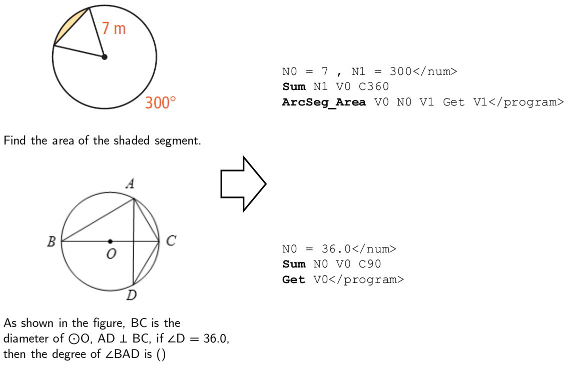
E GeoDANO

E.1 Modification of training data

Our fine-tuning strategy differs slightly from previous works (Chen et al., 2022; Zhang et al., 2023; Xia et al., 2024). Here, we clarify the difference between our approach and previous approaches. In previous works, the VLM is trained to produce the solution program given a diagram and problem description, as shown in Fig. 1. An interesting observation from GeoQA and PGPS9K datasets is that the numerical measurements, such as angles, lengths, and volumes, are not written in the problem description but given as additional conditions, and the numerals are substituted as a variable in the problem description as shown in Fig. 1a. Therefore, the VLM only needs to produce the solution program without having optical character recognition (OCR) from the diagram. The variables are automatically substituted with the actual numbers when the program is executed. Therefore, the vision encoders do not need to learn OCR from the image.

However, this approach cannot be generalized to a broader class of problems where the numerals are embedded in the diagram instead of being written in the problem description. Some variants of MathVerse, such as the vision-dominant problems, fall into this category as well. To incorporate OCR into the solution of the problem, we modify some problem statements in the training set, such that the numerical measurements are only shown in the diagram and not in the statements. We further modify the solution problem so that the solution contains OCR results as a part of the final output. Finally, we unify the language of the solution programs used in GeoQA and PGPS9K by converting GeoQA programs into PGPS9K format. The unification makes the output of VLM consistent since both datasets use different types of formal languages.

Fig. A7 shows examples of the modified input pairs and solutions, where the first problem statement does not have numerical measurements, and



Find the area of the shaded segment.

```
N0 = 7 , N1 = 300</num>
Sum N1 V0 C360
ArcSeg_Area V0 N0 V1 Get V1</program>
```

As shown in the figure, BC is the diameter of ⊙O, AD ⊥ BC, if ∠D = 36.0, then the degree of ∠BAD is ()

```
N0 = 36.0</num>
Sum N0 V0 C90
Get V0</program>
```

Figure A7: Examples of the training data for GeoDANO. While previous PGPS models require only predicting the solution steps and assuming the numerical values are explicitly given, GeoDANO is trained to predict both the solution steps and the numerical values in the diagram and text.

the OCR results are in the part of the output solution program.

E.2 Training details

Architectural details. We begin by summarizing the architecture of our VLM, a combination of a vision encoder and a language model. For the vision encoder, we use GeoCLIP-DA, with a two-layer MLP of GeLU activation as the projection layers following LLaVA-OneVision (Li et al., 2024a). For the language model, we employ LLama-3-8B-Instruct (Dubey et al., 2024). For a given diagram and question pair in PGPS, we feed the vision encoder with the given diagram, and then the output of the encoder is used as an input token of LLM through the projection layer. The question text is then fed into the LLM, followed by the diagram embedding.

Training approach. With the modified training data, we apply supervised fine-tuning on the VLM, i.e., the gradient only flows through the prediction of numerical values and solution steps, not the diagram and text. During the training of GeoDANO, the parameters of the vision encoder, i.e., GeoCLIP-DA, are frozen and remain unchanged. The projection layer, which maps visual embeddings to language model inputs, is randomly initialized and trained from scratch simultaneously with the language model.

Hyper-parameters. We train the VLM with AdamW optimizer (Loshchilov and Hutter, 2019) and cosine annealing scheduler with warmup ratio 0.03 and maximum learning rate 1e-5. We use LoRA (Hu et al., 2022) with rank 128. We set the batch size to 16 and train with 5 epochs. We train the VLM with four A100-80GB GPUs for approximately 24 hours.