TriGeoBench: A Multimodal Mathematical Geometry Benchmark with Figures and Tikz

Anonymous ACL submission

Abstract

Multimodal Large Language Models have shown strong capabilities across vision–language tasks, yet their ability to perform geometry-centric reasoning remains insufficiently understood. Existing multimodal mathematical benchmarks typically rely on rasterized figures and cover limited geometric knowledge areas and response formats, making it difficult to assess whether models genuinely reason over precise geometric structures or merely exploit superficial visual patterns. To address these limitations, we introduce **TriGeoBench**, a large-scale geometry-centric multimodal benchmark constructed from Chinese middle- and high-school mathematics examination problems. TriGeoBench contains approximately 7,000 question–answer pairs and 12,235 high-quality geometric figures with corresponding `TikZ` code, spanning plane geometry, solid geometry, analytic geometry, function, and vector geometry across diverse response formats. Each instance is represented as a unified {question–answer pair, image, `TikZ`} triplet, where the question–answer pair is stored in LaTeX, the image is provided as an RGB figure, and the `TikZ` code preserves fine-grained geometric topology, symbolic annotations, and relational structure. We evaluate representative open-source and closed-source MLLMs on image-to-`TikZ` generation and multimodal mathematical reasoning. Results show that current MLLMs still struggle with fine-grained geometric structure understanding and precise image–text alignment, while explicit structured representations such as `TikZ` provide useful geometric information for improving reasoning. TriGeoBench offers a comprehensive testbed for evaluating and advancing structure-aware multimodal mathematical reasoning.

1 Introduction

Multimodal Large Language Models (MLLMs) (Liu et al., 2023; Zhu et al., 2023; Hurst et al., 2024; Team et al., 2023; Li et al., 2023) have made significant progress, demonstrating strong understanding, generation, and reasoning capabilities across a wide range of vision–language tasks, including visual perception (Liu et al., 2024; Hudson and Manning, 2019), chart understanding (Masry et al., 2022; Yang et al., 2024), document comprehension (Mathew et al., 2021; Tang et al., 2025), visual question answering (Lu et al., 2022), and multimodal reasoning (Yue et al., 2025). By jointly modeling visual and textual inputs, these models can align natural-language instructions with complex visual content, enabling them to recognize objects, structures, and relations in images, understand complex scenes containing both textual and layout information, and perform cross-modal reasoning and decision-making. These advances have been largely driven by the construction of large-scale, high-quality vision–language data.

Existing multimodal benchmarks cover multiple dimensions of model capability, including visual recognition, reasoning, domain-specific understanding, hallucination detection, and multimodal instruction following (Li et al., 2024c; Caffagni et al., 2024). Among these capabilities, multimodal mathematical reasoning has attracted increasing attention, as it requires models to understand mathematical language and interpret visual figures to carry out reasoning and derive correct answers. Recently, a growing number of benchmarks for multimodal mathematical reasoning (Chen et al., 2025; Li et al., 2026; Wang et al., 2025b) have been proposed to evaluate models' integrated understanding and reasoning abilities on mathematical problems involving text, symbols, and visual figures. Unlike traditional text-only mathematical evaluation (Cobbe et al., 2021; Hendrycks et al., 2021a; Gao et al., 2024), these benchmarks examine whether models can parse mathematical language, recognize and understand diagrammatic information, establish correspondences between tex-

tual symbols and visual elements, and integrate multi-source information for mathematical derivation and problem solving. However, when applied to geometry-centric reasoning tasks, existing benchmarks still exhibit several limitations: (1) **Lack of precise structural representations of figures.** These typically rely on rasterized figures, which often fail to preserve exact topology, geometric relations, and symbolic annotations. As a result, models are evaluated largely on visual appearance rather than genuine reasoning over precise geometric structures, making it difficult to assess whether they understand the mapping between mathematical symbols and diagrammatic entities. (2) **Limited coverage of geometric knowledge areas and problem formats.** These datasets often cover only a subset of geometric knowledge and a restricted range of question types, with limited coverage of areas such as plane and solid geometry and insufficient diversity in response formats. Consequently, they provide only a coarse assessment of multimodal mathematical reasoning and may not fully reflect models' true performance on geometry reasoning tasks.

To address these limitations, we introduce **TriGeoBench**, a geometry-centric multimodal benchmark designed to evaluate both mathematical reasoning and structure-aware diagram understanding. TriGeoBench is constructed from publicly available Chinese middle- and high-school mathematics examination problems and their corresponding solution materials, with a focus on geometry-related problems that reflect authentic pedagogical distributions. TriGeoBench is constructed through large-model-assisted conversion and human verification, resulting in approximately 7,000 question–answer pairs and 12,235 high-quality geometric figures with corresponding `TikZ` code. The dataset spans five knowledge areas, including plane geometry, solid geometry, analytic geometry, function, and vector geometry, and covers multiple response formats, including single choice, fill-in-the-blank, proof and essay questions. A distinctive feature of TriGeoBench is its unified symbolic representation. Each problem statement is converted into LaTeX, while each corresponding figure is represented as editable `TikZ` code. This design preserves geometric topology, symbolic annotations, and relational structure, enabling more faithful evaluation of diagram-aware reasoning than rasterized figures alone.

We comprehensively evaluate TriGeoBench through image-to-`TikZ` generation and multimodal mathematical reasoning tasks, covering a diverse set of representative closed-source and open-source MLLMs. The results show that current MLLMs still struggle to understand fine-grained geometric structures in images. Their visual understanding largely remains at the level of coarse visual representations, making it difficult to accurately capture structural relations, symbolic annotations, and spatial constraints among geometric objects. This limitation further affects performance on multimodal mathematical reasoning. For geometry problems, relying only on text and image modalities is often insufficient to fully represent and parse key geometric information, leading to weaknesses in complex structural understanding and reasoning-chain construction. In contrast, explicitly introducing structured geometric information, such as `TikZ` code, provides models with clearer geometric objects, spatial relations, and symbolic constraints, thereby improving the accuracy of geometry problem solving. We hope our work encourages more systematic exploration of methods for enhancing image–text alignment in complex reasoning tasks. In summary, our key contributions are:

- **TriGeoBench Benchmark.** We introduce TriGeoBench, a large-scale geometry-centric multimodal benchmark constructed from Chinese middle- and high-school mathematics examination problems, covering diverse knowledge areas and response formats.

- **Unified Representation.** We provide a unified and editable representation for each instance, organized as a {question–answer pair, image, `TikZ`} triplet, where the question–answer pair is stored in LaTeX format, the image is provided as an RGB figure, and the corresponding `TikZ` code preserves fine-grained geometric structure and symbolic annotations.

- **Comprehensive Evaluation.** We systematically evaluate a broad range of state-of-the-art MLLMs on image-to-`TikZ` tasks and multimodal mathematical reasoning. Our results reveal critical limitations of current models in complex geometry reasoning, particularly when precise image–text alignment and structured figure understanding are required.

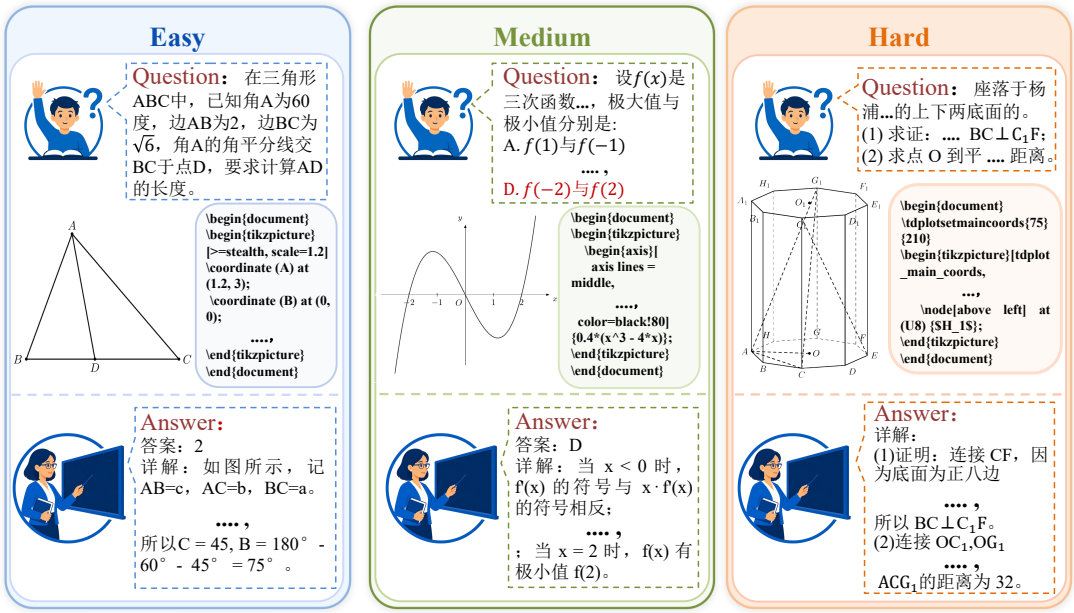


Figure 1: Illustrative examples from TriGeoBench. We divide the dataset into Easy, Medium, and Hard subsets based on problem difficulty.

2 Related Work

Multimodal mathematical and geometry benchmarks. Mathematical reasoning benchmarks have evolved from text-only problem solving to visually grounded multimodal reasoning. Early datasets such as MATH (Hendrycks et al., 2021b) focus on symbolic derivation, while MathVista (Lu et al., 2024), MATH-Vision (Wang et al., 2024), and MathVerse (Zhang et al., 2024b) extend evaluation to visual contexts. More specialized benchmarks, including VisioMath (Li et al., 2026), GeoEval (Zhang et al., 2024a), and SOLIDGEO (Wang et al., 2025b), show that MLLMs still struggle with fine-grained diagram understanding and spatial reasoning. Geometry-oriented datasets such as Geometry3K (Lu et al., 2021), GeoQA (Chen et al., 2021), GeoQA+ (Cao and Xiao, 2022), UniGeo (Chen et al., 2022), and PGPS9K (Zhang et al., 2023) further introduce formal languages, programs, proof annotations, or diagram clauses. However, most resources rely on rasterized diagrams or task-specific symbolic annotations, leaving general, editable, and compilable figure representations underexplored.

Image-to-LaTeX and graphics program synthesis. Another related direction converts visual inputs into symbolic markup or graphics programs, such as LaTeX, SVG, and TikZ. Compared with raster images, these representations are human-readable, editable, and executable, making them attractive for scientific and mathematical figures. Earlier image-to-LaTeX work mainly studies formula or handwritten expression recognition (Gervais et al., 2025), while vector-graphics benchmarks such as DeepSVG (Carlier et al., 2020) and VGBench (Zou et al., 2024) evaluate broader vector understanding. Recent TikZ-oriented works, including DeTikZify (Belouadi et al., 2024b), GeoGramBench (Luo et al., 2025), SGP-GenBench (Chen et al., 2025), OmniSVG (Yang et al., 2025), and InternSVG (Wang et al., 2025a), explore graphics program synthesis, but mainly emphasize reconstruction quality. TriGeoBench instead treats TikZ as a structured intermediate representation for diagram-grounded mathematical reasoning.

3 TriGeoBench

In this section, we present the construction pipeline of TriGeoBench, a benchmark for mathematical reasoning. TriGeoBench is a structured geometric mathematics dataset comprising question–answer pair, associated images, their corresponding TikZ code. The dataset is designed to evaluate whether multimodal models can perform geometry reasoning that relies on structured figure understanding, rather than superficial textual or visual processing.

Table 1: Comparison with existing mathematical reasoning and image-to-TikZ datasets. **Annotation:** A denotes Auto and M denotes Manual. **Knowledge:** Func: function; Vec: vector geometry; PG: Plane Geometry; SG: Solid Geometry; AG.: Analytic Geometry.

Benchmarks	Numbers	Data Type			Task Type		Knowledge					Annotation	Source
		Text	Image	TikZ	QA	Img2TikZ	Func	Vec	PG	SG	AG		
DeTikZv1 (Belouadi et al., 2024a)	117,832	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	A	TEX.SE&ArXiv
DeTikZv3 (Belouadi et al., 2025)	456,469	✗	✓	✓	✗	✓	✗	✗	✗	✗	✗	A	TEX.SE&ArXiv
SciTikZ (Lin et al., 2026)	230K	✗	✓	✓	✗	✓	✓	✓	✓	✓	✓	A M	Scientific diagrams
Geometry3K (Lu et al., 2021)	3,002	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	M	Textbooks
MATH (Hendrycks et al., 2021b)	12,500	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗	M	Competitions
UniGeo (Chen et al., 2022)	14,541	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	A M	GeoQA&Web
PGPS9K (Zhang et al., 2023)	9,022 / 4,000	✓	✓	✗	✓	✗	✗	✗	✓	✗	✗	M	Geometry3K&Web
MATH-Vision (Wang et al., 2024)	3,040	✓	✓	✗	✓	✗	✗	✗	✓	✓	✓	M	Competitions
MathVerse (Zhang et al., 2024b)	2,612	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓	M	Public Datasets
SolidGeo (Wang et al., 2025b)	3,113	✓	✓	✗	✓	✗	✗	✓	✗	✓	✗	A M	K12/Competitions
VisioMath (Li et al., 2026)	1,800 / 8,070	✓	✓	✗	✓	✗	✓	✗	✓	✓	✓	A M	CN Exams
TriGeoBench (Ours)	**7,000 / 12,235**	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	A M	**CN Exams**

3.1 Data Collection Principles

TriGeoBench is built around three principles: diversity, figure-critical reasoning, and structural representation. We collect geometry-related middle- and high-school problems covering plane geometry, solid geometry, analytic geometry, functions, and vectors, and retain only problems where figures provide necessary information for solution derivation. Each figure is paired with executable TikZ code to preserve topology, annotations, and geometric relations.

We focus on geometry problems from publicly available Chinese middle- and high-school mathematics exams and their corresponding solution materials from the past three years. These sources reflect authentic pedagogical distributions, enforce standardized mathematical notation, and exhibit naturally coupled text–figure reasoning patterns. As a result, the dataset is well suited not only for evaluating multimodal mathematical reasoning, but also for supporting auxiliary tasks such as image-to-`TikZ`. Tab. 1 presents a detailed comparison between TriGeoBench and existing benchmarks.

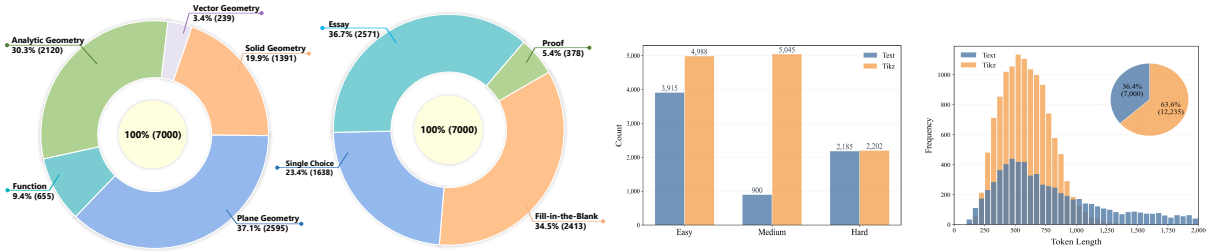
3.2 Annotation

TriGeoBench is constructed through a three-stage annotation pipeline: source collection and preprocessing, automatic conversion to unified symbolic representations, and human verification.

Collection and preprocessing. We collect publicly available Chinese mathematics examination papers from online educational platforms, school repositories, and open exam archives, covering both junior and senior secondary education. These materials typically contain question statements, answer keys, and detailed solution analyses. We extract figure-containing problems related to our target knowledge areas and segment each candidate into structured components, including the question stem, answer, solution, and associated figure(s). To reduce noise from heterogeneous layouts and document formats, we normalize mathematical notation, remove irrelevant document artifacts, and align each question with its corresponding figure region. Annotators further filter out problems without valid figures, visually degraded samples, and highly redundant templates, ensuring that the retained instances require meaningful text–figure reasoning. We also remove irrelevant metadata, personal information, and source-specific formatting artifacts during preprocessing. To mitigate data leakage and duplication, we perform near-duplicate filtering within the collected corpus and compare candidate items against commonly used public geometry and visual-math benchmarks when applicable; additional details are provided in Appendix A.

Automatic conversion to LaTeX and `TikZ`. To obtain machine-usable and editable representations, we convert each problem statement into LaTeX and each figure into `TikZ` code. Since direct manual conversion is labor-intensive, we use Gemini 3.1 Flash as the primary conversion engine, guided by carefully designed prompts detailed in Appendix D. This step provides an initial unified representation for each instance, but automatic conversion alone is insufficient for high-fidelity educational geometry figures, especially those involving nested geometric relations, solid-geometry hidden edges, function plots with labeled critical points, or precise incidence and positioning constraints.

Manual verification and correction. All automatically converted instances are manually verified against the original source materials. Annotators check the textual LaTeX for semantic correctness



(a) Distribution of knowledge point (b) Distribution of problem type (c) Distribution of TikZ and text difficulty levels (d) Distribution of token length

Figure 2: Detailed statistics of the TriGeoBench dataset. In (c) and (d), *Text* denotes the question–answer pair, while `TikZ` denotes the code corresponding to the image.

and revise the `TikZ` code to ensure compilability, visual fidelity, and structural consistency with the original figure. In particular, they correct notation errors, missing or distorted problem semantics, text–figure mismatches, and inaccurate geometric topology or label placement. When automatic `TikZ` outputs fail to preserve the intended geometric structure, annotators manually redraw the figure. This verification stage ensures that the final dataset is syntactically standardized, semantically faithful, and structurally aligned across question text, figures, and `TikZ` representations. Additional annotation details are provided in Appendix B.

3.3 Standardized Data Organization.

To support flexible downstream usage, we decouple the textual and graphical components of each instance. Specifically, the problem content is stored in LaTeX form, while each figure is stored separately in `TikZ` form. In the text, the original figure locations are replaced with explicit placeholders that specify the figure identity and local position, with each placeholder linked to the corresponding figure through the question ID and figure index. The unified representation enables flexible training and fine-tuning for a range of downstream tasks. To further support fine-grained evaluation and analysis, we categorize TriGeoBench along the following dimensions:

- **Knowledge point.** Each problem is labeled with one or more knowledge areas, including plane geometry, solid geometry, analytic geometry, function, and vector geometry. Since many problems involve multiple topics, we use both the question stem and the official solution for model-assisted labeling, and manually resolve ambiguous or overlapping cases. Fig. 2 (a) shows the distribution across knowledge areas.

- **Problem type.** We classify questions into four response formats: single choice, fill-in-the-blank, proof-based, and essay questions, with their statistical distribution shown in Fig. 2 (b). Initial labels are proposed by a large language model based on the question form and answer structure, and are subsequently verified by human annotators.

- **Figure complexity and problem difficulty.** We annotate figure complexity and problem difficulty in two stages. First, we derive initial labels using K-means clustering: problem difficulty is approximated by the length of the reference solution, while figure complexity is approximated by the length of the corresponding `TikZ` code, as shown in Fig. 2 (d). Both dimensions are divided into three levels: easy, medium, and hard. Second, these automatically generated labels are calibrated by experienced teachers and educational experts. The difficulty labels are adjusted according to the complexity of the solution strategy, the depth of mathematical inference, and the degree of text–figure interaction, while the complexity labels are refined based on the visual structure of the figure, including labeled entities, auxiliary constructions, geometric relations, and spatial dependencies. Fig. 2 (c) shows the difficulty distribution.

After large-model-assisted conversion and human verification, TriGeoBench comprises approximately 7,000 question–answer pairs paired with 12,235 high-quality geometric figures. Compared with prior benchmarks, TriGeoBench offers broader topical coverage, more diverse question formats, and native `TikZ`-based figure representations, making it well suited for both multimodal mathematical reasoning and figure generation tasks.

Table 2: Results of all models on the evaluation testset of TriGeoBench. Green highlights the best result in each column, while **bold** indicates the second-best result.

Model	Gen Acc ↑	Image Metrics				TikZ Metrics			Human Acc ↑
		SSIM $\times 10^2$↑	KID $\times 10^3$↓	CLIP $\times 10^2$↑	DINO $\times 10^2$↑	cBLEU ↑	MTE $\times 10^2$↑	TED ↓	
Open-Source Models									
Llama-3.2-90B-Vision-Instruct	80.5%	77.54	30.9	77.86	54.61	35.27	64.22	74.67	1.7%
GLM-4.5V-108B	76.4%	75.45	**4.0**	94.54	87.17	45.91	76.37	6.55	7.5%
Qwen2.5-VL-72B-Instruct	93.5%	80.66	67.8	70.47	41.60	50.39	76.52	7.26	12.8%
LLaVA-OneVision-72B	68.8%	**89.03**	96.3	81.56	66.62	22.39	50.61	56.60	4.1%
InternVL3-78B-Instruct	78.4%	74.52	6.9	92.80	82.52	39.53	70.92	9.47	8.3%
Closed-Source Models									
Qwen3.5-Plus	88.2%	86.33	6.7	**97.03**	**94.12**	**63.91**	**82.35**	**5.28**	**50.2%**
Seed-2.0-Pro	93.0%	85.70	8.2	94.70	89.66	52.44	78.44	5.83	42.9%
Gemini-3.1-Pro	**96.1%**	**86.44**	**2.2**	**97.76**	**95.92**	**63.78**	**82.73**	**3.89**	**73.7%**
GPT-5.4	**95.6%**	80.72	11.7	94.35	88.20	53.08	71.53	6.57	43.9%

4 Experiments

We evaluate TriGeoBench with two tasks: Image-to-TikZ, which tests whether models can convert geometric figures into editable and compilable code, and Multimodal Mathematical Reasoning, which tests whether textual, visual, and structured figure information improve answer accuracy.

Evaluation Models. We evaluate a broad spectrum of MLLMs on TriGeoBench, including both open-source and closed-source models. The open-source set covers eight model families: Qwen3-VL models (Bai et al., 2025) of different scales, Qwen2.5-VL (Bai Shuai et al., 2025), GLM-4.1V (Hong et al., 2025), Llama-3.2-Vision (Meta, 2024), Gemma3 (Aishwarya Kamath et al., 2025), LLaVA-OneVision (Li et al., 2024a), and InternVL3 (Zhu et al., 2025). The closed-source set includes four representative models: Gemini-3.1-Pro (Google DeepMind, 2026), GPT-5.4 (OpenAI, 2026), Doubao (ByteDance Seed Team, 2026), and Qwen3.5-Plus (Qwen Team, 2026).

Evaluation Metrics. For the Image-to-TikZ task, we use four complementary metrics. **Gen Acc** measures whether the generated TikZ code can be successfully compiled, evaluating executability. **Image Metrics** assess visual consistency between rendered predictions and ground-truth images. **TikZ Metrics** compare generated and reference code to quantify symbolic and structural similarity. **Human Acc** measures the proportion of rendered figures judged by humans as visually and mathematically faithful. For mathematical reasoning, we report answer accuracy using an LLM-based grader, counting each sub-question independently. Details are provided in Appendix C.

Evaluation Datasets. We construct the evaluation set by sampling 1,000 instances from TriGeoBench across different difficulty levels, including 500 Easy, 300 Medium, and 200 Hard problems. This stratified sampling strategy ensures that the test set covers a broad range of reasoning demands and figure complexities. The remaining instances are used as the training set for fine-tuning small-scale multimodal large language models.

4.1 Image-to-TikZ Task

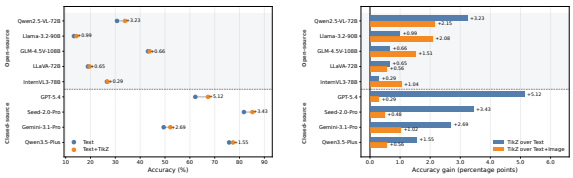
Evaluation of MLLMs on TriGeoBench. Tab. 2 summarizes the performance of representative open-source and closed-source MLLMs on the TriGeoBench evaluation set. Closed-source models substantially outperform open-source models in human-evaluated accuracy, indicating that current open-source MLLMs still struggle to understand geometric structures and generate correct TikZ programs. Among all models, Gemini-3.1-Pro achieves the best overall performance, with a generation accuracy (Gen Acc) of 96.1% and a human-evaluated accuracy (Human Acc) of 73.7%, suggesting strong compilation success and better geometric correctness. In contrast, GPT-5.4 also attains high Gen Acc of 95.6%, but its Human Acc is only 43.9%, revealing a clear gap between compilation success and geometric fidelity. These results show that automatic metrics alone cannot fully capture geometric correctness, making human evaluation necessary for assessing structural and semantic fidelity. Detailed complexity-wise results are provided in Fig. 6 in the appendix. Open-source models perform particularly poorly on Medium and Hard samples, whereas closed-source models remain stronger across all complexity levels. Nevertheless, accuracy consistently decreases as complexity increases, suggesting that compositional spatial relations and symbolic constraints remain challenging even for strong closed-source MLLMs.

Table 3: Comparison of 6 small-scale MLLMs before and after fine-tuning. **Bold** indicates the better result between the base model and its LoRA-SFT variant for each model, while green highlights the best result in each column.

Model	Stage	Gen Acc ↑	Image Metrics				TikZ Metrics			Human Acc ↑
			SSIM ×10²↑	KID ×10³↓	CLIP ×10²↑	DINO ×10²↑	cBLEU ↑	MTE ×10²↑	TED ↓	
Qwen3-VL-2B	Base model	**77.7%**	**86.95**	64.0	85.82	73.26	38.34	61.99	228.24	8.8%
	LoRA-SFT	72.7%	77.35	**2.9**	**95.53**	**88.01**	**69.39**	**85.67**	**4.26**	**12.0%**
Qwen3-VL-4B	Base model	**80.1%**	80.63	4.4	94.32	86.53	42.58	65.88	58.52	5.4%
	LoRA-SFT	71.60%	**82.49**	**2.5**	**95.69**	**89.25**	**69.71**	**85.88**	**4.11**	**13.4%**
Qwen3-VL-8B	Base model	60.10%	81.26	3.8	**95.38**	88.27	40.91	64.12	75.16	0.6%
	LoRA-SFT	**75.60%**	**82.89**	**3.8**	94.90	**88.86**	**69.85**	**85.86**	**4.27**	**27.0%**
GLM4.1VL-9B	Base model	74.6%	80.72	14.5	91.18	87.73	**64.50**	**83.65**	**8.28**	0.6%
	LoRA-SFT	**95.50%**	**81.66**	**3.7**	**95.12**	**88.36**	39.33	70.05	11.18	**4.2%**
Llama-3.2VL-11B	Base model	28.7%	**89.64**	139.5	75.16	57.10	8.45	22.75	139.14	0.3%
	LoRA-SFT	**88.00%**	82.40	**1.0**	**96.34**	**91.25**	**70.09**	**85.88**	**4.08**	**15.0%**
Gemma3-12B	Base model	**87.0%**	**87.81**	87.1	81.60	65.34	47.20	75.57	7.26	0.6%
	LoRA-SFT	72.70%	82.11	**2.1**	**95.84**	**89.81**	**62.93**	**80.86**	**5.42**	**11.3%**

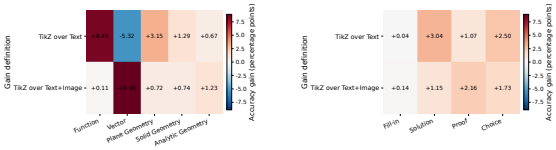
Inference and fine-tuning evaluation on small-scale MLLMs. To further validate the effectiveness and discriminative ability of TriGeoBench, we conduct LoRA fine-tuning experiments on several recent small-scale vision–language models, with results reported in Tab. 3. For fair comparison, all models use the same LoRA configuration, including learning rate, rank, and training epochs. The Gen Acc results show that LoRA fine-tuning does not consistently improve all models. For instance, Qwen3-VL-2B and Qwen3-VL-4B exhibit decreased Gen Acc after fine-tuning, indicating that supervised fine-tuning alone may not enforce the strict syntax, dependency, and compilation constraints required by TikZ. Future work may therefore benefit from compilation-aware signals, such as format rewards, compilation-success rewards, or image-consistency rewards.

For Image Metrics, fine-tuning also brings inconsistent improvements. Geometric figures often contain large blank regions, simple line segments, and sparse symbols, allowing general-purpose metrics such as CLIP and DINO to assign high scores even to imperfect outputs. Thus, image-level similarity metrics are useful but insufficient for fine-grained geometric faithfulness. In contrast, TikZ Metrics generally improve more substantially after fine-tuning, indicating that LoRA-SFT helps models learn domain-specific code patterns, command usage, and structured representations. Improvements in cBLEU, MTE, and TED suggest that fine-tuned models generate code closer to reference TikZ programs at the token, edit-distance, or tree-structure level. However, the same figure may have multiple valid TikZ implementations with different coordinate systems, drawing orders, macros, or equivalent constructions. Therefore, code-level metrics reflect



(a) Text vs. Text+TikZ. (b) TikZ gains under two settings.

Figure 3: Model-wise effect of TikZ on multimodal mathematical reasoning. Left: average accuracy under Text-only and Text+TikZ inputs. Right: gains from adding TikZ over Text-only and Text+Figure inputs. Models are grouped as open-source and closed-source MLLMs.



(a) By knowledge point. (b) By question type.

Figure 4: Fine-grained accuracy gains from adding TikZ. Heatmaps report average gains across models under two comparisons: Text+TikZ over Text-only, and Text+Figure+TikZ over Text+Figure. Positive values indicate useful complementary cues from structured TikZ.

reference-style alignment but cannot fully determine image correctness. Human evaluation further shows that all models achieve varying degrees of practical improvement after LoRA fine-tuning, suggesting that LoRA-SFT can improve adaptation to image-to-TikZ generation by enhancing geometric understanding, drawing-command organization, and structured code generation.

4.2 Multimodal Mathematical Reasoning Task

Following the widely adopted LLM-as-Judge paradigm, we employ an LLM-based evaluator

Table 4: Performance comparison of different models. Green highlights the best result in each column, while **bold** indicates the second-best result.

Model	Only Question				With Image				With TikZ				ALL			
	Easy	Medium	Hard	AVG	Easy	Medium	Hard	AVG	Easy	Medium	Hard	AVG	Easy	Medium	Hard	AVG
Open-source models																
Qwen2.5-VL-72B-Instruct	33.29	28.05	28.81	30.67	37.39	28.88	28.02	32.79	38.57	29.73	28.98	33.90	41.10	30.98	22.99	34.94
GLM-4.5V-108B	49.31	38.17	34.81	43.13	48.96	38.37	39.30	43.47	51.05	39.51	29.67	43.79	52.14	40.06	33.15	44.98
Llama-3.2-90B-Vision-Instruct	13.04	14.57	10.38	13.33	12.78	14.67	10.64	13.26	14.57	13.42	16.39	14.32	16.88	14.24	12.97	15.34
LLaVA-OneVision-72B	21.88	17.11	16.09	18.95	22.25	19.19	17.61	20.03	21.37	22.46	16.58	19.60	24.29	14.77	17.87	20.59
InternVL3-78B-Instruct	30.39	21.67	23.45	26.59	29.28	22.58	24.79	26.64	29.69	26.90	23.59	26.88	31.06	22.22	25.34	27.68
Closed-source models																
Qwen3.5-Plus	**73.87**	**80.13**	**68.57**	**75.81**	**73.67**	**77.69**	**64.44**	**74.20**	**75.65**	**80.49**	**73.63**	**77.36**	**75.37**	**76.86**	**65.57**	**74.76**
Seed-2.0-Pro	**79.10**	**84.63**	**82.70**	**81.76**	**83.29**	**84.44**	**85.56**	**84.02**	**84.88**	**86.46**	**81.87**	**85.19**	**83.88**	**85.20**	**84.66**	**84.50**
Gemini-3.1-Pro	64.18	40.84	21.13	49.45	60.32	32.33	13.30	43.00	67.89	44.30	18.69	52.14	60.91	31.97	19.10	44.02
GPT-5.4	60.82	64.23	60.92	62.20	69.67	65.64	59.57	66.84	69.59	67.27	58.66	67.32	70.15	66.61	56.99	67.13

to assess model responses, as carefully prompted LLM judges have been shown to provide reliable and scalable evaluation for open-ended generation tasks (Li et al., 2024b). To ensure consistency and reproducibility, we use a fixed grading prompt with explicit scoring rules and JSON-formatted outputs, as detailed in Appendix F. We also conduct a human-audit agreement study to validate the evaluator's reliability, with details in Appendix F.1.

Table 4 reports reasoning accuracy under four input settings: Text-only, Text+Figure, Text+TikZ, and Text+Figure+TikZ. Overall, closed-source models outperform open-source models across most settings. Doubao-Seed-2.0-Pro achieves the strongest overall performance, while GPT-5.4 shows clear gains from structured TikZ input. Among open-source models, GLM-4.5V-108B performs best on average, whereas Llama-3.2-90B and LLaVA-OneVision-72B remain much weaker, indicating that geometry-intensive reasoning with diagrams and symbolic structures remains challenging for current open-source MLLMs.

To isolate the effect of structured figure representations, Fig. 3 compares model-wise performance before and after adding TikZ. Fig. 3a shows that Text+TikZ consistently improves over Text-only across all models, while Fig. 3b shows positive gains both when TikZ is added to text-only inputs and when it is added on top of Text+Figure inputs. These results suggest that executable symbolic representations provide complementary geometric cues beyond natural-language statements and raw visual diagrams.

Fig. 4 further breaks down the gains by knowledge point and question type. As shown in Fig. 4a, TikZ yields gains across most knowledge areas, especially geometry-related categories involving explicit relations among points, lines, curves, and spatial configurations. Fig. 4b shows that solution-style and proof-style problems benefit more than short-answer cases, suggesting that structured diagrams are particularly useful for multi-step geometric reasoning.

However, the benefits of additional modalities are not always additive. Although TikZ consistently improves over Text-only, Text+Figure+TikZ does not always achieve the best absolute accuracy for every model, indicating that current MLLMs still struggle to fuse redundant visual and symbolic inputs. Thus, TriGeoBench demonstrates the value of TikZ as a structured intermediate representation while exposing the unresolved challenge of reliable text–figure–structure fusion.

5 Conclusion

We introduced **TriGeoBench**, a large-scale geometry-centric multimodal benchmark designed to evaluate structure-aware mathematical reasoning. By representing each instance as a {question–answer pair, image, TikZ} triplet, TriGeoBench provides both visual and symbolic access to geometric figures, preserving fine-grained topology, annotations, and relational structure. Through evaluations on image-to-TikZ generation and multimodal mathematical reasoning, we show that current MLLMs still struggle with precise geometric structure understanding and image–text alignment, especially for complex figures requiring symbolic and spatial reasoning. Importantly, our reasoning experiments show that adding TikZ representations improves accuracy across evaluated models, demonstrating that explicit structural figure information can provide useful complementary cues for diagram-grounded mathematical reasoning. These gains further suggest that executable symbolic representations are a promising intermediate modality for enhancing multimodal geometry problem solving.

Limitations

TriGeoBench is primarily built from Chinese middle- and high-school geometry problems, and may not fully reflect curricula or problem styles in other educational systems. In addition, multiple valid `TikZ` programs can represent the same figure, making automatic evaluation of diagram generation challenging. Future work will extend TriGeoBench to more languages, curricula, and mathematical domains, and develop more reliable structure-aware evaluation protocols. We also hope this benchmark encourages future MLLMs to incorporate explicit geometric structures, symbolic constraints, and feedback-driven training signals to improve robust multimodal mathematical reasoning.

References

Johan Ferret Aishwarya Kamath and 1 others. 2025. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*.

Shuai Bai, Yuxuan Cai, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Liu Xuejing Bai Shuai, Chen Keqin and 1 others. 2025. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*.

Jonas Belouadi, Eddy Ilg, Margret Keuper, Hideki Tanaka, Masao Utiyama, Raj Dabre, Steffen Eger, and Simone Ponzetto. 2025. TikZero: Zero-shot text-guided graphics program synthesis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 17793–17806.

Jonas Belouadi, Anne Lauscher, and Steffen Eger. 2024a. AutomaTikZ: Text-guided synthesis of scientific vector graphics with TikZ. In *The Twelfth International Conference on Learning Representations*.

Jonas Belouadi, Simone Paolo Ponzetto, and Steffen Eger. 2024b. DeTikZify: Synthesizing graphics programs for scientific figures and sketches with TikZ. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

ByteDance Seed Team. 2026. Doubao-1.5-pro / seed1.5. https://seed.bytedance.com/en/seed2.

Davide Caffagni, Federico Cocchi, Luca Barsellotti, Nicholas Moratelli, Sara Sarto, Lorenzo Baraldi, Marcella Cornia, and Rita Cucchiara. 2024. The revolution of multimodal large language models: A survey. *Findings of the association for computational linguistics: ACL 2024*, pages 13590–13618.

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, COLING 2022, Gyeongju, Republic of Korea, October 12-17, 2022*, pages 1511–1520. International Committee on Computational Linguistics.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. 2020. Deepsvg: A hierarchical generative network for vector graphics animation. In *Advances in Neural Information Processing Systems*, volume 33, pages 16351–16361. Curran Associates, Inc.

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.

Yamei Chen, Haoquan Zhang, Yangyi Huang, Zeju Qiu, Kaipeng Zhang, Yandong Wen, and Weiyang Liu. 2025. Symbolic graphics programming with large language models. *arXiv preprint arXiv:2509.05208*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, and 1 others. 2024. Omni-math: A universal olympiad level mathematic benchmark for large language models. *arXiv preprint arXiv:2410.07985*.

Philippe Gervais, Anastasiia Fadeeva, and Andrii Maksai. 2025. Mathwriting: A dataset for handwritten mathematical expression recognition. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, KDD '25, page 5459–5469, New York, NY, USA. Association for Computing Machinery.

Google DeepMind. 2026. Gemini 3.1 pro. https://deepmind.google/models/model-cards/gemini-3-1-pro/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and

Jacob Steinhardt. 2021a. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. Measuring mathematical problem solving with the math dataset. In *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, and 1 others. 2025. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*.

Drew A Hudson and Christopher D Manning. 2019. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 6700–6709.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and 1 others. 2024a. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*.

Can Li, Ying Liu, Ting Zhang, Mei Wang, and Hua Huang. 2026. Visiomath: Benchmarking figure-based mathematical reasoning in lmms. In *International Conference on Learning Representations*.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. 2024b. Llms-as-judges: A comprehensive survey on llm-based evaluation methods. *Preprint*, arXiv:2412.05579.

Jian Li, Weiheng Lu, Hao Fei, Meng Luo, Ming Dai, Min Xia, Yizhang Jin, Zhenye Gan, Ding Qi, Chaoyou Fu, and 1 others. 2024c. A survey on benchmarks of multimodal large language models. *arXiv preprint arXiv:2408.08632*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pages 19730–19742. PMLR.

Juekai Lin, Yun Zhu, Honglin Lin, Sijing Li, Tianwei Lin, Zheng Liu, Xiaoyang Wang, Wenqiao Zhang, and Lijun Wu. 2026. Scientific graphics program synthesis via dual self-consistency reinforcement learning. *Preprint*, arXiv:2604.06079.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations (ICLR)*.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. 2021. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The Joint Conference of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (ACL-IJCNLP 2021)*.

Pan Lu, Swaroop Mishra, Tanglin Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. 2022. Learn to explain: Multimodal reasoning via thought chains for science question answering. *Advances in neural information processing systems*, 35:2507–2521.

Shixian Luo, Zezhou Zhu, Yu Yuan, Yuncheng Yang, Lianlei Shan, and Yong Wu. 2025. Geogrambench: Benchmarking the geometric program reasoning in modern llms. *arXiv preprint arXiv:2505.17653*.

Ahmed Masry, Xuan Long Do, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the association for computational linguistics: ACL 2022*, pages 2263–2279.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Meta. 2024. Llama-3.2-11b-vision. https://huggingface.co/meta-llama/Llama-3.2-11B-Vision.

OpenAI. 2026. Gpt-5.4. https://openai.com/index/introducing-gpt-5-4/.

Qwen Team. 2026. Qwen3.5-plus. https://www.alibabacloud.com/help/en/model-studio/models.

Jingqun Tang, Qi Liu, Yongjie Ye, Jinghui Lu, Shu Wei, An-Lan Wang, Chunhui Lin, Hao Feng, Zhen Zhao, Yanjie Wang, and 1 others. 2025. Mtvqa: Benchmarking multilingual text-centric visual question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7748–7763.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Haomin Wang, Jinhui Yin, Qi Wei, Wenguang Zeng, Lixin Gu, Shenglong Ye, Zhangwei Gao, Yaohui Wang, Yanting Zhang, Yuanqi Li, and 1 others. 2025a. Internsvg: Towards unified svg tasks with multimodal large language models. *arXiv preprint arXiv:2510.11341*.

Ke Wang, Junting Pan, Weikang Shi, Zimu Lu, Houxing Ren, Aojun Zhou, Mingjie Zhan, and Hongsheng Li. 2024. Measuring multimodal mathematical reasoning with math-vision dataset. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Peijie Wang, Chao Yang, Zhong-Zhi Li, Fei Yin, Dekang Ran, Mi Tian, Zhilong Ji, Jinfeng Bai, and Cheng-Lin Liu. 2025b. Solidgeo: Measuring multimodal spatial math reasoning in solid geometry. *arXiv preprint arXiv:2505.21177*.

Xudong Yang, Yifan Wu, Yizhang Zhu, Nan Tang, and Yuyu Luo. 2024. Askchart: Universal chart understanding through textual enhancement. *arXiv preprint arXiv:2412.19146*.

Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. 2025. Omnisvg: A unified scalable vector graphics generation model. *arXiv preprint arXiv:2504.06263*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.

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. 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*, pages 3374–3382. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Renrui Zhang, Dongzhi Jiang, Yichi Zhang, Haokun Lin, Ziyu Guo, Pengshuo Qiu, Aojun Zhou, Pan Lu, Kai-Wei Chang, Yu Qiao, Peng Gao, and Hongsheng Li. 2024b. Mathverse: Does your multi-modal llm truly see the diagrams in visual math problems? In *Computer Vision – ECCV 2024: 18th European Conference, Milan, Italy, September 29–October 4, 2024, Proceedings, Part VIII*, page 169–186, Berlin, Heidelberg. Springer-Verlag.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. 2023. Minigpt-4: Enhancing vision-language understanding with advanced large language models. *arXiv preprint arXiv:2304.10592*.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, and 1 others. 2025. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

Bocheng Zou, Mu Cai, Jianrui Zhang, and Yong Jae Lee. 2024. VGBench: Evaluating large language models on vector graphics understanding and generation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 3647–3659, Miami, Florida, USA. Association for Computational Linguistics.

A Data Source, Licensing, and Leakage

Data sources. TriGeoBench is constructed from publicly accessible Chinese junior- and senior-secondary mathematics examination materials, including online educational platforms, school repositories, and open exam archives. The raw materials typically contain question statements, answer keys, solution analyses, and associated figures. During preprocessing, we remove irrelevant metadata, source-specific headers and footers, question indices that do not affect problem semantics, and any personal or institution-specific information when present.

Licensing and release. Since the raw examination materials are collected from heterogeneous public sources, their redistribution conditions may vary. We therefore release the processed dataset in a form that follows the applicable licensing constraints of the original sources. When necessary, we provide derived annotations, normalized LaTeX question–answer pairs, rendered figures, `TikZ` code, and metadata, while avoiding redistribution of source documents that are not explicitly

licensed for reuse. The final release will include a data statement describing source categories, preprocessing steps, and usage restrictions.

Deduplication and leakage control. To reduce duplicate or near-duplicate samples, we conduct filtering over the collected corpus using textual similarity and manual inspection. Problems based on highly redundant templates are removed during annotation. We further check candidate samples against representative public geometry and multimodal mathematical reasoning benchmarks when applicable. However, because some large multimodal models may have been trained on broad web-scale data, we cannot fully rule out prior exposure to public examination problems. We therefore report the dataset construction procedure transparently and encourage future evaluations to consider potential training-data contamination when interpreting results.

Privacy and sensitive information. The dataset is built from educational mathematics problems rather than personal records. We remove names, school-specific metadata, document headers, and other irrelevant identifying information when they appear in source documents. The retained content focuses on mathematical problem statements, solutions, figures, and structured annotations.

B Annotation Details

B.1 Collection and Preprocessing

We collect publicly available Chinese mathematics examination papers from online educational platforms, school repositories, and open exam archives. The collected materials cover both junior and senior secondary education and typically include question statements, answer keys, and detailed solution analyses. These materials provide not only the problem context but also the intended reasoning process, which is useful for verifying whether the extracted figures are necessary for solving the problem.

From the raw materials, we extract candidate problems that contain figures and are relevant to one or more target knowledge areas, including plane geometry, solid geometry, analytic geometry, functions, and vector geometry. Each candidate problem is segmented into structured components, including the question stem, answer, solution, and associated figure(s). Due to heterogeneous layouts and document styles, we perform careful normalization, including cleaning line breaks, removing question indices, deleting irrelevant headers and footers, standardizing mathematical notation, and aligning each question with its corresponding figure region.

After preprocessing, annotators screen the candidate pool by removing questions without valid figures, cases with severely degraded visual quality, and problems based on clearly redundant templates. This filtering step improves the quality and diversity of the dataset while ensuring that the retained instances require meaningful text–figure reasoning rather than relying only on textual shortcuts or superficial visual patterns.

B.2 Automatic Conversion to LaTeX and `TikZ`

To obtain unified and editable representations, each problem statement is converted into LaTeX and each corresponding figure is converted into `TikZ` code. We use Gemini 3.1 Flash Lite Preview as the primary conversion engine. The conversion prompt instructs the model to preserve mathematical notation, maintain the original problem semantics, and generate complete, compilable `TikZ` code for the figure. The full prompt is provided in Appendix D.

The automatic conversion step substantially reduces annotation cost, but it also introduces errors that require manual correction. Textual LaTeX conversion is usually usable after moderate revision, whereas `TikZ` generation is more challenging. Complex figures often require precise topology, geometric incidence, symbolic labels, hidden edges, auxiliary lines, and relative positioning. Current multimodal models frequently fail to recover these structures faithfully, especially in solid geometry diagrams, function plots with critical points, and figures involving multiple nested geometric relations.

B.3 Manual Verification and Correction

For every automatically converted instance, human annotators compare the generated LaTeX and `TikZ` code with the original source materials. The verification process focuses on four recurring categories of errors:

1. **Mathematical notation errors.** These include malformed fractions, missing operators, incorrect symbols, and wrong subscripts or superscripts.

2. **Semantic omissions or distortions.** These include missing conditions, incomplete solution steps, incorrect answer transcription, or

changes in the meaning of the original problem.

3. **Text–figure inconsistencies.** These include mismatched point labels, missing geometric objects, inconsistent references between the question and the figure, or incorrect association between a figure and its problem statement.

4. **TikZ structural errors.** These include incorrect topology, missing auxiliary lines, inaccurate relative positions, misplaced labels, missing hidden edges, and incorrect rendering of geometric or functional relations.

In many cases, the automatically generated TikZ output requires substantial rewriting rather than light editing. Annotators manually redraw figures when necessary to preserve the mathematical intent and visual reasoning structure of the original problem. The final TikZ code is required to be compilable and visually aligned with the original figure.

To support large-scale and traceable human verification, we further developed an internal annotation and review system, as illustrated in Figure 5. The system presents the original problem image, the automatically converted LaTeX text, and the generated TikZ code in a unified interface, allowing annotators to directly compare the source material with the converted representation. Annotators can edit the problem statement, solution, answer, and figure code, while immediately checking whether the corrected TikZ code compiles and visually matches the original diagram. This interface reduces context switching during annotation and helps annotators focus on mathematical fidelity rather than file-level operations.

The review system provides an additional quality-control layer. Each corrected instance is submitted for inspection, where reviewers check mathematical correctness, text–figure consistency, and rendering quality before accepting the item into the final dataset. Items with unresolved errors are returned for further correction. The system records annotation status and review decisions, enabling us to track the progress of each instance and avoid duplicated or conflicting edits. This system-based workflow makes the manual correction process more consistent, auditable, and scalable than independent offline editing.

C Evaluation Metrics

We evaluate the generated TikZ/LaTeX outputs from four complementary perspectives: renderability, visual fidelity, code-level similarity, and human-verified correctness. In addition, for the mathematical reasoning task, we report answer accuracy based on automatic grading by an LLM judge.

C.1 Render Success Rate

Render success rate, also referred to as generation accuracy (Gen Acc), measures whether the predicted TikZ/LaTeX code can be successfully compiled into a valid image. Given N test samples and N_{ok} successfully rendered samples, it is defined as:

$$\text{GenAcc} = \frac{N_{\text{ok}}}{N}. \tag{1}$$

This metric evaluates the syntactic correctness and executability of generated code. Since TikZ generation is sensitive to missing environments, invalid commands, and syntax errors, Gen Acc is an important indicator of practical usability.

C.2 Visual Similarity Metrics

To evaluate whether the rendered image from generated code matches the ground-truth figure, we use both low-level and high-level visual metrics.

SSIM. The Structural Similarity Index Measure (SSIM) measures local structural consistency between the predicted image x and the ground-truth image y:

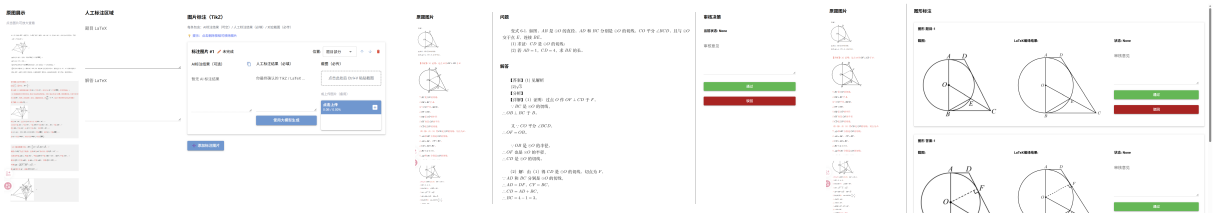
$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)}. \tag{2}$$

Here, μ, σ^2, and σ_{xy} denote image means, variances, and covariance, respectively. SSIM captures luminance, contrast, and structural similarity, making it suitable for checking whether lines, shapes, and local geometric patterns are preserved.

CLIP and DINO similarity. We further compute feature-level cosine similarity using CLIP and DINOv2 visual encoders:

$$\text{Sim}(x, y) = \frac{f_x^\top f_y}{\|f_x\|_2 \|f_y\|_2}, \tag{3}$$

where f_x and f_y denote visual embeddings of the predicted and reference images. CLIP similarity reflects high-level semantic alignment, while DINO similarity is more sensitive to visual layout and geometric structure, which is particularly useful for mathematical diagrams.



(a) Annotation interface (b) Review text (c) Review TikZ

Figure 5: Screenshots of the annotation and review system used for manual verification. The system supports side-by-side comparison between source materials and converted representations, direct correction of LaTeX and TikZ outputs, and reviewer-side quality control before final dataset inclusion.

KID. Kernel Inception Distance (KID) measures dataset-level distributional similarity between generated and reference images. It is based on the squared Maximum Mean Discrepancy:

$$\begin{aligned}\mathrm{MMD}^2(X,Y) = &\frac{1}{m^2}\sum_{i,j} k(x_i, x_j) \\ &+ \frac{1}{n^2}\sum_{i,j} k(y_i, y_j) \\ &- \frac{2}{mn}\sum_{i,j} k(x_i, y_j).\end{aligned} \tag{4}$$

where X and Y are feature sets extracted from generated and reference images. Unlike pairwise metrics such as SSIM, CLIP, and DINO, KID evaluates whether the overall generated image distribution matches the reference distribution. Lower KID indicates better distributional consistency.

C.3 Code-level Similarity Metrics

We evaluate the similarity between generated and reference TikZ code using lexical, token-level, and structural metrics.

Code BLEU. Code BLEU (cBLEU) is computed using sentence-level BLEU:

$$\mathrm{BLEU} = BP \cdot \exp\left(\sum_{n=1}^{N} w_n \log p_n\right), \tag{5}$$

where p_n is the modified n-gram precision, w_n is the weight for each n-gram order, and BP is the brevity penalty. cBLEU measures lexical and local syntactic overlap between generated and reference code.

Token F1. Token F1 measures token-level agreement after normalizing and tokenizing the generated and reference code:

$$P = \frac{TP}{TP+FP}, \quad R = \frac{TP}{TP+FN}, \quad F1 = \frac{2PR}{P+R} \tag{6}$$

Compared with BLEU, Token F1 is less sensitive to local ordering differences and is useful when semantically similar TikZ programs are written in different syntactic forms.

Tree Edit Distance. Tree Edit Distance (TED) measures structural discrepancy between generated and reference TikZ code. We construct lightweight syntax trees by extracting key graphical primitives and commands, such as `\draw`, `\node`, `\path`, `circle`, `rectangle`, `arc`, and `plot`. TED is defined as:

$$\mathrm{TED}(T_1, T_2) = \min \sum_i \mathrm{cost}(op_i), \tag{7}$$

where op_i denotes insertion, deletion, or substitution operations. Lower TED indicates stronger structural consistency between the predicted and reference programs.

C.4 Human Evaluation Accuracy

Automatic metrics cannot fully determine whether a generated diagram is mathematically faithful, especially when different TikZ programs may produce visually equivalent or semantically acceptable figures. Therefore, we additionally report human evaluation accuracy for the Image-to-TikZ task. Human annotators inspect the rendered prediction and judge whether it is visually and semantically consistent with the ground-truth figure. Given N evaluated samples and N_{pass} samples judged as acceptable, Human Acc is defined as:

$$\mathrm{HumanAcc} = \frac{N_{\text{pass}}}{N}. \tag{8}$$

A sample is considered acceptable only if the rendered figure preserves the key mathematical objects, labels, geometric relations, and visual structure required by the original problem.

C.5 Mathematical Reasoning Accuracy

For the mathematical reasoning task, we evaluate model answers using an LLM-based grader with a fixed grading prompt. The grader compares the model response against the reference solution and returns a structured judgment. Accuracy is computed as:

$$\mathrm{Acc} = \frac{N_{\mathrm{correct}}}{N_{\mathrm{total}}}, \tag{9}$$

where N_{correct} is the number of correctly answered questions and N_{total} is the total number of evaluated questions. If a problem contains multiple sub-questions, each sub-question is counted as an independent unit. Thus, a model receives credit for each correctly answered sub-question rather than only for the entire problem.

C.6 Overall Average Score

For the Image-to-`TikZ` task, we also report an aggregated average score by normalizing all metrics into a common higher-is-better scale. For example, cBLEU is divided by 100, KID and TED are transformed by reciprocal normalization, and cosine similarities are mapped to $[0, 1]$ when necessary:

$$\widetilde{\mathrm{KID}} = \frac{1}{1 + \max(\mathrm{KID}, 0)}, \quad \widetilde{\mathrm{TED}} = \frac{1}{1 + \max(\mathrm{TED}} \tag{10}$$

The final average score is:

$$\mathrm{AVG} = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \widetilde{m}, \tag{11}$$

where $\mathcal{M}$ denotes the set of normalized metrics. This score provides a compact summary of renderability, visual fidelity, code similarity, and structural consistency.

C.7 Quality Control

To ensure consistency across modalities, each finalized instance is checked from three perspectives. First, the problem text, answer, and solution must preserve the semantics of the original source. Second, the rendered figure from the `TikZ` code must match the original figure in terms of key geometric objects, labels, topology, and relative spatial relations. Third, textual references in the problem must be aligned with the corresponding visual entities in the figure. These checks ensure that each data instance forms a coherent {LaTeX question–answer pair, image, `TikZ`} triplet suitable for both image-to-`TikZ` generation and multimodal mathematical reasoning evaluation.

D Prompt for Image-to-TikZ Conversion

To convert mathematical figures into structured symbolic representations, we use a carefully designed prompt that guides the model to identify core visual elements and generate corresponding LaTeX/TikZ code. The prompt is shown below.

Prompt Template for Image-to-TikZ Conversion

Based on the uploaded problem image, extract the core visual information in the mathematical and geometric domains, including but not limited to the type of coordinate system, function expressions, graphical boundaries, coordinates of special points, auxiliary lines, filled regions, and legend descriptions. Then generate academic-style LaTeX drawing code.
The requirements are as follows:

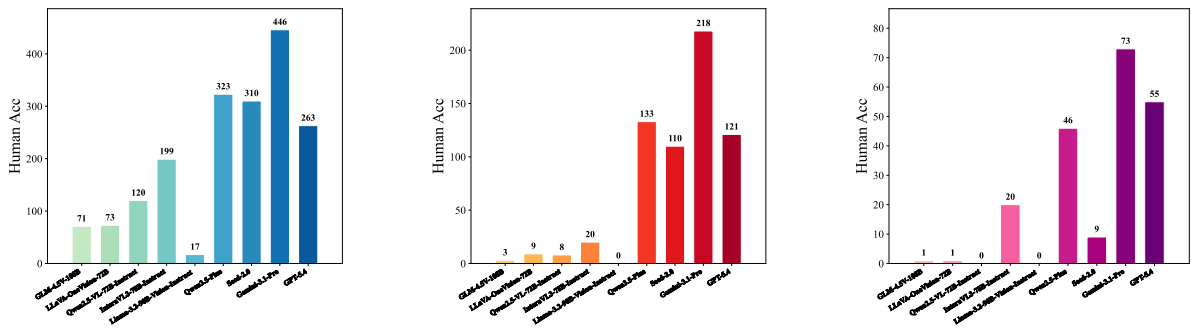
1. Prefer the TikZ package to ensure the clarity of vector graphics.
2. Strictly follow the parameter specifications in the original figure and do not omit any important details.
3. Keep the code structure clean and well organized, with detailed comments to facilitate subsequent parameter adjustment.
4. Provide a brief explanation indicating how the key parts of the code correspond to the visual elements in the original figure.

E Additional Results for Image-to-`TikZ`

Figure 6 reports human-evaluated Image-to-`TikZ` accuracy across different figure-complexity levels. The results show that model performance consistently decreases from Easy to Hard samples, indicating that complex geometric structures, dense symbolic annotations, and compositional spatial relations remain challenging for current MLLMs.

F Prompt for LLM-as-Judge Evaluation

To evaluate the correctness of model-generated answers, we adopt an LLM-as-Judge evaluation protocol. Given a problem statement, the reference solution, and the student's answer, the judge model is instructed to assign a score based on answer-level correctness. For a problem requiring multiple answers, the maximum score is defined as the number of required answers, and each correctly answered item receives one point. The prompt used for automatic evaluation is shown below.



(a) Easy (b) Medium (c) Hard

Figure 6: Human accuracy of different models on Image-to-TikZ tasks across difficulty levels.

Prompt Template for LLM-as-Judge Evaluation

You are a rigorous mathematics and geometry grading teacher. Please judge the student's solution according to the reference answer.
The grading rules are as follows:

1. First determine the total score of the problem. If the problem requires multiple answers, the maximum score is equal to the number of required answers.

2. Each correctly answered item receives 1 point, while incorrect items receive 0 points.

Please output the result in JSON format, including the following fields: `score` as an integer indicating the final score obtained by the student, `max_score` as an integer indicating the maximum score of the problem, and `comment` as a brief explanation.

Problem: {question}
Reference Answer: {solution}
Student Answer: {answer}

F.1 Reliability Analysis of LLM-based Evaluation

To assess whether the LLM-based evaluator provides stable and reliable scores, we conduct a human-audit agreement analysis on a randomly sampled subset of model responses. Human annotators independently score each response using the same scoring criteria as the LLM judge, and we compare the resulting human scores with the LLM-judge scores after normalization by the maximum score of each problem.

As shown in Figure 7a, the LLM-judge scores are highly aligned with human scores at the instance level, with most samples concentrated near the diagonal line. This indicates that the evaluator generally preserves the relative correctness of model responses across different score ranges. We further report exact agreement and within-one-point agreement across question types in Figure 7b. The LLM judge achieves strong agreement on objective formats such as choice and fill-in-the-blank questions, while slightly lower agreement is observed for proof-style and solution-style problems, where valid reasoning paths may be more diverse. Nevertheless, the high within-one-point agreement suggests that most discrepancies are minor rather than severe scoring failures.

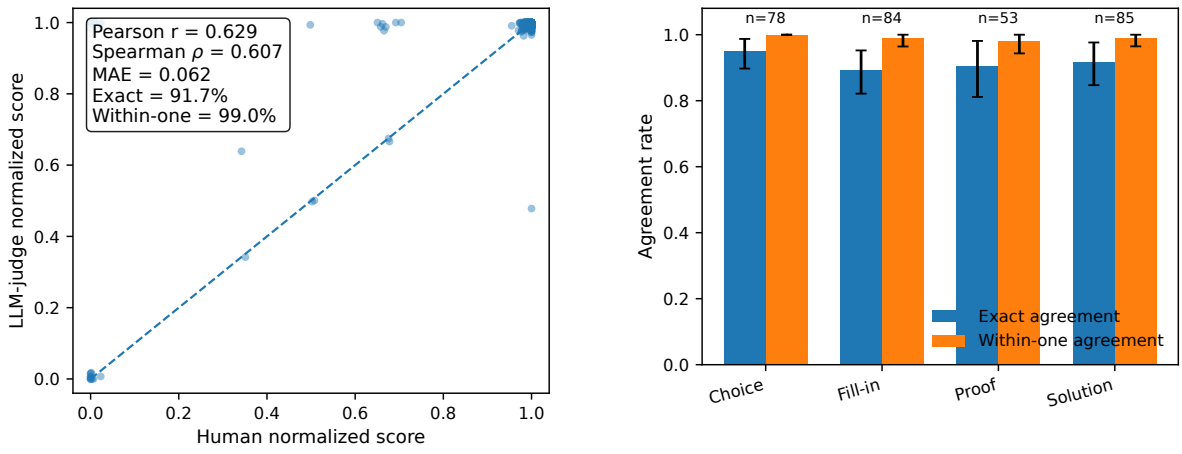
Overall, this audit supports the use of the LLM-based evaluator for large-scale evaluation in TriGeoBench, while also motivating our use of explicit JSON-formatted scoring rules and detailed grading prompts to improve consistency for open-ended mathematical responses.

G Generated Token Analysis

Figure 8 reports the average number of generated tokens across difficulty levels and input formats. The number of output tokens generally increases with problem difficulty, indicating that harder geometric problems tend to require longer reasoning chains. In contrast, introducing additional input modalities, such as figures or TikZ, does not substantially increase the number of generated tokens. This suggests that the performance changes observed in the main experiments are mainly associated with differences in available geometric information and multimodal fusion ability, rather than with a systematic increase in generation length or reasoning cost.

H Artifact License and Usage

TriGeoBench is constructed from publicly available Chinese educational materials and generated geometric figures. All released artifacts are intended solely for academic research purposes. We will



(a) Score-level agreement

(b) Agreement by question type

Figure 7: Agreement between the LLM-based evaluator and human audit results. Left: normalized score-level correlation between human scores and LLM-judge scores. Right: exact agreement and within-one-point agreement across question types, with 95% bootstrap confidence intervals.

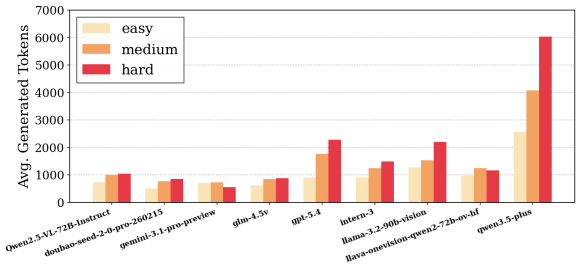
release the benchmark, annotations, and evaluation scripts under an open-source research license upon publication. The released code and benchmark will be distributed under the MIT License.

I Artifact Usage and Licensing

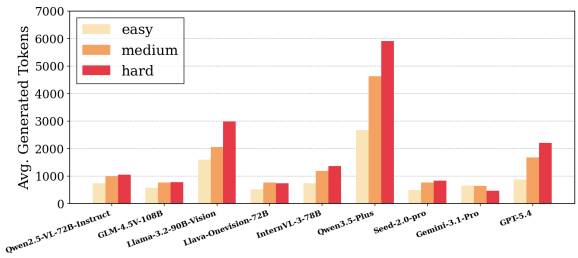
TriGeoBench is intended solely for academic research and benchmarking purposes. The benchmark is constructed from publicly available educational materials and generated geometric figures. Our use of existing resources follows their original research and educational usage conditions. Any redistributed artifacts will only contain processed benchmark data, annotations, and evaluation scripts necessary for research use. We do not authorize commercial deployment or redistribution of copyrighted source materials beyond their original access permissions.

J Training and Computational Resources

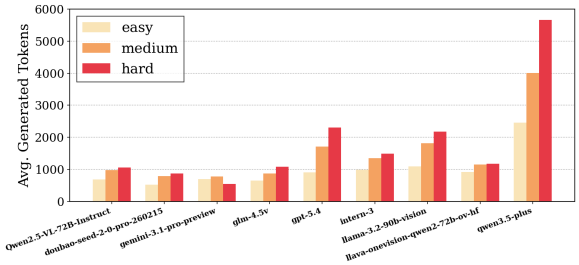
All LoRA fine-tuning experiments were conducted on two NVIDIA A100 80GB GPUs. We report model sizes in the corresponding model names (e.g., 2B, 4B, 7B, 72B, and 108B variants). Training was performed using a unified LoRA configuration across models, including the same learning rate, rank, and number of epochs for fair comparison. The total training cost for each model was approximately 2 GPU hours.



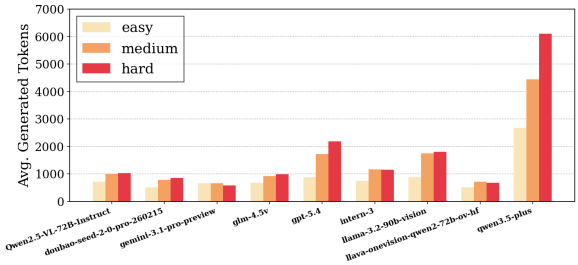
(a) Only Text



(b) Text+Figure



(c) Text+TikZ



(d) Text+Figure+TikZ

Figure 8: Average generated token count across difficulty levels and input formats.