Plane Geometry Diagram Formalization via Vision-Language Models

Xiaoteng Cui [1] **Yi Liu** [1 *]

Abstract

Large models such as vision language models (VLMs) have demonstrated robust world knowledge comprehension, inspiring advancements in automated mathematical problem-solving. In the domain of geometry problem-solving, the intricate and diverse abstract relationships inherent in geometry diagrams present significant challenges for leveraging large models. To enhance the accuracy of geometry problem-solving, we analyze existing problem-solving paradigms and propose leveraging VLMs for enhanced diagram autoformalization accuracy. First, we construct a multimodal instruction-tuning dataset named **G**eometry**D**iagram**F**ormalization**86K** (GDF86K) through data augmentation based on algebraic commutativity in the Geometry3K dataset. This dataset contains over 86,000 image-caption pairs to facilitate training of diagram autoformalization models. Utilizing GDF86K, we conduct supervised fine-tuning to implement Geo-TinyLLaVA, a vision-language model specialized in geometry diagram autoformalization. When input diagrams with complete point annotations, Geo-TinyLLaVA outperforms the conventional Inter-GPS diagram parser in autoformalization performance and can serve as a plugin to enhance the problem-solving accuracy of the geometry problem-solving system. Code and data are available at https://github.com/1509cxt/Geo-TinyLLaVA.

1. Introduction

The performance of large models in mathematical comprehension has garnered significant attention. Notably, the evaluation of large models with visual capabilities in the mathematical domain (Lu et al., 2024; Zhang et al., 2025) heavily relies on assessing their understanding of mathematical problems that incorporate visual information, such as plane geometry problems.

Advancements in automatic plane geometry problem-solving using Transformer-based large models can be categorized into two primary approaches:

- Large models functioning directly as complete solvers (e.g., G-LLaVA (Gao et al., 2023)), representing an end-to-end paradigm.
- Large models serving as auxiliary components to a mechanical symbolic deduction engine (e.g., AlphaGeometry (Trinh et al., 2024), Inter-GPS (Lu et al., 2021)), representing a non-end-to-end paradigm.

In the paradigm where large models directly as solvers, publicly available experimental results show that fine-tuned vision-language models like G-LLaVA exhibit good geometry problem-solving capabilities, surpassing many popular foundation models (e.g., LLaVA (Liu et al., 2023b), GPT-4V (OpenAI, 2023)) and the average human performance. However, enhancing the reasoning and computational capabilities of large models may require more meticulously designed post-training methods. Additionally, neural networks inherently have shortcomings in terms of overall stability and interpretability. Therefore, we are more inclined to rely on formal systems for reasoning and computation to ensure a certain level of mathematical correctness.

Large models may currently be best suited to an auxiliary role in the domain of plane geometry. We observe that AlphaGeometry employs a specifically designed mechanical deduction engine to handle the reasoning process for solving geometry problems, while using large models to assist in auxiliary construction, thereby achieving noteworthy performance on Olympiad-level geometry problems.

Mechanical deduction engines for solving plane geometry problems heavily depend on the accuracy of the formalized geometry information provided. For instance, AlphaGeometry relies on human experts to manually formalize the textual information of a limited set of test geometry problems. As a complete system for solving geometry problems, Inter-GPS incorporates a symbolic deduction engine along with two front-end modules: a text parser and a diagram parser,

*Corresponding author. [1]Beijing Key Laboratory of Traffic Data Mining and Embodied Intelligence, Beijing Jiaotong University, Beijing, China. Correspondence to: Yi Liu <yiliu@bjtu.edu.cn>.

The second AI for MATH Workshop at the 42nd International Conference on Machine Learning, Vancouver, Canada. Non-archival.

specifically designed to parse problem information as input to the deduction engine.

To enable the symbolic deduction engine to utilize more accurate formalized geometry information and thereby improve problem-solving accuracy, this work focuses on leveraging vision-language models to achieve more precise plane geometry diagram autoformalization.

A geometry diagram encompasses its fundamental geometric primitives (such as points, lines, circles, etc.), the relative coordinates of these points, and various textual elements and symbols that establish geometric relationships among all points. The task of geometry diagram parsing involves inputting a geometry diagram and outputting a structured textual representation of all the geometric information contained within the diagram, as described above. See Figure 1 for an example of a geometry diagram and its parsing result.

To enable more accurate autoformalization of geometry diagrams, we first construct a multimodal instruction-tuning dataset named GDF86K. This dataset is developed through quality refinement and algebraic commutativity-based data augmentation of the Geometry3K (Lu et al., 2021) training and validation sets. It comprises four subsets, each containing 86,080 image-caption pairs, where each instance includes a geometry diagram and its corresponding formal language description of geometric relationships. Leveraging GDF86K, we fine-tune a vision-language model with geometry diagram autoformalization capabilities, named Geo-TinyLLaVA. Specifically, given complete point annotation within diagrams, our Geo-TinyLLaVA demonstrates superior performance in geometry diagram autoformalization compared to the Inter-GPS diagram parser. The model can serve as a plug-in component to enhance the parsing capability of the Inter-GPS diagram parser, thereby improving the overall geometry problem-solving accuracy of the Inter-GPS system.

2. Related Work

Vision-Language Model. Recent years have witnessed remarkable advancements in Vision-Language Models (VLMs), more commonly referred to as Multimodal Large Language Models (MLLMs). The development of large language models (Brown et al., 2020; Ouyang et al., 2022; Touvron et al., 2023) has been significantly accelerated by the introduction of the Transformer architecture (Vaswani, 2017), thereby facilitating vision-language joint representation learning (Radford et al., 2021; Li et al., 2022; 2023). The continuous evolution of large-scale pretrained models has enabled more effective training paradigms for multimodal tasks, leading to the proliferation of diverse MLLMs (Liu et al., 2023b;a; 2024). Concurrently, novel frameworks (LLaMA-Factory (Zheng et al., 2024), TinyLLaVA Factory (Jia et al., 2024)) for efficient multimodal model training have been proposed, significantly reducing computational resource requirements and implementation complexity for associated tasks. However, current foundation models exhibit suboptimal performance in mathematical visual comprehension tasks (Zhang et al., 2025).

Geometry Diagram Parsing: Detection and Formalization. As mentioned earlier, the prerequisite for a mechanical deduction engine to correctly solve geometry problems is that the problem is accurately represented in certain machine language that the deduction engine need. Both the text and the geometry diagram of the problem need to be parsed to obtain the parsing result. Among these, parsing the geometry diagram is particularly challenging, as the vast amount of geometric information embedded in the diagram leads to a highly detailed parsing result. Any error in parsing results could cause the subsequent failure of the deduction engine. Previously, the conventional Inter-GPS diagram parser operated under a non-end-to-end paradigm: it first used Hough Transform (Stockman & Shapiro, 2001) to extract basic geometric elements, then employed an object detection network to extract the image regions of text and symbols, which were passed to an OCR tool. Finally, the geometric relationships were optimized based on the Euclidean distances between geometric elements and text-symbols. The parsing result (as shown in Figure 1) produced by this paradigm can be divided into two parts:

- Information about geometric elements such as points, lines, and circles, as well as the relative coordinates of points. This task can be considered a detection task for image segments, belonging to a mid-level vision task.
- Geometric relationships described in formal language (also known as logic forms), which use predicate logic to construct multiple literals that extract all the geometric relationships in the diagram. This task is categorized as a high-level vision task.

However, in generating logic forms, the conventional Inter-GPS diagram parser's paradigm falls short in terms of semantic understanding of the image. This work introduces a vision-language model that can construct more accurate logic forms, assisting the Inter-GPS diagram parser in producing better parsing results.

3. Data Preparation

3.1. Observations on the Dataset Geometry3K

In certain plane geometry problems, the original diagram may lack labels for some geometric points. However, during formalization, each point might be referenced; thus, every point in the diagram requires a consistent and explicit label.

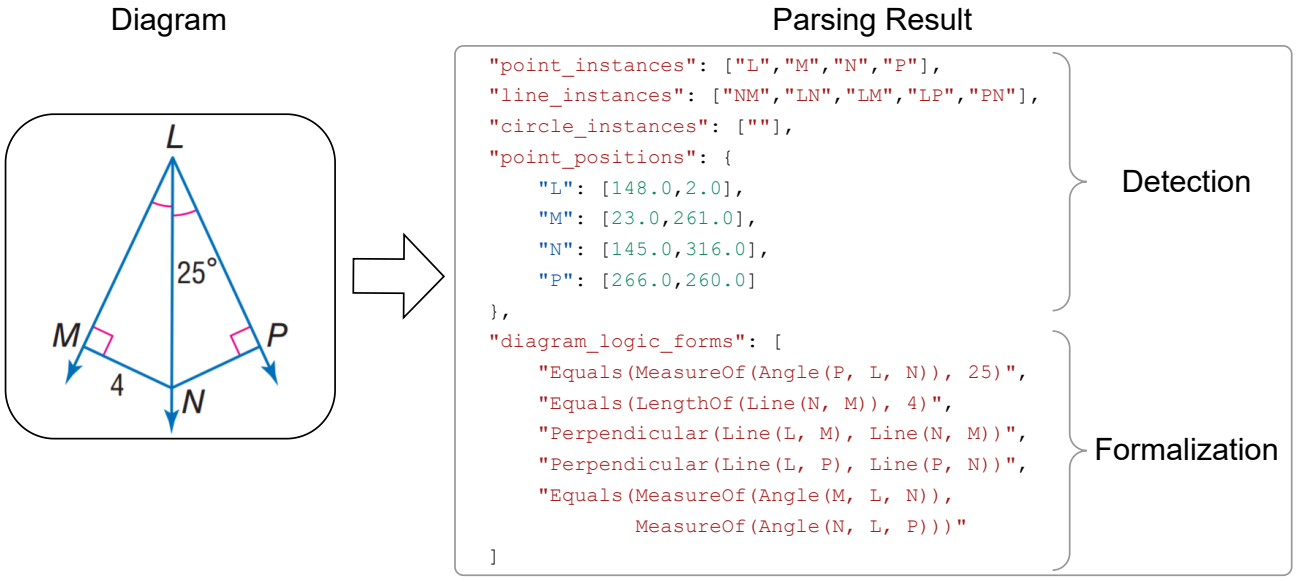


Figure 1. An example of a diagram and its parsing result from Geometry3K(Lu et al., 2021). On the left is the diagram to be parsed, while the right side shows its corresponding parsing result.

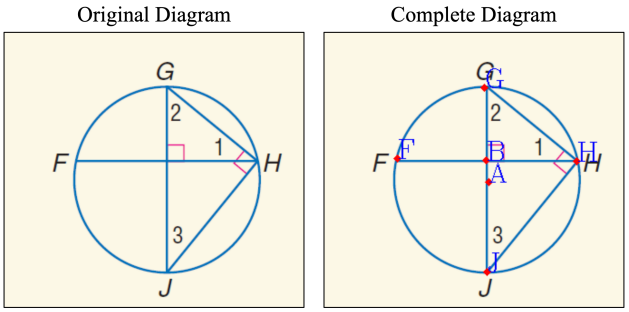


Figure 2. An example of an original diagram and its corresponding complete diagram in Geometry3K(Lu et al., 2021).

To achieve consistent and comprehensive geometric point annotations in the Geometry3K dataset, the annotation team re-labeled all geometric points in each problem's original diagram using blue uppercase letters. This approach resulted in the re-annotation of already labeled points and the identification of previously unlabeled points. The resulting diagram is referred to as `complete diagram` in this paper. Consequently, each problem in the Geometry3K dataset includes two images: the `original diagram` and `complete diagram`. See Figure 2 for an example of these two types of diagrams.

In Figure 2, based solely on the original diagram, it is challenging to deduce that the intersection of FH and GJ is labeled as point B, and the center of the circle is labeled as point A in the ground truth. For the unlabeled geometric points in the original diagram, it is difficult to infer the corresponding labels in the ground truth.

Our vision-language model also faces challenges in correctly naming unlabeled geometric points during formalization. Therefore, we predominantly utilize diagrams with complete point annotations as visual inputs for our model. Only when all geometric points in the original diagram are labeled do we provide the original diagram to the vision-language model.

For a geometry problem, it is preferable that its diagram assigns consistent names to all existing geometric points, facilitating the writing of solutions and the grading process. If a diagram lacks labels for certain points, completing these annotations can be done by the problem creator or considered within the scope of the detection task in diagram parsing.

3.2. Data Quality Refinement

While verifying the completeness of geometric point annotations in the Geometry3K diagrams, we identified certain expression errors in the logic forms of some diagrams that could be corrected (examples are provided in Appendix A).

We rectified these errors in the diagram logic forms, which consequently improved the problem-solving accuracy of Inter-GPS's symbolic solver (deduction engine) on the Geometry3K test set (see Table 1). Unless otherwise specified, all data referenced in this work refer to the dataset Geome-

try3K after our quality refinement process.

Table 1. Problem-solving accuracy of the Inter-GPS symbolic solver when directly using the ground truth parsing results from the Geometry3K test set (601 problems in total). The accuracy does not include cases where the solver failed and randomly selected one of the four multiple-choice options.

Data used	Accuracy
Uncorrected Data	67.22%
Corrected Data	**69.05%**

3.3. Instruction Data Construction

The training data for the Inter-GPS diagram parser is sourced from the training and validation sets of Geometry3K, comprising a total of 2,401 problems. This work also makes exclusive use of these data.

We employed heuristic methods, such as Hamming distance based on Average-hashing (Buchner, 2025) values of images, to deduplicate images in the training and validation sets of Geometry3K. Additionally, we manually filtered out data where unusable diagrams led to uncorrectable errors in their logic forms (examples are provided in Appendix B).

Following the aforementioned collation process, we retained a total of 1,345 problems from the Geometry3K training and validation sets for training the vision-language model.

3.3.1. Random Shuffle (RS)

When representing a diagram using formal language, multiple literals are generated to express geometric relationships. These literals are essentially conjunctive, and due to the commutativity of the conjunction operator, their order is inconsequential. Therefore, for a given diagram, we can apply a random shuffle to the existing literal list, producing different sequences that remain semantically equivalent. See Figure 3 for an example of data augmentation using the Random Shuffle technique.

Given a `(diagram, literal list)` pair containing n literals, this data augmentation technique allows for the generation of up to $n!$ distinct literal lists.

3.3.2. Predicate Commutativity Transformation (PCT)

Inspired by the performance evaluation methodology of the Inter-GPS diagram parser, we identify commutative properties in specific arguments of predicates and functions within its formal language definition. Based on the algebraic commutativity, individual literals may possess multiple equivalent representations conveying identical geometric information. For instance, the literal *"Perpendicular(Line(C, A), Line(B, A))"* exhibits 8 distinct equivalent forms, including *"Perpendicular(Line(A, B), Line(A, C))"* as a symmetric counterpart.

A single diagram typically corresponds to multiple literals, and applying Predicate Commutativity Transformation to each literal yields numerous distinct literal lists through combinatorial expansion. Notably, the Predicate Commutativity Transformation and Random Shuffle data augmentation techniques are orthogonal and can be applied concurrently for enhanced data diversity. For a more comprehensive example, refer to Figure 3.

4. Model Architecture and Training

4.1. Model Architecture

Our model, Geo-TinyLLaVA, utilizes the TinyLLaVA (Jia et al., 2024) framework based on the LLaVA (Liu et al., 2023b) architecture. An overview of our Geo-TinyLLaVA is depicted in Figure 4: it comprises a Large Language Model (LLM) F_θ, a modality connector P_ϕ, and a vision tower G_φ, where θ, ϕ, and φ are learnable parameters. The model takes as input a diagram and a fixed textual instruction for autoformalization, outputting a text sequence as formal language literals.

During the forward process, given an image $\mathbf{X_v}$ and a language instruction $\mathbf{X_q}$, the vision tower G_φ first encodes the image to obtain an embedding, which is then mapped to the LLM's dimension through the modality connector P_ϕ. The resulting image token embedding and the text embedding obtained by tokenizing the language instruction, are concatenated and input into the LLM. The LLM iteratively performs next-token prediction to generate output tokens. These output tokens are subsequently decoded by the tokenizer to produce the language response $\mathbf{Y}$.

4.2. Model Training

We train our Geo-TinyLLaVA in one stage of supervised fine-tuning. Given a single-turn conversation consisting of the image $\mathbf{X_v}$, the language instruction $\mathbf{X_q}$ and the target response $\mathbf{Y_t} = \{\mathbf{y}_i\}_{i=1}^N$ with sequence length of N, we compute the probability of generating $\mathbf{Y_t}$ as:

$$p(\mathbf{Y_t}|\mathbf{X_v}, \mathbf{X_q}) = \prod_{i=1}^{N} F_\theta(\mathbf{y}_i|P_\phi \circ G_\varphi(\mathbf{X_v}), \mathbf{X_q}) \quad (1)$$

and maximize its log-likelyhood autoregressively as training objective:

$$\max_{\phi,\theta,\varphi} \sum_{i=1}^{N} \log F_\theta(\mathbf{y}_i|P_\phi \circ G_\varphi(\mathbf{X_v}), \mathbf{X_q}) \quad (2)$$

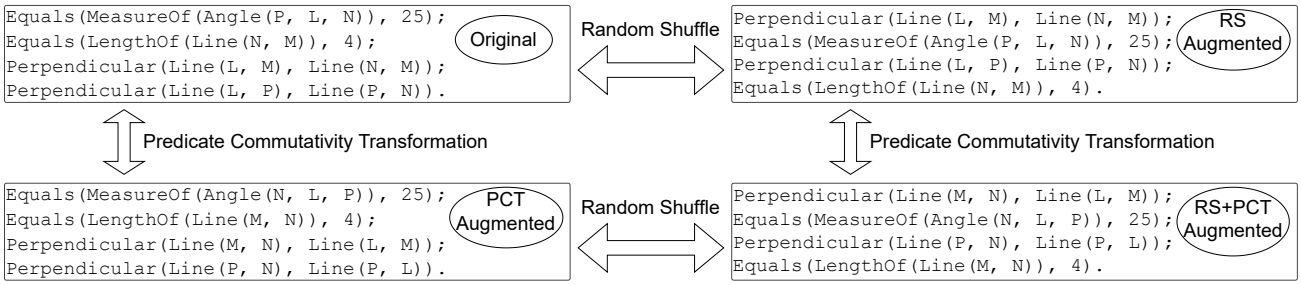


Figure 3. An example illustrating the application of the data augmentation techniques, Random Shuffle and Predicate Commutativity Transformation. The **Original** logic forms are selected from those presented in Figure 1. The **RS augmented**, **PCT augmented**, and **RS+PCT augmented** logic forms represent the results of applying Random Shuffle only, Predicate Commutativity Transformation only, and both techniques combined, respectively.

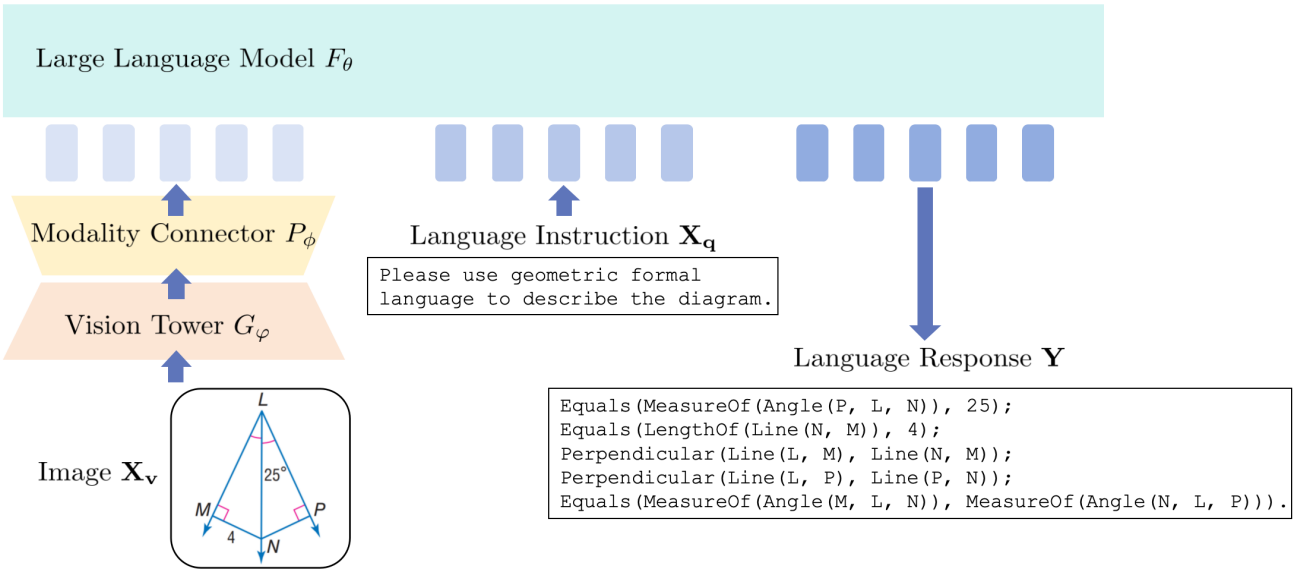


Figure 4. Overview of our Geo-TinyLLaVA.

5. Experiments

5.1. Setup

Evaluation Protocols. We employ two complementary evaluation protocols: one (**protocol 1**) directly assesses the autoformalization performance, and the other (**protocol 2**) evaluates the symbolic solver's problem-solving accuracy based on the parsing results derived from the model's autoformalization output.

To directly evaluate the autoformalization performance, we consider the commutative properties of predicates discussed in Section 3.3.2. We could match the formal language literals generated by the model for a given diagram with its ground truth literals, considering predicate commutativity equivalence. Metrics such as precision and recall are then calculated based on the number of correct literal matches.

Evaluating the problem-solving accuracy involves executing the solver on the complete parsing results of geometry problems. The textual component of a problem is processed using the Inter-GPS text parser to obtain text logic forms. The diagram parsing result comprises detection and formalization components: the detection part is obtained using the Inter-GPS diagram parser, while the formalization part is produced by the model.

Notably, when the symbolic solver fails to produce a solution, it randomly selects one of the four multiple-choice options. In such cases, we uniformly account for this chance accuracy in our evaluation.

Dataset. All instruction tuning data is constructed from the

1,345 curated problem instances (from training and validation sets from Geometry3K) described in Section 3.3. For each problem instance's diagram, we sample 64 literal lists through three augmentation strategies: 1) Random Shuffle (RS), 2) Predicate Commutativity Transformation (PCT), and 3) combined RS and PCT, yielding three augmented subsets each containing 86,080 `(diagram, literal list)` pairs. Additionally, we create a control group by replicating the original 1,345 data pairs 64 times without augmentation. These 4 subsets collectively constitute our instruction tuning dataset GDF86K.

For the evaluation data, we employ the Geometry3K test set containing 601 problems. It is crucial to reiterate that the task of identifying unnamed geometric points in diagrams and predicting their human-annotated ground truth labels falls outside the scope of formalization. For **protocol 1**, we evaluate all 601 problems, using diagrams with complete point annotations from Geometry3K when original diagrams lack full point labels. For **protocol 2**, we selected 259 test problems where all geometric points are explicitly labeled in original diagrams to facilitate ablation studies. These 259 diagrams ensure alignment in point labeling objectives across three parties: the conventional Inter-GPS diagram parser, our vision-language model, and human-annotated ground truth.

Implementation Details. We conduct supervised fine-tuning on the TinyLLaVA-Phi-2-SigLIP-3.1B pretrained model from the TinyLLaVA Factory (Jia et al., 2024), which integrates Microsoft Phi-2-2.7B (Javaheripi et al., 2023) as the language model, Google SigLIP ViT (Zhai et al., 2023) as the vision tower, and a two-layer Multi-Layer Perceptron (MLP) as the modality connector. The fine-tuned model Geo-TinyLLaVA-Phi-2-SigLIP-3.1B (hereafter abbreviated as Geo-TinyLLaVA for conciseness) is derived through this training process. During fine-tuning, all full parameters of the language model, vision tower, and modality connector are optimized with a learning rate of 1e-5, batch size of 1, and 1 training epoch across 4 NVIDIA A100 (40GB) GPUs.

For inference and evaluation, we standardized the decoding parameters of Geo-TinyLLaVA as follows: temperature set to 0, beam size of 4, and a maximum of 1024 new tokens.

5.2. Geometry Diagram Autoformalization

By performing literal matching with the ground truth, we compare the diagram autoformalization performance of our Geo-TinyLLaVA model against the Inter-GPS diagram parser, as presented in Tables 2 and 3. It is evident that Geo-TinyLLaVA surpasses the Inter-GPS diagram parser across all evaluation metrics in both tables. Furthermore, Geo-TinyLLaVA models fine-tuned with augmented data exhibit even better performance.

5.3. Geometry Problem Solving Based on Inter-GPS

In the geometry problem-solving system Inter-GPS, the execution of the symbolic solver relies on a structurally complete diagram parsing result. The diagram parsing result consists of a detection component and a formalization component. In evaluating problem-solving accuracy, we assessed the impact of different sources for the detection component by using both the detection function of the Inter-GPS diagram parser and the diagram detection ground truth. This allowed us to compare the problem-solving accuracy of the entire Inter-GPS system when utilizing autoformalization results from our Geo-TinyLLaVA and the Inter-GPS diagram parser, as shown in Tables 4 and 5.

In terms of symbolic solver problem-solving based on model's diagram autoformalization results, It reveals that our Geo-TinyLLaVA consistently outperforms the Inter-GPS diagram parser. Notably, the Geo-TinyLLaVA trained with data augmented solely through Predicate Commutativity Transformation data augmentation technique provides the most significant improvement in problem-solving accuracy within the Inter-GPS system. This observation may suggests that the Random Shuffle data augmentation technique may not substantially enhance data diversity.

Furthermore, the problem-solving performance of the autoformalization by the Inter-GPS diagram parser in these two tables indicates that using diagram detection ground truth (which is more accurate) led to a decrease in problem-solving accuracy. Upon detailed investigation of the solver execution process, we identified the primary cause: problems that originally resulted in solver failures when using detection information from the Inter-GPS diagram parser (where the system would randomly select one of four options, yielding a 25% accuracy rate) became more prone to incorrect computations when provided with diagram detection ground truth. This increased the likelihood of the solver attempting more extensive computations and deductions, ultimately producing incorrect results (leading to a 0% accuracy rate for these problems).

5.4. Comparison on Quality of Generated Formal Language Literals

During the experimental evaluation, it was observed that, in certain test problems where correct solutions were achievable with literals generated by our Geo-TinyLLaVA, the symbolic solver failed to derive correct answers when utilizing the formal language literals generated by Inter-GPS diagram parser and the ground truth annotations from the Geometry3K dataset. Specifically, as illustrated in Figure 5, both the Inter-GPS parser and the ground truth annotations exhibited ambiguity regarding whether the circle center A lies on chord GJ. While they implied this positioning through angle or arc annotations, they did not explic-

Table 2. Average value of autoformalization performance metrics over the formalization results on each evaluation data. This table compares the performance metrics of Geo-TinyLLaVA models trained with the four groups of instruction tuning data mentioned in Section 5.1 and the Inter-GPS diagram parser in diagram autoformalization. Subsequent experimental result tables follow this convention without further elaboration.

Autoformalization model	Precision	Recall	F1 Score	IoU
Inter-GPS diagram parser	61.48%	71.32%	66.03%	55.80%
Geo-TinyLLaVA (w/o)	81.80%	80.98%	81.39%	75.68%
Geo-TinyLLaVA (RS)	83.77%	84.46%	84.11%	78.51%
Geo-TinyLLaVA (PCT)	83.56%	82.55%	83.05%	77.76%
Geo-TinyLLaVA (RS+PCT)	**85.59%**	**85.68%**	**85.63%**	**80.61%**

Table 3. Autoformalization performance distribution across evaluation thresholds. The table presents the percentage of test cases achieving four literal-matching thresholds: **Totally Same** (F1 score = 100%), **Perfect Recall** (recall = 100%), **Almost Same** (F1 score >= 75%), **Likely Same** (F1 score >= 50%).

Autoformalization model	Totally Same	Perfect Recall	Almost Same	Likely Same
Inter-GPS diagram parser	32.61%	42.43%	45.09%	67.05%
Geo-TinyLLaVA (w/o)	57.17%	59.00%	71.67%	83.67%
Geo-TinyLLaVA (RS)	57.07%	60.73%	75.04%	88.02%
Geo-TinyLLaVA (PCT)	59.07%	62.06%	74.54%	86.02%
Geo-TinyLLaVA (RS+PCT)	**59.23%**	**63.06%**	**79.87%**	**89.02%**

Table 4. Problem-solving accuracy with Inter-GPS diagram parser in diagram detection across different diagram autoformalization models.

Autoformalization model	Inter-GPS text parser	Text Ground Truth
Inter-GPS diagram parser	61.29%	61.87%
Geo-TinyLLaVA (w/o)	64.58%	65.64%
Geo-TinyLLaVA (RS)	64.77%	65.93%
Geo-TinyLLaVA (PCT)	**66.22%**	**66.89%**
Geo-TinyLLaVA (RS+PCT)	64.58%	65.54%
Diagram logic form Ground Truth	69.59%	71.72%

Table 5. Problem-solving accuracy with ground truth in diagram detection across different diagram autoformalization models.

Autoformalization model	Inter-GPS text parser	Text Ground Truth
Inter-GPS diagram parser	56.37%	57.92%
Geo-TinyLLaVA (w/o)	70.95%	71.81%
Geo-TinyLLaVA (RS)	71.43%	71.72%
Geo-TinyLLaVA (PCT)	**72.01%**	**72.59%**
Geo-TinyLLaVA (RS+PCT)	71.14%	71.53%
Diagram logic form Ground Truth	78.76%	79.83%

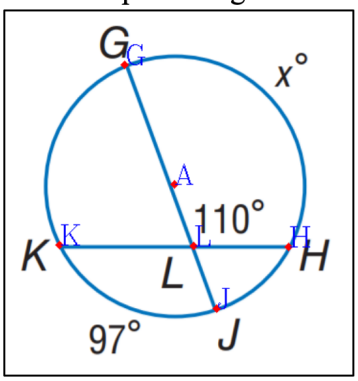
itly include the corresponding formal language statement *"PointLiesOnLine(A, Line(G, J))"*(even though solving the problem did not necessarily require this information). In contrast, Geo-TinyLLaVA generated literals that accurately represented the arc's measure, thereby conveying the geometric information more precisely.

This finding suggests that, as a diagram autoformalization model, Geo-TinyLLaVA can sometimes produce higher-quality formal language literals than the ground truth annotations in the Geometry3K dataset. Consequently, Geo-TinyLLaVA may contribute to the development of higher-quality datasets in this domain.

6. Conclusion

In this work, we focus on achieving more accurate geometry diagram autoformalization by leveraging vision-language models. We construct the instruction tuning dataset GDF86K by applying algebraic commutativity-based data augmentation techniques to the Geometry3K dataset and develop our vision-language model, Geo-TinyLLaVA. Experimental results demonstrate that, when diagrams include complete point annotations, Geo-TinyLLaVA outperforms

Complete Diagram



Geo-TinyLLaVA generated logic forms

```
Equals(MeasureOf(Angle(H, L, G)), 110);
Equals(MeasureOf(Arc(G, H)), x);
Equals(MeasureOf(Arc(K, J)), 97);
PointLiesOnLine(L, Line(K, H));
PointLiesOnLine(L, Line(G, J));
PointLiesOnCircle(H, Circle(A, radius_));
PointLiesOnCircle(G, Circle(A, radius_));
PointLiesOnCircle(J, Circle(A, radius_));
PointLiesOnCircle(K, Circle(A, radius_)).
```

Diagram logic forms of Ground Truth

```
Equals(MeasureOf(Angle(A, L, H)), 110);
Equals(Measureof(Angle(J, A, K)), 97);
Equals(MeasureOf(Angle(G, A, H)), x);
PointLiesOnLine(L, Line(H, K));
PointLiesOnLine(L, Line(G, J));
PointLiesOnCircle(H, Circle(A, radius_5_0));
PointLiesOnCircle(G, Circle(A, radius_5_0));
PointLiesOnCircle(K, Circle(A, radius_5_0));
PointLiesOnCircle(J, Circle(A, radius_5_0)).
```

Inter-GPS diagram parser generated logic forms

```
Equals(MeasureOf(Angle(J, L, K)), 97);
Equals(MeasureOf(Angle(A, L, H)), 110);
Equals(MeasureOf(Angle(A, L, H)), x);
PointLiesOnLine(L, Line(G, J));
PointLiesOnLine(L, Line(K, H));
PointLiesOnLine(L, Line(J, A));
PointLiesOnCircle(G, Circle(A, radius_5_0));
PointLiesOnCircle(K, Circle(A, radius_5_0));
PointLiesOnCircle(H, Circle(A, radius_5_0));
PointLiesOncircle(J, Circle(A, radius_5_0)).
```

Figure 5. Comparison of the formal language literals generated during evaluation for the diagram of Problem 2785 in Geometry3K(Lu et al., 2021). The literals highlighted in red indicate incorrect information.

the conventional Inter-GPS diagram parser in autoformalization tasks. Furthermore, integrating Geo-TinyLLaVA as a module within the geometry problem-solving system Inter-GPS enhances overall problem-solving accuracy. We hope that our commutativity-based data augmentation techniques and the Geo-TinyLLaVA model provide valuable insights to the mathematical formalization community.

7. Limitations

When employing Geo-TinyLLaVA as a plug-in module for Inter-GPS, the construction of diagram parsing (refer to Figure 1) results necessitates a collaborative workflow: the Inter-GPS diagram parser handles the detection part, while Geo-TinyLLaVA completes the formalization part. This integration requires both models to have a consistent naming convention for all geometric points within the diagram, necessitating the input of diagrams with complete point annotations. In future work, we plan to explore the development of a unified parser for geometry problems to eliminate the collaborative requirements between modules and enhance system flexibility.

Acknowledgements

We thank Dr. Mingming Sun and Dr. Yunfeng Cai of the Beijing Institute of Mathematical Sciences and Applications (BIMSA) for their invaluable assistance in selecting the research topic and discussing the main ideas. We also express our gratitude to Mingyang Jiao of Beijing Jiaotong University, for conducting the literature survey and technical assistance on plane geometry solvers.

Impact Statement

This paper presents work whose goal is to advance the field of Autoformalization. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33: 1877–1901, 2020.

Buchner, J. Imagehash: An image hashing library written in python. https://pypi.org/project/ImageHash/, 2025. Version 4.3.2.

Gao, J., Pi, R., Zhang, J., Ye, J., Zhong, W., Wang, Y., Hong, L., Han, J., Xu, H., Li, Z., and Kong, L. G-llava: Solving geometric problem with multi-modal large language model, 2023.

Javaheripi, M., Bubeck, S., Abdin, M., Aneja, J., Bubeck, S., Mendes, C. C. T., Chen, W., Del Giorno, A., Eldan, R., Gopi, S., et al. Phi-2: The surprising power of small language models. *Microsoft Research Blog*, 1(3):3, 2023.

Jia, J., Hu, Y., Weng, X., Shi, Y., Li, M., Zhang, X., Zhou, B., Liu, Z., Luo, J., Huang, L., and Wu, J. Tinyllava factory: A modularized codebase for small-scale large multimodal models. *arXiv preprint arXiv:2405.11788*, 2024.

Li, J., Li, D., Xiong, C., and Hoi, S. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International conference on machine learning*, pp. 12888–12900. PMLR, 2022.

Li, J., Li, D., Savarese, S., and Hoi, S. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PMLR, 2023.

Liu, H., Li, C., Li, Y., and Lee, Y. J. Improved baselines with visual instruction tuning, 2023a.

Liu, H., Li, C., Wu, Q., and Lee, Y. J. Visual instruction tuning, 2023b.

Liu, H., Li, C., Li, Y., Li, B., Zhang, Y., Shen, S., and Lee, Y. J. Llava-next: Improved reasoning, ocr, and world knowledge, January 2024. URL https://llava-vl.github.io/blog/2024-01-30-llava-next/.

Lu, P., Gong, R., Jiang, S., Qiu, L., Huang, S., Liang, X., and Zhu, S.-C. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*, 2021.

Lu, P., Bansal, H., Xia, T., Liu, J., Li, C., Hajishirzi, H., Cheng, H., Chang, K.-W., Galley, M., and Gao, J. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*, 2024.

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

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021.

Stockman, G. and Shapiro, L. G. *Computer vision*. Prentice Hall PTR, 2001.

Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023.

Trinh, T. H., Wu, Y., Le, Q. V., He, H., and Luong, T. Solving olympiad geometry without human demonstrations. *Nature*, 625(7995):476–482, 2024.

Vaswani, A. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

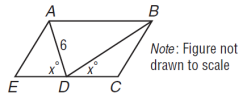
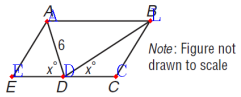
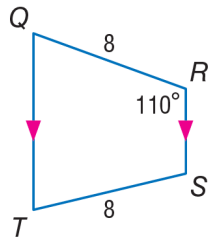
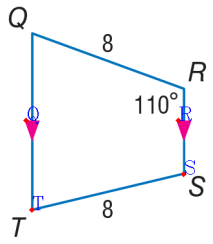
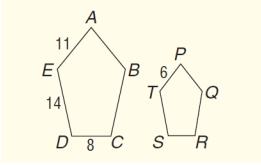
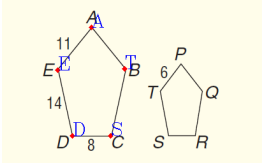
Zhai, X., Mustafa, B., Kolesnikov, A., and Beyer, L. Sigmoid loss for language image pre-training. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 11975–11986, 2023.

Zhang, R., Jiang, D., Zhang, Y., Lin, H., Guo, Z., Qiu, P., Zhou, A., Lu, P., Chang, K.-W., Qiao, Y., et al. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *European Conference on Computer Vision*, pp. 169–186. Springer, 2025.

Zheng, Y., Zhang, R., Zhang, J., Ye, Y., Luo, Z., Feng, Z., and Ma, Y. Llamafactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand, 2024. Association for Computational Linguistics. URL http://arxiv.org/abs/2403.13372.

A. Correctable Expression Errors in Diagram Logic Forms in Geometry3K

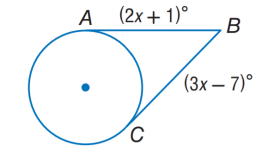
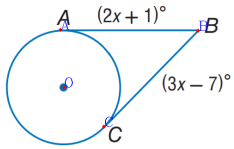
Figure 6. Examples of correctable expression errors in the diagram logic forms from the Geometry3K(Lu et al., 2021). **Uncorrected logic forms** denote the formal language expressions originally provided by Geometry3K for these diagrams, while **Corrected logic forms** represent the revised formal language expressions that we corrected based on the accurate information contained in the original diagram. The logic forms highlighted in red are incorrect and have been removed during correction, while the logic forms highlighted in green are correct information added during correction.

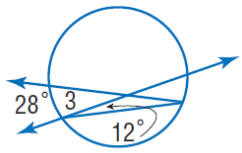
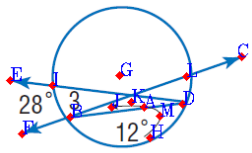
Original diagram	Complete diagram	Uncorrected logic forms	Corrected logic forms
		Equals(LengthOf(Line(D, A)), 6); Equals(LengthOf(Line(L, C)), 5); Equals(MeasureOf(Angle(E, D, A)), x); Equals(MeasureOf(Angle(L, D, C)), x); PointLiesOnLine(D, Line(C, E)).	Equals(LengthOf(Line(D, A)), 6); Equals(MeasureOf(Angle(E, D, A)), x); Equals(MeasureOf(Angle(B, D, C)), x); PointLiesOnLine(D, Line(C, E)).
		Equals(LengthOf(Line(R, Q)), 8); Equals(MeasureOf(Angle(S, R, Q)), 110); Equals(LengthOf(Line(S, T)), 8).	Equals(LengthOf(Line(R, Q)), 8); Equals(MeasureOf(Angle(S, R, Q)), 110); Equals(LengthOf(Line(S, T)), 8); Parallel(Line(Q, T), Line(R, T)).
		Equals(LengthOf(Line(A, E)), 11); Equals(LengthOf(Line(S, T)), 6); Equals(LengthOf(Line(S, D)), 8); Equals(LengthOf(Line(D, E)), 14).	Equals(LengthOf(Line(A, E)), 11); Equals(LengthOf(Line(T, P)), 6); Equals(LengthOf(Line(D, C)), 8); Equals(LengthOf(Line(D, E)), 14).

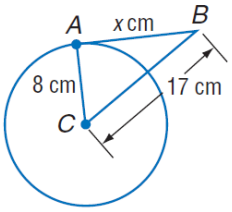
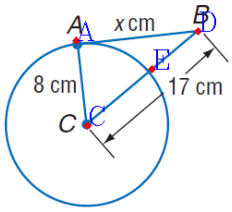
B. Unusable Diagrams with Uncorrectable Logic Form Errors in Geometry3K

Figure 7. Examples of unusable diagrams with uncorrectable logic form errors in Geometry3K(Lu et al., 2021).

Original diagram	Complete diagram	Logic forms	Notes

		`Equals(MeasureOf(Angle(A, B, C)), 2x+1);`
		`Equals(MeasureOf(Angle(A, B, D)), 3x-7);`
		`PointLiesOnCircle(A, Circle(O, radius_1_0));`
		`PointLiesOnCircle(C, Circle(O, radius_1_0));`
		`PointLiesOnLine(E, Line(B, A)).`
			The angular degree values adjacent to the straight lines were incorrectly labeled in both diagrams, leading to corresponding errors in the logic forms.

		`Equals(LengthOf(Line(A, E)), 3);`
		`Equals(MeasureOf(Angle(F, B, E)), 28);`
		`Equals(MeasureOf(Angle(H, M, B)), 12);`
		`PointLiesOnCircle(B, Circle(G, radius_6_0));`
		`PointLiesOnCircle(D, Circle(G, radius_6_0));`
		`PointLiesOnCircle(H, Circle(G, radius_6_0));`
		`...` **`(dozens of literals)`**
		`PointLiesOnLine(A, Line(K, M)).`
			The original diagram lacks labels for all geometric points, while the complete diagram excessively labels these points, resulting in disordered logic forms.

		`Equals(LengthOf(Line(C, A)), 8);`
		`Equals(LengthOf(Line(C, D)), 17);`
		`PointLiesOnCircle(E, Circle(C, radius_0_0));`
		`PointLiesOnCircle(A, Circle(C, radius_0_0));`
		`PointLiesOnLine(E, Line(C, D)).`
			The original diagram is missing the label for point E, the intersection of a line and a circle. The complete diagram erroneously labels point D adjacent to the already labeled point B in the original diagram. Both diagrams contain errors in point labeling.