GeoTikzBridge: Advancing Multimodal Code Generation for Geometric Perception and Reasoning

Supplementary Material

This supplementary material complements the main text: Firstly, Sec. A specifies seven benchmarks (five open-sourced, two in-house constructed) for geometric perception and reasoning, along with seven metrics: CLIP Score, FID, and Compilation Success Rate (CSR) for image-to-tikz generation; MSE, SSIM, PSNR for fine-grained auxiliary line addition; and Final Answer Accuracy (top-1 accuracy) for downstream reasoning. Secondly, Sec. B covers compilation details, including a post-processing workflow for syntax and logical errors and LaTeX compilation-rendering process. Thirdly, Sec. C provides Doubao prompt for filtering unreliable instruction annotations in constructing the GeoTikz-Instruct dataset. Then, Sec. D describes the image transformation strategy to enrich training image diversity. Next, Sec. E presents ablation studies analyzing impacts of maximum self-refinement iteration number K and the threshold τ in reliable code selection. Then, we provide failure analysis in Sec. F and analyze the model-agnostic effectiveness of our approach in Sec. G, respecitvely. Next, we supple quantitive comparison results with latest closed-source models in Sec. H. Finally, Sec. I provides supplementary visualization results, including additional examples of image-to-tikz generation, instruction-driven auxiliary line generation, as well as tikz assistant and auxiliary line assistant reasoning.

A. Benchmark & Metric Details

A.1. Benchmark Details

five open-sourced benchmarks (i.e., DaTikZ, MathVista-GPS, GAOKAO-MM-Math, RBench, and RBench-V) and two in-house constructed benchmarks named EDUBenchmark and the constructed GeoTikz-Instruct are used for evaluation, most of which are focused on geometric mathematical reasoning.

- *DaTikZ* is a large-scale image-tikz benchmark (360k+ graphics) for scientific diagram analysis, which requires LaTeX code parsing and text-graphics mapping. The GeoTikzBridge-Base results are reported on its test set, which contains 542 graphic images.
- *MathVista-GPS* focuses on visual-geometric reasoning, which is a subset of MathVista benchmark. It contains 208 samples covering 2D/3D geometry and dynamic transformations. It includes diverse visual contexts, such as natural images, geometric diagrams, abstract scenarios, synthetic scenes, and various charts and figures.
- *GAOKAO-MM-Math* is a subset of GAOKAO-MM with 80 questions that focus on mathematical reasoning and is a Gaokao-based benchmark. The GeoTikzBridge-Base results are reported on its 142 images.
- *RBench* is a graduate-level multi-disciplinary benchmark designed to evaluate the complex reasoning capabilities of MLLMs/LLMs, spanning 19 departments (e.g., mathematics, physics, biology) and over 100 subjects. It supports both English and Chinese, with 665 multimodal-specific questions integrating text and visual modals. We adopt its “Multimodal English (en)” subset (665 samples) for experiments.
- *RBench-V* is a vision-indispensable reasoning benchmark covering 4 core categories: math, physics, counting, and game. It contains 803 multimodal questions. In our work, we focus on two categories: the “Overall” category (totally 803 samples) for general reasoning, and the “Math” category (176 samples) for geometric reasoning.
- *EDUBenchmark* is an in-house constructed benchmark for multi-disciplinary reasoning. It contains 373 graphic images covering multiple disciplines (such as math, physics, chemistry, etc.) across age groups corresponding to primary school, junior high school, senior high school, university, and vocational exams, ensuring comprehensive coverage of multimodal mathematical reasoning scenarios.
- *GeoTikz-Instruct* is an in-house constructed benchmark for adding auxiliary lines in geometric figures, as mentioned before. It consists of 419k training samples and 789 testing samples.

A.2. Evaluation Metric Details

The image-to-tikz task focuses on generating valid tikz codes from geometric figures, so we adopt two metrics to measure both graphical consistency and code functionality:

- *CLIP Score*: We use the CLIP model with ViT-L/14@224px as vision encoder to quantify semantic consistency between the image that rendered from the generated tikz code and the original image. CLIP score is calculated as the cosine similarity between the two image embeddings.
- *FID*: Fréchet Inception Distance (FID) is also utilized for generated tikz quality evaluation. ViT-L/14@224px is re-employed to extract features from rendered and original images. FID computes the mean and covariance of the two feature distributions, and calculates the Fréchet distance between the corresponding multivariate Gaussians. Smaller values indicate higher distribution similarity and

better generation quality.

- *Compilation Success Rate*: We evaluate the functional correctness of the generated tikz code by compiling and rendering. The metric is defined as the percentage of code samples that successfully compile without errors and produce non-blank images.

Compared with geometric tikz code generation, the auxiliary line addition task is a more fine-grained task, as it requires precise localization of geometric primitives (e.g., vertices, line segments) and the generation of logically consistent auxiliary structures to connect known conditions, rather than merely reproducing the overall graphical layout. Hence, besides CLIP score and compilation success rate, we also adopt three fine-grained metrics to assess the similarity between each rendered image from the generated tikz and the corresponding ground-truth image in the test set:

- *MSE*: Mean Square Error (MSE) is a fundamental metric for quantifying pixel-wise differences between images rendered from generated tikz codes and ground-truth images. It is calculated as the average of squared errors between corresponding pixels of the two images. Lower MSE values indicate smaller pixel-level discrepancies and higher rendering fidelity.
- *SSIM*: Structural Similarity Index (SSIM) aligns with human visual perception by incorporating three key attributes of images: luminance, contrast, and local structure. It computes similarity between images rendered from generated tikz codes and ground-truth images through sliding window-based regional pixel analysis, with value ranging from 0 to 1. Values closer to 1 indicate greater structural and visual consistency between the two images.
- *PSNR*: Peak Signal to Noise Ratio (PSNR) is derived from MSE and measured in decibels (dB) to evaluate image quality by quantifying the ratio between signal peak amplitude and noise. It compares images rendered from generated tikz codes against ground-truth images, with higher PSNR values corresponding to less distortion.

In downstream mathematical reasoning task, *Final Answer Accuracy* is adopted, which is defined as the correctness of the generated final solution.

B. Compilation Details

Code Post-processing. Given that tikz codes predicted by the model are prone to format non-compliance, leading to compilation failures, a code post-processing strategy is implemented to fix common compilation errors. This integrated workflow combines syntax correction and logical optimization, mainly addressing unclosed environments and redundancy issues which not only enhances code compileability, but also improves the compactness and readability of the codes. Notably, the reported experimental results have employed the same post-processing operations for all models, including our proposed models and the compared models. The post-processing procedure is as follows:

1) *Package Deduplication:* For duplicate '\usepackage' declarations, retain one instance per unique package while preserving all non-redundant statements.
2) *Block-Aware Deduplication:* Parse and identify nested environments (e.g., 'tikzpicture', 'scope') with priority to innermost environments as well as '\foreach' code blocks. Within each identified block, remove redundant identical non-environment start/end lines, avoiding redundant code execution while maintaining structural integrity.
3) *Truncated Line Removal:* Detect syntactically incomplete lines, defined as lines with unclosed LaTeX commands (excluding environment directives) lacking terminating semicolons, or imbalanced curly braces/square brackets, and discard only the last non-symbol-only line (if incomplete) before the main environment end '\end{tikzpicture}'.
4) *Environment Balance:* Adopt stack-based nesting tracking to manage nested environments. Append missing environment end directives in reverse order of their opening (e.g., close 'scope' before 'tikzpicture' for nested structures).
5) *Document Structure Optimization:* Ensure the 'document' environment is fully defined with exactly one pair of start ('\begin{document}') and end ('\end{document}') directives, appending '\end{document}' at the end (after removing trailing blank lines) if absent. Relocate any graphical environments (e.g., 'tikzpicture') erroneously placed outside the 'document' scope into it for syntactic validity.
6) *Logical Refinement:* Extract redundant graphical commands (e.g., '\draw', '\node') from '\foreach' blocks that do not reference the loop variable (e.g., '\x', '\y'), relocating them outside the loop to avoid redundant execution. Besides, move commands misplaced outside their semantically required environments (e.g., '\draw' outside 'tikzpicture') to the correct scope.

After post-processing, the Compilation Success Rate (CSR) metrics of generated codes reflect the intrinsic code generation quality and syntax adaptation potential of different methods, highlighting inherent model capability in terms of code syntax correctness, structural rationality, and format compliance, eliminating interference from non-essential factors such as sequence disorder and redundant repetition.

Compiling and Rendering. Each post-processed tikz code snippet is embedded within a LaTeX code framework to enable standardized rendering of its defined geometric content. We also include the necessary language packages at the beginning of the LaTeX file to ensure that the text content in the corresponding languages can be displayed successfully in the compiled PDF. Besides, we set the LaTeX file type to 'standalone' to facilitate subsequent compiling and rendering into images. Then, the latex codes are compiled into PDFs, which are subsequently converted to Portable Network Graphics (PNG) images via the pdf2Image library.

C. Doubao Prompt for Filtering Unreliable Instruction Annotations

Doubao Prompt for Filtering Unreliable Instruction Annotations

Prompt:

<Image><Transformed_Image><tikz><Original TikZ Code></tikz><transformed_tikz><Transformed TikZ Code></transformed_tikz>

You are a geometric instruction discriminator for auxiliary line generation tasks. Your task is to assess the quality of an instruction Q by aligning it with the geometric transformation between a pre-transformed tikz code C_K (original) and a post-transformed tikz code $\tilde{C}_K$, along with their corresponding rendered images I_K and $\tilde{I}_K$. You should follow three rules for checking if the provided instruction is correct. If all of the rules are followed by the instruction, please output 'True'; Otherwise, please output 'False'. Please don't output other content. The checking rules are as follows:

1. Consistency of Code Difference: Does instruction Q accurately reference the modified primitives (points, lines, angles, etc.) in post-transformed tikz code $\tilde{C}_K$ (compared with pre-transformed code C_K)?

2. Consistency of Image Difference: Does instruction Q accurately reference the modified primitives (points, lines, angles, etc.) in post-transformed image $\tilde{I}_K$ (compared with pre-transformed image I_K)?

3. Professionalism of Geometric Terminology: Is instruction Q unambiguous and uses correct geometric terms?

Figure S1. Prompt input to Doubao for filtering unreliable instruction annotations in the GeoTikz-Instruct dataset construction.

To filter out low-quality GeoTikz-Instruct instructions annotated by Qwen2.5-VL-72B while ensuring geometric validity, we adopt a prompt-guided discriminative strategy using Doubao. Specifically, a targeted prompt is first designed for guiding Doubao to score the raw annotated instructions based on geometric consistency, as shown in Fig. S1. Then, high-quality outputs are selected to form the final instruction-image-tikz triples.

D. Image Transformation Details

As a key component of the localized geometric transformation strategy (introduced in Sec. 3.2 of the main text), image transformation complements code transformation to address the model's insensitivity to fine-grained geometric details. While code transformation enhances structural semantics via localized editing, fine-grained visual variations (e.g., slight rotations, scale adjustments) in geometric scenes still need handling. The core goal of image transformation is to enrich the visual diversity of training samples in images while preserving invariant geometric relationships in codes, forcing the model to focus on essential geometric properties rather than superficial visual features. The image transformation procedure is as follows:

1) *Canvas Expansion*: A 100-pixel replication border is appended around the original image to form an expanded canvas, which reserves sufficient edge space to prevent content truncation induced by subsequent perspective transformation.
2) *Perspective Transformation*: Source points are defined as the four corners of the original image on the expanded canvas, with target points generated through random offsets (ranging from -50 to 50 pixels, equivalent to half the border size). A perspective transformation matrix is estimated based on source-target correspondences and applied to the canvas to introduce subtle subtle distortion that simulates natural viewing angle variations.
3) *Optimized Cropping*: The transformed image is converted to gray-scale, binarized, and processed using morphological closing with a 15×15 kernel to fill holes in the foreground region. The maximum external contour is selected to compute the minimum enclosing rectangle, and the cropping boundary is expanded by 5 pixels. To preserve key content, the cropped region is constrained to be at least 50% of the original image size. Subsequently, the cropped image is resized to the original dimensions using area interpolation to ensure sharp down-sampling.
4) *Illumination Enhancement*: Random non-uniform illumination gradients are introduced, including horizontal (with brightness linearly transitioning from 0.4 to 0.6 on the left to 0.8 to 1.0 on the right), vertical (with brightness linearly transitioning from 0.4 to 0.6 at the top to 0.8 to 1.0 at the bottom), and central (with brightness decaying from the center to the edges). The central gradient followsgradient$(x, y) = \text{base_brightness} + \text{brightness_range} \times (1 - \frac{\text{dist}}{\text{max_dist}})$, where base_brightness ranges from 0.6 to 0.7, brightness_range ranges from 0.2 to 0.3, dist denotes the distance from a pixel to the center, and max_dist is the maximum diagonal distance from the center. The gradient matrix is multiplied with the image, and the resulting pixel values are clipped to the range [0, 255].
5) *Blur Augmentation*: Gaussian blur is applied with a 50% probability, using a radius ranging from 0.5 to 1.2 to simulate mild lens defocus.
6) *Color Adjustment*: Contrast (with a factor ranging from 0.7 to 0.8) and saturation (with a factor ranging from 0.7 to 0.85) are moderately reduced to mimic natural color attenuation.

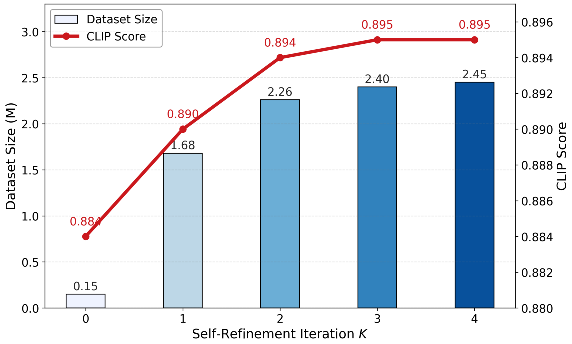


Figure S2. Ablation study of different values of self-refinement K for GeoTikzBridge-Base-8B, evaluated on the MathVista-GPS benchmark.

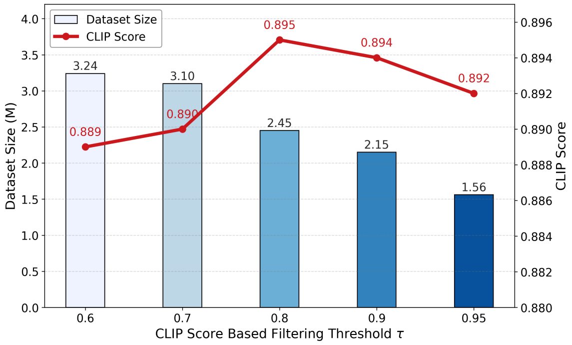


Figure S3. Ablation study of different values of CLIP score based filtering threshold τ for GeoTikzBridge-Base-8B, evaluated on the MathVista-GPS benchmark.

7) *Lens Distortion*: Subtle radial distortion is introduced, with distortion coefficients k_1 in the range [-0.03, 0.03] and k_2 in [-0.01, 0.01]. Image coordinates are normalized to the range [-1, 1], and the radial distance is computed as $r = x_{\text{norm}}^2 + y_{\text{norm}}^2$. Distorted coordinates are derived as $x' = x_{\text{norm}} \times (1 + k_1 r + k_2 r^2)$ and $y' = y_{\text{norm}} \times (1 + k_1 r + k_2 r^2)$, which are then remapped to the original image size.
8) *Rotation Augmentation*: Minor in-plane rotation (within the range $[-1.5°, 1.5°]$) is applied around the image center, with a replication border mode used to handle edge pixels.

E. Supplementary Ablations

Self-Refinement Iteration K. We evaluate the effect of iterative self-refinement on the image-to-tikz performance of GeoTikzBridge-Base-8B and the scalability of its training dataset, Fig. S2 presents the corresponding ablation study. Evaluated on the MathVista-GPS benchmark, the figure uses bars to represent training dataset size and lines to denote CLIP score.

$K = 0$ corresponds to the initial state where the pre-trained FigCodifier (denoted as M_0) utilizes the 145k-sample DaTikZ seed dataset. This model achieves a CLIP score of 0.884 on geometric perception, as it has not been trained on geometric-specific dataset. At $K = 1$, M_0 is employed to predict tikz codes for candidate geometric images sourced from nine public datasets. The limited ability of M_0 to parse fine-grained details (e.g., segment relations, angles) results in relatively few reliable samples that are filtered via the CLIP threshold of 0.8 (detailed in Sec. 3.2 of the main text). With this expanded dataset, the model trained on it (denoted as M_1) outperforms M_0 in CLIP score, as the added geometric-specific training samples enhance basic geometric perception of the model.

For $K = 2$, the improved geometric image-to-tikz ability of M_1 increases the number of reliable samples (including self-refined and transformation-augmented ones), driving a significant CLIP score gain for the subsequent model (denoted as M_2). At $K = 3$, dataset growth slows: Despite sampling with replacement from the candidate geometric image pool, the model already achieves relatively mature geometric image-to-tikz capability, so the gain from newly added reliable samples in driving performance improvement diminishes. When $K = 4$, dataset expansion diminishes further and the CLIP score of the model nearly converges. Hence, we set $K = 4$ as the maximum iteration number for the experiments, balancing training cost and performance.

CLIP Score Based Filtering Threshold τ. We evaluate the effect of the CLIP score-based filtering threshold τ on both dataset scalability and model performance, Fig. S3 presents the corresponding ablation study evaluated on the MathVista-GPS benchmark.

For $\tau = 0.6$, the dataset size reaches 3.24M (the largest across all thresholds), while the CLIP score is 0.889. This indicates loose filtering introduces massive low-quality samples, limiting performance. At $\tau = 0.7$, the dataset size decreases slightly to 3.10M, and the CLIP score increases to 0.890. When $\tau = 0.8$, the dataset size decreases to 2.45M, yet the CLIP score peaks at 0.895, achieving the optimal trade-off between dataset scale and sample quality, where sufficient samples enable effective training, and high-quality samples drive the best performance. For $\tau = 0.9$ and $\tau = 0.95$, the dataset size further reduces to 2.15M and 1.56M, respectively, and the CLIP score decreases to 0.894 and 0.892, as insufficient training dataset restricts performance. Nevertheless, the CLIP scores achieved with the 2.15M and 1.56M datasets (at $\tau = 0.9$ and $\tau = 0.95$) remain higher than those obtained with the larger 3.24M and 3.10M datasets (at $\tau = 0.6$ and $\tau = 0.7$), indicating that the quality of training data is more critical than its quan-

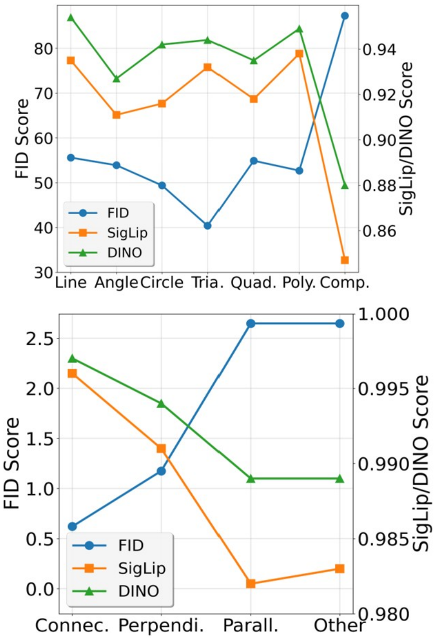


Figure S4. FID, SigLip/DINO scores of Base-8B model (up) and Instruct-8B model (bottom) across different geometric structures.

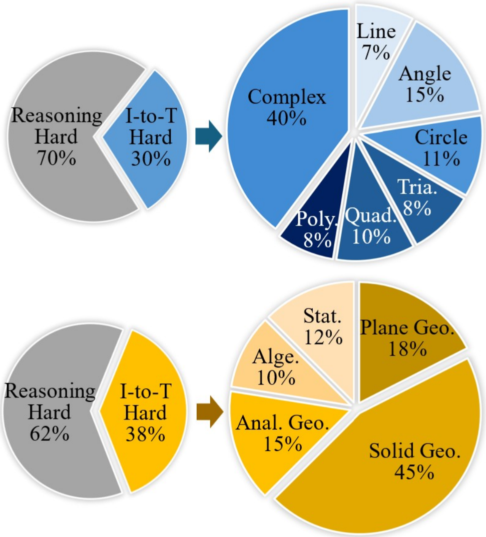
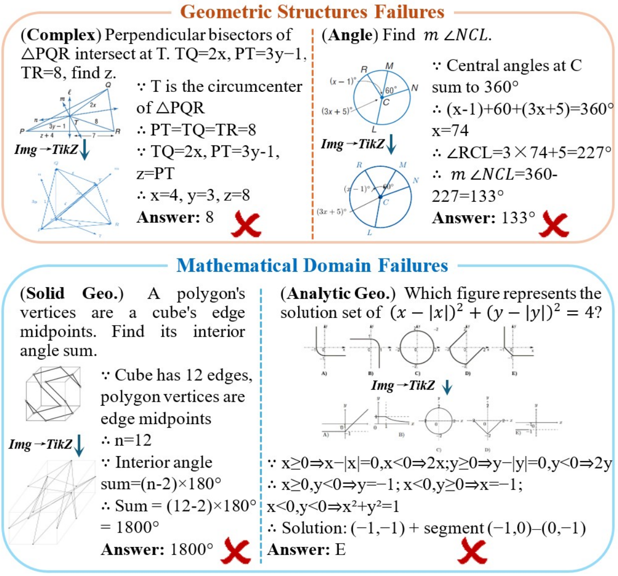


Figure S5. Average failure rate distributions of InternVL3.5-38B+GeoTikzBridge-Base-8B across different geometric structures (up) and mathematical domains (bottom).

tity for this task. Hence, we set $\tau = 0.8$ as the CLIP score based filtering threshold, balancing dataset size and sample quality.



Geometric Structures Failures

(**Complex**) Perpendicular bisectors of △PQR intersect at T. TQ=2x, PT=3y−1, TR=8, find z.

∵ T is the circumcenter of △PQR
∴ PT=TQ=TR=8
∵ TQ=2x, PT=3y-1, z=PT
∴ x=4, y=3, z=8
Answer: 8 ✗

(**Angle**) Find $m\angle NCL$.

∵ Central angles at C sum to 360°
∴ (x-1)+60+(3x+5)=360°
x=74
∴ ∠RCL=3×74+5=227°
∴ $m\angle NCL$=360-227=133°
Answer: 133° ✗

Mathematical Domain Failures

(**Solid Geo.**) A polygon's vertices are a cube's edge midpoints. Find its interior angle sum.

∵ Cube has 12 edges, polygon vertices are edge midpoints
∴ n=12
∵ Interior angle sum=(n-2)×180°
∴ Sum = (12-2)×180° = 1800°
Answer: 1800° ✗

(**Analytic Geo.**) Which figure represents the solution set of $(x-|x|)^2+(y-|y|)^2=4$?

∵ $x\geq0\Rightarrow x-|x|=0, x<0\Rightarrow 2x; y\geq0\Rightarrow y-|y|=0, y<0\Rightarrow 2y$
∴ $x\geq0, y<0\Rightarrow y=-1; x<0, y\geq0\Rightarrow x=-1;$
$x<0, y<0\Rightarrow x^2+y^2=1$
∴ Solution: (−1,−1) + segment (−1,0)–(0,−1)
Answer: E ✗

Figure S6. Failure cases of InternVL3.5-38B with tikz code input.

F. Failure Analysis

We analyze the failure cases of our proposed models on both geometric structure understanding and mathematical reasoning tasks. We first evaluate the GeoTikzBridge-Base-8B and GeoTikzBridge-Instruct-8B models across various geometric structures on the MathVista-GPS and GeoTikz-Instruct test sets, with results shown in Fig. S4. As shown from this figure, the GeoTikzBridge-Base-8B model suffers consistent performance degradation on complex figures, as overlong code tokens were truncated during training. The GeoTikzBridge-Instruct-8B model slightly underperforms on parallel line drawing tasks, due to the strict directional consistency constraint that raises higher requirements for generation precision.

Furthermore, we analyze problem-solving failure cases across geometric structures and mathematical domains, as shown in Fig. S5. I-to-T hard cases denote failure cases with an average score$\leq$0.9 across CLIP, SigLip and DINO metrics. The rest are reasoning hard cases. As shown from this figure, InternVL3.5-38B (with tikz code input) underperforms on tasks involving complex/angular geometric structures and solid/analytic geometry problems (examples in Fig. S6). These limitations stem from task difficulty, which we will address by developing an end-to-end reasoning paradigm in future work.

G. Metric Model-Agnostic Effectiveness

Though we use the CLIP model for evaluation, the performance gains of our models do not depend on any specific metric model. We provide additional SigLip

Table S1. Performance comparison of instructed code generation methods across CLIP, SigLip, and DINO model metrics.

Method	CLIP	SigLip	DINO
FigCodifier	0.936	0.940	0.957
ours-Base-8B	0.967	0.970	0.978
ours-Instruct-8B	**0.992**	**0.992**	**0.995**

Table S2. Performance comparison of image-to-tikz methods across CLIP, SigLip, and DINO model metrics.

Method	DaTikZ			MathVista-GPS		
	ClIP	SigLip	DINO	ClIP	SigLip	DINO
Qwen2.5-VL-72B	0.795	0.803	0.832	0.858	0.880	0.911
Qwen3-VL-30B	0.672	0.696	0.758	0.802	0.839	0.893
InternVL3.5-38B	0.733	0.752	0.786	0.786	0.793	0.855
GLM4.5V-106B	0.806	0.812	0.834	0.769	0.773	0.838
FigCodifier	0.785	0.791	0.819	0.884	0.906	0.922
ours-Base-8B	0.804	0.808	0.832	0.895	0.919	0.937
ours-Base-38B	**0.813**	**0.819**	**0.875**	**0.915**	**0.941**	**0.947**

Table S3. Performance comparison of image-to-tikz methods with GPT-5.0 on the MathVista-GPS benchmark across five metrics.

Method	FID↓	CSR↑	CLIP↑	SigLip↑	DINO↑
GPT-5.0	44.8	96.9%	0.880	0.903	0.921
ours-Base-8B	**36.0**	**97.1%**	**0.895**	**0.919**	**0.937**

and DINO evaluation results in Tables S1 and S2. As shown in these two tables, our GeoTikzBridge-Base-8B and GeoTikzBridge-Instruct-8B models consistently outperform the FigCodifier baseline across CLIP, SigLip and DINO metrics. They also maintain significant advantages over state-of-the-art baselines including Qwen2.5VL-72B and InternVL3.5-38B. These results demonstrate the metric model-agnostic effectiveness of our approach.

H. Comparison vs. Latest Closed-Source Models.

We compare the image-to-tikz performance of our GeoTikzBridge-Base-8B model with GPT-5.0 on the MathVista-GPS benchmark, with results reported in Table S3. As seen from this table, our model outperforms GPT-5.0 across all image-to-tikz metrics. Furthermore, we also provide the tikz codes generated by GeoTikzBridge-Base-8B to latest closed-source models for mathematical reasoning. The results on MathVista-GPS benchmark reported in Table S4 demonstrates that applying our approach to closed models also yields improvements.

Table S4. Mathematical reasoning performance of latest closed-source models on the MathVista-GPS benchmark. Scores are reported as original model / model + GeoTikzBridge-Base-8B.

Method	Accuracy (Base/+Ours)
GPT-5.0	0.891/**0.937**
Claude-4.5-Sonnect	0.774/**0.846**
Gemini-3-Pro	0.779/**0.923**

I. Supplementary Visualization

To supplement the results partially presented in the main text, we provide additional examples for the image-to-tikz task and the instruction-based image-to-tikz task in Fig. S7 and Fig. S8, respectively. As shown in Fig. S7, our model still maintains precise capture of key geometric components (line segments, angles, closed polygons, and semantic labels) in these additional examples. Fig. S8 further verifies that our model generates auxiliary lines more compliant with given instructions while preserving the characteristics of the original geometric figures.

As a complement to the quantitative results of both tikz assistant problem-solving task (Example I in Fig. S9 and Example II in Fig. S10) and auxiliary line assistant problem-solving task (Example III in Fig. S11 and Fig. S12, as well as Example IV in Fig. S13 and Fig. S14) presented in Table 2 and Table 4 of the main text (Sec. 4.2), we provide four representative qualitative examples. Consistent with the main text's evaluation setup: For the first two examples, InternVL3.5-38B is used for geometric reasoning; For the last two examples, InternVL3.5-38B first predicts auxiliary line instructions for original geometric problems, these instructions and raw images are then fed into our instruction-based model GeoTikzBridge-Instruct to generate tikz codes with auxiliary lines added, and the generated tikz codes and/or their rendered images are finally input to InternVL3.5-38B for geometric reasoning.

Notably, the examples align with key observations from Table 2 and Table 4: Examples I and II demonstrate that the model corrects its image perception errors after incorporating tikz codes; Example III yields an incorrect solution without auxiliary lines but correct ones with either augmented image or tikz code, demonstrating that introducing auxiliary lines, either in the form of augmented images or augmented images combined with tikz codes, benefit geometric problem solving; Example IV fails in both models "W/O Auxiliary Lines" and "With Auxiliary Line Image", with only "With Both Auxiliary Line Image And Tikz" leading to a correct solution, further demonstrating that tikz codes outperforms augmented images in facilitating precise geometric reasoning.

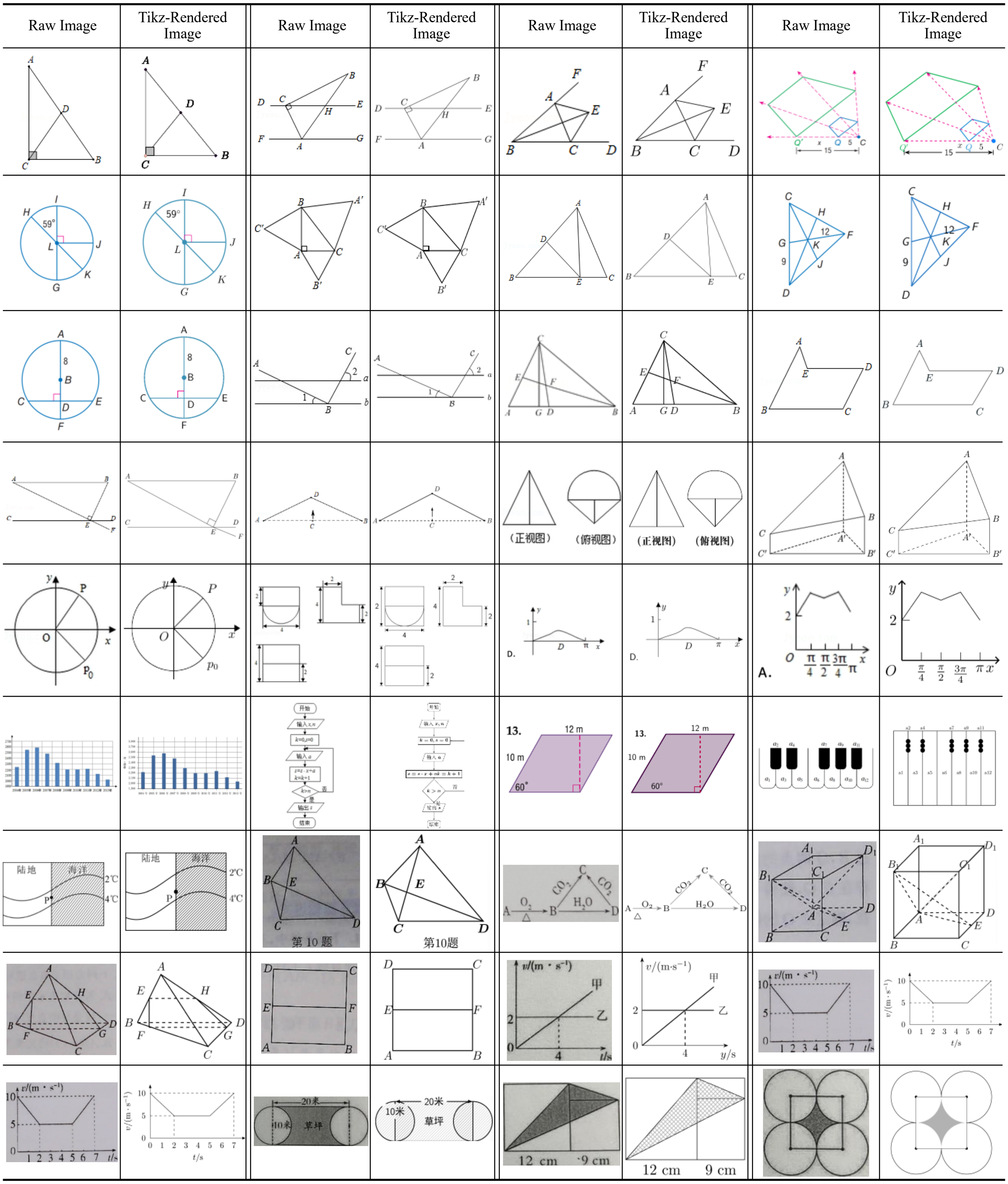


Figure S7. Visualization results of tikz codes generated by GeoTikzBridge-Base, consisting of input images (either geometric or non-geometric) and the corresponding figures rendered from our predicted tikz codes.

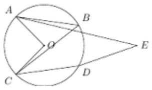
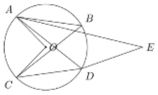
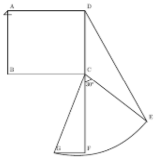
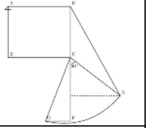
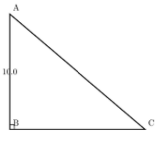
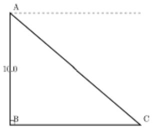
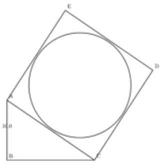
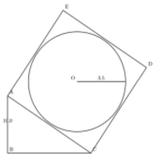
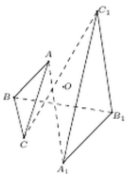
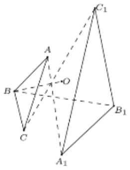
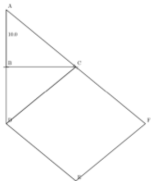
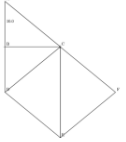
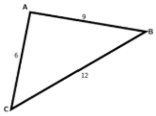
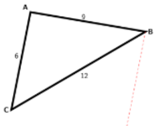
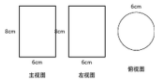
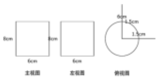
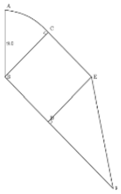
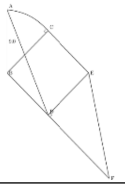
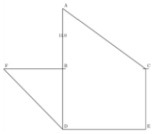
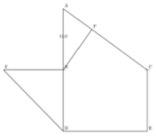
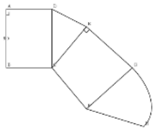
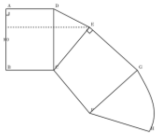
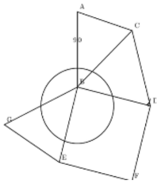
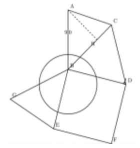
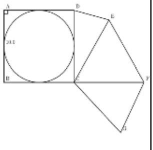
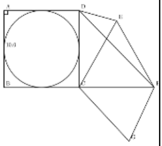
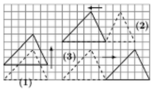
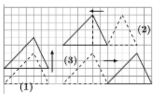
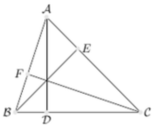
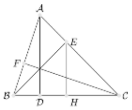
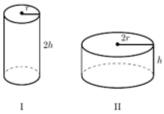
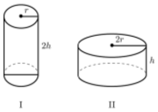
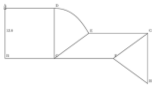
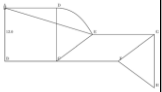
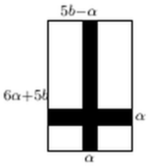
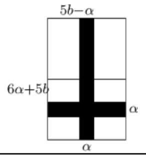
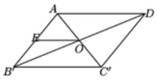
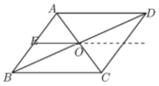
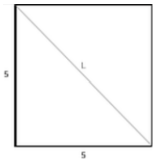
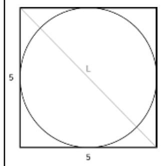
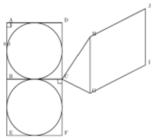
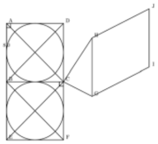
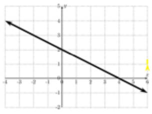
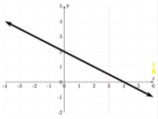
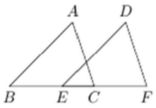
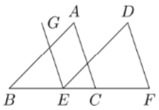
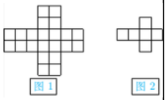
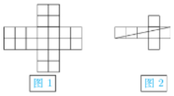
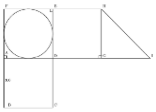
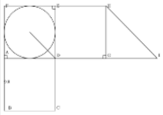
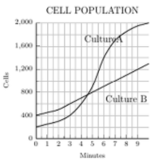
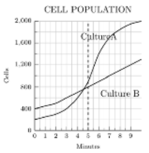
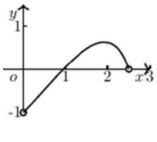
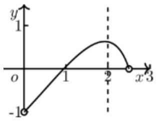
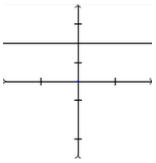
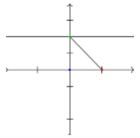
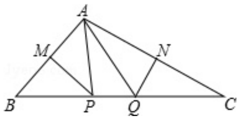
Connection of Two Points		Drawing a Perpendicular Line		Drawing a Parallel Line		Other Types of Auxiliary Lines	
Raw Image	Image With Auxiliary Lines Added	Raw Image	Image With Auxiliary Lines Added	Raw Image	Image With Auxiliary Lines Added	Raw Image	Image With Auxiliary Lines Added
	Connect AD 		Draw a perpendicular line to CD through point E 		Draw a line parallel to BC through point A 		Draw the radius of circle 
	Connect BO 		Draw a perpendicular line to BC through point E 		Draw a line parallel to AC through point B 		In the top view, draw the horizontal and vertical diameters of the circle. 
	Connect AD 		Draw a perpendicular line to AC through point B 		Draw a line parallel to BC through point E 		Draw the altitude of triangle ABC through point A 
	Connect points D and F 		Add an altitude perpendicular to the base in Figure (2) 		Draw a line parallel to AD through point E, intersecting BC at H. 		Draw the diameter on the base of Cylinder I 
	Connect points A and E 		Add a horizontal line perpendicular to this line on the middle vertical line 		Extend the line parallel to AD through point O 		Draw an incircle inside the square 
	Connect the center of the circle to the four vertices of the square 		Draw a straight line perpendicular to the x-axis at x=3 		Extend the line parallel to DF through point E 		Connect the lower-left corner of the square figure in Figure 2 to its upper-right corner 
	Connect the center of the circle to point D 		Draw a straight line perpendicular to the horizontal axis at x=5 		Draw a line parallel to the y-axis through point (2, 0) 		Mark an arbitrary point on the x-axis and connect it to an arbitrary point on the y-axis 

Figure S8. Visualization results of instruction-driven auxiliary line generation, including the input geometric images, the natural language instructions for auxiliary line addition, and the rendered figures from the generated tikz codes.

Problem Solving Example I: Tikz Code Improves Angle Calculation Reasoning

Question: As shown in the figure, ∠BAC = 110°, if A and B are symmetrical with respect to the line MP, A and C are symmetrical with respect to the line NQ, then the size of ∠PAQ is ()



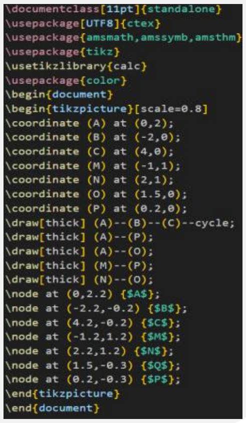
GT Answer: 40°

Tikz Generation by GeoTikzBridge-Base

Prompt:

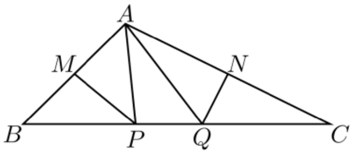
Please generate the corresponding TikZ code that accurately represents the visual elements in the image. TikZ is a powerful tool for creating vector graphics within LaTeX documents. Your generated code should be precise, well-structured, and should recreate the image as faithfully as possible.

Response:



```
\documentclass[11pt]{standalone}
\usepackage[UTF8]{ctex}
\usepackage{amsmath,amssymb,amsthm}
\usepackage{tikz}
\usetikzlibrary{calc}
\usepackage{color}
\begin{document}
\begin{tikzpicture}[scale=0.8]
\coordinate (A) at (0,2);
\coordinate (B) at (-2,0);
\coordinate (C) at (4,0);
\coordinate (M) at (-1,1);
\coordinate (N) at (2,1);
\coordinate (O) at (1.5,0);
\coordinate (P) at (0.2,0);
\draw[thick] (A)--(B)--(C)--cycle;
\draw[thick] (A)--(P);
\draw[thick] (A)--(O);
\draw[thick] (M)--(P);
\draw[thick] (N)--(O);
\node at (0,2.2) {$A$};
\node at (-2.2,-0.2) {$B$};
\node at (4.2,-0.2) {$C$};
\node at (-1.2,1.2) {$M$};
\node at (2.2,1.2) {$N$};
\node at (1.5,-0.3) {$Q$};
\node at (0.2,-0.3) {$P$};
\end{tikzpicture}
\end{document}
```

Image Rendered From the Generated Tikz Code:



Problem Solving by InternVL3.5-38B W/O Tikz

Prompt:

<Image> Please answer this question based on the given image. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think> To determine ∠PAQ: 1. A and B are symmetrical about MP, so MP bisects AB perpendicularly, hence ∠BAP = ∠ABP. 2. A and C are symmetrical about NQ, so NQ bisects AC perpendicularly, hence ∠CAP = ∠ACQ. 3. In △ABC, ∠B + ∠C = 180° - 110° = 70°. 4. Since ∠BAP = ∠ABP and ∠CAP = ∠ACQ, ∠BAP + ∠CAP = 140° (as 2(∠BAP + ∠CAP) = 280°). 5. ∠PAQ = 360° - 2(∠BAP + ∠CAP) = 360° - 280° = 80°. Thus, ∠PAQ is 80°.</think>
<answer>**80°**</answer>

Problem Solving by InternVL3.5-38B With Tikz

Prompt:

<Image> <tikz><Generated TikZ Code></tikz> Please answer this question based on the given image and the corresponding Tikz code. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

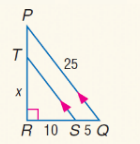
Response:

<think> To determine ∠PAQ: 1. A and B are symmetrical about MP, so MP bisects AB perpendicularly, hence ∠BAP = ∠ABP. 2. A and C are symmetrical about NQ, so NQ bisects AC perpendicularly, hence ∠QAC = ∠C. 3. In △ABC, ∠B + ∠C = 180° - 110° = 70°. 4. Since ∠PAB = ∠B and ∠QAC = ∠C, ∠PAB + ∠QAC = 70°. 5. ∠PAQ = ∠BAC - (∠PAB + ∠QAC) = 110° - 70° = 40°. Thus, ∠PAQ is 40°.</think>
<answer>**40°**</answer>

Figure S9. Example I: Angle calculation problem solving comparison with and without tikz code. Tikz clarifies geometric relationships to avoid perceptual errors in angle reasoning.

Problem Solving Example II: Tikz Code Enables Precise Segment Length Calculation

Question: Find PT.



GT Answer: **20/3**

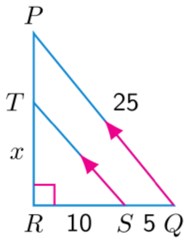
Tikz Generation by GeoTikzBridge-Base

Prompt: Please generate the corresponding TikZ code that accurately represents the visual elements in the image. TikZ is a powerful tool for creating vector graphics within LaTeX documents. Your generated code should be precise, well-structured, and should recreate the image as faithfully as possible.

Response:

```
\documentclass[tikz,border=3.14mm]{standalone}
\usepackage[UTF8]{ctex}
\usetikzlibrary{calc,angles,quotes,arrows.meta}
\begin{document}
    \begin{tikzpicture}[font=\sffamily,>=Stealth]
        \path (0,0) coordinate (R) node[below]{$R$}
        (2,0) coordinate (Q) node[below]{$Q$}
        (0,2.5) coordinate (P) node[above]{$P$}
        (1.3,0) coordinate (S) node[below]{$S$}
        (0,1.5) coordinate (T) node[left]{$T$};
        \draw[thick,cyan!70!blue] (R) -- (Q) -- (P) -- cycle;
        \draw[thick,cyan!70!blue] (R) -- (S) -- (T) -- cycle;
        \draw[magenta,thick] (R) ++ (0,0.3) coordinate (R') -- ++ (0.3,0) coordinate (R'') -- ++ (0,-0.3);
        \draw[magenta,thick,-{Latex[length=3mm]}] (S) -- ($(S)!0.5!(T)$);
        \draw[magenta,thick,-{Latex[length=3mm]}] (Q) -- ($(Q)!0.5!(P)$);
        \path (T) -- (R) node[midway,left]{$x$}
        (Q) -- (P) node[midway,above right]{25}
        (R) -- (S) node[midway,below]{10}
        (Q) -- (S) node[midway,below]{5};
    \end{tikzpicture}
\end{document}
```

Image Rendered From the Generated Tikz Code:



Problem Solving by InternVL3.5-38B W/O Tikz

Prompt: <Image> Please answer this question based on the given image. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response: <think>To find PT, use similar triangles: triangles PRT and PRQ are similar as they share angle P and both have a right angle. Given RQ = RS + SQ = 10 + 5 = 15 and PQ = 25, use the Pythagorean theorem in triangle PRQ: $PR^2 + 15^2 = 25^2 \rightarrow PR^2 + 225 = 625 \rightarrow PR^2 = 400 \rightarrow PR = 20$. Use the similarity ratio: $20/25 = RT/15 \rightarrow 4/5 = RT/15 \rightarrow RT = (4\times15)/5 = 12$. Find PT: $PT = \sqrt{(20^2 + 12^2)} = \sqrt{(400 + 144)} = \sqrt{544} = 4\sqrt{34}$. So, $PT = 4\sqrt{34}$.</think>
<answer>**4√34**</answer> ✗

Problem Solving by InternVL3.5-38B With Tikz

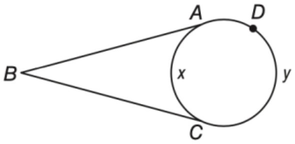
Prompt: <Image> <tikz><Generated TikZ Code></tikz> Please answer this question based on the given image and the corresponding Tikz code. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response: <think>To find PT, use similar triangles: triangles PRT and PRQ are similar as they share angle P and both have a right angle. Given RS = 10, SQ = 5, so RQ = 10 + 5 = 15 and PQ = 25. Use the Pythagorean theorem in triangle PRQ: $PR^2 + 15^2 = 25^2 \rightarrow PR^2 + 225 = 625 \rightarrow PR^2 = 400 \rightarrow PR = 20$. Use the similarity ratio: $TR/20 = 10/15 \rightarrow TR/20 = 2/3 \rightarrow TR = 40/3$. Find PT: $PT = PR - TR = 20 - 40/3 = 20/3$. So, $PT = 20/3$.</think>
<answer>**20/3**</answer> ✓

Figure S10. Example II: Segment length problem solving comparison with and without tikz code. Tikz facilitates accurate proportion and theorem application for length reasoning.

Problem Solving Example III (1/2): Auxiliary Line & Tikz Generation (Circle Center-Tangent Connection)

Question: The degree measures of minor arc $\widehat{AC}$ and major arc $\widehat{ADC}$ are x and y respectively. If ∠ABC = 70°, find x.



GT Answer: **110**

Auxiliary Line Instruction Generation by InternVL3.5-38B

Prompt:

<Image> Please provide the most appropriate instruction for adding auxiliary lines to solve the problem based on the given problem and the provided geometric image. The instruction should be simple and clear, such as "Connect AC", "Draw a perpendicular line from point B to AC", "Draw a line through point B parallel to AC", etc. If no auxiliary line is needed, output "None". Do not include the thinking process. Additionally, place the instruction for adding auxiliary lines between <instruction> and </instruction>. <Problem Text>

Response:

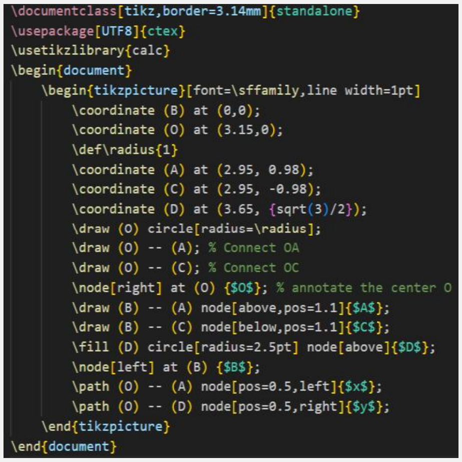
<instruction>Connect the center O to points A and C.</instruction>

Tikz Generation by GeoTikzBridge-Instruct

Prompt:

Modify the given image based on the following requirements: Connect the center O to points A and C. Please generate the corresponding TikZ code that accurately represents the visual elements in the modified image. TikZ is a powerful tool for creating vector graphics within LaTeX documents. Your generated code should be precise, well-structured, and should recreate the image as faithfully as possible.

Response:



```
\documentclass[tikz,border=3.14mm]{standalone}
\usepackage[UTF8]{ctex}
\usetikzlibrary{calc}
\begin{document}
    \begin{tikzpicture}[font=\sffamily,line width=1pt]
        \coordinate (B) at (0,0);
        \coordinate (O) at (3.15,0);
        \def\radius{1}
        \coordinate (A) at (2.95, 0.98);
        \coordinate (C) at (2.95, -0.98);
        \coordinate (D) at (3.65, {sqrt(3)/2});
        \draw (O) circle[radius=\radius];
        \draw (O) -- (A); % Connect OA
        \draw (O) -- (C); % Connect OC
        \node[right] at (O) {$O$}; % annotate the center O
        \draw (B) -- (A) node[above,pos=1.1]{$A$};
        \draw (B) -- (C) node[below,pos=1.1]{$C$};
        \fill (D) circle[radius=2.5pt] node[above]{$D$};
        \node[left] at (B) {$B$};
        \path (O) -- (A) node[pos=0.5,left]{$x$};
        \path (O) -- (D) node[pos=0.5,right]{$y$};
    \end{tikzpicture}
\end{document}
```

Image Rendered From the Generated Tikz Code:

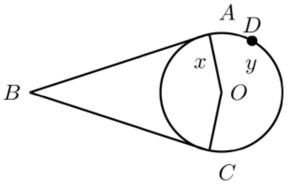
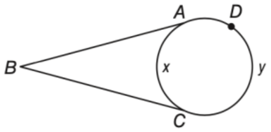


Figure S11. Example III (1/2): Auxiliary line instruction (connecting circle center and tangent points) and tikz code generation. The line manifests radius-tangent perpendicularity.

Problem Solving Example III (2/2): Auxiliary Line Assistant Reasoning (Circle Center-Tangent Connection: Dual Formats Effective)

Question: The degree measures of minor arc $\widehat{AC}$ and major arc $\widehat{ADC}$ are x and y respectively. If ∠ABC = 70°, find x.



GT Answer: **110**

Problem Solving by InternVL3.5-38B W/O Auxiliary Lines

Prompt:

<Image> Please answer this question based on the given image. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think>To solve for x, use the Inscribed Angle Theorem: an inscribed angle's measure is half its intercepted arc's measure. ∠ABC is an inscribed angle intercepting arc AC (measure x). Given ∠ABC = 70°, so 70° = 1/2 x. Solve: x = 70° × 2 = 140°. Thus, x = 140.</think>
<answer>**140**</answer> ✗

Problem Solving by InternVL3.5-38B With Auxiliary Line Image

Prompt:

<Image><Rendered Image> Please answer this question based on the given image and the image with auxiliary lines added. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think>To solve for x, use tangent properties and central angle relationships: By tangent property, OA⊥BA and OC⊥BC, so ∠OAB = ∠OCB = 90°. In quadrilateral OABC, interior angles sum to 360°. Given ∠ABC = 70°, so 90° + 90° + 70° + ∠AOC = 360°, thus ∠AOC = 110°. The central angle ∠AOC corresponds to minor arc AC (measure x), so x = 110°. Thus, x = 110.</think>
<answer>**110**</answer> ✓

Problem Solving by InternVL3.5-38B With Both Auxiliary Line Image And Tikz

Prompt:

<Image><Rendered Image> <tikz><Generated TikZ Code></tikz> Please answer this question based on the given image, the image with auxiliary lines added, and the corresponding Tikz code. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

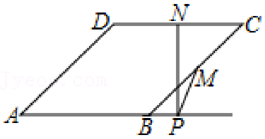
Response:

<think>To solve for x, use tangent properties and central angle relationships: By the tangent property, OA ⊥ BA and OC ⊥ BC, so ∠OAB = ∠OCB = 90°. In quadrilateral OABC, the sum of interior angles is 360°. Given ∠ABC = 70°, we have 90° + 90° + 70° + ∠AOC = 360°, so ∠AOC = 110°. The central angle ∠AOC corresponds to minor arc AC (measure x), so x = 110°. Thus, x = 110.</think>
<answer>**110**</answer> ✓

Figure S12. Example III (2/2): Problem solving comparison under different forms, with incorrect analysis/answers highlighted in red and key correct ones in green. Both auxiliary line image and its tikz combination enable correct reasoning via explicit perpendicularity recognition.

Problem Solving Example IV (1/2): Auxiliary Line & Tikz Generation (Key Right Triangle Construction)

Question: In the figure, in rhombus ABCD, M and N are the midpoints of BC and CD respectively. NP is perpendicular to AB at point P, and MP is connected. If ∠DAB = 40°, then ∠MPB = ( ) ?



GT Answer: **110°**

Auxiliary Line Instruction Generation by InternVL3.5-38B

Prompt:

<Image> Please provide the most appropriate instruction for adding auxiliary lines to solve the problem based on the given problem and the provided geometric image. The instruction should be simple and clear, such as "Connect AC", "Draw a perpendicular line from point B to AC", "Draw a line through point B parallel to AC", etc. If no auxiliary line is needed, output "None". Do not include the thinking process. Additionally, place the instruction for adding auxiliary lines between <instruction> and </instruction>. <Problem Text>

Response:

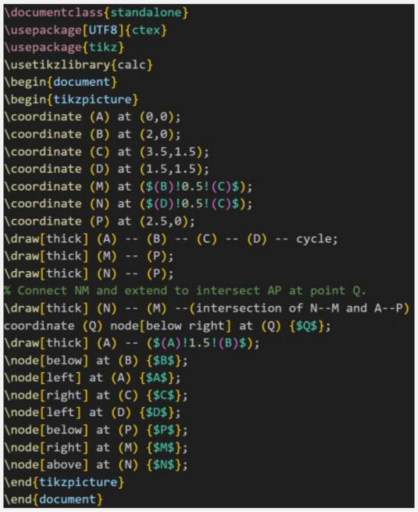
<instruction>Connect NM and extend to intersect AP at point Q.</instruction>

Tikz Generation by GeoTikzBridge-Instruct

Prompt:

Modify the given image based on the following requirements: Connect NM and extend to intersect AP at point Q. Please generate the corresponding TikZ code that accurately represents the visual elements in the modified image. TikZ is a powerful tool for creating vector graphics within LaTeX documents. Your generated code should be precise, well-structured, and should recreate the image as faithfully as possible.

Response:



```
\documentclass{standalone}
\usepackage[UTF8]{ctex}
\usepackage{tikz}
\usetikzlibrary{calc}
\begin{document}
\begin{tikzpicture}
\coordinate (A) at (0,0);
\coordinate (B) at (2,0);
\coordinate (C) at (3.5,1.5);
\coordinate (D) at (1.5,1.5);
\coordinate (M) at ($(B)!0.5!(C)$);
\coordinate (N) at ($(D)!0.5!(C)$);
\coordinate (P) at (2.5,0);
\draw[thick] (A) -- (B) -- (C) -- (D) -- cycle;
\draw[thick] (M) -- (P);
\draw[thick] (N) -- (P);
% Connect NM and extend to intersect AP at point Q.
\draw[thick] (N) -- (M) --(intersection of N--M and A--P)
coordinate (Q) node[below right] at (Q) {$Q$};
\draw[thick] (A) -- ($(A)!1.5!(B)$);
\node[below] at (B) {$B$};
\node[left] at (A) {$A$};
\node[right] at (C) {$C$};
\node[left] at (D) {$D$};
\node[below] at (P) {$P$};
\node[right] at (M) {$M$};
\node[above] at (N) {$N$};
\end{tikzpicture}
\end{document}
```

Image Rendered From the Generated Tikz Code:

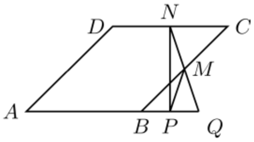
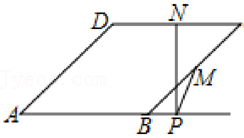


Figure S13. Example IV (1/2): Auxiliary line instruction (extending midpoint connection) and tikz code generation. The line constructs a key right triangle by forming intersection Q.

Problem Solving Example IV (2/2): Auxiliary Line Assistant Reasoning (Key Right Triangle Construction: Tikz-Augmented Combination Effective)

Question: In the figure, in rhombus ABCD, M and N are the midpoints of BC and CD respectively. NP is perpendicular to AB at point P, and MP is connected. If ∠DAB = 40°, then ∠MPB = ( ) ?



GT Answer: **110°**

Problem Solving by InternVL3.5-38B W/O Auxiliary Lines

Prompt:

<Image> Please answer this question based on the given image. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think>To find ∠MPB in rhombus ABCD: 1. ABCD is a rhombus, ∠DAB = 40°, so ∠DBC = ∠DBA = 0.5×∠DAB = 20°, AB ∥ CD, CD = CB. 2. N is the midpoint of CD, so CN = DN. Thus BN = CN = DN, ∠NBD = ∠NDB = 20°, ∠NBP = 40°. 3. NP⊥AB, so ∠NPB = 90°, thus ∠BNP = 50°. 4. M is the midpoint of BC, BN = CN, so MN ∥ BD. 5. By MN ∥ BD, ∠MPB = ∠DBA = 20°. Thus, ∠MPB = 20°.</think>
<answer>**20°**</answer> ✗

Problem Solving by InternVL3.5-38B With Auxiliary Line Image

Prompt:

<Image><Rendered Image> Please answer this question based on the given image and the image with auxiliary lines added. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think>To find ∠MPB in rhombus ABCD: 1. Adjacent angles in a rhombus are supplementary. Given ∠DAB = 40°, ∠ABC = 140° and ∠C = 40°. 2. M, N are midpoints of BC, CD, so CM = CN (BC = CD in rhombus). △CMN is isosceles, so ∠CNM = 70°. 3. AB ∥ CD. By parallel line property, ∠Q = ∠CNM = 70°. 4. NP ⊥ AB, so △NPQ is right-angled. M is the midpoint of PQ, by hypotenuse median theorem, MP = MQ. 5. △MPQ is isosceles (∠MPQ = ∠Q = 70°), so ∠MPB = ∠MPQ = 70°. Thus, ∠MPB = 70°.</think>
<answer>**70°**</answer> ✗

Problem Solving by InternVL3.5-38B With Both Auxiliary Line Image And Tikz

Prompt:

<Image><Rendered Image> <tikz><Generated TikZ Code></tikz> Please answer this question based on the given image, the image with auxiliary lines added, and the corresponding Tikz code. Provide the correct answer, place the thinking process between <think> and </think>, and place the final answer between <answer> and </answer>. <Problem Text>

Response:

<think>To find ∠MPB in rhombus ABCD: 1. Adjacent angles in a rhombus are supplementary. Given ∠DAB = 40°, ∠ABC = 140° and ∠C = 40°. 2. M, N are midpoints of BC, CD, so CM = CN (BC = CD in rhombus). △CMN is isosceles, so ∠CMN = ∠CNM = 70°. 3. AB ∥ CD, so ∠Q = ∠CNM = 70°. 4. NP ⊥ AB, so △NPQ is right-angled. By hypotenuse median theorem, MP = MQ. 5. △MPQ is isosceles (∠MPQ = ∠Q = 70°), so ∠MPB = 180° - 70° = 110°. Thus, ∠MPB = 110°.</think>
<answer>**110°**</answer> ✓

Figure S14. Example IV (2/2): Problem solving comparison under different forms, with incorrect analysis/answers highlighted in red and key correct ones in green. The auxiliary line image provides a visual foundation for constructing the right triangle △NPQ. Its combination with tikz code achieves correct reasoning, since tikz explicitly encodes the spatial structure of the constructed right triangle.