Fine-Grained Geometric Visual Perception: Recovering Precise Positions, Boundaries, and Trajectories

Anonymous authors
Paper under double-blind review

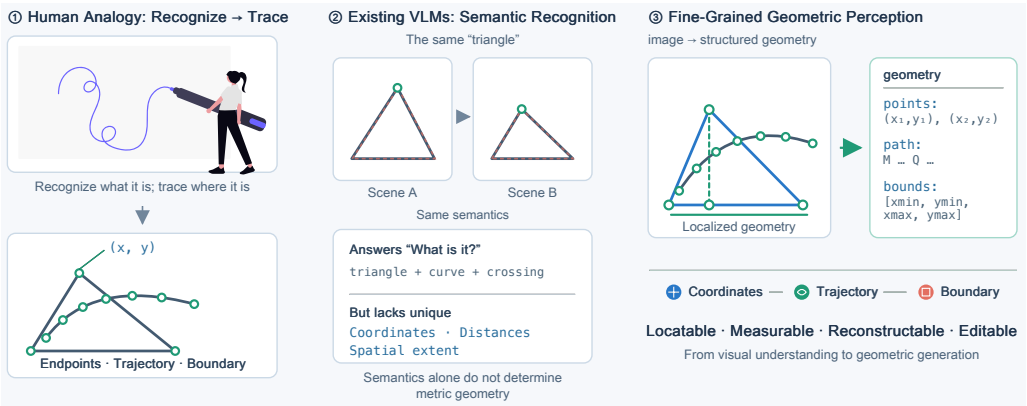

Figure 1: Motivation for fine-grained geometric visual perception. Recognizing a figure and tracing its geometry require different levels of precision. Scenes with the same semantics and topology can still differ in coordinates, distances, and spatial extent; recovering coordinates, trajectories, and boundaries turns a visual input into a measurable and reconstructable geometric representation.

Abstract

Vision–language models (VLMs) can recognize geometric objects and coarse relations, yet they remain unreliable when asked to recover precise positions, endpoints, boundaries, and continuous trajectories in a common canvas coordinate system. We formulate this capability as fine-grained geometric visual perception (FGGVP) and introduce FGGVP-22K, a dataset containing approximately 20,000 synthetic samples and 1,500 real samples. The data cover explicit primitive localization, semantically specified components, and derived targets determined by geometric relations. We fine-tune Qwen3-VL-4B with native coordinate tokens and multitask supervision, combining autoregressive answer loss with ordinal-coordinate, expected pixel-distance, and coordinate-point conditional value-at-risk losses. Experiments on synthetic, real, and external geometric datasets show substantial improvements in fine-grained localization over the base model and stronger Image2Code reconstruction. The results indicate that targeted supervision of positions, boundaries, and trajectories improves geometric visual perception and transfers to downstream reconstruction.

1 Introduction

In recent years, vision–language models have developed strong image-level semantic understanding (Alayrac et al., 2022; Li et al., 2023; Liu et al., 2023; Bai et al., 2023; Wang et al., 2024). For geometric images, they can usually recognize basic primitives such as points, line segments, circles, and curves, and identify some relative or topological relations such as adjacency, intersection, connection, and parallelism. However, when a task requires accurate recovery of a primitive's absolute canvas position, endpoints and boundary, continuous trajectory, or occupied spatial extent, existing models often produce unstable or large errors.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

A model may recognize that an image contains a line segment or that two lines intersect, while still failing to identify the segment endpoints, the precise intersection, or the geometric information needed for accurate localization and reconstruction. Slow Perception similarly shows that accurate geometric copying is difficult for current multimodal models (Wei et al., 2024).

We call this capability fine-grained geometric visual perception (FGGVP). As illustrated in Figure 1, FGGVP binds a concrete geometric object in a visual input to a common canvas coordinate system and recovers its precise position, boundary, and ordered trajectory. The recovered representation should support subsequent geometric computation and structured operations. Existing visual grounding and coordinate-modeling approaches provide a foundation for object localization, but typically output boxes, masks, or generic points (Chen et al., 2023; Peng et al., 2023; He et al., 2024; Chen et al., 2022). FGGVP instead requires corresponding coordinates for endpoints, ordered trajectories, and targets defined by semantic or geometric relations.

We address this problem with FGGVP-22K and a multitask coordinate-supervised fine-tuning scheme. The dataset follows a five-stage construction process: primitive generation, primitive composition, shape composition and target specification, semantic-target construction, and rendering with quality control. Programs retain primitive parameters, object relations, and coordinate transforms, while analytic geometry provides supervision for midpoints, division points, perpendicular feet, and intersections that are not rendered as independent objects. We further augment the synthetic data with real geometric scenes from Geo-Code and add semantic and derived-target annotations to capture appearance and structural variation in real images.

During training, endpoints, boundary keypoints, and ordered trajectories are represented with native coordinate tokens. In addition to answer-token cross-entropy, ordinal-coordinate, expected pixel-distance, and coordinate-point conditional value-at-risk losses constrain coordinate ordering, spatial distance, and large local errors. We evaluate fine-grained localization on synthetic, real, and external PGDP5K test sets, and measure downstream utility on Image2Code. Despite using a 4B model, the proposed training substantially lowers average pixel error relative to the base model across the reported localization tasks and improves geometric reconstruction. These findings show that targeted supervision for positions, boundaries, and trajectories can strengthen VLM geometric perception and its downstream use.

Our contributions are:

- We formulate FGGVP as a measurable capability that unifies explicit position and trajectory recovery, semantic target selection, and derived-object localization in a common canvas coordinate system.
- We construct FGGVP-22K through parameterized five-stage synthesis and real-scene annotation, aligning primitive structure, query semantics, and analytic coordinates to provide traceable supervision for the three capability levels.
- Experiments show that FGGVP-22K training substantially improves fine-grained localization on synthetic, real, and external geometric data and improves Image2Code reconstruction. Classical fitting remains more precise on clean primitives, but does not directly address semantic references or relation-defined targets.

2 Related Work

2.1 Fine-grained geometric visual perception

Geometric datasets study primitive parsing and reasoning from complementary perspectives. PGDP5K annotates primitive locations and relations (Hao et al., 2022); PGPS9K couples diagram parsing with textual clauses for problem solving (Zhang et al., 2023); Geometry3K/Inter-GPS connects diagram representations to symbolic reasoning (Lu et al., 2021); and GeomVerse evaluates reasoning at controlled difficulty levels (Kazemi et al., 2023). These resources support object and relation understanding, but a correct symbolic

relation does not specify the endpoints, trajectory, or extent of its visual referent. We directly supervise these coordinates, including geometrically defined targets that are not separately drawn.

Slow Perception decomposes figures into point–line units and traces strokes incrementally with a perceptual ruler (Wei et al., 2024). We extend geometric recovery to query-conditioned component selection and implicit derived targets, jointly representing endpoints, arc keypoints, text-box corners, and ordered curves through coordinate tokens. This interface supports heterogeneous targets without organizing inference around incremental stroke tracing. CogAlign trains VLMs to recognize invariants under visual transformations, improving visual arithmetic and geometric reasoning (Huang et al., 2025). Our supervision instead specifies absolute canvas coordinates and directly addresses localization error.

Grounding methods link semantic references to spatial outputs. Shikra expresses coordinates in natural language, while Kosmos-2 links text spans to boxes through location tokens (Chen et al., 2023; Peng et al., 2023). AnyRef and LISA predict pixel-level masks from multimodal references or reasoning queries (He et al., 2024; Lai et al., 2023). Such outputs locate objects or regions but do not directly encode ordered trajectories or unmarked construction points. Pix2Seq establishes discrete autoregressive modeling of visual coordinates (Chen et al., 2022). Building on this perspective, our distinction is task-aligned geometric supervision: FGGVP-22K connects object structure and query semantics to explicit and derived coordinates, and coordinate-aware auxiliary losses complement token cross-entropy with spatial supervision.

2.2 Image2Code and geometric reconstruction

DeepSVG learns hierarchical representations of shapes and drawing commands (Carlier et al., 2020). StarVector generates SVG code from images and learns diverse primitives through SVG-Stack (Rodriguez et al., 2023). Chat2SVG combines language-generated templates with diffusion-guided optimization for text-to-SVG generation (Wu et al., 2024). These approaches emphasize structured graphics and rendered quality. Our task recovers query-specified geometry from an image and measures coordinate error directly, so semantic resemblance or valid code alone cannot establish geometric accuracy.

Geo-Code combines pixel-anchored geometric modeling with metric-driven code evolution in a synthesis–rendering–validation loop, and provides real geometric data and GeoCodeLM (Wu et al., 2026). Its agent framework improves executable representations through operators and iterative feedback. Our method instead explicitly trains the VLM to recover positions, boundaries, and trajectories before downstream reconstruction. The difference concerns both the optimization stage and the supervised representation: Geo-Code refines programs, while our coordinate tasks directly diagnose perceptual errors. We retain the Geo-Code agent framework and GeoCodeLM as downstream comparisons to assess whether improved geometric perception benefits full-scene reconstruction.

3 Method

3.1 Data requirements

As illustrated in Figure 2, fine-grained geometric visual perception consists of three progressively structured capabilities: fine-grained positional perception, semantic perception, and derived-object perception.

Fine-grained positional perception concerns the spatial information of primitives, including their absolute positions on the canvas, endpoints and extents, continuous trajectories, and occupied spatial regions. It provides the basis for the other capabilities. Semantic perception requires the model to bind precise geometric locations to the correct primitive, role, or relation specified by a query. Derived-object perception requires the model to locate targets determined by existing primitives and their geometric relations, even when those targets are not rendered as independent objects, such as midpoints, division points, perpendicular feet, and intersections.

Table 1 compares whether representative datasets define tasks corresponding to these three capabilities.

Table 1: Coverage of the three fine-grained geometric visual perception capabilities in representative datasets.

Dataset	Position	Semantic	Derived
PGDP5K	✓	✓	–
PGPS9K	✓	✓	–
Geometry3K / Inter-GPS	✓	✓	–
Slow Perception	✓	–	–
GeomVerse	–	–	–
SVG-Stack / StarVector	✓	–	–
RefCOCO / RefCOCO+ / RefCOCOg	–	–	–
AnyRef	–	–	–
PixMo-Points / Molmo	–	–	–

3.2 Data construction

To obtain supervision aligned with the three capabilities, we use the construction pipeline shown in Figure 2.

Primitive generation. A parameterized program generates line segments, arcs, irregular curves, and other basic primitives, while recording geometric parameters, object types, and coordinate transforms.

Primitive composition. Under constraints on object count, position, scale, orientation, and density, primitives are composed within and across families to form scenes containing multiple segments, arcs, and mixed primitives.

Shape composition and target specification. Triangles, rectangles, circles, and other basic shapes are introduced into the primitive scene. Membership and topological relations between primitives and shapes are recorded. A visible label or an explicit object description specifies the query target, producing paired image, query, and target-coordinate samples.

Semantic target construction. Midpoints, division points, perpendicular feet, and intersections are constructed on the validated scene graph. The process first specifies the target relation, samples parent objects satisfying that relation, and computes the target coordinates using analytic geometry. Semantic targets are provided as ground-truth supervision and are not rendered as additional independent objects.

Rendering and quality control. Fonts, font sizes, and stroke widths are randomized within predefined ranges and recorded. Deterministic checks verify geometric relations, coordinate transforms, and target uniqueness; sampled visual checks verify target readability and consistency between the query and scene structure.

To improve generalization to real scenes, we use real geometric data released with GeoCode and add semantic and derived-object annotations on top of these scenes. The final dataset contains approximately 20,000 synthetic samples and 1,500 real samples and is named FGGVP-22K.

3.3 Coordinate representation and task serialization

As shown in Figure 3, each image is converted to a 448×448 coordinate frame with origin at the upper-left corner. For a coordinate $c \in [0, 447]$, we quantize it into 1024 discrete bins:

$$k(c) = \text{clip}\left(\text{round}\left(\frac{c}{447} \times 1023\right), 0, 1023\right). \tag{1}$$

The decoded pixel coordinate is $\hat{c} = k(c) \times 447/1023$. With radix $R = 32$, $k = Rh + \ell$ where $h, \ell \in \{0, \ldots, 31\}$. Each two-dimensional point is represented by four tokens:

$$(x_{\text{high}}, x_{\text{low}}, y_{\text{high}}, y_{\text{low}}). \tag{2}$$

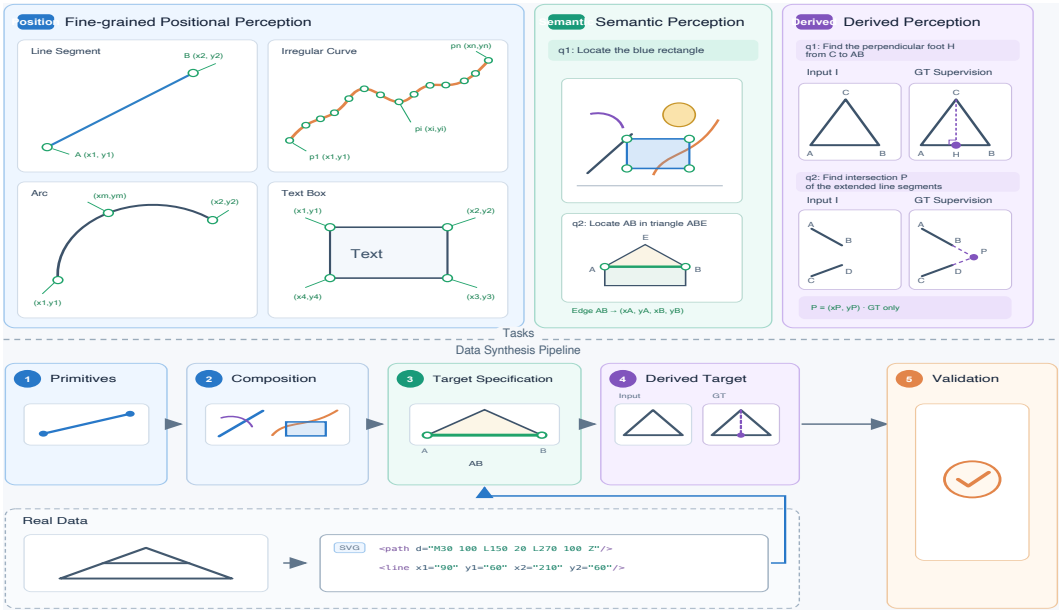

Figure 2: Three levels of fine-grained geometric visual perception and the data construction pipeline.

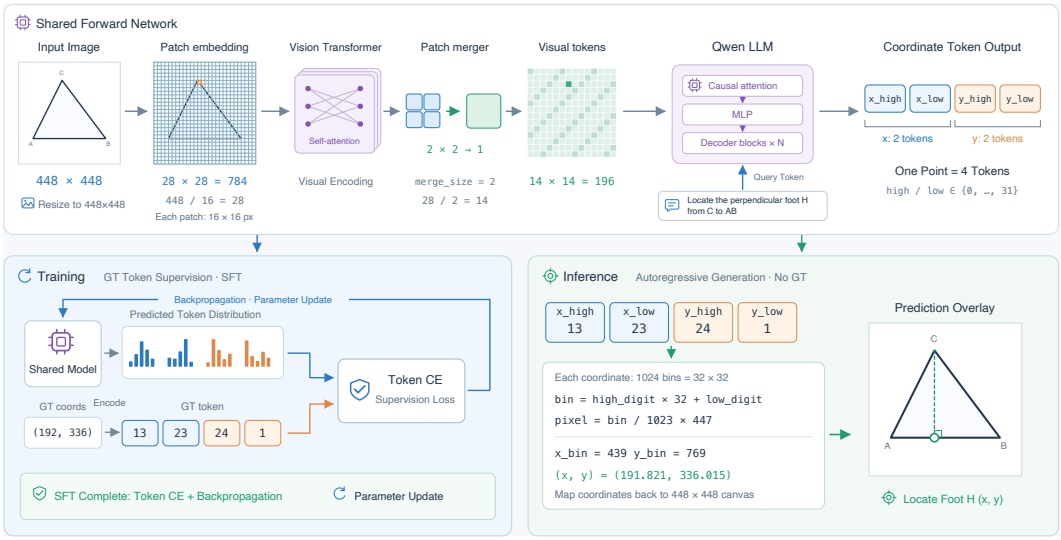

Figure 3: Coordinate-token representation and training/inference flow.

A segment outputs two endpoints, an arc three key points, and a regular trajectory eight sampled points.

3.4 Training objective and multitask replay

We use autoregressive supervised fine-tuning with native coordinate tokens. The model is initialized from Qwen3-VL-4B and loads a pretrained geometric adapter before multitask replay. Given an image I_i, query q_i, and target sequence z_i, the model autoregressively generates the target tokens. Prompt and padding positions are masked, while answer tokens and the answer termination token are supervised. The primary loss is the per-example

normalized autoregressive cross-entropy:

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{N}\sum_{i=1}^{N}\frac{1}{|\mathcal{A}_i|}\sum_{t\in\mathcal{A}_i}\log p_\theta(z_{i,t} \mid I_i, q_i, z_{i,<t}). \tag{3}$$

To jointly optimize token correctness and spatial accuracy, we add three coordinate losses. The ordinal coordinate loss uses the order of coordinate bins to assign softer penalties to nearby errors. The expected pixel-distance loss constrains the pixel distance induced by the predicted coordinate distribution. The coordinate-point conditional value-at-risk loss emphasizes large local errors among the target points, preventing the average loss from hiding severe outliers. Let e_j denote the pixel error of target point j, let M be the number of target points, and let α be the fixed risk level. We define

$$\mathcal{L}_{\mathrm{tail},\alpha} = \min_{\tau}\left[\tau + \frac{1}{1-\alpha}\frac{1}{M}\sum_{j=1}^{M}\max(e_j-\tau, 0)\right]. \tag{4}$$

For the full objective, let $\mathcal{L}_{\mathrm{tail}} = \mathcal{L}_{\mathrm{tail},\alpha}$ at the fixed risk level. The objective is

$$\mathcal{L} = \lambda_{\mathrm{CE}}\mathcal{L}_{\mathrm{CE}} + \lambda_{\mathrm{ord}}\mathcal{L}_{\mathrm{ord}} + \lambda_{\mathrm{pix}}\mathcal{L}_{\mathrm{pix}} + \lambda_{\mathrm{tail}}\mathcal{L}_{\mathrm{tail}}. \tag{5}$$

The auxiliary losses are used only during training. At inference, a grammar constraint restricts the model to legal coordinate tokens and output structures; it does not provide geometric information or post-hoc coordinate correction.

4 Experiments

This section presents three experiments: fine-grained geometric visual perception, downstream Image2Code reconstruction, and training-data ablation with primitive-type analysis.

4.1 Experimental setup

Training. All experiments use Qwen3-VL-4B as the base model. Training uses AdamW, gradient accumulation, bfloat16, gradient checkpointing, and SDPA; images are resized to 448×448. The complete learning-rate, regularization, clipping, batch, microbatch, and accumulation settings are provided in the appendix.

Baselines. We compare the base Qwen3-VL-4B, our model, classical geometric algorithms, and API models including GPT-5.6-terra, Gemini-3.6-flash, GLM-5.3-flash, GLM-5.3, Qwen3.8-flash, and Qwen3.8-max. Classical methods use TLS/PCA endpoint fitting for line segments, least-squares circle fitting for circles and arcs, classical ellipse fitting for ellipses, and skeleton tracking followed by resampling for irregular curves. They are evaluated only on clean-primitive tasks; other cells are marked with dashes.

Metric. Average pixel difference is the sole core metric for the fine-grained perception experiments. For predicted points $\hat{p}_i$ and corresponding targets p_i,

$$D_{\mathrm{pixel}} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{p}_i - p_i\|_2, \tag{6}$$

where N is the number of corresponding target points. Lower values indicate more accurate localization. Image2Code additionally reports VGF (visual comparison score), CD, HD, and SSIM. VGF is a visual comparison score assigned by an evaluation model comparing the rendered reconstruction with the target image; higher values indicate better visual agreement.

4.2 Fine-grained geometric visual perception

We evaluate on three test sets. The FGGVP-22K synthetic test split is generated independently from the construction pipeline with a different random seed and contains approximately 1,000 samples. The Image2Code test set is based on Geo-Code and contains 1,172 samples. PGDP5K is a fully external test set. Each dataset reports three columns aligned with the capability levels: Clean denotes fine-grained positional perception, Complex denotes semantic perception in a composed scene, and Specified denotes derived-object perception for a relation-defined target.

Table 2: Fine-grained geometric visual perception results. Clean, Complex, and Specified correspond respectively to fine-grained positional perception, semantic perception, and derived-object perception. Values are average pixel differences (px; lower is better). Classical methods are evaluated only on clean primitives; dashes indicate unreported settings.

	FGGVP-22K			Image2Code			PGDP5K		
Method	Clean	Complex	Specified	Clean	Complex	Specified	Clean	Complex	Specified
Classical algorithms	0.475	–	–	0.592	–	–	0.583	–	–
Qwen3-VL-4B base	46.29	52.58	58.41	73.40	80.06	76.59	88.06	90.30	82.35
GPT-5.6-terra	6.51	5.65	5.46	5.57	6.22	8.47	5.75	6.10	6.36
Gemini-3.6-flash	6.24	5.96	5.63	5.68	5.70	6.89	5.71	6.46	6.74
GLM-5.3-flash	8.04	9.68	10.03	8.13	8.00	9.42	8.47	7.73	8.69
GLM-5.3	7.34	8.81	9.43	6.10	8.66	10.18	6.98	9.23	12.44
Qwen3.8-flash	7.50	62.68	10.35	6.53	11.00	29.90	7.85	12.32	8.75
Qwen3.8-max	6.56	6.57	22.71	6.51	6.68	29.57	6.75	6.82	7.74
Ours	1.386	1.827	1.331	4.389	3.311	2.390	4.467	2.867	3.561

Compared with the base model, our method achieves lower average pixel error on every reported cell. On FGGVP-22K, errors decrease from 46.29, 52.58, and 58.41 px to 1.386, 1.827, and 1.331 px. Errors on Image2Code and PGDP5K also remain below 4.5 px for all three task types. The improvement at the same 4B scale shows that targeted supervision of geometric positions and coordinate outputs directly improves fine-grained localization.

The API models display large variation across datasets and query types. Strong visual description and semantic association do not automatically yield stable pixel-level coordinates; dedicated supervision for coordinates, trajectories, and semantic targets is therefore useful. Classical fitting is more accurate on clean primitives, with errors of 0.475, 0.592, and 0.583 px. This advantage reflects the precision of specialized fitting once the target pixels are isolated. Such fitting does not by itself resolve natural-language references, component selection, or relation-defined targets such as the foot of a perpendicular or an intersection. Extending it to those settings would require additional semantic parsing, object selection, and geometric construction mechanisms.

4.3 Downstream Image2Code

Following the data and evaluation protocol of Geo-Code, we evaluate geometric reconstruction with VGF (visual comparison score), Chamfer Distance (CD), Hausdorff Distance (HD), and SSIM. VGF is a visual comparison score assigned by an evaluation model comparing the rendered reconstruction with the target image; higher values indicate better visual agreement. VGF and SSIM are higher-is-better; CD and HD are lower-is-better.

Table 3: Image2Code reconstruction results.

Method	VGF ↑	CD ↓	HD ↓	SSIM ↑
Geo-Code	95.65	9.836	86.092	0.8520
GeoCodeLM	69.58	13.625	57.625	0.6742
Ours	97.24	3.640	42.380	0.9560

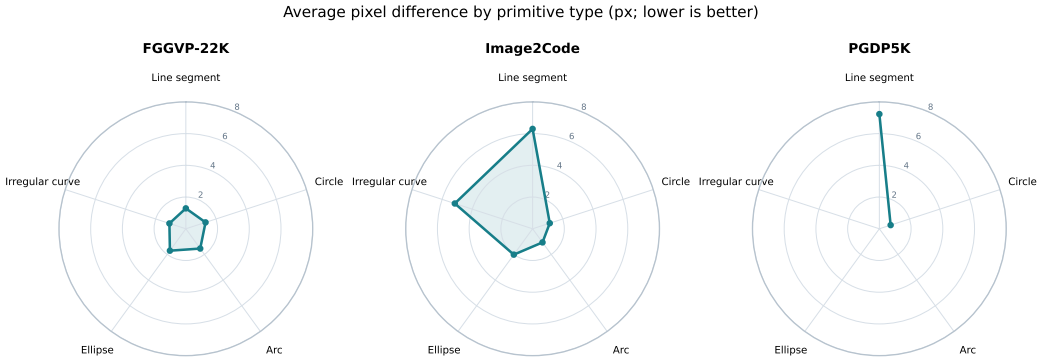

Figure 4: Average pixel difference by primitive type (px; lower is better). Categories unavailable for PGDP5K are left blank.

Our method outperforms both Geo-Code and GeoCodeLM on all four metrics. Relative to Geo-Code, VGF increases by 1.59 points, CD decreases from 9.836 to 3.640, HD decreases from 86.092 to 42.380, and SSIM increases from 0.8520 to 0.9560. Although our model has only 4B parameters, its explicit optimization for fine-grained geometric perception produces a clear gain in Image2Code reconstruction.

4.4 Ablation and analysis

4.4.1 Training-data ablation

We compare the base model, training with synthetic data only, and the complete setting using both synthetic and real data.

Table 4: Training-data ablation results (average pixel difference, px; lower is better). Clean, Complex, and Specified denote fine-grained positional, semantic, and derived-object perception, respectively.

	FGGVP-22K			Image2Code			PGDP5K		
Setting	Clean	Complex	Specified	Clean	Complex	Specified	Clean	Complex	Specified
Base	46.29	52.58	58.41	73.40	80.06	76.59	88.06	90.30	82.35
+ synthetic data	33.29	43.65	99.80	47.80	71.01	74.16	35.18	66.86	75.26
Complete training	1.386	1.827	1.331	4.389	3.311	2.390	4.467	2.867	3.561

Adding synthetic data lowers the error in eight of nine cells, including a decrease from 88.06 to 35.18 px on PGDP5K clean primitives. The specified-component error on FGGVP-22K increases from 58.41 to 99.80 px, showing that synthetic-only training does not improve every capability uniformly. The complete setting obtains the lowest error in all nine cells, supporting the value of combining synthetic and real data.

4.4.2 Primitive-type analysis

Figure 4 reports average pixel differences for line segments, circles, arcs, ellipses, and irregular curves. The FGGVP-22K trace denotes its synthetic test split.

On FGGVP-22K, errors range from 1.089 to 1.708 px; irregular curves have the lowest error and ellipses the highest. On Image2Code, circle and arc errors are 1.138 and 1.062 px, while line-segment and irregular-curve errors are 6.297 and 5.158 px. On PGDP5K, the reported line-segment and circle errors are 7.231 and 0.748 px. The distribution differs across datasets: categories are relatively balanced on FGGVP-22K, whereas line segments and irregular curves are harder on real and external data. Circles remain comparatively accurate across datasets.

5 Limitations

The present study focuses on improving fine-grained geometric perception through supervised fine-tuning (SFT). We have not yet explored reinforcement learning, preference optimization, iterative feedback, or other potentially stronger post-training strategies that could further improve coordinate precision. The current pixel errors also remain non-negligible: several real and external categories still have errors of multiple pixels, and the primitive-level analysis reports errors above seven pixels for some PGDP5K categories. These results show clear progress over the base model while leaving substantial room for more precise recovery of boundaries and trajectories. Rasterization, antialiasing, stroke width, occlusion, and annotation conventions can additionally make image geometry non-identifiable in some cases; centerline targets should not be interpreted as thick-stroke boundary segmentation.

6 Conclusion

We studied fine-grained geometric visual perception as the recovery of primitive positions, boundaries, and ordered trajectories in a shared canvas coordinate system. FGGVP-22K combines five-stage parameterized synthesis with real geometric scenes and semantic and derived-target annotations. Multitask replay with native coordinate tokens and coordinate-aware auxiliary losses substantially reduces average pixel error on synthetic, real, and external datasets and improves Image2Code reconstruction. Classical fitting is more accurate for isolated clean primitives, whereas the proposed model also handles semantic references and relation-defined targets.

Reproducibility Statement

The main text specifies the coordinate representation, training objective, data construction, and evaluation metrics. The appendix provides the synthetic-data pseudocode template, training settings, and prediction examples to support reproduction of the reported procedure.

AI Use Statement

Generative AI was used only to further polish the writing and language of this manuscript. The authors reviewed the revised text and remain responsible for all claims, methods, data, figures, and experimental results.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. In Advances in Neural Information Processing Systems, 2022. URL https://arxiv.org/abs/2204.14198.

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, Jingren Zhou, et al. Qwen-vl: A versatile vision-language model for understanding, localization, text reading, and beyond. arXiv preprint arXiv:2308.12966, 2023. URL https://arxiv.org/abs/2308.12966.

Alexandre Carlier, Martin Danelljan, Alexandre Alahi, and Radu Timofte. Deepsvg: A hierarchical generative network for vector graphics animation. arXiv preprint arXiv:2007.11301, 2020.

Keqin Chen, Zhao Zhang, Weili Zeng, Richong Zhang, Feng Zhu, and Rui Zhao. Shikra: Unleashing multimodal llm's referential dialogue magic. arXiv preprint arXiv:2306.15195, 2023.

Ting Chen, Saurabh Saxena, Lala Li, David J. Fleet, and Geoffrey Hinton. Pix2seq: A language modeling framework for object detection. In International Conference on Learning Representations, 2022.

Yihan Hao, Mingliang Zhang, Fei Yin, and Linlin Huang. Pgdp5k: A diagram parsing dataset for plane geometry problems. arXiv preprint arXiv:2205.09947, 2022. URL https://arxiv.org/abs/2205.09947.

Junwen He, Yifan Wang, Lijun Wang, Huchuan Lu, Jun-Yan He, Jin-Peng Lan, Bin Luo, and Xuansong Xie. Multi-modal instruction tuned llms with fine-grained visual perception. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2024. doi: 10.1109/CVPR52733.2024.01326.

Kung-Hsiang Huang, Can Qin, Haoyi Qiu, Philippe Laban, Shafiq Joty, Caiming Xiong, and Chien-Sheng Wu. Why vision language models struggle with visual arithmetic? towards enhanced chart and geometry understanding. arXiv preprint arXiv:2502.11492, 2025.

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.

Xin Lai, Zhuotao Tian, Yukang Chen, Yanwei Li, Yuhui Yuan, Shu Liu, and Jiaya Jia. Lisa: Reasoning segmentation via large language model. arXiv preprint arXiv:2308.00692, 2023.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Cheng-Hsin Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In International Conference on Machine Learning, 2023. URL https://arxiv.org/abs/2301.12597.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In Advances in Neural Information Processing Systems, 2023. URL https://arxiv.org/abs/2304.08485.

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. URL https://arxiv.org/abs/2105.04165.

Zhiliang Peng, Wenhui Wang, Li Dong, Yaru Hao, Shaohan Huang, Shuming Ma, and Furu Wei. Kosmos-2: Grounding multimodal large language models to the world. arXiv preprint arXiv:2306.14824, 2023.

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. arXiv preprint arXiv:2312.11556, 2023.

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

Haoran Wei, Youyang Yin, Yumeng Li, Jia Wang, Liang Zhao, Jianjian Sun, Zheng Ge, Xiangyu Zhang, and Daxin Jiang. Slow perception: Let's perceive geometric figures step-by-step. arXiv preprint arXiv:2412.20631, 2024.

Ronghuan Wu, Wanchao Su, and Jing Liao. Chat2svg: Vector graphics generation with large language models and image diffusion models. arXiv preprint arXiv:2411.16602, 2024.

Zhenyu Wu, Yanxi Long, Jian Li, and Hua Huang. Geo-code: A code framework for reverse code generation from geometric images based on two-stage multi-agent evolution. arXiv preprint arXiv:2602.07749, 2026. URL https://arxiv.org/abs/2602.07749.

Ming-Liang Zhang, Fei Yin, and Cheng-Lin Liu. A multi-modal neural geometric solver with textual clauses parsed from diagram. arXiv preprint arXiv:2302.11097, 2023. URL https://arxiv.org/abs/2302.11097.