GeoFM: Enhancing Geometric Reasoning of MLLMs via Synthetic Data Generation through Formal Language

Anonymous ACL submission

Abstract

Multi-modal Large Language Models (MLLMs) have gained significant attention in both academia and industry for their capabilities in handling multi-modal tasks. However, these models face challenges in mathematical geometric reasoning due to the scarcity of high-quality geometric data. To address this issue, synthetic geometric data has become an essential strategy. Current methods for generating synthetic geometric data involve rephrasing or expanding existing problems and utilizing predefined rules and templates to create geometric images and problems. However, these approaches often produce data that lacks diversity or is prone to noise. Additionally, the geometric images synthesized by existing methods tend to exhibit limited variation and deviate significantly from authentic geometric diagrams. To overcome these limitations, we propose GeoFM, a novel method for synthesizing geometric data. GeoFM uses formal languages to explore combinations of conditions within metric space, generating high-fidelity geometric problems that differ from the originals while ensuring correctness through a symbolic engine. Experimental results show that our synthetic data significantly outperforms existing methods. Models trained with our data surpass the proprietary GPT-4o model by 18.7% on geometry problem-solving tasks in MathVista and by 16.5% on GeoQA. Additionally, our approach exceeds the performance of the state-of-the-art open-source model by 5.7% on MathVista and by 2.7% on GeoQA.

1 Introduction

Large language models (LLMs) exhibit excellent reasoning capabilities. There has been a significant amount of research dedicated to applying large language models to solve text-based mathematical problems, resulting in substantial progress (Aaron Hurst, 2024; Luo et al., 2023; Shao et al., 2024; Yang et al., 2024). Recently, there has also been a growing focus on using Multi-modal Large Language Models (MLLMs) to address multi-modal mathematical problems that include images (Gao et al., 2023; Shi et al., 2024; Zhang et al., 2024a; Li et al., 2024a). Although MLLMs perform well in general tasks such as Visual Question Answering (VQA), their performance often falls short when tackling multi-modal mathematical problems (Lu et al., 2024; Wang et al., 2024a). In particular, geometry problems, which are a typical example of multi-modal mathematical problems with wide-ranging applications, require the integration of both visual and textual information for reasoning and solution. However, MLLMs struggle with these problems. One of the primary reasons

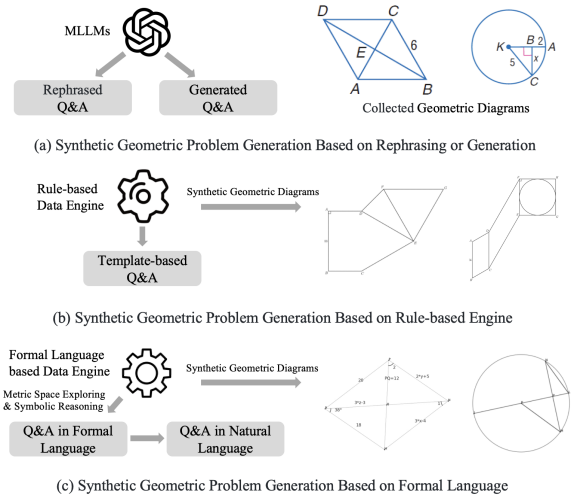


Figure 1: Comparison of different methods for synthesizing geometric data. (a) Generate geometric Q&A data by using MLLMs to rephrase existing problems or create new Q&A from collected geometric images. (b) Utilize a rule-based data engine to generate template-based Q&A and low-fidelity images. (c) Employ formal language to explore the combinations of geometric metric conditions and synthesize new problems, ensuring solution accuracy through symbolic reasoning, and generate high-fidelity geometric images.

for this difficulty is the lack of high-quality geometric data for training MLLMs. Compared to natural scene tasks like VQA, the sources and quantity of geometric data are relatively limited, which hinders the advancement of MLLMs' abilities in geometry.

To address the shortage of geometric data, some approaches have employed synthetic data generation. A straightforward method involves rewriting the problem statements and answers (Gao et al., 2023). However, simple rewrites do not alter the underlying meaning of the problems. Although this increases the quantity of problems, it does not enhance the diversity. Other approaches have attempted to use MLLMs to modify original geometric problems and generate answers (Gao et al., 2023), or to directly create new problems and corresponding responses based on collected geometric images (Shi et al., 2024), as shown in Figure 1(a). Nevertheless, these methods rely on the geometric reasoning capabilities of MLLMs. Given the current limitations of MLLMs in solving geometric problems, these approaches are prone to introducing noise into the synthetic data. Recently, there have been attempts to synthesize geometric problems using predefined rules and templates (Kazemi et al., 2023; Zhang et al., 2024a). For example, new shapes are generated by continuously extending basic geometric figures such as triangles and quadrilaterals outward along their edges. The reasoning paths and final answers are obtained through programming, as illustrated in Figure 1(b). While this method ensures the correctness of the reasoning and answers, the low fidelity of the synthesized images and the restricted variety of problems resulting in a significant disparity from real geometric problems. This discrepancy limits the progress of MLLMs in developing geometric capabilities.

To address the challenges present in current approaches, we propose a novel method for synthesizing geometric data. We have observed that existing geometric datasets often associate a single geometric diagram with only one or two problems, despite the fact that geometric diagrams often contain rich metric information that are not fully covered by the existing problems. Therefore, we propose GeoFM, a method that employs formal languages to explore the combinations of conditions within metric spaces of geometric diagrams, thereby generating high-fidelity geometric problems differ from the original ones but whose correctness is guaranteed using a symbolic engine. Existing work on geometric formal languages is scattered across different fields, such as geometric problem solving (Lu et al., 2021; Peng et al., 2023; Zhang et al., 2024b), theorems proving (Trinh et al., 2024) and geometric drawing (Krueger et al., 2021). Furthermore, these studies frequently necessitate human intervention, such as manual formalization, to accomplish the associated tasks (Zhang et al., 2024b; Krueger et al., 2021), which prevents their application for large-scale automatic synthesis of geometric data. To address this issue, we propose a comprehensive framework for geometric data synthesis that automates the formalization of seed problems, the synthesis of new problems, and the generation of images. Utilizing this approach, we have developed a highly accurate and realistic geometric synthetic dataset GeoFM80K. Experimental results demonstrate that our synthetic data can effectively enhance the geometric capabilities of MLLMs. The dataset will be released soon.

Our contributions are summarized as follows:

1. We propose GeoFM, a geometric data synthesis method using formal languages and symbolic reasoning to generate accurate solutions and geometric diagrams, addressing data noise and discrepancies in existing data synthesis methods.

2. We introduce a strategy for synthesizing new geometric problems through the combination of geometric metric conditions, resulting in the GeoFM80K dataset. Models trained on GeoFM80K outperform those trained on representative synthetic data by 8.2% on MathVista-GPS (Lu et al., 2024) and 11.1% on GeoQA (Chen et al., 2021).

3. Experimental results show our method enhances the geometric reasoning of MLLMs. The GeoFM-8B model surpasses GPT-4o by 18.7% on MathVista-GPS and 16.5% on GeoQA, and exceeds the best open-source model by 5.7% on MathVista-GPS and 2.7% on GeoQA.

2 Method

2.1 Overview

In this section, we introduce our method for generating synthetic geometric problems. We first collect a set of seed problems and then automatically convert them into a formal language used for geometric problem solving. Next, within the formal language space, we generate new problems by arbitrarily combining the metric geometric conditions of the seed problems. These new problems can be solved through symbolic reasoning, which aids in synthesizing natural language solutions and verify-

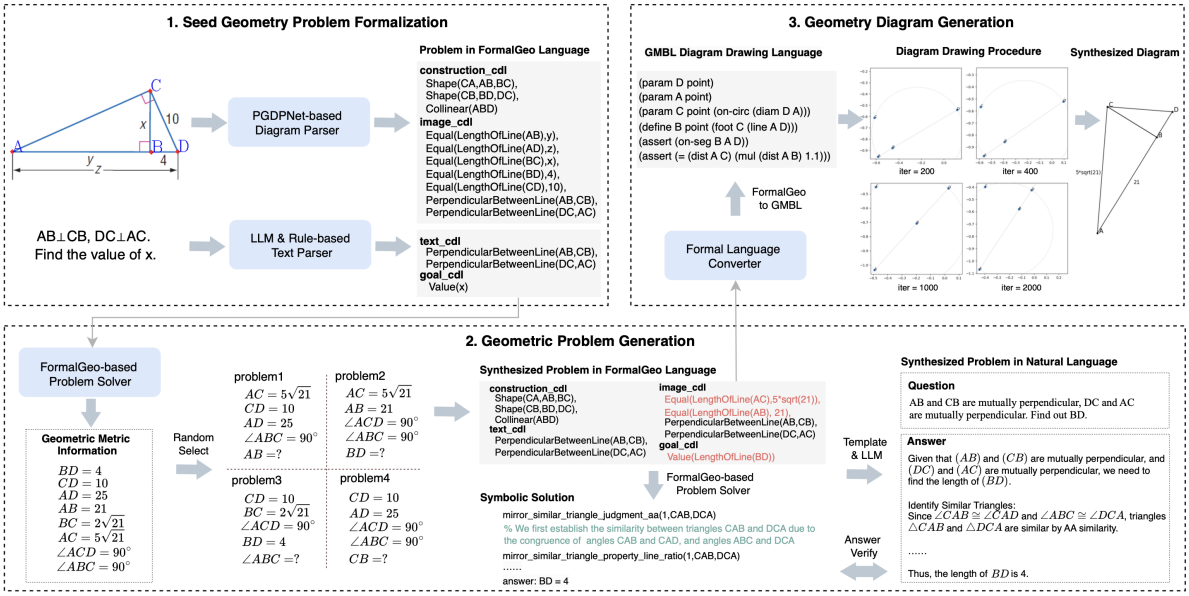


Figure 2: The Framework of Geometric Data Synthesis GeoFM

ing results. Finally, we convert the formal language representations of the new problems into a drawing language to produce corresponding geometric diagrams. The framework of the data generation process is illustrated in Figure 2.

2.2 Seed Geometry Problem Formalization

Formalizing geometric problems is a significant research area in geometry. Various formalization schemes have been proposed, including InterGPS (Lu et al., 2021), AlphaGeometry (Trinh et al., 2024), and FormalGeo (Zhang et al., 2024b), each employing different approaches. In this study, we utilize FormalGeo as it more effectively represents metric geometry than AlphaGeometry and offers a broader range of geometric theorems than InterGPS. FormalGeo employs the Conditional Declaration Language (CDL) to represent geometric problems, which includes construction CDL, text CDL, image CDL, and goal CDL. Construction CDL conveys geometric structure information, such as basic shapes, collinearity, and cocircularity. Text CDL and image CDL capture geometric and algebraic relations from the problem statement and diagram, respectively, while goal CDL defines the problem-solving objective. An illustrative example is shown in Figure 2.

For the text parser, we propose a new construction method based on training a large language model with synthetic data. Since the text parser focuses on mapping natural language to formal language without considering the validity or solvability of the problem, we propose a method for generating synthetic training data based on formal language back-translation. Initially, for each formal language expression in FormalGeo, we use GPT-4o to generate 20 corresponding natural language templates, which are then manually reviewed and corrected. During data synthesis, we randomly select formal language conditions and goals to be solved, insert randomly generated geometric points to create a formal language problem, and then convert it into a natural language problem description using the natural language templates. This description is rewritten using the large language model Qwen2.5-72B-Instruct (An Yang, 2025) to increase the diversity of expressions. In this way, we construct synthetic training data for the text parser that maps natural language problems to formal language problems. Using this method, we synthesized 30k training data samples and trained Llama-3-8B-Instruct (Aaron Grattafiori, 2024), resulting in the development of a text parser.

For the diagram parser, we constructed it by integrating the geometric shape parsing method PGDPNet (Zhang et al., 2022), OCR tool (Du et al., 2021), and rule-based processing. PGDPNet can identify various geometric elements, including points and lines, their coordinates, and geometric relationships like parallelism and perpendicularity. To enhance the accuracy of text and symbol recognition, we employ OCR to re-recognize the information within the detection boxes extracted by PGDPNet. Based on all the parsed information, we convert it into

construction CDL and image CDL through rule-based processing.

The seed problems are processed using the text parser and the diagram parser to derive their formal representations. After filtering out invalid conditions using formal language grammar validation, seed problems represented in formal language are generated. These seed problems are then used for subsequent geometric problem synthesis. It is important to note that while parsing errors by the text parser and diagram parser may cause discrepancies between the formalized problems and the original ones, the final synthesized data remains consistent and error-free. This is because both the new problems and the corresponding images are generated solely based on the formalized seed problems, rather than the original ones.

Algorithm 1 Geometric Problem Generation

Input formalized seed problem set $\mathcal{FS}$, number of synthetic problems m
Output synthetic problem set $\mathcal{S}$

for $\mathcal{P} \in \mathcal{FS}$ **do**
  $\mathcal{M}_p \leftarrow$ MetricInfoOfProblemStatement($\mathcal{P}$)
  $\mathcal{M}_{all} \leftarrow$ GatheringMetricInfo($\mathcal{P}$)
  $m_p = m$
  while $m_p > 1$ **do**
    $n \leftarrow \text{Random}(1, \min(|\mathcal{M}_p|, |\mathcal{M}_{all}| - |\mathcal{M}_p|))$
    $\mathcal{M}_{del} \leftarrow$ RandomSelect($\mathcal{M}_p, n$)
    $\mathcal{M}_{add} \leftarrow$ RandomSelect($\mathcal{M}_{all} - \mathcal{M}_p, n$)
    $\mathcal{P}_{new} \leftarrow \mathcal{P} - \mathcal{M}_{del} + \mathcal{M}_{add}$
    $\mathcal{A}_{new} \leftarrow$ FormalGeoSolver($\mathcal{P}_{new}$)
    $\mathcal{P}_{syn}, \mathcal{A}_{syn} \leftarrow$ Template&LLM($\mathcal{P}_{new}, \mathcal{A}_{new}$)
    if AnswerVerify ($\mathcal{A}_{syn}, \mathcal{A}_{new}$) **then**
      $\mathcal{S}$.add([$\mathcal{P}_{syn}, \mathcal{A}_{syn}$])
      $m_p \leftarrow m_p - 1$
    end if
  end while
end for
Return $\mathcal{S}$

2.3 Geometric Problem Generation

In this section, we will introduce the process of generating new geometry problems based on formalized seed problems. Since each geometric diagram contains rich metric information such as lengths, angles, and areas, we can utilize the formal language representation to combine the metric information in various ways, thereby generating new problems with different conditions and goals. Specifically, the synthesis process primarily consists of three components: calculating the geometric metric information of the seed problems, synthesizing data in formal language, and converting this data into natural language geometric instruction data. The process is detailed in Algorithm 1.

2.3.1 Gathering Geometric Metrics

To extract as much metric information as possible from the seed problems, we utilize the FormalGeo problem solving engine. During the solving process, we employ a breadth-first search approach to determine the applicability of predefined geometric theorems to the problems, continuing until a solution is found or a timeout occurs. Regardless of whether the solution is ultimately successful, the reasoning process yields substantial metric information about various geometric elements in the problem. We extract this metric information $\mathcal{M}_{all}$ for the subsequent synthesis of new problems.

2.3.2 Synthesizing Data in Formal Language

After obtaining geometric metric conditions $\mathcal{M}_{all}$ for a seed problem $\mathcal{P}$, we can combine these conditions to generate new geometric problems. Let $\mathcal{M}_p$ be the set of metric conditions of the original problem statement. We first sample a random number n (where $n \leq \min(|\mathcal{M}_p|, |\mathcal{M}_{all}| - |\mathcal{M}_p|)$). Next, we replace n metric conditions from $\mathcal{M}_p$ with n new conditions sampled from the remaining metric set $\mathcal{M}_{all} - \mathcal{M}_p$ and randomly choose one metric condition different from the new problem statement as the goal, thereby creating a new problem. This ensures that the new problem has the same number of metric conditions as the seed problem, minimizing issues related to insufficient metric conditions for deriving valid conclusions and avoiding redundancy from having too many conditions. Furthermore, we randomly allocate the metric conditions to text CDL and image CDL. The metric conditions in image CDL will only appear in the synthesized images and not in the problem statements, thereby forcing the model to interpret the problem by reading the images rather than relying solely on textual information.

Once the formal language problem is obtained, we solve the synthesized problem using the FormalGeo symbolic engine to derive the corresponding symbolic solutions. The symbolic solution includes the geometric theorems applied and the derivation process. Since the goal of the synthesized problem is randomly selected and may not always be solvable, if the goal is not achieved, we select the last valid inference from the symbolic engine's reasoning path as the new goal. This ensures the validity of the problem. Through this process, we can synthesize multiple formal language problems with symbolic solutions from each seed problem.

2.3.3 Geometric Instruction Data Synthesis

After obtaining the formalized problems and their symbolic solutions, it is necessary to convert them into natural language instruction data to facilitate subsequent training of the MLLMs. This conversion process begins by transforming all FormalGeo formalized language and the geometric theorems used in problem-solving into natural language templates. These templates are manually verified to ensure their accuracy. Subsequently, we use these templates to convert the formalized problems and their symbolic solutions into natural language.

The lack of diversity in template-based solutions can lead to mode collapse when used directly for model training. To address this issue, we employ the large language model Qwen2.5-72B-Instruct to rewrite the template-generated solutions, producing more fluent and varied problem-solving solutions. The prompt for rewriting is provided in Appendix A. To minimize rewriting errors, we also use the LLM to compare the final answers of the rewritten problems with the results derived from FormalGeo through answer extraction and verification following the MathVista (Lu et al., 2024) evaluation methodology, retaining only those problems where the answers are consistent. Compared to directly generating problem solutions using a strong MLLM, our method references the reasoning process of a symbolic engine during solution generation and the final answers are cross-verified for consistency with the results from the symbolic engine, thereby significantly reducing the probability of errors in the synthesized problem solutions.

2.4 Geometry Diagram Generation

Synthesizing geometric images for each generated problem is challenging due to the need to meet geometric constraints. Some methods use specialized drawing programs, but these often produce a limited variety of images that conform to predefined patterns (Kazemi et al., 2023; Zhang et al., 2024a). Tools like GeoGebra (Hohenwarter and Preiner, 2007) require manual manipulation for drawing. The Geometry Model Building Language (GMBL) uses a formal language and computational geometry to approximate target images through numerical optimization. However, it requires manually creating the formal language for the target image and evaluating if the synthesized image meets expectations, making it impractical for large-scale automated synthesis.

To address the limitations of existing methods, we developed a new engine capable of automatically synthesizing large-scale geometric images based on GMBL. This engine contains a formal language converter that automatically transforms construction CDL and image CDL statements, which illustrate geometric diagrams, into GMBL formal language. This conversion requires the prior construction of a mapping table from the FormalGeo language to the GMBL language. When generating the GMBL description of a problem, a heuristic rule-based method is first employed to determine the definition order of geometric points. Subsequently, the relevant geometric constraints represented in the FormalGeo language for each geometric point are translated into the GMBL language based on predefined rules and the mapping table.

We categorize the computational geometry objects in GMBL used to assess whether geometric constraints are met based on the strictness of these constraints. For example, the requirement for a point to lie on a line is stricter than that for two line segments to be of equal length, as deviations from the former are more apparent. We then establish different loss thresholds for each group, filtering out images that do not meet these thresholds after numerical optimization to maintain the quality of synthetic images. For geometric images that satisfy the constraints, we incorporate image CDL information, such as segment lengths and angles, into the diagram. This inclusion ensures that MLLMs must interpret the image to extract necessary information for problem-solving, thereby enhancing the model's image perception capabilities. This approach allows us to automatically generate images corresponding to synthesized geometric problems represented by the FormalGeo formal language.

3 Experiments

3.1 Experimental Setup

We synthesized 80k data points for our experiments based on the training sets of the formalgeo7k (Zhang et al., 2024b) and PGPS9K (Zhang et al., 2023) geometric datasets. The effectiveness of our synthesized data was validated using the LLaVA-NeXT-8B (Liu et al., 2024), a model trained with limited geometric data, which facilitates the assessment of how the addition of various geometric data affects the model's geometric capabilities. Additionally, we employed InternVL2-8B-MPO (Wang et al., 2024c), a model trained with a larger amount

Model	D_{origin}	$D_{synthetic}$
LLaVA-NeXT-8B	11.2	9.5
Qwen2-VL-7B	28.2	15.8
InternVL2-8B-MPO	40.7	27.7

Table 1: Comparison of MLLM performance on open source geometric data D_{origin} and synthetic geometric data $D_{synthetic}$.

Training Data	Vol.	MathVista	GeoQA
Seed Data	5k	17.8	22.7
w/ GPT-4o CoT	5k	25.9	22.9
w/ CoT + Rephrase	25k	20.7	23.5
w/ CoT + MLLM Aug	25k	26.3	25.8
w/ GeoFM Data	25k	**27.9**	**32.0**

Table 2: Results of different geometric seed data utilization methods on MathVista for geometry problem solving (GPS) and GeoQA.

of geometric data, to determine whether synthesized data can further enhance the performance of models with higher geometric capabilities. Both models are trained for two epochs. The LLaVA-NeXT-8B model utilizes a batch size of 64 and a learning rate of 3e-5, while the InternVL2-8B-MPO model employs a batch size of 128 and a learning rate of 1e-5. We utilized the test mini set of MathVista for geometry problem-solving (GPS) (Lu et al., 2024) and the test set of GeoQA (Chen et al., 2021) for evaluation. Model performance was assessed through response generation, answer extraction, and score calculation, following the MathVista methodology. Top-1 accuracy was used as the evaluation metric.

3.2 Necessity of Metric Space Exploration

Some MLLMs are trained using open-source geometric datasets, where each image is associated with only a few questions. This raises the question of whether MLLMs can generalize to other variations of questions related to the same geometric diagram. To investigate this, we conducted an experiment using synthetic data. We sampled 500 questions each from two commonly used open-source geometric datasets, GeoQA (Chen et al., 2021) and Geometry3k (Lu et al., 2021), to create a test set D_{origin}. Correspondingly, we generated a synthetic test set $D_{synthetic}$, by creating an equal number of problems based on D_{origin} but with different conditions or problem-solving objectives.

As shown in Table 1, models with limited geometric capabilities, such as LLaVA-NeXT-8B (Liu et al., 2024), performed similarly on both test sets. In contrast, models trained on open-source geometric data such as Qwen2-VL-7B (Wang et al., 2024b) and InternVL2-8B-MPO (Wang et al., 2024c) showed overall performance improvement but exhibited significantly lower performance on $D_{synthetic}$ compared to D_{origin}. This indicates that these models have difficulty generalizing from previously encountered problems to related problem-solving scenarios. Since $D_{synthetic}$ is generated through metric space exploration, the suboptimal performance of existing models on this dataset suggests that employing a similar method for large-scale data synthesis in model training could boost geometric capabilities. This hypothesis will be validated in subsequent sections.

3.3 Effectiveness of GeoFM

3.3.1 More Effective Utilization of Seed Data

Effectively utilizing a set of geometric seed data to enhance the geometric problem-solving abilities of MLLMs is a significant research question. Traditional approaches include learning the Chain of Thought (CoT) process from more advanced models, augmenting original problems through rewriting, and using MLLMs to generate new problems and solutions. In this section, we compare these methods with our data synthesis method and conduct experiments based on LLaVA-NeXT-8B. We sampled 5k geometric problems from the formalgeo7k dataset as seed data. Various data construction methods were experimentally compared, including directly using the seed data, constructing CoT solutions based on GPT-4o (Aaron Hurst, 2024), rewriting problems and CoT solutions, augmenting existing problems and solutions with MLLMs similar as (Gao et al., 2023), and employing the GeoFM data synthesis method. The results are presented in Table 2.

As demonstrated, utilizing GPT-4o's CoT data enhances model performance. Though simple rewrites show varying effectiveness across datasets, synthesizing new problems improve performance. The most significant improvement is achieved with the GeoFM data synthesis method, which increases performance by 10.1% on the MathVista-GPS and 9.3% on the GeoQA compared to the seed data. This indicates that our data synthesis method can more effectively utilize existing geometric data to help enhance model performance.

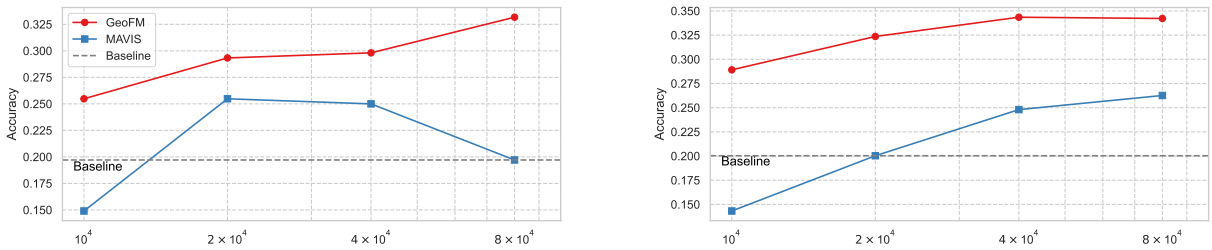


Figure 3: Comparison with existing geometric synthesis data at different data scales using LLaVA-NeXT-8B. The baseline corresponds to the performance of the original model.

3.3.2 Comparison with Existing Geometric Synthetic Datasets

To assess the impact of using solely synthesized data, we compare GeoFM with existing geometric synthetic datasets. The GeoGPT4V (Cai et al., 2024) dataset contains 4.9k synthetic data points, which is small in quantity. The GermVerse (Kazemi et al., 2023) dataset performs suboptimally on benchmarks. Therefore, our primary comparison is between GeoFM and the recently proposed MAVIS-Geometry (Zhang et al., 2024a) dataset, a representative dataset generated through rule-based data engine. To evaluate the model's performance across various data scales, we sampled 10k, 20k, 40k, and 80k data points from each dataset. The experimental results presented in Figure 3 evident that both datasets show performance improvements after training. However, GeoFM significantly outperforms MAVIS-Geometry, with an average improvement of 8.2% on MathVista-GPS and 11.1% on GeoQA. We speculate that this is primarily due to the rule-based synthetic geometric problems in MAVIS-Geometry differing substantially from real data, as illustrated in Appendix D, thereby limiting its effectiveness.

3.3.3 Performance Boost from GeoFM

To assess the benefits of adding GeoFM synthetic data to existing open-source datasets, we conducted experiments using the Geo170k-QA (Gao et al., 2023) and MathV360K-GPS (Shi et al., 2024) geometric datasets. We trained two base models, LLaVA-NeXT-8B and InternVL2-8B-MPO, using both the open-source data alone and the open-source data combined with GeoFM data. The experimental results, presented in Table 3, demonstrate that models trained with the addition of GeoFM data achieved consistent improvements on the MathVista-GPS and GeoQA benchmarks. Specifically, LLaVA-NeXT-8B showed improvements of 1.9% and 2.3%, while InternVL2-8B-MPO exhibited gains of 4.8% and 3.2%, respectively.

Model	MathVista	GeoQA
GM-LLaVA-NeXT-8B	54.8	68.3
GeoFM-LLaVA-NeXT-8B	**56.7**	**70.6**
GM-InternVL2-8B-MPO	74.5	74.7
GeoFM-InternVL2-8B-MPO	**79.3**	**77.9**

Table 3: Performance improvements from GeoFM: Models prefixed with 'GM-' are trained on the Geo170k-QA and MathV360K-GPS datasets, while 'GeoFM-' models include an additional 80k GeoFM data.

Model	**MathVista**	**GeoQA**
Closed-source MLLMs		
GPT-4o (Aaron Hurst, 2024)	60.6	61.4
GPT-4V (OpenAI, 2023)	50.5	-
Gemini 1.0 Ultra (Rohan Anil, 2024)	56.2	-
Open-source MLLMs		
LLaVA-LLaMA-2-13B (Liu et al., 2023)	29.3	20.3
Qwen-VL-Chat-7B (Bai et al., 2023)	35.6	26.1
InternVL2-Pro (InternVL, 2024)	65.4	-
InternVL2-8B-MPO (Wang et al., 2024c)	73.6	53.1
Mathematical MLLMs		
Math-LLaVA-13B (Shi et al., 2024)	57.7	47.8
G-LLaVA-7B (Gao et al., 2023)	53.4	62.8
MAVIS-7B (Zhang et al., 2024a)	-	66.7
EAGLE (Li et al., 2024a)	54.3	67.1
GeoGPT4V (Cai et al., 2024)	64.4	-
GOLD (Zhang and Moshfeghi, 2024)	-	75.2
GeoFM-8B	**79.3**	**77.9**

Table 4: Comparison of GeoFM model with existing MLLMs on MathVista-GPS and GeoQA

We compare GeoFM-8B which trained on the InternVL2-8B-MPO backbone with GeoFM data against existing MLLMs, including both proprietary and open-source representative models. The results, presented in Table 4, indicate that the GeoFM-8B model significantly outperforms existing models on the MathVista-GPS and GeoQA benchmarks. Specifically, it exceeds GPT-4o's accuracy by 18.7% on MathVista-GPS and 16.5% on

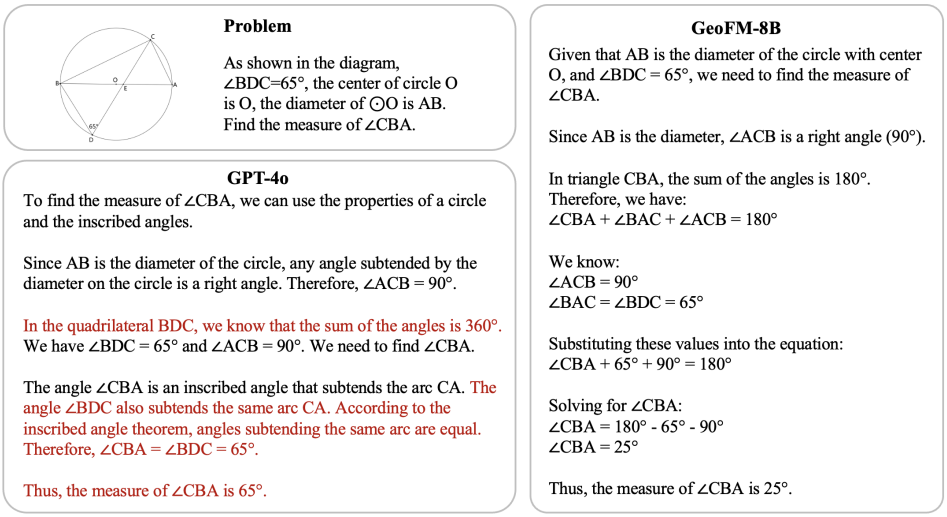


Figure 4: Demonstration of geometric problem solving using GPT-4o and GeoFM-8B

GeoQA, and surpasses the state-of-the-art model by 5.7% on MathVista-GPS and 2.7% on GeoQA.

3.4 Qualitative Analysis

We conduct a qualitative analysis by comparing our model, GeoFM, with the representative model GPT-4o, as illustrated in Figure 4. Our model effectively captures the geometric features of the problems and provides an accurate reasoning process. In contrast, GPT-4o demonstrates errors in understanding geometric figures and exhibits hallucinations that lead to incorrect answers. This comparison highlights the advantages of our synthetic data method.

4 Related Work

Geometry Problem Solving Solving geometry problems is a challenging multi-modal mathematical task. Some studies have employed symbolic solvers to address geometric problems by first formalizing them and then performing symbolic reasoning (Lu et al., 2021; Li et al., 2024b; Zhang et al., 2024b). However, these symbolic solvers are limited to solving specific geometric problems and cannot transfer geometric capabilities across different scenarios like MLLMs. Recently, research aimed at enhancing the geometric capabilities of MLLMs has emerged, primarily by improving model performance through high-quality geometric data. Early geometric datasets such as GeoQA (Chen et al., 2021), GeoQA+ (Cao and Xiao, 2022), UniGeo (Chen et al., 2022), and PGPS9K (Zhang et al., 2023) were manually collected and curated, which often limited their scale. G-LLaVA (Gao et al., 2023) expanded existing geometric datasets using a large language model for rewriting and augmentation, but this method lacked diversity and was prone to introducing noise due to the limitations of the rewriting model. GeoGPT4V (Cai et al., 2024) enhances this approach by incorporating image synthesis, generating Wolfram code via GPT-4 (Josh Achiam, 2024), and using this tool to create geometric images. However, this method's image synthesis is insufficiently stable. GeomVerse (Kazemi et al., 2023) and MAVIS (Zhang et al., 2024a) utilized rule-based data engines to generate geometric problems, but the data produced often differed significantly from real-world data, affecting their effectiveness. To address these shortcomings, we propose GeoFM, which employs formal languages to explore combinations of conditions within metric spaces, thereby generating high quality geometric data that can effectively enhance the geometric reasoning capabilities of MLLMs.

5 Conclusion

In this paper, we present GeoFM, a novel method for generating high-quality geometric problems to enhance the geometric reasoning abilities of MLLMs. GeoFM uses formal languages to systematically explore condition combinations within metric spaces. Our approach involves formalizing seed problems, generating new geometric problems through the combination of metric conditions, and creating geometric diagrams corresponding to the problems. Experimental results show that our method significantly outperforms existing approaches, achieving state-of-the-art results on the MathVista and GeoQA benchmarks.

6 Limitations

In this study, we employ formal languages to explore various condition combinations within metric spaces of seed problems and synthesize high-quality geometric data to enhance the performance of multimodal large language models. During the synthesis process, we use seed problems to generate synthetic data, which need manual collection. Additionally, certain types of geometric problems, such as word problems or those lacking geometric point identifiers, are challenging to formalize. Therefore, designing new methods for synthesizing geometric problems from scratch is a direction worth further exploration.

References

Abhinav Jauhri Abhinav Pandey Abhishek Kadian Ahmad Al-Dahle Aiesha Letman et al. Aaron Grattafiori, Abhimanyu Dubey. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Adam P. Goucher Adam Perelman Aditya Ramesh Aidan Clark AJ Ostrow et al. Aaron Hurst, Adam Lerer. 2024. Gpt-4o system card. *Preprint*, arXiv:2410.21276.

Beichen Zhang Binyuan Hui Bo Zheng Bowen Yu Chengyuan Li et al. An Yang, Baosong Yang. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. 2023. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. *Preprint*, arXiv:2308.12966.

Shihao Cai, Keqin Bao, Hangyu Guo, Jizhi Zhang, Jun Song, and Bo Zheng. 2024. GeoGPT4V: Towards geometric multi-modal large language models with geometric image generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 750–766, Miami, Florida, USA. Association for Computational Linguistics.

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.

Yuning Du, Chenxia Li, Ruoyu Guo, Cheng Cui, Weiwei Liu, Jun Zhou, Bin Lu, Yehua Yang, Qiwen Liu, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. 2021. Pp-ocrv2: Bag of tricks for ultra lightweight ocr system. *Preprint*, arXiv:2109.03144.

Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, and Lingpeng Kong. 2023. G-llava: Solving geometric problem with multi-modal large language model. *Preprint*, arXiv:2312.11370.

M. Hohenwarter and J. Preiner. 2007. Dynamic mathematics with geogebra. *JOMA*, 7:1448.

InternVL. 2024. Internvl2: Better than the best—expanding performance boundaries of open-source multimodal models with the progressive scaling strategy.

Sandhini Agarwal Lama Ahmad Ilge Akkaya Florencia Leoni Aleman et al. Josh Achiam, Steven Adler. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. 2023. Geomverse: A systematic evaluation of large models for geometric reasoning. *Preprint*, arXiv:2312.12241.

Ryan Krueger, Jesse Michael Han, and Daniel Selsam. 2021. Automatically building diagrams for olympiad geometry problems. *Preprint*, arXiv:2012.02590.

Zhihao Li, Yao Du, Yang Liu, Yan Zhang, Yufang Liu, Mengdi Zhang, and Xunliang Cai. 2024a. Eagle: Elevating geometric reasoning through llm-empowered visual instruction tuning. *Preprint*, arXiv:2408.11397.

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.

Haotian Liu, Chunyuan Li, Yuheng Li, Bo Li, Yuanhan Zhang, Sheng Shen, and Yong Jae Lee. 2024. Llava-next: Improved reasoning, ocr, and world knowledge.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pages 34892–34916. Curran Associates, Inc.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *Preprint*, arXiv:2310.02255.

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.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

OpenAI. 2023. Gpt-4v system card.

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.

Jean-Baptiste Alayrac Jiahui Yu Radu Soricut Johan Schalkwyk Andrew M. Dai et al. Rohan Anil, Sebastian Borgeaud. 2024. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

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

Trieu H. Trinh, Yuhuai Wu, Quoc V. Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*, 625:476 – 482.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Mingjie Zhan, and Hongsheng Li. 2024a. Measuring multimodal mathematical reasoning with math-vision dataset. *Preprint*, arXiv:2402.14804.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024b. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Weiyun Wang, Zhe Chen, Wenhai Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Jinguo Zhu, Xizhou Zhu, Lewei Lu, Yu Qiao, and Jifeng Dai. 2024c. Enhancing the reasoning ability of multimodal large language models via mixed preference optimization. *Preprint*, arXiv:2411.10442.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. 2024. Qwen2.5-math technical report: Toward mathematical expert model via self-improvement. *Preprint*, arXiv:2409.12122.

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, Xinyu Wei, Dongzhi Jiang, Ziyu Guo, Shicheng Li, Yichi Zhang, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, Peng Gao, Chunyuan Li, and Hongsheng Li. 2024a. Mavis: Mathematical visual instruction tuning with an automatic data engine. *Preprint*, arXiv:2407.08739.

Xiaokai Zhang, Na Zhu, Yiming He, Jia Zou, Qike Huang, Xiaoxiao Jin, Yanjun Guo, Chenyang Mao, Yang Li, Zhe Zhu, Dengfeng Yue, Fangzhen Zhu, Yifan Wang, Yiwen Huang, Runan Wang, Cheng Qin, Zhenbing Zeng, Shaorong Xie, Xiangfeng Luo, and Tuo Leng. 2024b. Formalgeo: An extensible formalized framework for olympiad geometric problem solving. *Preprint*, arXiv:2310.18021.

A Template-based Solution Rewriting Prompt

Prompt: Rewrite Template-based Solution

Given a geometry problem and its answer hint, write a answer to the problem. Ensure the answer is correct, concise, easy to understand, and written with clarity and natural flow.

Guidelines
1. Refer to the answer hint, but do not use the information in it as given conditions.
2. Only output the solution, without any additional information.

Problem
<problem>

Hint
<template-based solution>

B Illustration of Geometric Problem and Solution Synthesis

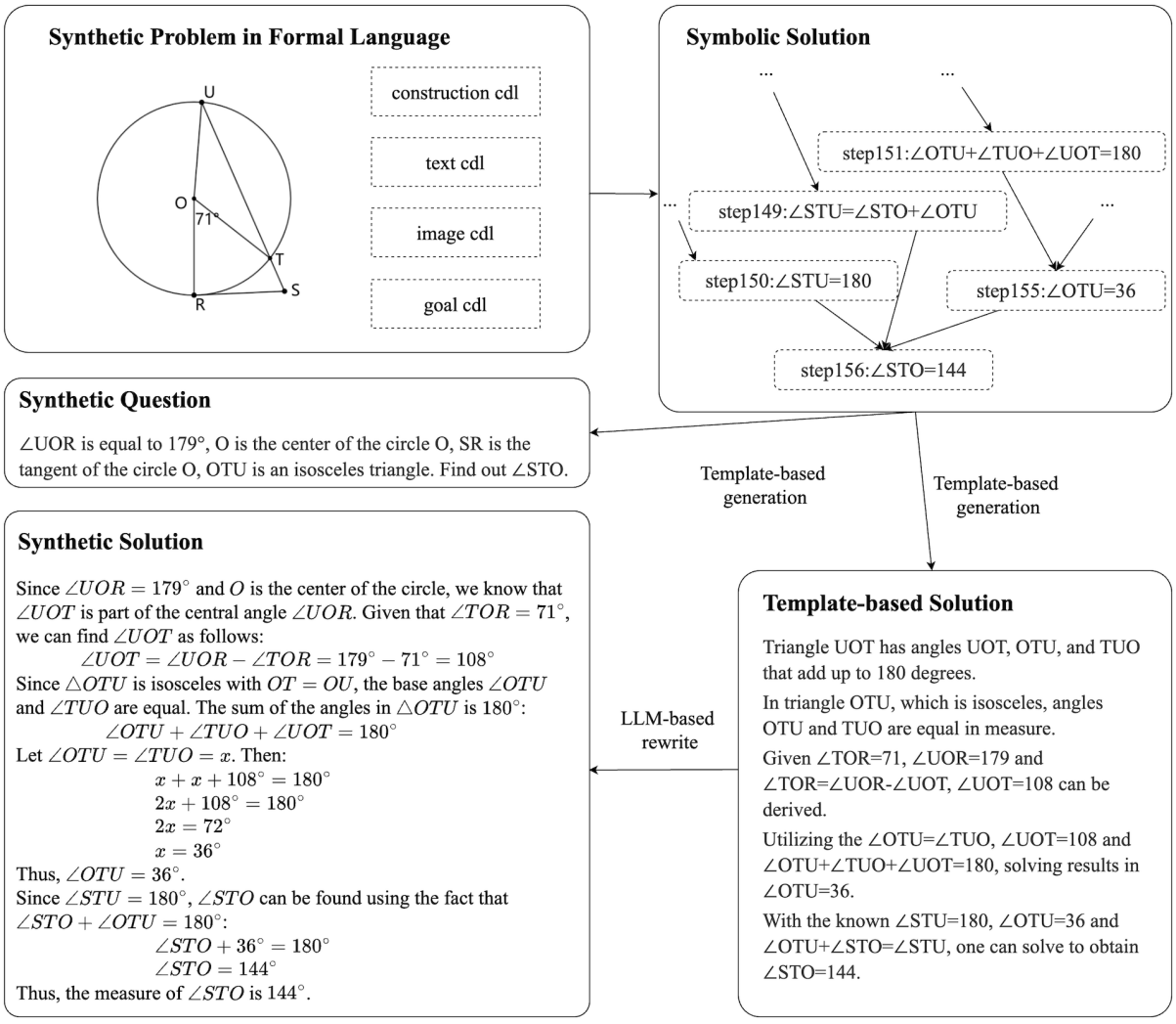


Figure 5: Convert a synthesized formal language geometric problem into natural language instruction data

C Examples of Synthetic Data

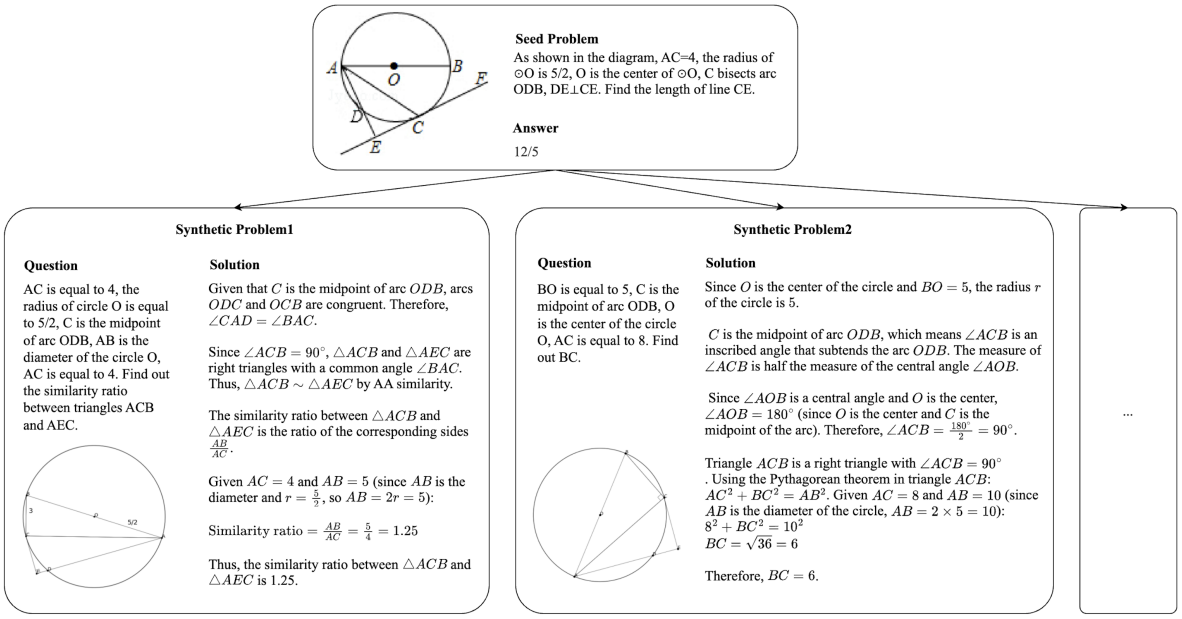
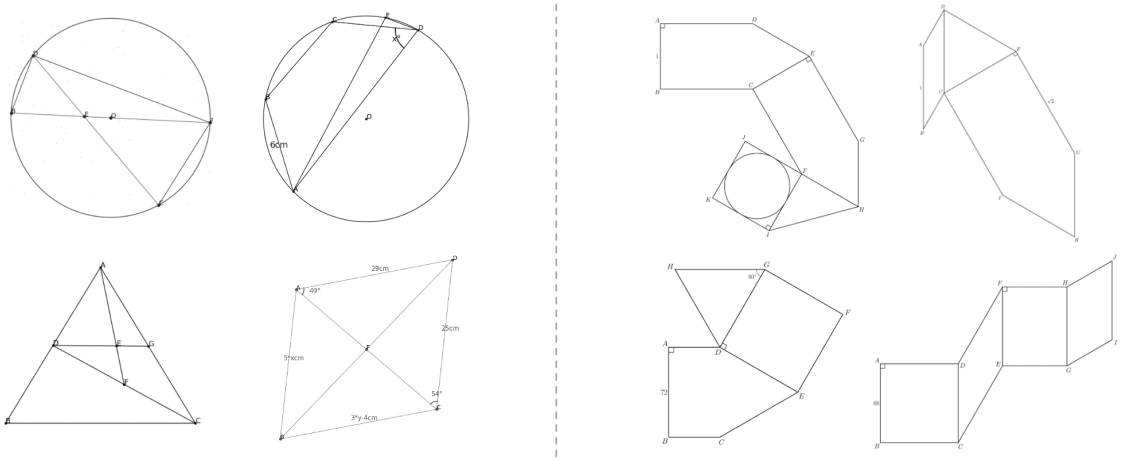


Figure 6: Examples of GeoFM Synthetic Data

D Comparison of Geometric Images in Synthetic Datasets



(a) GeoFM (b) MAVIS-Geometry

Figure 7: Comparison of Synthetic Images between GeoFM and MAVIS-Geometry