Theorem-Validated Reverse Chain-of-Thought Problem Generation for Geometric Reasoning

Anonymous ACL submission

Abstract

Large Multimodal Models (LMMs) face limitations in geometric reasoning due to insufficient Chain of Thought (CoT) image-text training data. While existing approaches leverage template-based or LLM-assisted methods for geometric CoT data creation, they often face challenges in achieving both diversity and precision. To bridge this gap, we introduce a two-stage Theorem-Validated Reverse Chain-of-Thought Reasoning Synthesis (TR-CoT) framework. The first stage, TR-Engine, synthesizes theorem-grounded geometric diagrams with structured descriptions and properties. The second stage, TR-Reasoner, employs reverse reasoning to iteratively refine question-answer pairs by cross-validating geometric properties and description fragments. Our approach expands theorem-type coverage, corrects long-standing misunderstandings, and enhances geometric reasoning. Fine-grained CoT improves theorem understanding and increases logical consistency by 24.5%. Our best models surpass the baselines in MathVista and GeoQA by 10.1% and 4.7%, outperforming advanced closed-source models like GPT-4o.

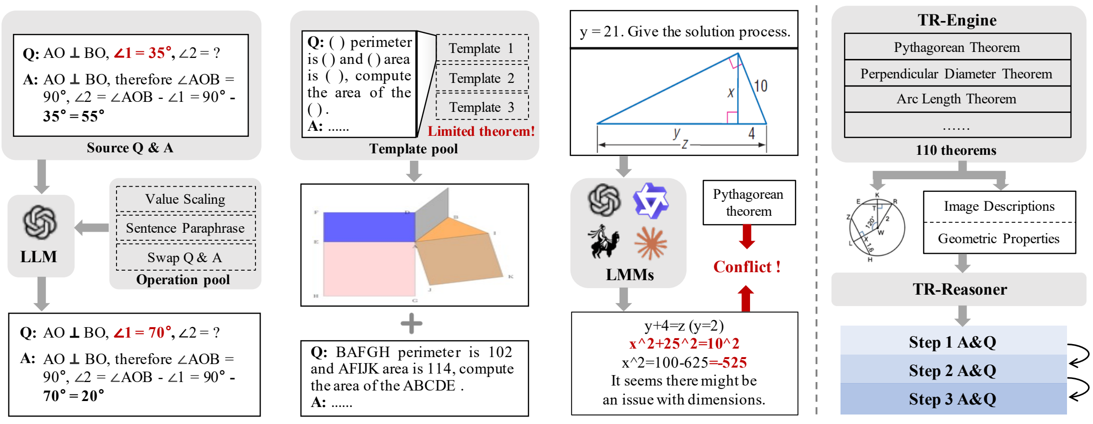
1 Introduction

Large Language Models (LLMs) (OpenAI, 2024; Guo et al., 2025) have revolutionized textual mathematical reasoning through advanced inferential mechanisms. While architectural innovations now enable these models to process multimodal inputs via parameter-efficient vision-language alignment (e.g., GPT-4o (Islam and Moushi, 2024), Gemini (Team et al., 2023)), achieving human-competitive VQA performance (Fan et al., 2024), their geometric reasoning remains constrained. This limitation stems from training data dominated by natural scenes, which lack the geometric specificity required for rigorous spatial problem-solving.

Current methods for generating geometric reasoning data through Chain-of-Thought (CoT) frameworks face three fundamental limitations. First, rephrasing approaches (Gao et al., 2023b) use LLM to transform the CoT format of existing problems, which requires scarce high-quality annotations and domain-specific expertise to ensure theorem consistency (Fig. 1 (a)). Second, template-based methods (Kazemi et al., 2023a; Zhang et al., 2024b) generate geometrically oversimplified images by combining predefined polygons in rigid configurations, lacking theorem-aware element interactions, limiting their applicability to advanced reasoning, as shown in Fig. 1 (b). Thirdly, while LMM-based resoning (Peng et al., 2024) ensures reasoning diversity, but insufficient mathematical priors often lead to icorrect reasoning, e.g., it uses the Pythagorean theorem for the three sides of different triangles, leading to logically invalid chains of reasoning(Fig. 1 (c)).

We introduce Theorem-Validated Reverse Chain-of-Thought (TR-CoT), a two-stage framework designed to generate geometric reasoning data and verify logical flows, as shown in Fig. 1 (d). In the first stage, we develop the theorem-driven image and property generation engine (TR-Engine), which creates images paired with geometric properties, ensuring dependencies among elements. In the second stage, TR-Reasoner derives questions from answers by segmenting image descriptions, generating single-step reasoning, and combining them into multi-step reasoning chains. Each step is verified against geometric properties, discarding pairs that violate mathematical rules. This ensures the logical rigor in the generated data.

With TR-CoT, we create TR-GeoMM and TR-GeoSup, comprehensive datasets of diverse geometric theorems, which fully leverage CoT information. TR-CoT can bring notable and consistent improvements across a range of LMM baselines such as LLaVA, Qwen, and InternVL. Using the recent LMM baselines, we achieve a new performance record in 2B, 7B, and 8B settings for solving



(a) CoT Rephrasing (b) CoT Template (c) CoT Generation (d) TR-CoT (Ours)

Figure 1: Comparison of TR-CoT with existing CoT data generation approaches. (a) Rephrase existing Q&A pairs using LLMs, relying on existing CoT data. (b) Generate images and CoT data using pre-defined templates containing a limited number of theorems. (c) Generate CoT using LMMs, where accuracy is limited by the performance of the LMMs. (d) Design TR-Engine to generate images, corresponding descriptions, and geometric properties from theorems. And input the descriptions and properties into TR-Reasoner to generate reliable CoT Q&A pairs.

geometry problems. The main advantages of our method are summarized as follows:

- Compared to traditional template-based methods, our approach covers twice the number of theorem types, effectively correcting long-standing theorem misunderstandings in models and enhancing their geometric reasoning.
- Generating geometric data with fine-grained CoTs enhances the model's understanding of theorems, increasing the proportion of logically consistent and clear outputs by 24.5%.
- Our most advanced models achieve a 10.1% performance gain on MathVista and 4.7% on GeoQA over the baseline, outperforming advanced closed-source models such as GPT-4o.

2 Related Work

Enhancing Reasoning with CoT in Inference. Recent works leverage chain-of-thought (CoT) reasoning to improve mathematical and geometric problem-solving. KQG-CoT (Liang et al., 2023a) is a prompting method for few-shot Question Generation over Knowledge Bases (KBQG) that uses reasoning chains to select logical forms from an unlabeled data pool. In general math tasks, (Zhou et al., 2023) introduces code-based self-verification (CSV) to validate reasoning steps, while (Zhao et al., 2024b) (SSC-CoT) combines multiple reasoning chains with knowledge graph queries to reduce errors. Other approaches, such as Problem Elaboration Prompting (PEP) (Liao et al., 2024) Plan-and-Solve Prompting (PS) (Wang et al., 2023), and in-context example solutions (Didolkar et al., 2024), further refine reasoning accuracy across datasets. In geometry, (Zhao et al., 2024a) employs dual visual-symbolic CoT reasoning, and (Hu et al., 2024) generates code-based diagrams paired with visual CoT to align multimodal understanding.

Enhancing Reasoning in Geometry Training. Training robust geometric solvers faces two key challenges: scalability of datasets and diversity of geometric representations. Early symbolic systems like GeoS (Seo et al., 2015), Inter-GPS (Lu et al., 2021), and S2G (hung Tsai et al., 2021) established foundational deductive frameworks but were limited to small benchmarks (e.g., GeoS, Geometry3K). Subsequent neural solvers, such as UniGeo (Chen et al., 2022), PGPS9K (Zhang et al., 2023a) and LANS (Li et al., 2024b), expanded problem coverage to 36K examples, yet manual annotation costs remains. Recent efforts to automate data generation, including G-LLaVA (Gao et al., 2023a), leverage existing datasets like GeoQA to synthesize reasoning traces, while code-based engines (Kazemi et al., 2023b; Zhang et al., 2024b) prioritize procedural diagram generation. Additionally, GeoEval(Zhang et al., 2024a) assesses model performance across various subsets, including standard, reverse reasoning, augmented, and hard problems. Advances in LLM-generated CoT data (e.g., (Peng et al., 2024)) show promise for evolving reasoning capabilities of base models.

Recently, reverse engineering has emerged to diagnose and refine reasoning errors in LLMs. Meth-

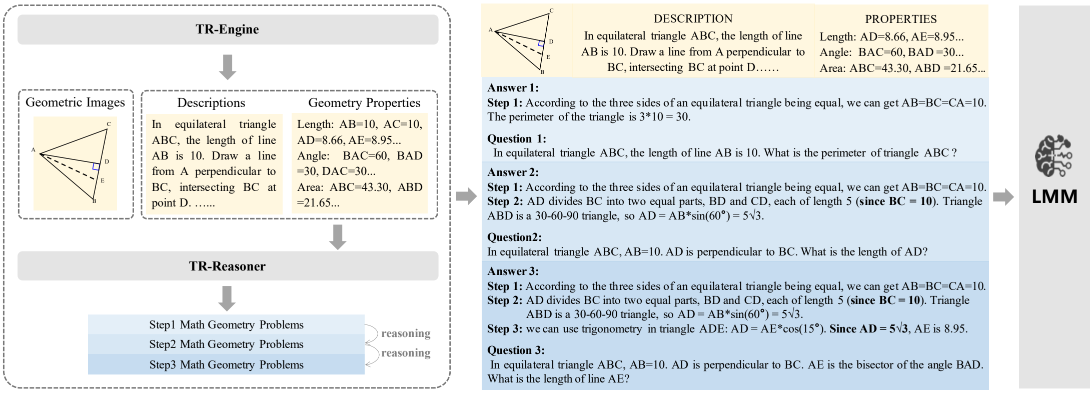


Figure 2: The TR-Engine generates diverse images, corresponding descriptions, and geometric properties step by step based on geometric theorems. Subsequently, the TR-Reasoner is utilized to obtain accurate geometric Q&A pairs from descriptions and properties.

ods like condition-answer swapping (Jiang et al., 2024; Weng et al., 2023) and error localization (Xue et al., 2023) verify logical consistency without updating model weights, while (Yuan et al., 2024) optimizes prompts based on initial reasoning traces. However, these approaches lack direct intervention in model training. Our work integrates reverse engineering into the CoT generation process, forming high-quality fine-grained reasoning process with accurate theorem comprehension.

3 Theorem-Validated Reverse Chain-of-Thought

There are two key challenges for generating geometry reasoning data: (1) Direct generation of question-answer pairs often leads to errors or unsolvable problems due to oversimplified scenarios. (2) Single-step reasoning processes lack validation of intermediate steps, compromising reliability.

We propose Theorem-Validated Reverse Chain-of-Thought (TR-CoT), a two-stage framework for creating geometry reasoning data with verified logical flow, as shown in Fig. 2. The pseudo-code of TR-CoT is shown in Appendix A.

1) Stage 1: **Theorem-Driven Image & Property Generation.** We collect 110 fundamental geometry theorems (the complete theorems are shown in Appendix G) and develop **TR-Engine**, a structured method to generate images paired with textual descriptions and geometric properties (e.g., angles, lengths). Unlike random image generation, TR-Engine enforces dependencies between geometric elements across generation steps. Current step's operation must perform on geometric objects(lines, angles, points, etc.) generated in the previous step.

2) Stage 2: **Q&A Generation with Stepwise Validation.** Using the descriptions and properties from Stage 1, we implement **TR-Reasoner** to derive questions from answers through three steps: First, the image description is divided into logical segments (e.g., "Triangle ABC is isosceles with AB=AC"). A language model processes these sequentially, generating single-step inferences. These inferences are progressively combined into multi-step reasoning chains. Secondely, for each reasoning step, the system creates corresponding questions. For instance, the inference "$\angle B = \angle C$" generates the question: "If triangle ABC is isosceles with AB=AC, which angles are equal?" Finally, all Q&A pairs are cross-checked against the wgeometric properties from Stage 1. Pairs violating mathematical rules (e.g., claiming "$\angle A = 90°$" for a non-right isosceles triangle) are automatically discarded.

3.1 TR-Engine

TR-Engine is a theorem-guided framework for synthesizing geometrically valid images with rich relational structures. TR-Engine operates through three key components (Fig. 3):

1) Geometric Substrate Library. We curate 20 foundational shapes (substrates) such as triangles, circles, and quadrilaterals, each paired with multiple description templates. These templates encode geometric conditions (e.g., "Triangle ABC has AB = 5 cm, BC = 6 cm") to anchor subsequent reasoning steps.

2) Theorem-Based Dynamic Element Injection. TR-Engine integrates 110 geometric theorems and defines element-combination rules, strategically injecting elements to enable complex reasoning scenarios based on theorem requirements. For example: Adding parallel lines to invoke properties of alternate angles. Introducing auxiliary lines (e.g.,

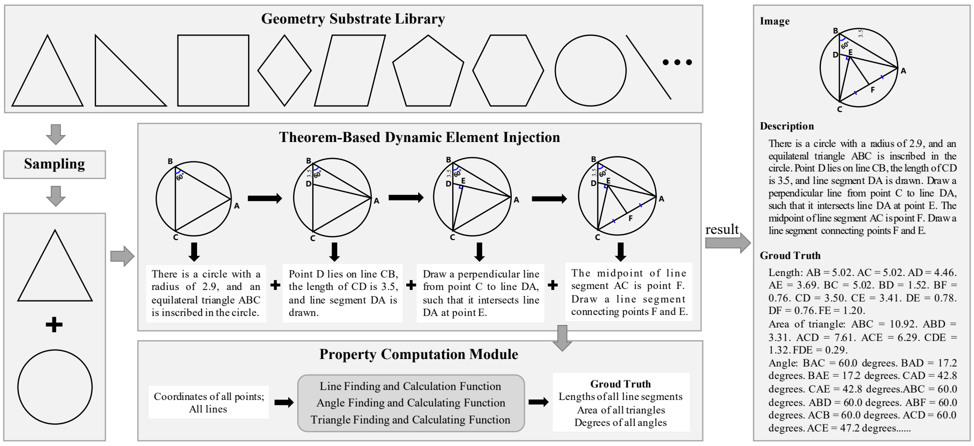


Figure 3: Overview of the TR-Engine. Starting from a Geometric Substrate Library, dynamically injecting elements based on theorems, and integrating a property computation module to enable multi-step geometric reasoning and validation in image generation.

medians, altitudes) to create congruent sub-shapes. Such operations expand reasoning opportunities while maintaining geometric validity.

3) Property Computation Module. As elements are added, coordinates of vertices are used to automatically calculate: Metric properties: Lengths, angles, areas. Relational properties: Parallelism, congruence, symmetry. These properties serve as ground truth for verifying generated Q&A pairs. By integrating theorem-driven construction with stepwise validation, TR-Engine ensures images inherently support multi-step geometric reasoning, which is a critical advance over prior generation methods in practice.

3.2 TR-Reasoner

Despite advances in large language models (LLMs), generating accurate and educationally viable geometric question-answer (Q&A) pairs remains challenging due to three persistent issues: (1) misapplication of geometric theorems in multi-step proofs, (2) diagram-text misalignment in problem formulation, and (3) inability to maintain answerability constraints during question generation. To address these limitations, we propose the TR-Reasoner to generate theorem-grounded Q&A pairs through coordinated interaction between geometric properties and structured reasoning chains (Fig. 4).

Description Patch Reasoning Fusion Building on the geometrically valid descriptions from TR-Engine, this module enforces logical coherence through causal dependencies between reasoning steps. Let $D = \{p_1, p_2, ..., p_x\}$ denote the x description patches extracted from an image, where each patch p_i corresponds to a geometrically meaningful component (e.g., "Circle O with chord AB and tangent CD"). The single-step reasoning r_i for patch p_i is generated through theorem-constrained transformation:

$$r_i = \mathcal{F}_{\text{LLM}}(p_i | r_{<i}, \mathcal{T}), \tag{1}$$

where $r_{<i} = \{r_1, ..., r_{i-1}\}$ represents preceding reasoning states, and $\mathcal{T}$ denotes the applicable theorem set (e.g., intersecting chords theorem for patch p_i describing chord intersections). This chained formulation ensures cumulative reasoning: later steps automatically inherit and extend prior conclusions (e.g., deriving arc lengths after establishing chord congruence).

Reverse Question Generation To prevent answerability drift, we implement *answer-constrained reverse generation* rather than open-ended question synthesis. Given a verified reasoning chain $R = \{r_1, r_2, ..., r_n\}$, each step r_i undergoes answerability assessment through a theorem-aware discriminator:

$$f_{aq}(r_i) = \begin{cases} f_q(r_i; \Phi_{\text{geo}}), & \text{if} \mathcal{V}(r_i, G_{\text{props}}) = \text{True} \\ \emptyset, & \text{otherwise} \end{cases} \tag{2}$$

where G_{props} denotes geometric properties from TR-Engine (e.g., coordinate-derived lengths), $\mathcal{V}$ performs theorem-based validation (e.g., checking triangle congruence rules), and f_q generates questions using a geometry-specialized LLM with instruction prompt Φ_{geo}. This approach leverages the granular reasoning steps from patch reasoning stage to generate fine-grained theorem-aware Q&A pairs.

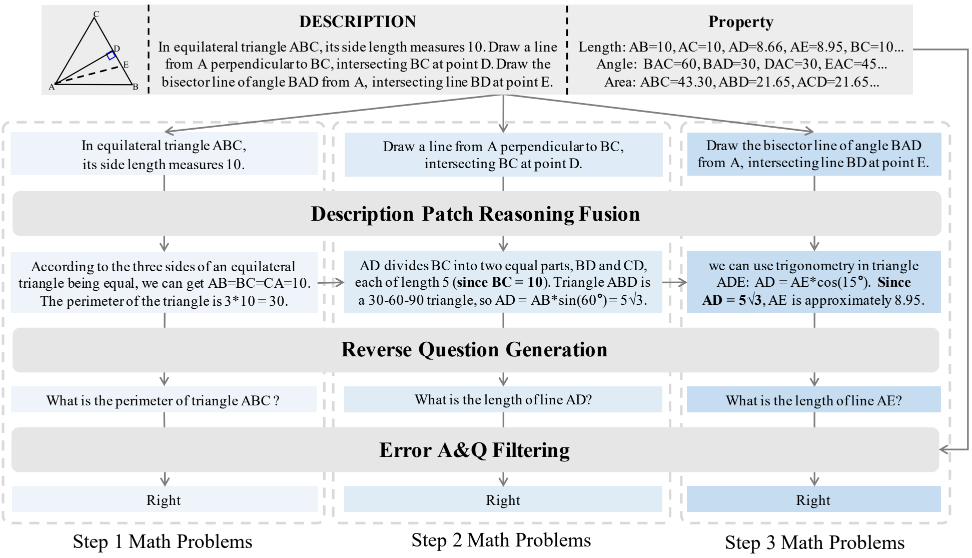


Figure 4: Overview of the TR-Reasoner. Image descriptions are segmented into patches to generate single-step reasoning results. Single-step reasoning results are fused progressively to get multi-step reasoning results. Then questions are generated based on the multi-step reasoning results. Finally, Q&A pairs that contradict geometric properties are filtered.

Error A&Q Filtering The final verification stage employs bidirectional theorem cross-checking to align generated content with ground-truth geometric properties. Forward validation ensures strict adherence of all Q&A pairs to the precomputed geometric properties (G_{props}), such as rejecting questions like "Find (AB)" if the model-generated answer contradicts coordinate-derived lengths. Reverse consistency analysis enforces answer uniqueness under theorem constraints, eliminating ambiguous questions that permit multiple valid answers under differing geometric conditions (e.g., congruence proofs lacking sufficient criteria). This pipeline systematically filters four categories of errors: theorem violations (36.3%), such as incorrectly citing the SSA congruence rule; metric discrepancies (24.9%), including arithmetic inconsistencies between stated lengths (e.g., claiming (AB + BC = 12) cm when (AB = 5) cm and (BC = 8) cm); diagram-text mismatches (12.2%), where questions reference nonexistent elements; and ambiguous answerability(26.5%), exemplified by underspecified tasks like "Prove similarity" without requisite premises.

Context-Aware Prompt Engineering To optimize reasoning, we deploy an instruction-based context-aware prompting strategy. Specifically, we pre-construct a reasoning instruction template pool, which includes a series of typical geometric problems with corresponding reasoning process. For each input, we sample three to four instruction templates that are most relevant based on the geometric figure and theorem included in the input. The sampled instruction templates serve as examples to assist the LLM to perform correct reasoning. In practice, such instruction-based context-aware prompt engineering ensures a relatively ideal reasoning accuracy, improving the efficiency of data generation. Details of prompt templates in Appendix B.

3.3 TR-GeoMM

Through the TR-CoT pipeline, we construct TR-GeoMM dataset containing diverse knowledge, designed to enhance LMM's geometric reasoning ability. We generate Q&A pairs from 15k figure, obtaining 45k high-quality Q&A pairs as the final dataset after error filtering, with an average of 3.49 questions per figure. Comprehensive statistical details of TR-GeoMM are illustrated in Fig. 6.

At the image level, TR-GeoMM comprises 20 substrate shapes, primarily triangles, quadrilaterals, and circles. Departing from conventional polygon-combining approaches, TR-Engine adopts lines as fundamental geometric elements. Key lines with distinctive properties such as midlines, angle bisectors, and radii are central to geometric theorems (e.g., midline theorems). As illustrated in Fig. 5 (a), TR-GeoMM contains 1.7k image patterns through sequential line additions derived from diverse theorem combinations. Each step ensures interaction

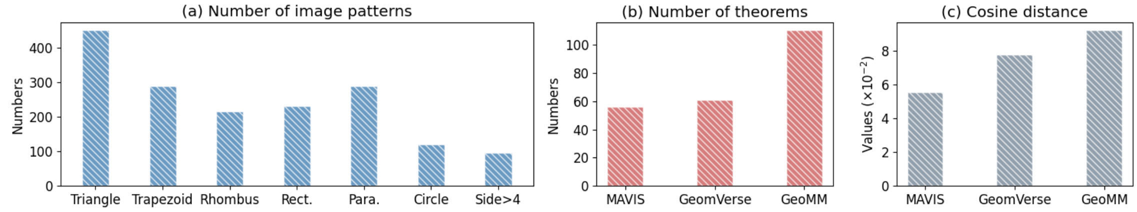


Figure 5: Diversity analysis of TR-GeoMM.

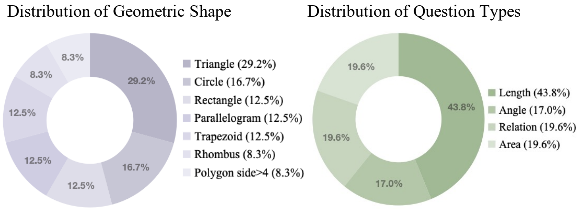


Figure 6: Statistical information about TR-GeoMM.

with existing elements (e.g., a new line's vertex must align with previously generated lines).

At the text level, the TR-GeoMM dataset organizes questions into four core problem types: side length, angle measurement, area calculation, and geometric relationships. The hierarchical structure of generated images leads to interdependent questions across stages, where initial solutions form prerequisites for subsequent problems. Such "subproblem" characteristic facilitates progressive learning of geometric concepts and complex relationships. As shown in Fig. 5 (b), TR-GeoMM contains a theorem repository twice as large as existing synthetic datasets (MAVIS and GeomVerse). Furthermore, Fig. 5 (c) demonstrates superior data diversity through higher Q&A pair cosine distances. Detailed information are provided in Appendix C.

3.4 TR-GeoSup

TR-CoT can not only synthesize reliable CoT geometric data but also be used to augment existing CoT data. Current real-world geometry CoT data usually contain several key intermediate steps which also contain abundant problem-solving knowledge. However, these intermediate steps are often implicitly or simplistically expressed, assuming human examinees' prior knowledge, which may hinder model training efficiency due to models' limited knowledge and reasoning abilities. By leveraging TR-CoT pipeline, we decompose the original CoT process into explicit theorem-aware steps, then reversely generate new Q&A pairs with TR-Reasoner.

Specifically, our augmentation involves three steps: generating a comprehensive multi-step analysis of the geometric figure, segmenting it into essential problem-solving steps, and creating new QA pairs for each step. These fine-grained Q&A pairs explicitly guide the model with theorems and knowledge implicitly expressed in original data, leading to improvement in comprehension and reasoning abilities. We applied TR-Reasoner to the GeoQA dataset, producing the TR-GeoSup dataset with 20k new entries. Final TR-GeoSup dataset does not contain the original GeoQA data. Examples of TR-GeoSup are shown in Appendix D.

4 Experiments

4.1 Setup

We train multiple LMMs (Wang et al., 2024; Liu et al., 2024; Chen et al., 2024c) using existing geometric instruction datasets (Chen et al., 2021; Gao et al., 2023b) and our TR-CoT generated data (TR-GeoMM and TR-GeoSup). Both the projected linear layer and the language model are trainable. The models are trained for two epochs with a batch size of 128 on 16$\times$ 64G NPU, and learning rate set to 5e-6. For evaluation, we assess these models on the geometry problem solving on the testmini set of MathVista (Lu et al., 2023) and GeoQA (Chen et al., 2021) following Gao et al. (2023b). Top-1 accuracy serves as the metric, with predictions and ground truth evaluated via ERNIE Bot 4.0. Ablation experiments were done on Intern-VL-2.0-8B.

4.2 Ablation Study

Data generating procedures. To assess the effectiveness of the key designs in TR-CoT, we construct several ablated variants by selectively removing different proposed components, as summarized in Tab. 1. Each variant is used to generate training data, and the resulting models are evaluated on MathVista and GeoQA. Q&A pairs generated based on descriptions outperform those generated directly from images, with performance improvements of 5.3% on MathVista and 6.3% on GeoQA. Reverse generation is designed to improve the accuracy of Q&A pairs. When using the reverse generation strategy, the accuracy is improved by 2.9% on MathVista and 2.6% on GeoQA. As a result, the full setting achieves the highest result on both datasets, demonstrating the effectiveness of each procedure in the TR-CoT pipeline.

Table 1: Ablation study on the data generating procedures. 'Description' represents generation based on descriptions. 'Reverse' represents generating reasoning followed by reverse question generation. 'Filter' represents filtering errors based on geometric properties.

Configurations			MathVista	GeoQA
Description	Reverse	Filter		
✗	✗	✗	55.3	44.2
✓	✗	✗	60.6	50.5
✓	✓	✗	63.5	53.1
✓	✓	✓	64.4	54.0

Separate validity of synthetic and augmented data. We evaluated the impact of the TR-GeoSup and TR-GeoMM datasets on model performance, as shown in Tab. 2. First, training with TR-GeoSup improved performance by 1.4% on MathVista and 7.9% on GeoQA compared to baseline. Furthermore, combining GeoQA with TR-GeoSup improves performance by 2.9% on MathVista and 3.9% on GeoQA compared to GeoQA alone, indicating their complementarity. This suggests that TR-CoT-augmented data enhances in-domain performance by better-extracting knowledge from existing data. Additionally, a deeper understanding of knowledge may contributes to improved generalization on mixed out-of-domain datasets.

Table 2: Ablation study on the TR-CoT generated data.

Configurations			MathVista	GeoQA
GeoQA	TR-GeoSup	TR-GeoMM		
✗	✗	✗	63.0	52.4
✓	✗	✗	64.9	64.8
✗	✓	✗	64.4	60.3
✗	✗	✓	64.4	54.0
✓	✓	✗	67.8	68.7
✓	✗	✓	65.4	67.9
✓	✓	✓	68.3	69.0

Second, training with TR-GeoMM improved performance by 1.4% on MathVista and 1.6% on GeoQA, confirming the strong generalization of TR-CoT synthetic data to real data. Moreover, joint training with GeoQA further improved performance, highlighting the effectiveness of synthetic data in supplementing real data. Finally, when jointly training on all three datasets (GeoQA, TR-GeoSup, and TR-GeoMM), the model achieved the best performance, with improvements of 5.3% on MathVista and 6.6% on GeoQA over the baseline. These results support that TR-CoT-generated data compensate for the limitations of existing datasets and enhance the model's reasoning capability.

Compared with other synthesis datasets. We train InternVL-2.0-8B using TR-GeoMM and two recent synthetic datasets for geometric problems, *i.e.* MAVIS (synthesis part) (Zhang et al., 2024b) and GeomVerse (Kazemi et al., 2023a), as summarized in Tab. 3. Compared to the baseline, models trained with GeomVerse or MAVIS show a slight performance gain on GeoQA and a decline on MathVista, both lower than TR-GeoMM. We attribute this to the limited diversity of image and Q&A pairs in these datasets, which benefits the simpler distribution of GeoQA but struggles with the diverse distributions in MathVista. In contrast, TR-GeoMM, with its diverse image and Q&A pairs, improves performance on both datasets.

Table 3: Compared with other synthesis datasets.

Dataset	MathVista	GeoQA
/	63.0	52.4
GeomVerse	58.2	53.6
MAVIS	57.2	53.2
TR-GeoMM k	64.4	54.0

4.3 Comparison with Previous State-of-the-Art

With the proposed method, we train three specialized models for geometry problem solving named TR-CoT-InternVL-2.0-2B, TR-CoT-Qwen2.5-VL-7B, and TR-CoT-InternVL-2.5-8B on joint dataset of Geo170K and TR-CoT-generated data (TR-GeoMM and TR-GeoSup). We compare our models with both general and mathematical LMMs on the geometry problems from testmini set of MathVista and the test set of GeoQA. As shown in Tab. 4, TR-CoT-InternVL-2.5-8B outperforms GPT-4o by 17.3% on MathVista and TR-CoT-Qwen2.5-VL-7B outperforms GPT-4o by 17.8% on GeoQA. Compared to mathematical LMMs, TR-CoT-InternVL-2.5-8B maintains a 11.1% lead on MathVista, and TR-CoT-Qwen2.5-VL-7B achieves a 12.5% advantage on GeoQA. For performance analysis on more baselines, please refer to Appendix H and Tab. 7.

5 Discussion

Fig. 7 highlights consistent improvements: post-trained models produce concise, logical CoTs with

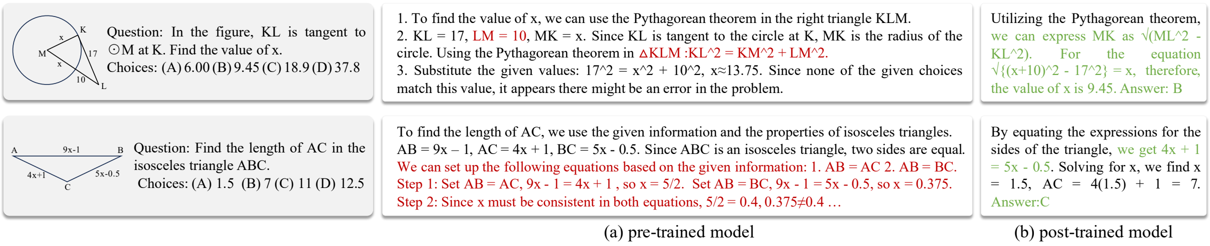


Figure 7: Comparison of model problem solving before and after training.

Table 4: Top-1 Accuracy (%) on geometry problem solving on the testmini set of MathVista and the GeoQA test set. * represents the results from the existing papers.

Model	MathVista	GeoQA
Closed-source LMMs		
GPT-4o (Islam and Moushi, 2024)	60.6	61.4
GPT-4V	51.0*	43.4*
Gemini Ultra (Team et al., 2023)	56.3*	-
Open-source LMMs		
LLaVA2-13B (Liu et al., 2024)	29.3*	20.3*
mPLUG-Owl2-7B (Ye et al., 2024)	25.5	21.4
Qwen-VL-Chat-7B (Bai et al., 2023)	35.6	26.1
Monkey-Chat-7B (Li et al., 2024a)	24.5	28.5
Deepseek-VL-7B (Lu et al., 2024)	34.6	33.7
InternVL-2.0-2B (Chen et al., 2024c)	46.2	38.2
InternLM-XC2-7B (Zhang et al., 2023b)	51.4	44.7
InternVL-1.5-20B (Chen et al., 2024b)	60.1	49.7
Qwen2-VL-7B (Wang et al., 2024)	55.1	55.7
InternVL-2.0-8B (Chen et al., 2024c)	65.9	56.5
InternVL-2.5-8B (Chen et al., 2024a)	67.8	59.0
Qwen2.5-VL-7B (Wang et al., 2024)	71.6	74.5
Open-source Mathematical LMMs		
UniMath (Liang et al., 2023b)	-	50.0*
Math-LLaVA-13B (Shi et al., 2024)	56.5*	47.8
G-LLaVA-7B (Gao et al., 2023b)	53.4*	62.8*
MAVIS-7B (Zhang et al., 2024b)	-	66.7*
PUMA-Qwen2-7B (Zhuang et al., 2024)	48.1*	-
MultiMath-7B (Peng et al., 2024)	66.8*	-
TR-CoT-InternVL-2.0-2B	56.3	63.4
TR-CoT-Qwen2.5-VL-7B	74.5	**79.2**
TR-CoT-InternVL-2.5-8B	**77.9**	76.7

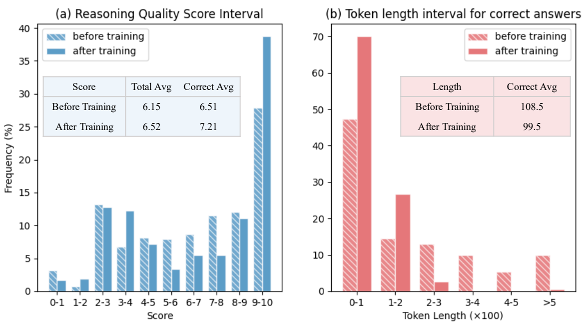


Figure 8: Comparison of model output quality and token length before and after training.

accurate conclusions, demonstrating robust geometric understanding. Pre-trained models show recurring errors (e.g., misdefining isosceles triangles as having two equal sides), reflecting foundational gaps in theorem comprehension. Our approach trains models on diverse theorems with structured reasoning, addressing these errors and enhancing general geometric problem-solving.

We use DeepSeek R1 and ERNIE Bot 4.0 to quantitatively evaluate model outputs before and after training, focusing on logical consistency, clarity, and lack of ambiguity (see Appendix I for detailed information). We use average score of the two models as final score. As shown in Fig. 8 (a), the total mean score increased by 0.37 after training, the mean score for correct answers increased by 0.70, and outputs with scores of 8 or higher increased by 24.5%. We attribute these improvements to TR-CoT's explicit focus on the reasoning process, where step decomposition enhances the model's logical consistency and rigor.

We further compare the token usage for correct answers before and after training. As shown in Fig. 8 (b), the model after training requires fewer tokens on average, with the percentage of correct answers within 200 tokens increasing by 35%. We assume this improvement results from the data diversity, which enables the model to find more efficient solutions across different theorems, while a deeper understanding of the theorems allows for more concise reasoning.

6 Conclusion

We propose TR-CoT, a novel theorem-based reverse generation pipeline that enhances theorem coverage and supports fine-grained theorem understanding in geometric datasets. Models trained on TR-CoT data demonstrate a significant improvement in geometric problem solving with more concise and rigorous reasoning. We will extend this approach to other mathematical domains to further analyze the impact of theorem mastery on problem-solving, offering insights for future research.

7 Limitations

During our practice, there still exist limitations that can be further improved.

For our method, one major constraint is that there is still room for further improvement in the generation efficiency. The overall efficiency can be divided into time efficiency and data efficiency. First, in our process, LLM is called multiple times for reasoning generation. The limited reasoning speed of LLM becomes the bottleneck of time efficiency. In addition, although we have adopted various methods to improve the reasoning accuracy of LLM, due to the limitations of model performance, there is still a certain proportion of errors in the direct output of the model. We observe that about 10% of the direct output is deleted in the Error A&Q Filtering stage.

References

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. *arXiv preprint arXiv:2308.12966*.

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. *Preprint*, arXiv:2212.02746.

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.

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

Zhe Chen, Weiyun Wang, Hao Tian, Shenglong Ye, Zhangwei Gao, Erfei Cui, Wenwen Tong, Kongzhi Hu, Jiapeng Luo, Zheng Ma, et al. 2024b. How far are we to gpt-4v? closing the gap to commercial multimodal models with open-source suites. *arXiv preprint arXiv:2404.16821*.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. 2024c. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 24185–24198.

Jacob Devlin. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Aniket Didolkar, Anirudh Goyal, Nan Rosemary Ke, Siyuan Guo, Michal Valko, Timothy Lillicrap, Danilo Rezende, Yoshua Bengio, Michael Mozer, and Sanjeev Arora. 2024. Metacognitive capabilities of llms: An exploration in mathematical problem solving. *Preprint*, arXiv:2405.12205.

Yue Fan, Jing Gu, Kaiwen Zhou, Qianqi Yan, Shan Jiang, Ching-Chen Kuo, Yang Zhao, Xinze Guan, and Xin Wang. 2024. Muffin or chihuahua? challenging multimodal large language models with multipanel vqa. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6845–6863.

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

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. *Preprint*, arXiv:2406.09403.

Shih hung Tsai, Chao-Chun Liang, Hsin-Min Wang, and Keh-Yih Su. 2021. Sequence to general tree: Knowledge-guided geometry word problem solving. *Preprint*, arXiv:2106.00990.

Raisa Islam and Owana Marzia Moushi. 2024. Gpt-4o: The cutting-edge advancement in multimodal llm. *Authorea Preprints*.

Weisen Jiang, Han Shi, Longhui Yu, Zhengying Liu, Yu Zhang, Zhenguo Li, and James T. Kwok. 2024. Forward-backward reasoning in large language models for mathematical verification. *Preprint*, arXiv:2308.07758.

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

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

Zhang Li, Biao Yang, Qiang Liu, Zhiyin Ma, Shuo Zhang, Jingxu Yang, Yabo Sun, Yuliang Liu, and Xiang Bai. 2024a. Monkey: Image resolution and text label are important things for large multi-modal models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 26763–26773.

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.

Yuanyuan Liang, Jianing Wang, Hanlun Zhu, Lei Wang, Weining Qian, and Yunshi Lan. 2023a. Prompting large language models with chain-of-thought for few-shot knowledge base question generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4329–4343, Singapore. Association for Computational Linguistics.

Zhenwen Liang, Tianyu Yang, Jipeng Zhang, and Xiangliang Zhang. 2023b. Unimath: A foundational and multimodal mathematical reasoner. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7126–7133.

Haoran Liao, Jidong Tian, Shaohua Hu, Hao He, and Yaohui Jin. 2024. Look before you leap: Problem elaboration prompting improves mathematical reasoning in large language models. *Preprint*, arXiv:2402.15764.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2024. Visual instruction tuning. *Advances in neural information processing systems*, 36.

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Hao Yang, et al. 2024. Deepseek-vl: Towards real-world vision-language understanding. *CoRR*.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2023. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The 3rd Workshop on Mathematical Reasoning and AI at NeurIPS'23*.

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. *Preprint*, arXiv:2105.04165.

OpenAI. 2024. Openai o1 system card. *preprint*.

Shuai Peng, Di Fu, Liangcai Gao, Xiuqin Zhong, Hongguang Fu, and Zhi Tang. 2024. Multimath: Bridging visual and mathematical reasoning for large language models. *arXiv preprint arXiv:2409.00147*.

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.

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. *arXiv preprint arXiv:2406.17294*.

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

Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2609–2634, Toronto, Canada. Association for Computational Linguistics.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yixuan Weng, Minjun Zhu, Fei Xia, Bin Li, Shizhu He, Shengping Liu, Bin Sun, Kang Liu, and Jun Zhao. 2023. Large language models are better reasoners with self-verification. *Preprint*, arXiv:2212.09561.

Tianci Xue, Ziqi Wang, Zhenhailong Wang, Chi Han, Pengfei Yu, and Heng Ji. 2023. Rcot: Detecting and rectifying factual inconsistency in reasoning by reversing chain-of-thought. *Preprint*, arXiv:2305.11499.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. 2024. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13040–13051.

Jiahao Yuan, Dehui Du, Hao Zhang, Zixiang Di, and Usman Naseem. 2024. Reversal of thought: Enhancing large language models with preference-guided reverse reasoning warm-up. *Preprint*, arXiv:2410.12323.

Jiaxin Zhang, Zhong-Zhi Li, Ming-Liang Zhang, Fei Yin, Cheng-Lin Liu, and Yashar Moshfeghi. 2024a. GeoEval: Benchmark for evaluating LLMs and multi-modal models on geometry problem-solving. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1258–1276, Bangkok, Thailand. Association for Computational Linguistics.

Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. 2023a. A multi-modal neural geometric solver with textual clauses parsed from diagram. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*, pages 3374–3382.

Pan Zhang, Xiaoyi Dong Bin Wang, Yuhang Cao, Chao Xu, Linke Ouyang, Zhiyuan Zhao, Shuangrui Ding, Songyang Zhang, Haodong Duan, Hang Yan, et al. 2023b. Internlm-xcomposer: A vision-language large model for advanced text-image comprehension and composition. *arXiv preprint arXiv:2309.15112*.

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

Xueliang Zhao, Xinting Huang, Tingchen Fu, Qintong Li, Shansan Gong, Lemao Liu, Wei Bi, and Lingpeng Kong. 2024a. Bba: Bi-modal behavioral alignment for reasoning with large vision-language models. *Preprint*, arXiv:2402.13577.

Zilong Zhao, Yao Rong, Dongyang Guo, Emek Gözlüklü, Emir Gülboy, and Enkelejda Kasneci. 2024b. Stepwise self-consistent mathematical reasoning with large language models. *Preprint*, arXiv:2402.17786.

Aojun Zhou, Ke Wang, Zimu Lu, Weikang Shi, Sichun Luo, Zipeng Qin, Shaoqing Lu, Anya Jia, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2023. Solving challenging math word problems using gpt-4 code interpreter with code-based self-verification. *Preprint*, arXiv:2308.07921.

Wenwen Zhuang, Xin Huang, Xiantao Zhang, and Jin Zeng. 2024. Math-puma: Progressive upward multimodal alignment to enhance mathematical reasoning. *arXiv preprint arXiv:2408.08640*.

A Pseudo Code

We have written pseudo-code for the overall flow of TR-CoT, the details of which are given in Algor. 1.

Algorithm 1: Pseudo-code of TR-CoT

Input: Geometry substrates sampling rounds n, plot function f, image-description pair sets $\mathcal{S}$, line sampling rounds k, geometric property calculation module $\mathcal{V}$, large language model $\mathcal{M}$

Output: Generated Image $\mathcal{I}$, Description $\mathcal{D}$, Geometric Properties $\mathcal{T}$, Question $\mathcal{Q}$; Answer $\mathcal{A}$

Initialization: $\mathcal{I} \leftarrow \emptyset$, $\mathcal{D} \leftarrow \emptyset$, $\mathcal{T} \leftarrow \emptyset$, vertex coordinate $\mathcal{C} \leftarrow \emptyset$, $r_s \leftarrow \emptyset$

for $i \leftarrow 1$ *to* n **do**

  Sample geometry substrate $\mathcal{G}_i$ and description $\mathcal{D}_i$ from image-description pair sets $\mathcal{S}$

  Refresh $\mathcal{I}$ using plot function: $\mathcal{I} \leftarrow f(\mathcal{I}, \mathcal{G}_i)$

  Refresh corresponding description: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{D}_i$

  Refresh vertex coordinate: $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_i$

end

for $j \leftarrow 1$ *to* k **do**

  Select line drawing position $\mathcal{P}_j$

  Draw line and label length: $\mathcal{I} \leftarrow f(\mathcal{I}, \mathcal{P}_j)$

  Refresh corresponding description: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{P}_j$

  Refresh vertex coordinate: $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}_i$

  if $j = k$ **then**

    Calculate all angle information $\mathcal{R}$

    Draw angles and label degrees: $\mathcal{I} \leftarrow f(\mathcal{I}, \mathcal{R})$

    Refresh corresponding description: $\mathcal{D} \leftarrow \mathcal{D} \cup \mathcal{R}$

  end

end

Refresh Geometric Properties: $\mathcal{T} \leftarrow \mathcal{V}(C)$

Produce single-step reasoning result r_c using prompt P_s: $r_c \leftarrow \mathcal{M}(\mathcal{D}, P_s)$

Generate answer A_e and its corresponding question Q_e using prompt P_q: $A_e, Q_e \leftarrow \mathcal{M}(r_c, P_q)$

Filtering for correct answer A and its corresponding question Q using prompt P_e: $A, Q \leftarrow \mathcal{M}(A_e, Q_e, T, P_e)$

Return: $\mathcal{I}$, $\mathcal{D}$, $\mathcal{Q}$, $\mathcal{A}$

B Details of prompt in TR-Reasoner

We used ERNIE Bot 4.0 to implement TR-Reasoner. We describe the prompts used in TR-Reasoner, including the prompts for the Description Patch Reasoning Fusion (Fig. 9), the Reverse Question Generation (Fig. 10), and the Error A&Q Filtering (Fig. 11). In these figures, the texts in blue represent the Task Description, while the texts in orange represent the input information. Each prompt includes three contextual examples, and we show only one of them, with the remaining examples replaced by ellipses. In addition to the examples, some prompts also include an instruction section that specifies more detailed requirements, some incorporate additional basic knowledge, and others outline more specific goals that must be achieved.

Description Patch Reasoning Fusion Prompt

Assume expertise in geometry. Perform accurate inferences based on image descriptions, summarize analysis, and provide logical, correct reasoning for each level of description based on the provided data.

Instructions:

1. *Inference Process*: Analyze each description step by step and derive conclusions.
2. *Relevance*: If a description is irrelevant, state why and ignore it.
3. *Output Format*: Present inference as numbered points, concise and precise.

Input:

Step 1: In triangle ABC, length of side AB is 4, length of side AC is 6, length of side BC is 6.Step 2: CD is the bisector of angle ACB, D lies on line segment AB. Step 3: E is the midpoint of CD, connect E and B

Output:

Reasoning 1: Triangle ABC has sides AB= 4, AC = 6, BC = 6. Since AC = BC, triangle ABC is isosceles. The perimeter of triangle ABC is AB + AC + BC = 4 + 6 + 6 = 16.Reasoning 2: CD bisects angle ACB, forming 2 equal angle ACD and BCD. Also, according to the property of isosceles triangle, CD is also Perpendicular bisector of AB. We can know AD = BD = 0.5 \times AB = 2, and using the Pythagorean theorem, we get CD = √(AC^2 - AD^2) CD = 4 √ 2. Reasoning 3: As E being the midpoint of CD, CE = ED = 0.5 \times CD = 2 √ 2. Connecting BE form a new triangle BDE, and angle BDE = 90 degrees as CD is perpendicular to AB. Using the Pythagorean theorem, we can get BE = √(BD^2 + DE^2) = 2\sqrt(3)

……

Input: [DESCRIPTION]

Output: [INFERENCE]

Basic knowledge:

1. sin(15 °) =cos(75 °) ≈ 0.2588 , cos(15 °)= sin(75 °) ≈ 0.9659.
2. The radius of circle equals the side length of its inscribed regular hexagon.
4. In a hexagon, diagonals CA, AC, EA, AE, DB, BD, FB, BF, EC, CE are √ 3*side length; diagonals DA, AD, EB, BE, FC, CF are 2 *side length.
5. In a square, the radius of the circle is r, and the side length of the inscribed square is √ 2*r.
6. In an equilateral triangle, the radius of the circle is r , and the side length of the inscribed triangle is √ 3*r.

Figure 9: The prompt of the Description Patch Reasoning Fusion.

Reverse Question Generation Prompt

Assume you're an expert in geometry. Your task is to generate questions and step-by-step answers from provided hierarchical geometry descriptions. For each level of detail, create a specific, calculation-based question with a precise answer. Ensure questions and answers align with the given reasoning.

Instructions:

1. *Analysis*: Analyze hierarchical geometry descriptions for key properties and relationships, ensuring logical consistency.
2. *Question and Answer Generation*: For each step, create specific, calculation-based questions with detailed answers, using appropriate geometric principles.
3. *Output Format*: Provide questions and answers in a clear, structured format as shown in the example below.

Input:

Step 1: Equilateral triangle ATW has side length AT = 9.Step 2: AG is perpendicular to TW, with G on TW. Step 3: M is the midpoint of AT, and MG is drawn. Reasoning 1: Reasoning 1: All sides of the equilateral triangle are equal, so AW = TW = 9. Reasoning 2: Since AG is perpendicular to TW and G lies on TW, AG is the height of the equilateral triangle ATW. The height h of an equilateral triangle can be calculated as h = $\frac{\sqrt{3}}{2}$ *side length. Therefore, AG = $\frac{9\sqrt{3}}{2}$ .G is the midpoint of TW, so TG = GW = 9/2 = 4.5. Reasoning 3: GM is the midline of triangle ATW, parallel to AW, and half its length. Therefore, GM = 4.5.

Output:

Question 1: In equilateral triangle ATW, with AT = 9, what are the lengths of sides AW and TW? Answer 1: Since all sides of an equilateral triangle are equal, AT = AW = TW = 9.. The final answer: AW = TW = 9 Question 2: In equilateral triangle ATW, with AT = 9, and AG perpendicular to TW at G, what is the length of AG? Answer 2: All three sides of the equilateral triangle are equal, so AT = AW = TW = 9. Since AG is the height of the triangle, h = $\frac{\sqrt{3}}{2}$ *side length. Substituting, AG = $\frac{9\sqrt{3}}{2}$. Question 3: In equilateral triangle ATW, with AT = 9, AG perpendicular to TW, G on TW, and M the midpoint of AT, what is the length of GM? Answer 3: With M as the midpoint of AT and G as the midpoint of TW, GM is the midline of triangle ATW. The midline is parallel to one side and has half its length. Since AW = 9, GM = $\frac{9}{2}$ = 4.5. The final answer is GM = 4.5

……

Input: [DESCRIPTION AND REASONING]

Output: [QUESTION AND ANSWER]

Figure 10: The prompt of the Reverse Question Generation.

Error A&Q Filtering Prompt

All the correct answers are given in Ground Truth, if the result obtained in Answer contradicts the Ground Truth, return Score: 0, otherwise return Score: 1.

Input:

Ground Truth: Length of segment OA = 2.00. Length of segment OB = 2.00. Length of segment AC = 1.04. Length of segment BC = 4.00. Length of segment BD = 1.04. Angle OAC measures 75.0 degrees. Angle AOB measures 150.0 degrees. Angle ACB measures 75.0 degrees. Angle OBD measures 75.0 degrees. Angle CBD measures 75.0 degrees.Question: There is a circle O with a radius of 2. The central angle AOB measures 150 degrees. Line BD is parallel to line CA. What is the relationship between angle DBC and angle ACB?Answer: Since line BD is parallel to line CA, according to the properties of parallel lines, the corresponding angles are equal. Therefore, angle DBC is equal to angle ACB.

Output:

Analysis: From the ground truth we can get angle ACB measures 75.0 degrees and angle CBD measures 75.0 degrees. So angle ACB = angle DBC.Score: 1

......

Input: [GROUND TRUTH AND ANSWER]

Output: [SCORE]

What must be achieved:

1. When Ground Truth is the same as the content of the Answer, Score: 1.
2. When Ground Truth contradicts the content of the Answer, Score: 0.
3. When Ground Truth is not related to the content of the Answer, Score:
4. You don't need to focus on the solution process, as long as the result agrees with the Ground Truth, return Score: 1.

Figure 11: The prompt of the Error A&Q Filtering.

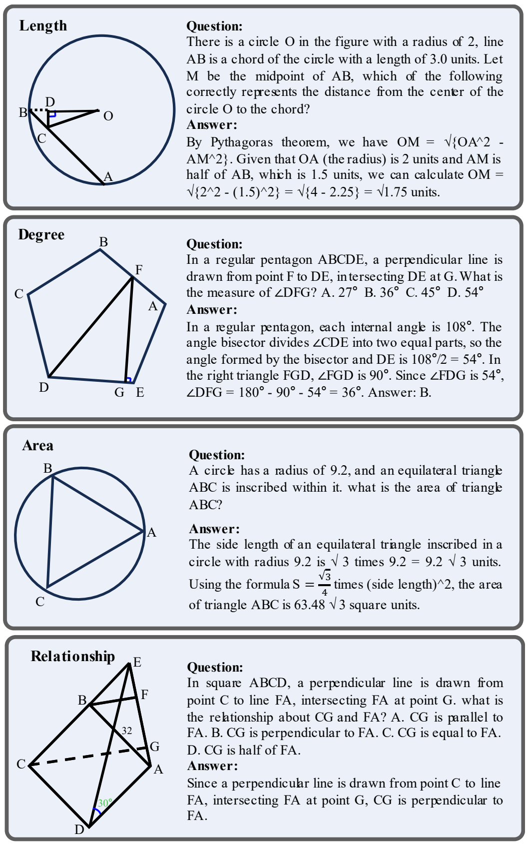


Figure 12: Examples of TR-GeoMM dataset.

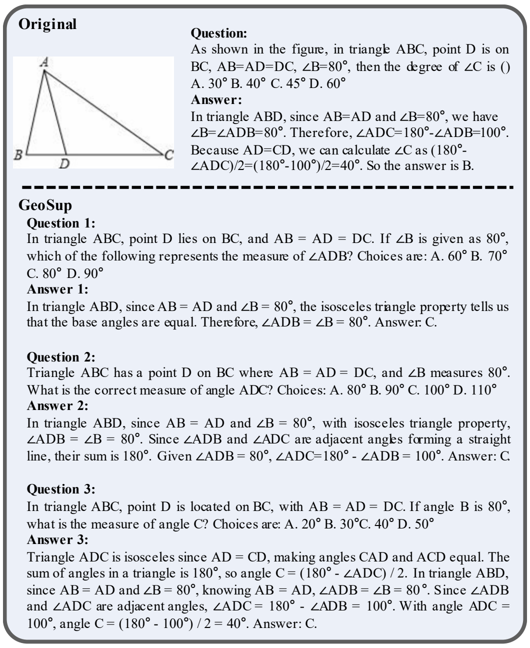


Figure 13: Examples of TR-GeoSup dataset.

C More information of TR-GeoMM

Through the TR-CoT, we construct a high-quality geometric dataset, TR-GeoMM. In Fig. 12, we provide a detailed overview of specific cases from TR-GeoMM. These cases demonstrate the variety of mathematical geometry question types covered by TR-GeoMM, including solving for lengths, angles, areas, and geometry elemental relations. Each of these categories is critical for improving the geometric reasoning ability of LMMs.

For Cosine distance based data diversity, we first randomly sample 5000 instances from each dataset(MAVIS, GeomVerse and TR-GeoMM), then we encode the instances into embedding features using pretrained BERT model (Devlin, 2018). Finally, we calculate the average cosine distance of each dataset using the BERT output features. Higher distance score indicates better diversity, and our TR-GeoMM has the highest distance score among the three dataset.

D Examples of TR-GeoSup dataset

Fig. 13 illustrates an example from the TR-GeoSup dataset, showcasing the transformation of a multi-step reasoning problem from the original GeoQA dataset. In the original Q&A pair, the reasoning process is condensed and lacks explicit intermediate steps, relying on implicit knowledge. TR-GeoSup decomposes the original reasoning process into three hierarchical sub-questions, each accompanied by a detailed and theorem-aware reasoning chain. This augmentation not only clarifies the implicit knowledge embedded in the original data but also provides a step-by-step guide for model training.

E Detail of polygon distribution

We conducted robustness experiments for different polygon distributions, where the details of the polygon distributions are shown in Tab. 5. From top to bottom, the percentage of triangles and quads gradually decreases, and the percentage of pentagons and hexagons gradually increases. There is also a clear difference in the percentage of circles.

Similar quantitative results within 0.6% in Tab. 6 show the impact of polygon distributions is almost negligible, demonstrating the strong robustness of our method to different polygon distributions. Therefore, the performance gain is mainly attributed to the diverse geometry representation and reasoning knowledge provided by our method.

Table 5: Details of polygon distribution for distributional robust ablation studies.

Method	Polygon Distribution				
	triangle	quad	circle	pentagon	hexagon
Group I	29%	46%	17%	5%	3%
Group II	32%	40%	14%	8%	6%
Group III	25%	33%	21%	12%	8%

Table 6: Ablation study on the robustness to polygonal distributions.

Polygon Distribution	MathVista	GeoQA
Group I	64.4	54.0
Group II	64.4	53.7
Group III	63.9	53.4

F The Case of Direct Generation and TR-Reasoner Generation

The core idea of the TR-Reasoner is to improve the accuracy of Q&A pairs by simplifying the reasoning based on descriptions and then generating corresponding questions from the answers in a reversed manner. A straightforward approach is directly prompting ERNIE Bot 4.0 to generate Q&A pairs from the input image description. However, as shown in the left of Fig. 15, this approach often fails to determine the correct answer. In contrast, the Q&A pairs produced by TR-Reasoner are correct for all three instances with our design.

G Details of the theorems

The support of mathematical theorems is crucial for the accuracy of TR-Engine. In Tab. 8, we present the geometric theorems and properties that we used. These define the rules for combining elements, establishing a logically coherent chain throughout the figure construction process. They serve as the foundation for extending reasoning scenarios and also assist in the computation and verification of question-answer pairs.

Quality Judgment Prompt

You are provided with a language model's response to a geometric question. Your mission is to judge the quality of the response based on the following standards, and give a score between 0 to 10.

Judging Standards:

1.*Logic consistency*. Assess whether the response is self-consistent, logically coherent, and free from contradictions or illogical reasoning.

2.*Clarity*. Evaluate whether the response is clear and easy to understand, avoiding ambiguity or vague expressions.

3.*Output format*: Score: your score(from 0 to 10).

Figure 14: Comparison of model problem solving before and after training.

H Effectiveness of TR-CoT

As shown in Fig. 7, models jointly trained on Geo170K and TR-CoT-generated data (TR-GeoMM and TR-GeoSup) consistently outperform those trained solely on Geo170K ('Geo-'). InternVL2.5-8B receives a 1.5% improvement on MathVista and GeoQA, and Qwen2.5-VL-7B improves by 1.0% and 2.0% on MathVista and GeoQA respectively. These results indicate that TR-CoT-generated data can supplement existing datasets and is widely effective in various LMMs.

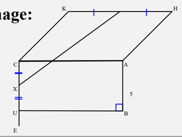
Table 7: TR-CoT generated data effectiveness validation on different models. 'Geo-' indicates the model is fine-tuned only with geometric instruction data of Geo170K. Consistent and significant improvement without adding any additional parameters.

Model	MathVista	GeoQA
Geo-InternVL-2.0-2B	51.9	62.5
TR-CoT-InternVL-2.0-2B	56.3 (4.4↑)	63.4 (0.9↑)
Geo-LLaVA-1.5-7B	27.9	47.6
TR-CoT-LLaVA-7B	29.3 (1.4↑)	51.7 (4.1↑)
Geo-Qwen2-VL-7B	59.9	69.1
TR-CoT-Qwen2-VL-7B	67.6 (7.7↑)	70.4 (1.3↑)
Geo-InternVL-2.0-8B	70.2	74.9
TR-CoT-InternVL-2.0-8B	72.1 (1.9↑)	76.7 (1.8↑)
Geo-InternVL-2.5-8B	76.4	75.2
TR-CoT-InternVL-2.5-8B	77.9 (1.5↑)	76.7 (1.5↑)
Geo-Qwen2.5-VL-7B	73.5	77.2
TR-CoT-Qwen2.5-VL-7B	74.5 (1.0↑)	79.2 (2.0↑)

I Details of CoT quality evaluation

We used ERNIE Bot 4.0 and DeepSeek R1 to evaluate model outputs. For each response, the evaluation model gives a score between 0 and 10 to judge the logical consistency, clarity and lack of ambiguity. We use average score of the two models as final score. To ensure more accurate evaluation, we include specific judging standards. The prompts used are shown in Fig. 14. The blue part represents the Task Description.

Image:



Description:

There is a parallelogram in the diagram, represented by the symbol AHKC. There is also a rectangle in the diagram, represented by the symbol UBAC. The length of line BA is 5. Line QX is the line connecting the midpoint Q of line HK and the midpoint X of line CU. Point Q lies on line HK. Point X lies on line CU. The length of line QH equals the length of line QK. The length of line XC equals the length of line XU.UE is the extension line of CU, that is, C, U, and E are collinear. Line AB is perpendicular to line UB. Point B lies on line UB.

There is a parallelogram in the diagram, represented by the symbol AHKC. There is also a rectangle in the diagram, represented by the symbol UBAC. The length of line BA is 5. Line QX is the line connecting the midpoint Q of line HK and the midpoint X of line CU... Generate reasonable math QA pairs based on image descriptions.

Q1: What is the length of line QX in the diagram?
A1: Since QX connects the midpoints of lines HK and CU in a parallelogram AHKC and rectangle UBAC, the length of QX corresponds to half the diagonal of rectangle UBAC. However, without additional information about the rectangle' s dimensions, we cannot determine the exact length of QX.

Q2: Are the lines AB and UE parallel in the diagram?
A2: No, the AB and UE are not parallel because AB is perpendicular to line UB, which means it is also perpendicular to CU since UBAC is a rectangle. Line UE, being an extension of CU, would also be perpendicular to AB.

Q3: Line AB is perpendicular to line UB, what can be said about the angle at point B?
A3: Since AB is perpendicular to UB, the angle B is 90 degrees.

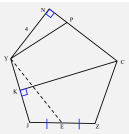
TR-Reasoner

Q1: There is a parallelogram in the diagram, represented by the symbol AHKC. There is also a rectangle in the diagram, represented by the symbol UBAC. The length of line BA is 5. What is the length of line CU?
A1: Since BA is a side of the rectangle UBAC, and its length is given as 5, we can infer that the length of CU is also 5.

Q2: There is a parallelogram...The length of line BA is 5. Line QX is the line connecting the midpoint Q of line HK and the midpoint X of line CU. What is the length of line XU?
A2: Since BA is a side of the rectangle UBAC, and its length is given as 5, we can infer that the length of CU (the opposite side in the rectangle) is also 5. X being the midpoint of CU means that XU=XC=CU/2=2.5. The answer is 2.5.

Q3: There is a parallelogram...Line QX is the line connecting the midpoint Q of line HK and the midpoint X of line CU. What is the length of line QH if the length of line HK is 8?
A3: Q being the midpoint of HK means that QH=QK=HK/2. Substituting the given value HK=8, we get QH=8/2=4.

Image:



Description:

There is a regular pentagon in the diagram, represented by the symbol CPYJZ. The length of line PY is 4. Line YE is the line connecting point Y and the midpoint E of line JZ. Point E lies on line JZ. The length of line EJ equals the length of line EZ.Line YN is perpendicular to line CP. Point N lies on line CP. Line CK is perpendicular to line YJ. Point K lies on line YJ.

There is a regular pentagon in the diagram, represented by the symbol ... Generate reasonable math QA pairs based on image descriptions.

Q1: What is the length of line YE in the diagram?
A1: Since E is the midpoint of line JZ in a regular pentagon CPYJZ with side length 4, line YE can be considered as the apothem of the pentagon. The length of the apothem in a regular pentagon can be calculated using the formula a = s / (2 * tan(π / 5)), where s is the side length of the pentagon. Substituting s = 4, we get a ≈ 2.75.

Q2: What is the length of line YN in the diagram?
A2: Line YN is perpendicular to line CP and passes through point Y, which is a vertex of the regular pentagon CPYJZ. In a regular pentagon, the length of a perpendicular from any vertex to the opposite side is the same as the apothem. Therefore, the length of line YN is approximately 2.75.

Q3: What is the length of line CK in the diagram?
A3: Line CK is perpendicular to line YJ and passes through point C, which is a vertex of the regular pentagon CPYJZ. Since CPYJZ is a regular pentagon, all its sides are of equal length, and all its angles are equal. The length of line CK can be calculated by considering the right triangle formed by line CK, line KC (which is a part of line YJ), and line YK (which is a part of line JZ). However, without additional information or assumptions, we cannot determine the exact length of line CK.

TR-Reasoner

Q1: There is a regular pentagon in the diagram, represented by the symbol CPYJZ. The length of line PY is 4. Line YE is the line connecting point Y and the midpoint E of line JZ. What is the length of line EZ?
A1: According to the five sides of a regular pentagon being equal, we can get CP=PJ=JY=YZ=ZC=PY=4. According to E is the midpoint of JZ, we can get JE=EZ=JZ*1/2=2.

Q2: There is a regular pentagon in the diagram, represented by the symbol CPYJZ. Line YN is perpendicular to line CP. What is the product of the slopes of line YN and line CP?"
A2: According to the product of the slopes of two perpendicular lines being -1, we can get the slope of YN*the slope of CP=-1.

Q3: There is a regular pentagon in the diagram, represented by the symbol CPYJZ. Line CK is perpendicular to line YJ. What is the product of the slopes of line CK and line YJ?
A3: According to the product of the slopes of two perpendicular lines being -1, we can get the slope of CK*the slope of YJ=-1.

Figure 15: The Case of Direct Generation and TR-Reasoner Generation.

Table 8: Summary of Geometric Theorems and Properties

Category	**Properties**	**Criteria**
Parallel Lines	Corresponding angles equal; Alternate interior angles equal; Consecutive interior angles supplementary	Equal corresponding angles; Supplementary consecutive angles; Equal alternate angles; Parallel to same line
General Triangles	Interior angles sum to 180°	AA similarity; SSS/SAS/ASA/AAS/HL congruence
Isosceles Triangles	Equal base angles; Three-line coincidence (angle bisector, median, altitude) ;Base angles are 45° in right-isosceles case	Two equal angles ; Two equal sides
Equilateral Triangles	All angles are 60° ; Three - line coincidence	Three equal sides ; Three equal angles ; Isosceles triangle with a 60° angle
Right Triangles	Acute angles are complementary ; Side opposite 30° angle is half of the hypotenuse ; Median on the hypotenuse is half of the hypotenuse ; Pythagorean theorem: $a^2+b^2=c^2$	Contains a right angle ; HL congruence for right - triangles
Angle Bisector	Points on the perpendicular bisector are equidistant from the endpoints	A ray that divides an angle into two equal parts
Triangle Midline	Parallel to the third side and half of its length	Connects the mid-points of two sides
Parallelogram	Opposite sides are equal ; Diagonals bisect each other ; Area $= base \times height$	Both pairs of opposite sides are parallel; Diagonals bisect each other; Opposite sides are equal
Rectangle	All angles are 90° ; Diagonals are equal	A parallelogram with a right angle; A quadrilateral with three right angles
Rhombus	All sides are equal ; Diagonals are perpendicular to each other	A parallelogram with adjacent sides equal; A quadrilateral with four equal sides
Square	All sides and angles are equal; Diagonals are equal and perpendicular	Prove it is both a rectangle and a rhombus
Isosceles Trapezoid	Legs are equal; Base angles on the same base are equal	Two equal legs; Equal base angles on the same base
Trigonometric Functions	$\sin 30^\circ = \frac{1}{2}$; $\sin 45^\circ = \frac{\sqrt{2}}{2}$; $\sin 60^\circ = \frac{\sqrt{3}}{2}$; $\sin 90^\circ = 1$; $\cos 30^\circ = \frac{\sqrt{3}}{2}$; $\cos 45^\circ = \frac{\sqrt{2}}{2}$; $\cos 60^\circ = \frac{1}{2}$; $\cos 90^\circ = 0$; $\tan 30^\circ = \frac{\sqrt{3}}{3}$; $\tan 45^\circ = 1$; $\tan 60^\circ = \sqrt{3}$	/
Circle	The perpendicular bisector of a chord is perpendicular to the chord; The perpendicular bisector of a chord passes through the center	/
Central Angle	Equal central angles subtend equal chords and arcs	/
Inscribed Angle	An inscribed angle is half of the central angle subtended by the same arc; An angle subtended by a diameter is a right angle	/
Cyclic Quadrilateral	Opposite angles are supplementary	/
Tangent	A tangent is perpendicular to the radius at the point of contact; Tangents from an external point to a circle are equal in length	A line perpendicular to the radius at the endpoint on the circle is a tangent
Regular Polygon	For an equilateral triangle inscribed in a circle of radius R, side length $a = R\sqrt{3}$; For a square inscribed in a circle of radius R, side length $a = R\sqrt{2}$	/