This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

GeoTikzBridge: Advancing Multimodal Code Generation for Geometric Perception and Reasoning

Jiayin Sun, Caixia Sun, Boyu Yang*, Hailin Li, Xiao Chen, Yi Zhang
Errui Ding, Liang Li, Chao Deng, Junlan Feng

JIUTIAN Research
{sunjiayin, yangboyu, liliang}@cmjt.chinamobile.com

Abstract

Multimodal Large Language Models (MLLMs) have recently demonstrated remarkable perceptual and reasoning abilities. However, they struggle to perceive fine-grained geometric structures, constraining their ability of geometric understanding and visual reasoning. To address this, we propose GeoTikzBridge, a framework that enhances local geometric perception and visual reasoning through tikz-based code generation. Within this framework, we build two models supported by two complementary datasets. The GeoTikzBridge-Base model is trained on GeoTikz-Base dataset, the largest image-to-tikz dataset to date with 2.5M pairs (16 × larger than existing open-sourced datasets). This process is achieved via iterative data expansion and a localized geometric transformation strategy. Subsequently, GeoTikzBridge-Instruct is fine-tuned on GeoTikz-Instruct dataset which is the first instruction-augmented tikz dataset supporting visual reasoning. Extensive experimental results demonstrate that our models achieve state-of-the-art performance among open-sourced MLLMs. Furthermore, GeoTikzBridge models can serve as plug-and-play reasoning modules for any MLLM(LLM), enhancing reasoning performance in geometric problem-solving.

1. Introduction

Multimodal Large Language Models (MLLMs) have made remarkable progress in cross-modal perception and reasoning, demonstrating capabilities covering from fine-grained visual understanding to complex mathematical problem scenarios. In contrast, geometric problem remains challenging, as it demands the integration of fine-grained geometric perception and structured symbolic reasoning. This capability is fundamental for high-level cognitive and scientific tasks such as diagram-based mathematics, spatial reasoning and STEM diagram interpretation. Bridging this gap is crucial for advancing MLLMs toward deeper geometric cognition and interpretable reasoning.

Recent studies have explored solutions from the perspective of image-to-code generation and geometric reasoning. Existing Img2Code methods usually focus on web UI-to-HTML/CSS or data charts-to-Python[6, 20, 21, 35, 44, 45, 47, 52, 55]. Though a few works have touched on reconstructible code generation for graphics, they rarely involve geometric contents[5, 36, 42, 50]. For the downstream reasoning task, existing mathematical reasoning models primarily rely on text-based reasoning[4, 40, 41, 43, 57], while often overlooking the geometric relational transfer required for visual reasoning in geometric problem-solving.

Despite these advancements, MLLMs still exhibit limited local geometric perception, struggling to parse fine-grained visual details such as segment relations, angle magnitudes, and shape constraints (Fig. 1(1)). This limitation is primarily due to the scarcity of large-scale geometric image–code datasets and insufficient modeling of subtle geometric variations.

To address these challenges, we propose GeoTikzBridge, a framework designed for enhancing multimodal geometric perception via structured tikz code. To alleviate data scarcity, we first construct the GeoTikz-Base dataset, the largest image-to-tikz dataset containing 2.5M image–tikz pairs, by iteratively expanding a high-quality seed dataset. To further enhance the model perception on geometric details, we propose a localized geometric transformation strategy, including a code transformation method that leverages localized code editing to enhance geometric details in codes. Building upon this strategy, we additionally develop GeoTikz-Instruct, an auxiliary dataset that enables models to generate and interpret instruction-guided auxiliary lines.

Notably, the generated tikz codes can be seamlessly inte-

*Corresponding author.

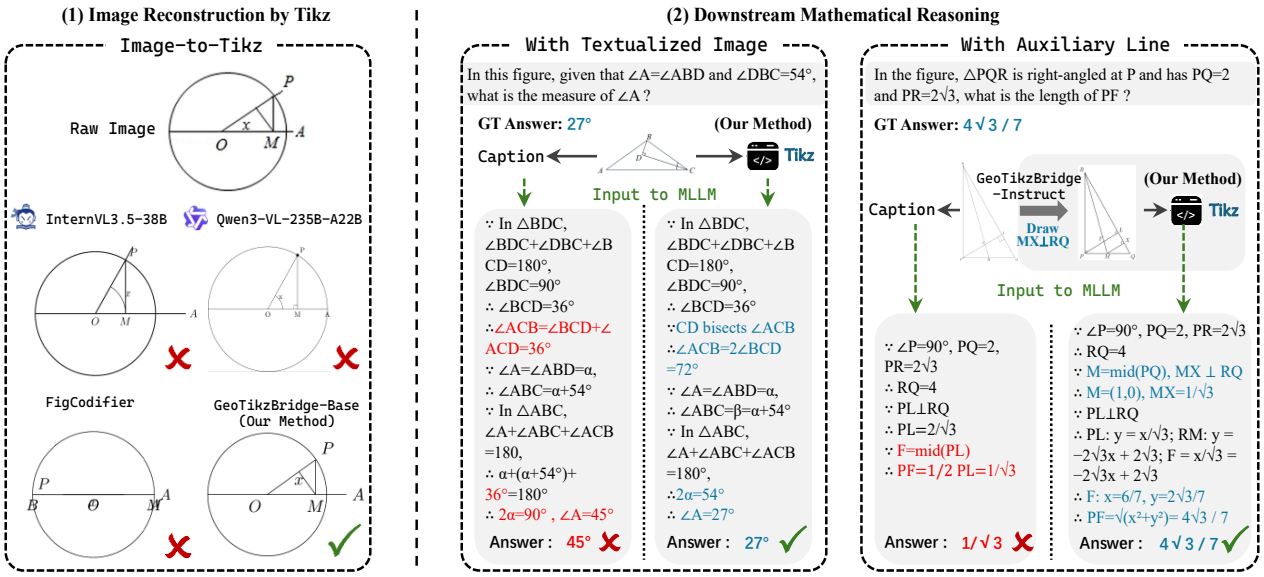


Figure 1. GeoTikzBridge demonstrates advantages in geometric perception and mathematical reasoning. (1) GeoTikzBridge-Base achieves the most accurate reconstruction of local geometric structures compared to existing approaches. (2) The generated tikz representations enhance MLLMs' mathematical and visual geometric reasoning.

grated into downstream geometric reasoning tasks, complementing MLLMs' geometric perception capabilities through structured textual encoding of geometric elements and relationships, as illustrated in Fig. 1(2).

The main contributions of this paper can be summarized as follows:

- We propose GeoTikzBridge, a novel framework that bridges geometric perception and symbolic reasoning via tikz-based code generation, thereby enabling visual reasoning in MLLMs.
- We construct two datasets: GeoTikz-Base, the largest image-to-tikz dataset to date and GeoTikz-Instruct, the first instruction-augmented tikz dataset for visual reasoning.
- We develop GeoTikzBridge models which achieve state-of-the-art performance among open-sourced MLLMs and can serve as plug-and-play reasoning modules to enhance geometric reasoning capabilities.

2. Related Works

Image-to-Code Generation. Image-to-Code Generation allows multimodal systems to translate visual content into structured symbolic forms. These forms can be executed or further edited. Recent studies have achieved notable progress across various code forms, including vector graphics[7, 36, 46, 50], HTML/CSS[6, 20, 21, 35, 45], Python[44, 47, 52, 55], and other codes[12, 32]. However, most of these efforts focus on converting webpage or chart images into executable code, leaving critical gaps in geometric code generation. Only a few works[5, 36, 42, 50] have explored converting more diverse graphic images into codes that reproduce the visual elements. Among these, tikz[23], a LaTeX-based graphic language, is superior to formats like SVG for geometric reasoning because its structured, code-based syntax inherently documents the logical steps and dependencies of geometric constructions, not just their final visual appearance. DaTikZ[5] pioneered pairing graphic images with tikz codes, enabling precise coding of elements and their spatial relationships. However, its geometric samples remain limited, focusing mainly on general graphics rather than the precise line segments, angles, and shapes needed for geometric tasks. To address this gap, we propose GeoTikz-Base, a large-scale geometric image-to-tikz dataset to alleviate the scarcity of high-quality geometric-dominant image-tikz data.

Multimodal Mathematical Reasoning. Mathematical reasoning has become an important capability of MLLMs, yet geometric problem-solving remains challenging due to the need for fine-grained visual perception. Existing mathematical reasoning works could be roughly divided into generalized and specialized methods. Generalized MLLMs[4, 40, 41, 43, 57] exhibit strong cross-modal performance, however, they are limited by scarce high-quality geometric visual-code data and poor sensitivity to local structures, which constrain their geometric-specific performance[28, 56]. Specialized methods focus on geometry-specific optimizations[8, 15, 24, 31, 38, 49, 51, 56] through improved visual-text alignment, fine-tuning on fine-grained datasets, symbolic based intermediate rea-

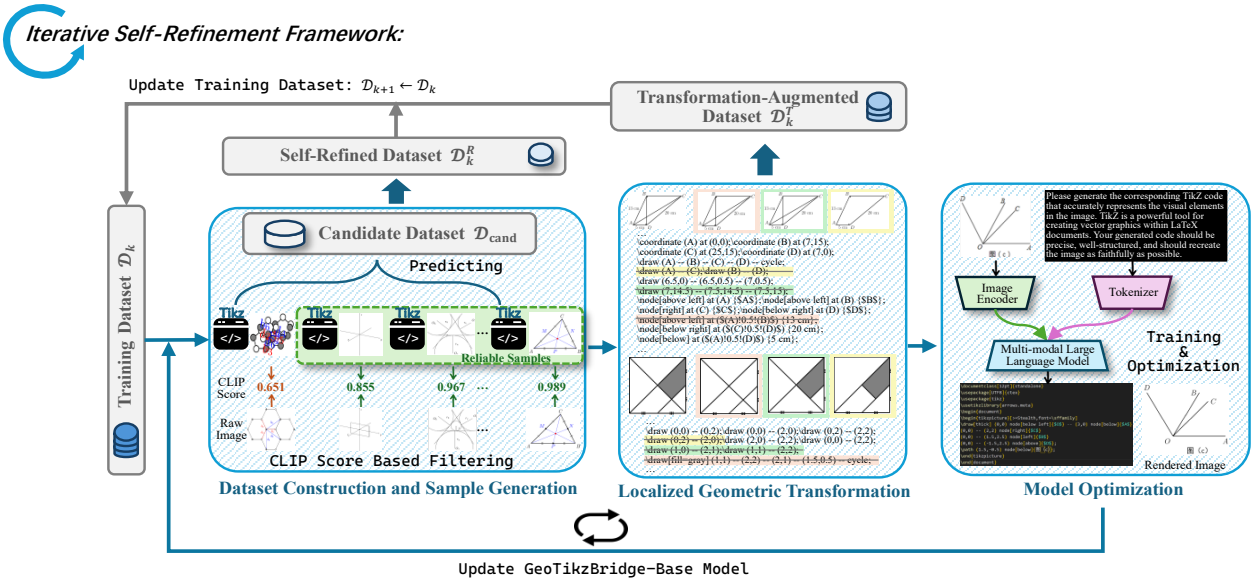


Figure 2. Framework of the proposed iterative self-refinement for GeoTikz-Base dataset and GeoTikzBridge-Base model. It iteratively conducts three steps: dataset construction and sample generation, localized geometric transformation, and model optimization.

soning, etc. However, they lack local geometric perception and auxiliary-line reasoning, limiting their effectiveness in geometric tasks. To address this, we develop GeoTikzBridge-Base and GeoTikzBridge-Instruct models based on the constructed image-to-tikz datasets, which could boost geometric-dominant reasoning by the generated tikz codes and auxiliary lines.

3. Method

3.1. Overview

We present the GeoTikzBridge framework, which advances visual reasoning by bridging the gap between geometric images and structured representations, specifically through the generation of tikz code. The GeoTikzBridge framework consists of three components: (i) an iterative self-refinement framework for constructing the image-to-tikz dataset called GeoTikz-Base and training the corresponding model called GeoTikzBridge-Base, (ii) an instruction guided image-to-tikz dataset called GeoTikz-Instruct and the corresponding trained model called GeoTikzBridge-Instruct, and (iii) a training-free visual reasoning pipeline in geometric problem-solving.

Specifically, the self-refinement framework begins with a high-quality seed dataset and iteratively expands the training data by generating new image-tikz pairs. In each iteration, tikz code is predicted by the model and unreliable samples are filtered by comparing the original image with its rendering. A novel localized code transformation strategy is introduced to enhance the model sensitivity to fine-grained geometric details. Building upon this foundation, we construct a specialized dataset and train a model for generating auxiliary lines, making it easier to visual reasoning on mathematical problems. Finally, we explore the broader applicability of our approach by proposing a training-free plug-and-play pipeline for visual reasoning.

3.2. Iterative Self-Refinement for Image-to-Tikz Dataset Construction and Model Training

The scarcity of high-quality image-tikz code pairs for model training necessitates the adoption of a self-refinement strategy. This approach iteratively generates reliable data and retrains the image-to-tikz model. The reliability of the data is assessed by comparing the rendered image with the original image.

During the iterative self-refinement process, both the training data and the model are continuously updated. For the k-th iteration ($k > 1$), the updated training dataset $\mathcal{D}_k$ is formed by combining the dataset from the previous iteration $\mathcal{D}_{k-1}$ with the two newly acquired datasets: the self-refined set $\mathcal{D}_{k-1}^R$ and the transformation-augmented set $\mathcal{D}_{k-1}^T$. Here, $\mathcal{D}_k = \{(I_k, C_k)\}$, where I_k and C_k denote an image and the corresponding tikz code in the updated training dataset at the k-th iteration, respectively. The model M_k is optimized on the training dataset $\mathcal{D}_k$ under the maximum likelihood objective $\mathcal{L}_{\text{gen}}$. The iterative process can be formalized as follows:

$$\begin{cases} \mathcal{D}_k = \mathcal{D}_{k-1} \cup \mathcal{D}_{k-1}^R \cup \mathcal{D}_{k-1}^T, \\ M_k = \underset{M \leftarrow M_{k-1}}{\arg\min}\ \mathbb{E}_{(I,C)\sim\mathcal{D}_k}\left[\mathcal{L}_{\text{gen}}(M(I), C)\right] \end{cases} \tag{1}$$

where $k = 1, 2, \ldots, K$, and K denotes the maximum iteration number, which is set to 4 in our experiments for a

balance between cost and performance. The overall process for image-to-tikz data construction and model training consists of the following three steps:

Step1. Dataset Construction and Sample Generation

As the only open-sourced and large-scale image-tikz code pairing dataset sourced from public arXiv papers and TeX repositories, DaTikZ[5] serves as the seed dataset (denoted as $\mathcal{D}_0$). The model initially trained on its training set with 145k samples is referred to as M_0.

Due to the limited variety of geometric images in DaTikZ, which cannot effectively cover diverse geometric scenarios. The goal of this step is to expand the seed dataset with reliable generated samples from a large collection of candidate geometric images.

Dataset Construction We construct a candidate dataset $\mathcal{D}_{\text{cand}}$ by collecting and organizing geometric images from nine public datasets. This candidate dataset comprises a rich collection of unannotated tikz geometric diagrams from diverse sources. The datasets utilized are as follows: MMathCoT-1M[27], GEOS[37], MAVIS-Instruct-Geometry[54], MAVIS-Instruct-Function[54], geo170k-alignment[13], geo170k-qa-tuning[13], Geometry3K[25], GeomVerse[22], UniGeo[9]. These datasets contain both synthetic and real-world images, with a primary focus on basic geometric elements, spatial relationships, and topological structures.

Sample Generation. Guided by the self-refinement paradigm, we select reliable samples from the constructed candidate dataset $\mathcal{D}_{\text{cand}}$ as potential additions to the training set by applying a CLIP score-based filtering strategy. The process begins with the generation of tikz pseudo-label set $\{\hat{C}\}$ for the geometric images, which are predicted by the model M_k. Then, the CLIP score (a normalized cosine similarity metric) s between the image $\hat{I}$ rendered from the predicted tikz code $\hat{C}$ and the original image I is computed. If s exceeds a threshold τ, the code $\hat{C}$ and the rendered image $\hat{I}$ would be considered as a reliable sample and incorporated in the self-refined set $\mathcal{D}_k^R$. $\mathcal{D}_k^R$ is defined as:

$$\mathcal{D}_k^R = \left\{ (\hat{I}, \hat{C}) \mid \hat{C} = M_k(I), \hat{I} = \mathcal{R}(\hat{C}), s(I, \hat{I}) > \tau \right\} \quad (2)$$

where $\mathcal{R}(\cdot)$ denotes the render operation and $I \in \mathcal{D}_{\text{cand}}$.

Step2. Localized Transformation Strategy

We observed that complex images often cause the model to neglect fine-grained geometric details, resulting in the omission or hallucination of critical code lines. To address this, we propose a localized transformation strategy applied to the tikz codes and their corresponding images to better capture geometric features.

The transformation strategy consists of two components: code transformation applied during the iterative dataset expansion, and image transformation applied during the model optimization (see Supplementary Material for details). Code transformation is operated on each tikz code in the self-refined set $\mathcal{D}_k^R$ at the k-th iteration, which randomly removes 1 to n lines (where n does not exceed 40% of the total tikz code lines splitted with the semicolon). Each compilable modified code $\tilde{C}$ along with its rendered image $\tilde{I}$ serve as a sample pair in the transformation-based incremental dataset $\mathcal{D}_k^T$ at the k-th iteration. Denote the transformation operation as $\mathcal{T}(\cdot)$, $\mathcal{D}_k^T$ can be defined as:

$$\mathcal{D}_k^T = \left\{ (\widetilde{I}, \widetilde{C}) \mid \widetilde{I} = \mathcal{R}(\widetilde{C}), \widetilde{C} = \mathcal{T}(\hat{C}), \hat{C} \in \mathcal{D}_k^R \right\} \quad (3)$$

Through code transformation, the model is forced to learn the structural and fine-grained semantics of the code rather than memorizing specific text sequences. It can also be regarded as injecting code noise, which enhances the generalization and robustness of the code generation. Leveraging this strategy, the code repetition prediction rate decreases by 15%. The finally expanded 2.5M training samples is called GeoTikz-Base. Fig. 3 shows that the distribution of GeoTikz-Base predominantly features Plane Geometry and Analytic Geometry as core components, while also incorporating Algebra, Calculus, and other mathematical domains, exhibiting comprehensive coverage of geometric figures.

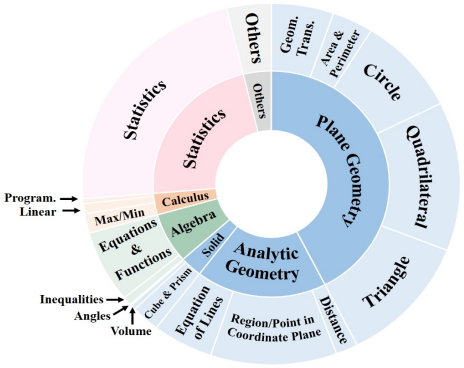


Figure 3. Distribution of the 2.5M GeoTikz-Base dataset.

Step3. Model Optimization

For the k-th iteration, the model M_k is initialized from the previous model M_{k-1} and optimized on the k-th training dataset $\mathcal{D}_k$. The training follows the causal auto-regressive modeling paradigm, where each code token in the tikz code sequence $c_i \in C$ is predicted conditioned on the image I and all previous tokens $c_{<i}$. The corresponding training objective is defined as:

$$\mathcal{L}_{\text{gen}}\ (M\,(I)\,, C) = -\sum_i \log \mathrm{P}_M\,(c_i \mid I, c_{<i}), \quad (4)$$

where P_M denotes the probability prediction by model M.

3.3. Instruction Guided Image-to-Tikz Dataset Construction and Model Training

To address the challenge of auxiliary line generation in geometric problems, we not only construct the GeoTikz-Instruct dataset $\mathcal{D}_{\text{ins}}$ based on the built $\mathcal{D}_K$, but also train the auxiliary line generation model GeoTikzBridge-Instruct M_{ins} on $\mathcal{D}_{\text{ins}}$.

Firstly, we apply the code transformation strategy $\mathcal{T}(\cdot)$ (introduced in Sec. 3.2) to each sample in $\mathcal{D}_K$, yielding the transformed tikz code $\tilde{C}_K$ and its rendered image $\tilde{I}_K$. Then, these images are fed into Qwen2.5-VL-72B to annotate the instruction Q describing what auxiliary lines have been added or what geometric elements have been modified. Finally, to ensure dataset quality, we use Doubao to filter unreliable annotations via the VLM-based filter strategy $\mathcal{F}(\cdot)$. Each retained annotated instruction Q' along with its corresponding pre-transformed tikz code C' and post-transformed image $\tilde{I}'$ (rendered from post-transformed tikz code $\tilde{C}'$) form a sample pair in the GeoTikz-Instruct dataset $\mathcal{D}_{\text{ins}}$. $\mathcal{D}_{\text{ins}}$ is defined as follows:

$$\mathcal{D}_{\text{ins}} = \left\{ (Q', \tilde{I}', C') \,\middle|\, \begin{matrix} \tilde{I}' = \mathcal{F}(\tilde{I}_K), \tilde{I}_K = \mathcal{R}(\mathcal{T}(C_K)), \\ Q' = \mathcal{F}(Q), C' = \mathcal{F}(C_K) \end{matrix} \right\} \quad (5)$$

where $(I_K, C_K) \in \mathcal{D}_K$. The built GeoTikz-Instruct dataset contains 419k training samples.

Then, we develop an instruction-guided auxiliary line generation model GeoTikzBridge-Instruct M_{ins} based on the built $\mathcal{D}_{\text{ins}}$ by performing SFT on the GeoTikzBridge-Base model. The optimization is defined as follows:

$$M_{\text{ins}} = \arg\min_{M} \mathbb{E}_{(Q', \tilde{I}', C') \sim \mathcal{D}_{\text{ins}}} \left[\mathcal{L}_{\text{gen}}(M(\tilde{I}', Q'), C') \right] \quad (6)$$

3.4. Training-Free Visual Reasoning in Geometric Problem-Solving

VLMs struggle with geometric problems due to two issues: visual feature compression causes critical detail loss, and auxiliary line requirements complicate COT reasoning. Besides, LLMs excel at reasoning but cannot directly process geometric images (e.g., via OCR), limiting their applicability. To address these issues, we design a training-free visual reasoning pipeline integrating GeoTikzBridge-Base or GeoTikzBridge-Instruct as plug-and-play modules for VLMs and LLMs.

For VLM reasoning with tikz codes, GeoTikzBridge-Base is utilized for translating the geometric problem image I_p to tikz code C_p. It is noted that for LLMs, prompts only include problem text T_p and tikz code C_p.

For VLM reasoning with auxiliary lines, Algorithm 1 shows the pipeline, where the VLM itself uses original image I_p and problem text T_p to predict the auxiliary line instruction P_{aux} (None if auxiliary lines are unnecessary determined by the VLM itself) via prompt P_{ins} (see Supplementary Material for details). If no auxiliary lines are needed, the VLM then solves the problem with T_p, I_p, and prompt for problem solving P_{ps} directly; Otherwise, GeoTikzBridge-Instruct generates the tikz code C_{aux} with auxiliary lines added based on P_{aux}, which is then rendered to I_{aux}. The VLM then utilizes T_p, I_p, P_{ps} together with I_{aux} and C_{aux} for problem solving.

Algorithm 1 Geometric Reasoning with Auxiliary Line

Require: Problem text T_p, problem image I_p, prompt P_{ins} for generating auxiliary lines, prompt P_{ps} for problem solving, and Vision-Language Model (VLM)
Ensure: Solution A to the problem

$P_{\text{aux}} \leftarrow \text{VLM}(I_p, T_p, P_{\text{ins}})$ ▷ *Query VLM for auxiliary lines*
if P_{aux} is None **then**
  $A \leftarrow \text{VLM}(I_p, T_p, P_{\text{ps}})$ ▷ *Solve without auxiliary lines*
else
  $C_{\text{aux}} \leftarrow \text{GeoTikzBridge-Instruct}(I_p, P_{\text{aux}})$
  $I_{\text{aux}} \leftarrow \mathcal{R}(C_{\text{aux}})$
  $A \leftarrow \text{VLM}(T_p, I_p, P_{\text{ps}}, I_{\text{aux}}, C_{\text{aux}})$ ▷ *Solve with auxiliary lines*
end if
return Solution A

4. Experiments

In this section, we introduce the experimental results in both image-to-tikz and downstream visual reasoning tasks.

4.1. Settings

Implementation Details. We develop two GeoTikzBridge-Base models, initialized from pretrained FigCodifier-8B and InternVL3.5-38B-Instruct respectively. For the 8B model, full-parameter SFT is adopted with learning rate of 4e-7, batch size per device of 4, and gradient accumulation steps of 4. For the 38B model, LoRA fine-tuning is applied to all QKV layers, using learning rate of 1e-4, batch size per device of 1, and gradient accumulation steps of 4. DeepSpeed (ZeRO-3 stage[34]) and flash attention[10] are employed. Both models are trained for 1 epoch on 8 NVIDIA H100 GPUs (80GB). For reproducibility, greedy decoding with temperature 0 is used to generate tikz codes during testing, and the threshold τ for selecting reliable generated code is set to 0.8. To advance geometric visual reasoning, we further introduce a model that edits tikz code to add auxiliary lines, which is fine-tuned from GeoTikzBridge-Base-8B for 3 epochs with a learning rate of 4e-7. Training costs are ∼96 GPU hours (8B) and ∼488 GPU hours (38B). The choice of $K = 4$ is analyzed in Sec. E of supplementary.

Benchmarks and Metrics. We evaluate our models on five open-sourced benchmarks (DaTikZ[5], MathVista-GPS[26], GAOKAO-MM-Math[58], R-Bench[17], RBench-V[16]) and two in-house benchmarks (EDUBenchmark and GeoTikz-Instruct), focusing on mathematical visual reasoning. Following[36, 50], we

Table 1. Performance comparison of image-to-tikz methods across four datasets. The best and second-best results are marked with bold and underlines, respectively.

Method	DaTikZ			MathVista-GPS			GAOKAO-MM-Math			EDUBenchmark		
	CLIP-S↑	FID↓	CSR↑	CLIP-S↑	FID↓	CSR↑	CLIP-S↑	FID↓	CSR↑	CLIP-S↑	FID↓	CSR↑
Open-sourced VLMs												
Qwen2.5-VL-32B[4]	0.788	77.2	80.6%	0.832	61.0	97.6%	0.808	92.2	94.4%	0.748	87.0	81.2%
Qwen2.5-VL-72B[4]	0.795	49.8	81.5%	0.858	46.9	**98.1%**	0.854	61.3	90.1%	0.781	60.5	81.8%
Qwen3-VL-30B-A3B[40]	0.672	174.4	89.1%	0.802	65.7	81.0%	0.826	107.9	95.2%	0.748	95.9	80.2%
InternVL3-78B[57]	0.747	62.7	78.8%	0.860	47.9	92.3%	0.871	55.4	93.7%	0.801	58.3	81.5%
InternVL3.5-38B[43]	0.733	50.6	94.7%	0.786	104.0	93.8%	0.834	77.2	91.5%	0.768	75.6	83.1%
GLM4.5-V-106B-A12B[41]	0.806	50.9	93.8%	0.769	122.7	88.9%	0.835	75.9	77.5%	0.809	61.4	62.7%
Existing Image-to-Tikz Model												
FigCodifier-8B[42]	0.785	45.8	86.7%	0.884	42.5	88.9%	0.849	59.1	92.3%	0.675	67.9	67.3%
GeoTikzBridge-Base-8B	0.804	43.6	88.7%	0.895	36.0	97.1%	0.890	44.9	93.0%	0.795	54.1	80.7%
GeoTikzBridge-Base-38B	**0.813**	**39.7**	**95.1%**	**0.915**	**30.6**	95.2%	**0.900**	**40.7**	**95.8%**	**0.821**	**47.6**	**83.4%**

Table 2. Performance comparison of downstream mathematical reasoning across five benchmarks. The best and second-best results are marked with bold and underlines, respectively.

Method	MathVista (GPS)	GAOKAO-MM (Math)	RBench-M (en)	RBench-V (Math)	RBench-V (Overall)
Closed-sourced VLMs					
GPT-4V[29]	0.505	0.450	-	-	-
GPT-4-turbo[1]	0.583	0.500	-	-	-
GPT-4o[19]	0.647	-	0.334	0.244	0.141
Claude3.5-Sonnet[3]	0.644	-	0.397	-	-
Gemini-1.5-Pro[39]	0.589	-	0.355	-	-
*Open-sourced VLMs (Base or Base/**ours**)*					
Qwen2.5-VL-32B[4]	0.654	0.500	0.359	0.227	0.100
Qwen2.5-VL-72B[4]	0.644	0.613	0.304	0.153	0.106
InternVL3-78B[57]	0.663	0.638	0.397	0.159	0.101
Qwen3-VL-30B-A3B[40]/+GeoTikzBridge-Base	0.697/0.745	0.550/0.588	0.314/0.320	0.218/0.236	0.110/0.117
InternVL3.5-38B[43]/+GeoTikzBridge-Base	0.688/0.718	0.600/0.635	0.353/0.369	0.216/0.232	0.118/0.120
GLM4.5-V-106B-A12B[41]/+GeoTikzBridge-Base	0.745/0.764	0.613/0.663	0.434/**0.498**	0.230/0.295	0.125/0.133
*LLMs + Tikz (**ours**)*					
Skywork-OR1-32B[18] + GeoTikzBridge-Base	0.861	0.663	0.436	0.253	0.150
Qwen3-30B-A3B[48] + GeoTikzBridge-Base	0.856	0.550	0.397	0.267	0.128
GLM-Z1-32B[14] + GeoTikzBridge-Base	0.875	0.675	0.305	0.201	0.105
DS-R1-Distill-Qwen-32B[11] + GeoTikzBridge-Base	0.841	**0.688**	0.364	0.213	0.112
GLM-4.5-Air-106B-A12B[53] + GeoTikzBridge-Base	**0.889**	**0.688**	0.359	0.250	0.138
GPT-OSS-120B[2] + GeoTikzBridge-Base	0.880	0.625	0.430	**0.301**	**0.164**

adopt CLIP Score[33] and FID[33] for image-to-tikz task, along with fine-grained scores including MSE, SSIM, and PSNR for image-to-auxiliary-line task. Besides, we introduce a stricter metric, Compilation Success Rate, which is defined as the percentage of codes that can be successfully compiled and produce valid non-blank images, since most existing methods neglect the compilation errors that may occur in the code-to-image process. Details are introduced in Supplementary Material.

4.2. Evaluation

Evaluation on Image-to-Tikz. Table 1 presents CLIP scores (CLIP-S), FID, and Compilation Success Rates (CSR) of models across four benchmarks in the GeoTikzBridge-Base task. As shown in Table 1, our proposed GeoTikzBridge-Base models achieve state-of-the-art (SOTA) performance in the GeoTikzBridge-Base task in most cases. Notably, GeoTikzBridge-Base-38B achieves the best performance across all benchmarks, as measured by the CLIP Score, significantly outperforming 6 open-sourced VLMs and the existing image-to-Tikz model.

Evaluation on LLMs/VLMs Multimodal Reasoning. We evaluate the performance of various LLMs and VLMs on multimodal reasoning tasks when provided with tikz codes generated by our GeoTikzBridge-Base-8B model. The results are summarized in Table 2, from which three main points can be observed: (i) Models provided with tikz codes from our GeoTikzBridge-Base achieve the best performance in most cases, demonstrating the effectiveness of our tikz-enhanced reasoning method; (ii) Supplementing with tikz codes significantly boosts VLMs' performance, indicating that symbolic geometric representations can compensate for VLMs' limitations in visual perception and understanding; (iii) LLMs provided with tikz codes generally outperform their VLM counterparts under the same setting. This can be attributed to catastrophic forgetting: When

Table 3. Performance comparison of instructed code generation in the downstream auxiliary line generation task. The best results are marked with bold.

Method	MSE↓	SSIM↑	PSNR↑	CLIP Score↑	FID↓	CSR↑
FigCodifier[42]	1435.9	0.135	17.6dB	0.936	9.798	88.0%
GeoTikzBridge-Base-8B	692.6	0.549	23.1dB	0.967	5.476	95.4%
GeoTikzBridge-Instruct	**211.7**	**0.844**	**89.6dB**	**0.992**	**1.158**	**96.7%**

Table 4. Performance comparison of four different visual reasoning formats on the MathVista-GPS benchmark. The baseline model is InternVL3.5-38B.

Exp ID	I	II	III	IV
Problem Text	✓	✓	✓	✓
Raw Geometric Image	✓	✓	✓	✓
Tikz Code With Aux. Line Added	×	×	✓	✓
Image With Aux. Line Added	×	✓	×	✓
Answer Accuracy	0.688	0.697	0.707	**0.736**

VLMs are trained on visual-language alignment, their inherent language capabilities often degrade.

Evaluation on Instructed Code Generation. We evaluate our GeoTikzBridge-Instruct in generating tikz code from geometric figures with specified instructions, e.g., "Draw a line through point C parallel to AB". The model takes raw geometric images and auxiliary line instructions as input, outputting tikz codes for figures with added auxiliary lines. As shown in Table 3, GeoTikzBridge-Instruct achieves strong performance across multiple metrics, demonstrating that it can capture the geometric semantic and structural details more accurately.

Evaluation on Problem Solving With Auxiliary Line. We evaluate how auxiliary lines from our GeoTikzBridge-Instruct affect geometric problem solving with different visual reasoning formats. Firstly, InternVL3.5-38B predicts auxiliary line instructions for original geometric problems, excluding those unnecessary for auxiliary lines. Then, these instructions and raw images are fed into GeoTikzBridge-Instruct to generate tikz codes with added auxiliary lines. Finally, the generated tikz codes and/or rendered images are fed into InternVL3.5-38B for mathematical reasoning. The corresponding results on the MathVista-GPS benchmark are reported in Table 4. Two points can be seen from this table: (i) Adding auxiliary lines is helpful for problem solving, no matter in the form of rendered image or tikz code; (ii) Tikz code is still more effective than image in this task.

4.3. Ablations

Geometric Representations in Reasoning Task. To investigate how different geometric representations influence the geometric reasoning performance of models, we selected two typical types of representations for comparative analysis: one is natural language captions generated by open-sourced VLMs (Qwen2.5-VL and InternVL3 series), and the other is tikz codes produced by the same VLMs and the proposed GeoTikzBridge-Base-8B model. All these representation forms need to convert original geometric images into geometric representations, which are then input into a specific VLM (Qwen3-VL-30B-A3B) together with problem texts and original images for reasoning.

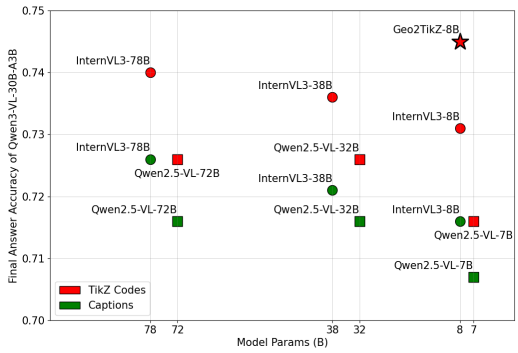


Figure 4. Ablation study of geometric representation on the MathVista-GPS benchmark. The baseline model is Qwen3-VL-30B-A3B.

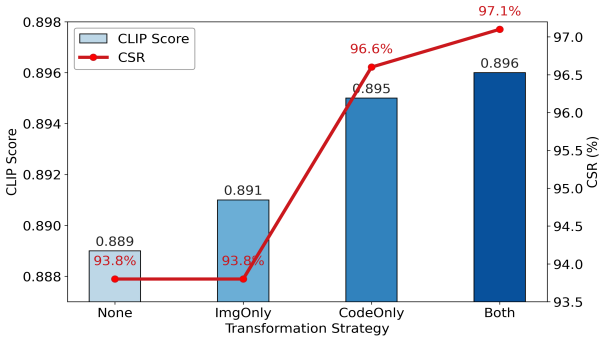


Figure 5. Ablation study of localized geometric transformation strategies for GeoTikzBridge-Base-8B, evaluated on the MathVista-GPS benchmark.

Fig. 4 presents the ablation results on the MathVista-GPS benchmark. The results indicate that, for the same model, the generated tikz codes can drive VLM to handle geometric reasoning more effectively than natural language descriptions. This fully demonstrates the advantages of tikz codes in representing geometric information. Meanwhile, the tikz codes generated by the proposed GeoTikzBridge-Base-8B model contribute more significantly to the improvement of VLMs' geometric reasoning performance compared with the tikz codes generated by models with larger parameter sizes, demonstrating the effectiveness of our method.

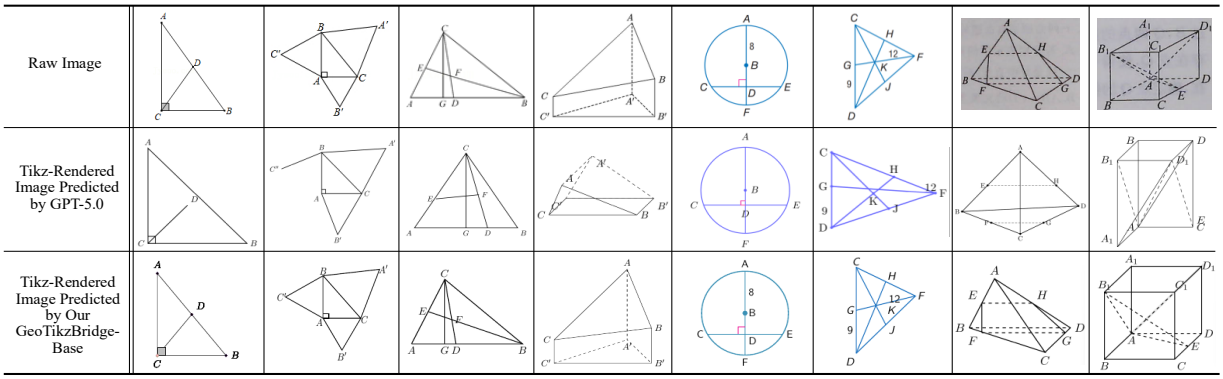


Figure 6. Visualization results of tikz codes generated by GeoTikzBridge-Base compared with GPT-5.0[30], each figure shows the input image (either geometric or non-geometric) and the corresponding image rendered from our predicted tikz code.

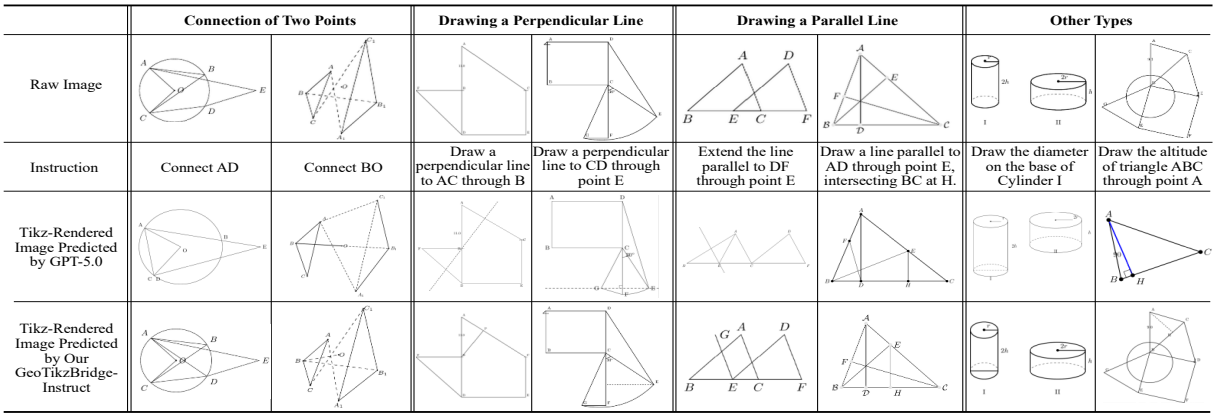


Figure 7. Visualization results of instruction-driven auxiliary line generation compared with GPT-5.0[30], each figure shows the input geometric image, the natural language instruction for auxiliary line addition, and the rendered image from the generated tikz code.

Effect of Localized Geometric Transformation. We analyze the effectiveness of the proposed localized geometric transformation strategy, including image and code transformation. Experiments are conducted using GeoTikzBridge-Base-8B on the MathVista-GPS benchmark, with results presented in Fig. 5. As shown in this figure, the code transformation strategy achieves substantial gains in both CLIP score and compilation success rate, primarily due to its capacity to not only reduce redundant code generation but also enhance geometric details via localized code editing.

4.4. Visualization

We present generated examples for the image-to-tikz and image-to-auxiliary-line tasks, with comparative analysis against GPT-5.0[30] conducted for each task. As illustrated in Fig. 6, our model more precisely captures key geometric components, including line segments, angles, closed polygons, and semantic labels. Fig. 7 demonstrates that our model outperforms GPT-5.0 across diverse geometric problem scenarios: It generates auxiliary lines more compliant with given instructions while maintaining better preservation of the original geometric figure characteristics. Additional results are provided in the Supplementary Material.

5. Conclusion

In this paper, we present GeoTikzBridge, a framework that enhances geometric perception reasoning in MLLMs and LLMs. Within this framework, we introduce two datasets. GeoTikz-Base is the largest image-to-tikz dataset to date, containing 2.5M pairs—16× larger than existing open-source datasets. We also present GeoTikz-Instruct, the first instruction-augmented tikz dataset designed for visual reasoning. Trained on these datasets, GeoTikzBridge models achieve state-of-the-art performance among open-source MLLMs. They can also serve as plug-and-play modules to enhance geometric reasoning in VLMs and LLMs without extensive retraining. The fine-tuned model further boosts MLLMs' reasoning capabilities through instructed code generation, bridging the gap between visual understanding and symbolic reasoning.

In the future, we will expand this approach to cover technical diagrams and schematics, including engineering circuit diagrams and advanced educational reasoning problems. Additionally, we aim to explore integrating GeoTikzBridge as a code-generation tool for multimodal agents in an Agentic RL manner to further enhance reasoning capabilities.

References

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023. 6

[2] Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025. 6

[3] Anthropic. Model card for claude 3.5-sonnet. https://www-cdn.anthropic.com/de8ba9b01c9ab7cbabf5c33b80b7bbc618857627/Model_Card_Claude_3.pdf, 2024. Accessed: 2024-10-14. 6

[4] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025. 1, 2, 6

[5] Jonas Belouadi, Eddy Ilg, Margret Keuper, Hideki Tanaka, Masao Utiyama, Raj Dabre, Steffen Eger, and Simone Paolo Ponzetto. TikZero: Zero-shot text-guided graphics program synthesis, 2025. 1, 2, 4, 5

[6] Tony Beltramelli. pix2code: Generating code from a graphical user interface screenshot. In *Proceedings of the ACM SIGCHI symposium on engineering interactive computing systems*, pages 1–6, 2018. 1, 2

[7] Mu Cai, Zeyi Huang, Yuheng Li, Utkarsh Ojha, Haohan Wang, and Yong Jae Lee. Leveraging large language models for scalable vector graphics-driven image understanding. *arXiv preprint arXiv:2306.06094*, 2023. 2

[8] Guizhen Chen, Weiwen Xu, Hao Zhang, Hou Pong Chan, Deli Zhao, Anh Tuan Luu, and Yu Rong. Geopqa: Bridging the visual perception gap in mllms for geometric reasoning. *arXiv preprint arXiv:2509.17437*, 2025. 2

[9] Jiaqi Chen, Tong Li, Jinghui Qin, Pan Lu, Liang Lin, Chongyu Chen, and Xiaodan Liang. Unigeo: Unifying geometry logical reasoning via reformulating mathematical expression. *arXiv preprint arXiv:2212.02746*, 2022. 4

[10] Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in neural information processing systems*, 35:16344–16359, 2022. 5

[11] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. 6

[12] Youqiang Deng, Cai Fu, and Yang Li. Codeegan: Code generation via adversarial training. In *International Conference on Dependability in Sensor, Cloud, and Big Data Systems and Applications*, pages 18–30. Springer, 2019. 2

[13] Jiahui Gao, Renjie Pi, Jipeng Zhang, Jiacheng Ye, Wanjun Zhong, Yufei Wang, Lanqing Hong, Jianhua Han, Hang Xu, Zhenguo Li, et al. G-llava: Solving geometric problem with multi-modal large language model. *arXiv preprint arXiv:2312.11370*, 2023. 4

[14] Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Diego Rojas, Guanyu Feng, Hanlin Zhao, Hanyu Lai, Hao Yu, Hongning Wang, Jiadai Sun, Jiajie Zhang, Jiale Cheng, Jiayi Gui, Jie Tang, Jing Zhang, Juanzi Li, Lei Zhao, Lindong Wu, Lucen Zhong, Mingdao Liu, Minlie Huang, Peng Zhang, Qinkai Zheng, Rui Lu, Shuaiqi Duan, Shudan Zhang, Shulin Cao, Shuxun Yang, Weng Lam Tam, Wenyi Zhao, Xiao Liu, Xiao Xia, Xiaohan Zhang, Xiaotao Gu, Xin Lv, Xinghan Liu, Xinyi Liu, Xinyue Yang, Xixuan Song, Xunkai Zhang, Yifan An, Yifan Xu, Yilin Niu, Yuantao Yang, Yueyan Li, Yushi Bai, Yuxiao Dong, Zehan Qi, Zhaoyu Wang, Zhen Yang, Zhengxiao Du, Zhenyu Hou, and Zihan Wang. Chatglm: A family of large language models from glm-130b to glm-4 all tools, 2024. 6

[15] Jarvis Guo, Tuney Zheng, Yuelin Bai, Bo Li, Yubo Wang, King Zhu, Yizhi Li, Graham Neubig, Wenhu Chen, and Xiang Yue. Mammoth-vl: Eliciting multimodal reasoning with instruction tuning at scale. *arXiv preprint arXiv:2412.05237*, 2024. 2

[16] Meng-Hao Guo, Xuanyu Chu, Qianrui Yang, Zhe-Han Mo, Yiqing Shen, Pei-lin Li, Xinjie Lin, Jinnian Zhang, Xin-Sheng Chen, Yi Zhang, et al. Rbench-v: A primary assessment for visual reasoning models with multi-modal outputs. *arXiv preprint arXiv:2505.16770*, 2025. 5

[17] Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, et al. R-bench: Graduate-level multi-disciplinary benchmarks for llm & mllm complex reasoning evaluation. *arXiv preprint arXiv:2505.02018*, 2025. 5

[18] Jujie He, Jiacai Liu, Chris Yuhao Liu, Rui Yan, Chaojie Wang, Peng Cheng, Xiaoyu Zhang, Fuxiang Zhang, Jiacheng Xu, Wei Shen, et al. Skywork open reasoner 1 technical report. *arXiv preprint arXiv:2505.22312*, 2025. 6

[19] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024. 6

[20] Yilei Jiang, Yaozhi Zheng, Yuxuan Wan, Jiaming Han, Qunzhong Wang, Michael R Lyu, and Xiangyu Yue. Screencoder: Advancing visual-to-code generation for front-end automation via modular multimodal agents. *arXiv preprint arXiv:2507.22827*, 2025. 1, 2

[21] Thisaranie Kaluarachchi and Manjusri Wickramasinghe. Webdraw: A machine learning-driven tool for automatic website prototyping. *Science of Computer Programming*, 233:103056, 2024. 1, 2

[22] Mehran Kazemi, Hamidreza Alvari, Ankit Anand, Jialin Wu, Xi Chen, and Radu Soricut. Geomverse: A systematic evaluation of large models for geometric reasoning. *arXiv preprint arXiv:2312.12241*, 2023. 4

[23] Stefan Kottwitz. Latex graphics with tikz. Webpage, 2025. Accessed: [2025-11-14]. 2

[24] Zhihao Li, Yao Du, Yang Liu, Yan Zhang, Yufang Liu, Mengdi Zhang, and Xunliang Cai. Eagle: Elevating geo-

metric reasoning through llm-empowered visual instruction tuning. *arXiv preprint arXiv:2408.11397*, 2024. 2

[25] Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. *arXiv preprint arXiv:2105.04165*, 2021. 4

[26] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. *arXiv preprint arXiv:2310.02255*, 2023. 5

[27] Ruilin Luo, Zhuofan Zheng, Yifan Wang, Yiyao Yu, Xinzhe Ni, Zicheng Lin, Jin Zeng, and Yujiu Yang. Ursa: Understanding and verifying chain-of-thought reasoning in multimodal mathematics. *arXiv preprint arXiv:2501.04686*, 2025. 4

[28] Spyridon Mouselinos, Henryk Michalewski, and Mateusz Malinowski. Beyond lines and circles: Unveiling the geometric reasoning gap in large language models. *arXiv preprint arXiv:2402.03877*, 2024. 2

[29] OpenAI. Gpt-4v(ision) system card. Technical report, OpenAI, 2023. https://cdn.openai.com/papers/GPTV_System_Card.pdf. 6

[30] OpenAI. Gpt-5 system card. Technical report, OpenAI, 2025. https://cdn.openai.com/gpt-5-system-card.pdf. 8

[31] Yicheng Pan, Zhenrong Zhang, Pengfei Hu, Jiefeng Ma, Jun Du, Jianshu Zhang, Quan Liu, Jianqing Gao, and Feng Ma. Enhancing the geometric problem-solving ability of multimodal llms via symbolic-neural integration. *arXiv preprint arXiv:2504.12773*, 2025. 2

[32] Xiongwen Pang, Yanqiang Zhou, Pengcheng Li, Weiwei Lin, Wentai Wu, and James Z Wang. A novel syntax-aware automatic graphics code generation with attention-based deep neural network. *Journal of Network and Computer Applications*, 161:102636, 2020. 2

[33] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 6

[34] Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. Zero: Memory optimizations toward training trillion parameter models. In *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–16. IEEE, 2020. 5

[35] Josselin S Roberts, Tony Lee, Chi H Wong, Michihiro Yasunaga, Yifan Mai, and Percy Liang. Image2struct: Benchmarking structure extraction for vision-language models. *Advances in Neural Information Processing Systems*, 37: 115058–115097, 2024. 1, 2

[36] Juan A Rodriguez, Abhay Puri, Shubham Agarwal, Issam H Laradji, Pau Rodriguez, Sai Rajeswar, David Vazquez, Christopher Pal, and Marco Pedersoli. Starvector: Generating scalable vector graphics code from images and text. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 16175–16186, 2025. 1, 2, 5

[37] Min Joon Seo, Hannaneh Hajishirzi, Ali Farhadi, Oren Etzioni, and Clint Malcolm. Solving geometry problems: Combining text and diagram interpretation. In *EMNLP*, page 3, 2015. 4

[38] Yanpeng Sun, Shan Zhang, Wei Tang, Aotian Chen, Piotr Koniusz, Kai Zou, Yuan Xue, and Anton van den Hengel. Mathglance: Multimodal large language models do not know where to look in mathematical diagrams. *arXiv preprint arXiv:2503.20745*, 2025. 2

[39] Gemini Team, Petko Georgiev, Ving Ian Lei, Ryan Burnell, Libin Bai, Anmol Gulati, Garrett Tanzer, Damien Vincent, Zhufeng Pan, Shibo Wang, et al. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *arXiv preprint arXiv:2403.05530*, 2024. 6

[40] Qwen Team. Qwen3 technical report, 2025. 1, 2, 6

[41] V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Jiazheng Xu, Jiale Zhu, Jiali Chen, Jing Chen, Jinhao Chen, Jinghao Lin, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Sheng Yang, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Shangqin Tu, Shengbiao Meng, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wenkai Li, Wei Jia, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyue Fan, Xuancheng Huang, Yanling Wang, Yadong Xue, Yanfeng Wang, Yanzi Wang, Yifan An, Yifan Du, Yiming Shi, Yiheng Huang, Yilin Niu, Yuan Wang, Yuanchang Yue, Yuchen Li, Yutao Zhang, Yuting Wang, Yu Wang, Yuxuan Zhang, Zhao Xue, Zhenyu Hou, Zhengxiao Du, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning, 2025. 1, 2, 6

[42] Ke Wang, Junting Pan, Linda Wei, Aojun Zhou, Weikang Shi, Zimu Lu, Han Xiao, Yunqiao Yang, Houxing Ren, Mingjie Zhan, et al. Mathcoder-vl: Bridging vision and code for enhanced multimodal mathematical reasoning. *arXiv preprint arXiv:2505.10557*, 2025. 1, 2, 6, 7

[43] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025. 1, 2, 6

[44] Chengyue Wu, Yixiao Ge, Qiushan Guo, Jiahao Wang, Zhixuan Liang, Zeyu Lu, Ying Shan, and Ping Luo. Plot2code: A comprehensive benchmark for evaluating multi-modal large language models in code generation from scientific plots. *arXiv preprint arXiv:2405.07990*, 2024. 1, 2

[45] Jialong Wu, Baixuan Li, Runnan Fang, Wenbiao Yin, Liwen Zhang, Zhengwei Tao, Dingchu Zhang, Zekun Xi, Gang Fu, Yong Jiang, et al. Webdancer: Towards autonomous information seeking agency. *arXiv preprint arXiv:2505.22648*, 2025. 1, 2

[46] Ronghuan Wu, Wanchao Su, Kede Ma, and Jing Liao. Iconshop: Text-guided vector icon synthesis with autoregressive transformers. *ACM Transactions on Graphics (TOG)*, 42(6): 1–14, 2023. 2

[47] Guangming Xing, Tawfiq Salem, and Gongbo Liang. Chartcode: A flowchart-based tool for introductory programming courses. In *Proceedings of the 56th ACM Technical Symposium on Computer Science Education V. 2*, pages 1665–1666, 2025. 1, 2

[48] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. 6

[49] Tianyun Yang, Yunwen Li, Ziniu Li, Zhihang Lin, Ruoyu Sun, and Tian Ding. Bridging formal language with chain-of-thought reasoning to geometry problem solving. *arXiv preprint arXiv:2508.09099*, 2025. 2

[50] Yiying Yang, Wei Cheng, Sijin Chen, Xianfang Zeng, Fukun Yin, Jiaxu Zhang, Liao Wang, Gang Yu, Xingjun Ma, and Yu-Gang Jiang. Omnisvg: A unified scalable vector graphics generation model. *arXiv preprint arXiv:2504.06263*, 2025. 1, 2, 5

[51] Zhen Yang, Jinhao Chen, Zhengxiao Du, Wenmeng Yu, Weihan Wang, Wenyi Hong, Zhihuan Jiang, Bin Xu, and Jie Tang. Mathglm-vision: solving mathematical problems with multi-modal large language model. *arXiv preprint arXiv:2409.13729*, 2024. 2

[52] Ryan Yen, Jian Zhao, and Daniel Vogel. Code shaping: Iterative code editing with free-form ai-interpreted sketching. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, pages 1–17, 2025. 1, 2

[53] Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025. 6

[54] Renrui Zhang, Xinyu Wei, Dongzhi Jiang, Yichi Zhang, Ziyu Guo, Chengzhuo Tong, Jiaming Liu, Aojun Zhou, Bin Wei, Shanghang Zhang, Peng Gao, and Hongsheng Li. Mavis: Mathematical visual instruction tuning, 2024. 4

[55] Xuanle Zhao, Xianzhen Luo, Qi Shi, Chi Chen, Shuo Wang, Zhiyuan Liu, and Maosong Sun. Chartcoder: Advancing multimodal large language model for chart-to-code generation. *arXiv preprint arXiv:2501.06598*, 2025. 1, 2

[56] Yurui Zhao, Xiang Wang, Jiahong Liu, Irwin King, and Zhitao Huang. Towards geometry problem solving in the large model era: A survey. *arXiv preprint arXiv:2506.02690*, 2025. 2

[57] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 1, 2, 6

[58] Yi Zong and Xipeng Qiu. Gaokao-mm: A chinese human-level benchmark for multimodal models evaluation. *arXiv preprint arXiv:2402.15745*, 2024. 5