THE CONTINUOUS SPACE GAP: WHY VLMS FAIL IN CONTINUOUS GEOMETRIC REASONING

Yikun Zong *
Department of Engineering
University of Cambridge
Cambridge, CB3 0DG, United Kingdom
{yz977}@cam.ac.uk

Cheston Tan
Centre for Frontier AI Research
A*STAR
138632, Singapore
{cheston-tan}@a-star.edu.sg

ABSTRACT

This paper presents a **clear negative result**: despite their success in discrete reasoning, Vision–Language Models (VLMs) **fail significantly** in continuous geometric reasoning. On the full benchmark, VLMs achieve only 0.41 IoU (baseline) on single-piece tasks and 0.23 on two-piece composition, far below human performance (Bohning & Althouse, 1997). Experiments across 5 state-of-the-art VLMs show that while test-time self-improvement through reward-guided refinement loops **does improve** predictions on single-piece cases, this refinement is **far from sufficient** to close the gap: refinement results (0.65→0.93 IoU) are reported on the *medium triangle* subset only (see appendix), and even there, performance remains below human level, gains do not reliably generalize, and multi-piece tasks would face even greater challenges. Thus, our negative result targets **VLMs' continuous-space reasoning ability**, not the existence of test-time refinement itself. We posit five underlying limitations, with the most critical being training distribution mismatch and output format constraints (see Section 5 for details), and document **boundary conditions** where refinement helps on single-piece tasks but saturates quickly, while multi-piece tasks remain far from human performance, indicating systematic rather than correctable errors. Our work is available at this anonymous link https://anonymous.4open.science/r/TangramVLM-F582/.

1 INTRODUCTION

Vision Language Models (VLMs) have achieved remarkable success in discrete reasoning tasks, from mathematical problem-solving (Cobbe et al., 2021) to code generation. However, their performance in *continuous geometric reasoning*, tasks requiring precise spatial alignment with metric precision, which remains largely unexplored. This gap is critical: real-world applications such as robotic manipulation, puzzle assembly, and spatial planning demand accurate coordinate predictions in continuous space, where small errors can lead to catastrophic failures. Unlike discrete reasoning where approximate answers may suffice, continuous geometric tasks require exact spatial relationships: positions must align within pixel-level precision, angles must match within degrees, and scales must preserve relative proportions.

Tangram puzzle assembly, where small positional/angular deviations disrupt configurations, provides an ideal testbed for evaluating continuous geometric reasoning (Shepard & Metzler, 1971; Bohning & Althouse, 1997; Yamada & Batagelo, 2017). We evaluate using explicit geometric metrics and test-time self-improvement through reward-guided refinement loops (Madaan et al., 2023; Shinn et al., 2023).

Despite success in discrete reasoning (Cobbe et al., 2021), experiments across five VLMs (Qwen (Bai et al., 2023), GPT-4o (Hurst et al., 2024; Islam & Moushi, 2025), LLaMA (Gao et al., 2023), Gemini, Claude) show **systematic failures**: on the full benchmark, **baseline** (zero-shot/few-shot) performance is only 0.41 IoU on single-piece tasks and 0.23 on two-piece composition, significantly low compared to human performance (Bohning & Althouse, 1997). **Refinement** (ICL +

*Corresponding author

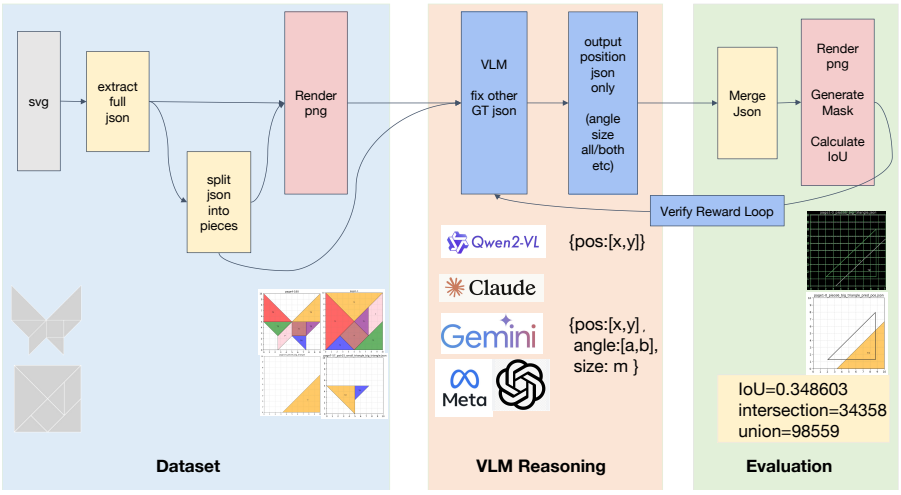


Figure 1: The diagram shows how SVG tangram are parsed into JSON annotations (type, position, angle, size), rendered into evaluation images, and split into single/two-piece, or full-tangram subsets.

reward loop on the medium triangle subset; see Table 3 in appendix) improves from 0.65 to 0.93 IoU on that subset, but this **cannot bridge the fundamental gap**: performance remains below human level, improvements fail to generalize across diverse configurations, and refinement saturates within 6 iterations; multi-piece tasks would face even greater challenges.

2 Related Work

Failures in spatial reasoning and VLM limitations. Prior work on spatial reasoning in multimodal systems shows systematic limitations rather than successes. Benchmarks like Winoground (Thrush et al., 2022) and large-scale evaluations (Ma et al., 2024; Stogiannidis et al., 2025) document failures in complex spatial settings, while specialized models like SpatialVLM (Chen et al., 2024) still struggle with continuous geometric tasks. Critically, most protocols reduce spatial reasoning to discrete judgments (e.g., multiple choice), without measuring *continuous* geometric errors—precisely the failure modes we expose. Tangram assembly (Bohning & Althouse, 1997; Yamada & Batagelo, 2017) provides a classical testbed for spatial cognition, but prior AI work emphasizes symbolic inference in discrete settings, whereas our evaluation operates in *continuity space*, explicitly quantifying errors in position, angle, and size.

Limitations of test-time self-improvement. Recent work on self-improving AI explores test-time adaptation (Sun et al., 2020), in-context learning with feedback loops (Madaan et al., 2023), and reward-guided refinement. Methods like ReAct (Yao et al., 2022) and Reflexion (Shinn et al., 2023) show improvements in discrete reasoning domains, but their success does not generalize to continuous geometric reasoning. Our work shows that extending this paradigm to *continuous geometric reasoning* exposes fundamental limitations: while reward-based feedback can iteratively refine spatial predictions, improvements are bounded and fail to reach human-level performance, indicating boundary conditions that prior work did not expose.

3 Methodology

DATASET AND TASKS. We build a Tangram benchmark with two splits: **single-piece** (one canonical piece with GT $(\mathbf{p}, \alpha, s)$) and **two-piece** (two pieces requiring mutual non-overlap coverage). Each sample is annotated in JSON format (`type`, `pos`$= [x, y]$, `angle`, `size`> 0) and rendered from canonical templates to enable exact polygon-level IoU computation (see Algorithm 2 in appendix). We design four tasks: **pos-only** (predict $\hat{\mathbf{p}}$), **angle-only** (predict $\hat{\alpha}$), **size-only** (predict $\hat{s}$), and **two-piece arrangement** (predict $(\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2)$). We also include a **joint** setting predicting all three fields simultaneously to expose compounding errors.

MODELS, INFERENCE PROTOCOL, AND METRICS. We evaluate four Vision–Language Models (VLMs): Qwen-3B, Qwen-72B, GPT-4o mini, and LLaMA Maverick. Each model receives a Tangram silhouette image and is prompted to output a minimal JSON containing the requested field(s): position (`pos`), orientation (`angle`), or scale (`size`). All predictions are evaluated in a normalized $[0, 1]^2$ coordinate frame. Geometric consistency is assessed using `geometry`, which computes Euclidean position error, angular deviation, and scale difference, and `overlay`, which renders predicted and ground-truth polygons to measure intersection-over-union (IoU). We report rasterized IoU on a 512×512 canvas with 1–2 px dilation; for two-piece assembly, IoU is computed over the union of both pieces. We report results under both zero-shot and few-shot ICL settings (typically $k = 15$), following a unified inference and evaluation protocol across all models. The full evaluation pipeline is given in Algorithm 1.

Algorithm 1 Evaluation Pipeline

1: **Input:** `IN_DIR` (PNGs), `GT_DIR` (JSONs), `OUT_DIR`, model, mode
2: **Output:** Pred JSONs, metrics (L2/angle/size/IoU), visualizations
3: **for all** $I \in$ `IN_DIR` **do**
4: $G \leftarrow$ pair JSON; `two` $\leftarrow$ IsTwoPiece(G)
5: Predict: call model; parse & validate by `mode`
6: Compute metrics (if G exists): L2/angle/size/IoU
7: Render GT/PRED/OVERLAY; save outputs
8: **end for**

Self-Improvement Loop via Reward-Guided Refinement. For a single tangram piece with pose $\Theta = (\mathbf{p}, \alpha, s)$, we use a reward that trades off geometric coverage (IoU) against position error:

$$\mathcal{R}(\Theta) = \mathrm{IoU}(\mathcal{U}(\Theta),\, S) - \lambda \cdot \frac{\|\hat{\mathbf{p}} - \mathbf{p}\|_2}{10}. \tag{1}$$

Here $\hat{\mathbf{p}}$ is the GT position, the canvas side length is 10 (hence the division by 10 for normalization), and $\lambda > 0$ is a small weight. *No* overlap penalty, edge-shape term, angle/scale regularizer, or global loss is used in our implementation. We run T iterations of self-refinement through ICL + feedback, treating the VLM as a proposal generator with a geometry-based verifier that guides iterative improvement at test time. The refinement loop is formalized in Algorithm 3 (appendix). Complete experimental details, including setup and results, are provided in Section A.4 (appendix).

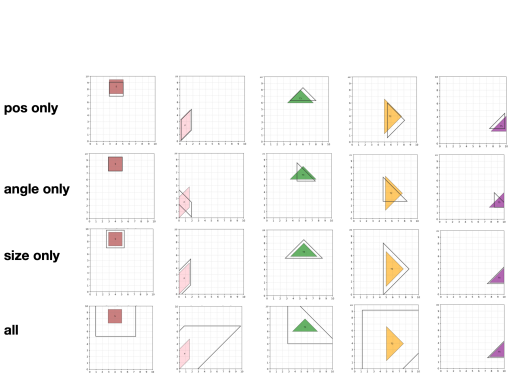
4 Results and Analysis

4.1 Part I: Cross-Model Comparison on Spatial Reasoning

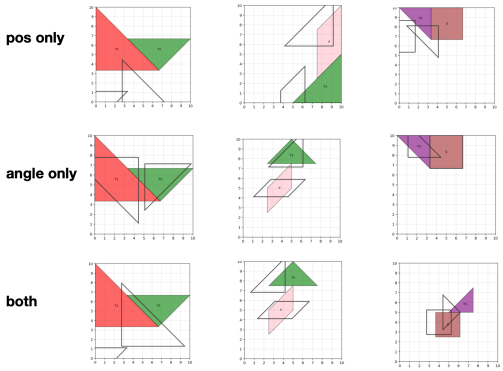
Setup. We evaluate *pos-only*, *angle-only*, *size-only*, and *joint (all)* predictions across five models: Qwen-3B, Qwen-72B, GPT-4o mini, LLaMA Maverick, and Gemini-2.5-pro. We report mean±95%CI over the test set and include equivariance stress-tests (rotation, mirror, and scale). Unlike prior sections, we unify cross-model and factorized testing into a single comparison table, where each row is a model and each column corresponds to a prediction task. Metrics reported are task-specific errors (L2, angular degrees, relative scale) and IoU (higher is better).

Table 1: Unified results for VLM one-piece spatial reasoning (↑ higher is better). For the human baseline, humans can complete tangram tasks even in childhood, demonstrating significantly high continuous spatial reasoning ability (Bohning & Althouse, 1997).

Method	Pos IoU ↑	Angle IoU ↑	Size IoU ↑	All IoU ↑
Claude-Sonnet-4	0.419	0.394	0.372	0.395
Gemini-2.5-pro	**0.443**	**0.434**	**0.432**	**0.417**
GPT-4o mini-8B	0.427	0.429	0.393	0.413
LLaMA Maverick 17B	0.424	0.427	0.371	0.377
Qwen-3B	0.236	0.414	0.369	0.219
Qwen-72B	0.415	0.432	0.425	0.408



(a) Single-piece setup and outputs: position, angle, size, and all.



(b) Two-Piece setup and outputs: position, angle and both.

Figure 2: Spatial reasoning tasks: single-piece and two-piece Tangram assembly.

Table 2: Unified results for VLM two-piece spatial reasoning (↑ higher is better). Human baseline follows the same reference as Table 1, that human perform well in these tasks

Model	Pos IoU ↑	Angle IoU ↑	All IoU ↑
Claude-Sonnet-4	0.318	0.394	0.235
Gemini-2.5-pro	**0.340**	0.397	0.340
GPT-4o mini	0.276	0.394	0.278
LLaMA Maverick	0.220	0.427	**0.371**
Qwen-3B	0.192	0.317	0.214
Qwen-72B	0.253	**0.495**	0.248

Findings. (i) Larger models reduce L2 and scale errors, but angle remains fragile across all models; (ii) joint prediction aggregates noise across multiple axes, amplifying errors versus factorized tasks; (iii) IoU is highly sensitive to angular mismatch even when position errors are small.

4.2 Part II: Spatial Arrangement (Two-Piece Composition)

Setup. We test *arrangement* with two pieces. We consider three modes: (A) fix both (α, s), predict $(\mathbf{p}_1, \mathbf{p}_2)$; (B) fix $\mathbf{p}, s$ and predict *angles*; (C) predict positions + angles jointly (scaled fixed). Metrics: union IoU and overlap penalty.

Findings. Two-piece tasks, being inherently more complex, show even worse baseline performance (0.23 IoU vs 0.41 for single-piece). Given that refinement on single-piece tasks already shows fundamental limitations (saturating at 0.93 IoU, still below human level), multi-piece tasks would face even greater challenges even with refinement. Typical failure modes: mutual collision, near-miss adjacency, and mirrored angles (see Figure 2).

5 Discussion

We posit five fundamental reasons why current VLMs cannot close the continuous-space gap. (1) **Training distribution mismatch**: models are trained on discrete semantic tasks without continuous coordinate supervision, encouraging pattern matching instead of true geometric computation. (2) **Output format constraints**: autoregressive decoders represent coordinates as text strings, with no inherent notion of metric space or Euclidean distance. (3) **Visual encoder geometric invariance**: visual encoders are optimized to be invariant to small geometric changes for classification, which conflicts with the need for precise geometric sensitivity. (4) **Limited positional precision**: positional encodings provide only coarse location signals, insufficient to distinguish near-perfect from slightly misaligned configurations. (5) **Lack of geometry-aware feedback and inductive bias**: training lacks geometric consistency rewards and biases such as rotation equivariance, creat-

ing a ceiling that iterative feedback cannot overcome. These factors define **boundary conditions**: refinement helps on single-piece tasks but saturates within a few iterations, while multi-piece compositions remain far from human performance, indicating systematic rather than correctable errors. **Limitations.** Refinement experiments are limited to the medium triangle subset; evaluation covers one- and two-piece configurations rather than full seven-piece assembly.

6 CONCLUSION

This work demonstrates a **fundamental limitation of current VLMs in continuous geometric reasoning**. While reward-guided refinement improves IoU on the medium triangle subset (0.65→0.93), these gains are **insufficient**: VLMs remain far below human-level accuracy, fail to generalize across diverse configurations, and multi-piece tasks face even greater challenges. The gap is driven by **task complexity** (multi-piece compositions accumulate errors) and **saturation** (refinement plateaus regardless of iteration count), indicating systematic rather than correctable errors. These failures impact real-world applications requiring precise spatial reasoning, suggesting that test-time adaptation must be complemented with training-time geometry-aware supervision.

REFERENCES

Jinze Bai, Shuai Bai, Shusheng Yang, Shijie Wang, Sinan Tan, Peng Wang, Junyang Lin, Chang Zhou, and Jingren Zhou. Qwen-vl: A versatile vision-language model for understanding, localization. *Text Reading, and Beyond*, 2:1, 2023.

Gerry Bohning and Jody Kosack Althouse. Using tangrams to teach geometry to young children. *Early childhood education journal*, 24(4):239–242, 1997.

Boyuan Chen, Zhuo Xu, Sean Kirmani, Brain Ichter, Dorsa Sadigh, Leonidas Guibas, and Fei Xia. Spatialvlm: Endowing vision-language models with spatial reasoning capabilities. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14455–14465, 2024.

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

Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al. Llama-adapter v2: Parameter-efficient visual instruction model. *arXiv preprint arXiv:2304.15010*, 2023.

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

Raisa Islam and Owana Marzia Moushi. Gpt-4o: The cutting-edge advancement in multimodal llm. In *Intelligent Computing-Proceedings of the Computing Conference*, pp. 47–60. Springer, 2025.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, et al. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *Advances in Neural Information Processing Systems*, 37:95963–96010, 2024.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback, 2023. *URL https://arxiv. org/abs/2303.17651*, 2023.

Roger N Shepard and Jacqueline Metzler. Mental rotation of three-dimensional objects. *Science*, 171(3972):701–703, 1971.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652, 2023.

Ilias Stogiannidis, Steven McDonagh, and Sotirios A Tsaftaris. Mind the gap: Benchmarking spatial reasoning in vision-language models. *arXiv preprint arXiv:2503.19707*, 2025.

Yu Sun, Xiaolong Wang, Zhuang Liu, John Miller, Alexei Efros, and Moritz Hardt. Test-time training with self-supervision for generalization under distribution shifts. In *International conference on machine learning*, pp. 9229–9248. PMLR, 2020.

Tristan Thrush, Ryan Jiang, Max Bartolo, Amanpreet Singh, Adina Williams, Douwe Kiela, and Candace Ross. Winoground: Probing vision and language models for visio-linguistic compositionality. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5238–5248, 2022.

Fernanda Miyuki Yamada and Harlen Costa Batagelo. A comparative study on computational methods to solve tangram puzzles. In *Workshop of Works in Progress (WIP) in the 30th Conference on Graphics, Patterns and Images (SIBGRAPI'17)*, 2017.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*, 2022.

A Appendix

A.1 Figures

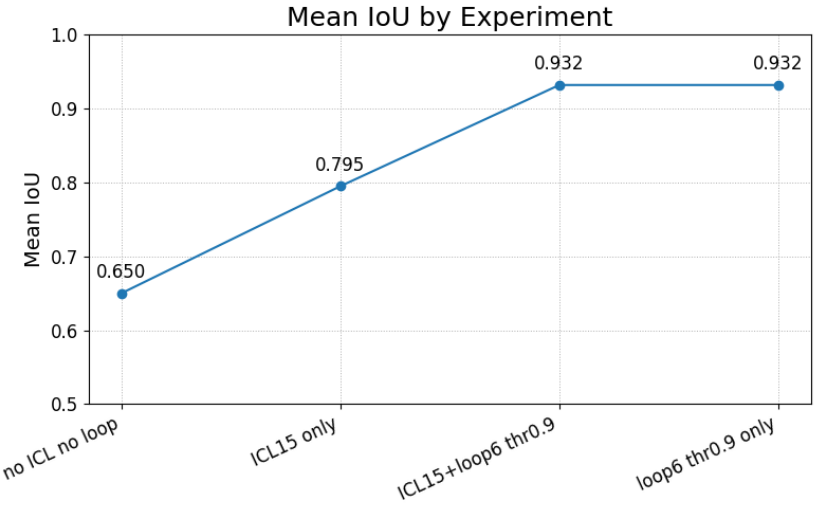


Figure 3: Mean IoU across ablations on the *medium triangle*. The test-time self-refinement loop (ICL + reward) yields the largest gain.

A.2 Dataset Construction Pipeline

Algorithm 2 Tangram Dataset Pipeline (SVG → JSON → PNG)

Input: Directory of SVG files (`IN_SVG_DIR`)
Output: JSON annotations and optional rendered PNGs
for all `svg_path` ∈ `IN_SVG_DIR` **do**
    Parse SVG into polygon list (`polys`, W, H)
    Fit polygons to canonical tangram templates
    Save piece parameters as JSON (`pos`, `angle`, `flip`, `scale`)
    if rendering enabled **then**
        Render shapes via geometry engine and save PNG
    end if
    if aligned outline available **then**
        Compute IoU between rendered union and outline
    end if
end for
return dataset (`JSON`, `PNG`, optional IoU logs)

A.3 Refinement Loop Algorithm

Algorithm 3 VLM + ICL + Reward Loop (simplified)

Input: image I, model M, mode $\in \{\text{pos}, \text{angle}, \text{size}, \text{all}\}$, ICL size k, loop iters T, threshold τ
Output: best JSON prediction $J^\star$, best IoU
$\mathcal{S} \leftarrow$ sample k few-shot (image, JSON) pairs for ICL
Initialize best $\leftarrow$ (iou $= 0,\ J = \varnothing$)
for $t = 1$ to T **do**
    Query M with I + $\mathcal{S}$ + refinement hint
    Parse output $\rightarrow J_t$ (JSON fields)
    Compute $\text{iou}_t = \text{IoU}(J_t,\ G)$
    if $\text{iou}_t >$ best.iou **then**
        best $\leftarrow (J_t,\ \text{iou}_t)$
    end if
    if best.iou $\geq \tau$ **then**
        break
    end if
end for
Optionally run small local search around best.pos
return $J^\star =$ best.J, best.iou

A.4 Test-Time Self-Improvement via Reward-Guided Refinement

Setting. For a single tangram piece, the pose is $\Theta = (\mathbf{p}, \alpha, s)$ with position $\mathbf{p} \in [0, 10]^2$, angle α (deg), and size $s > 0$. Let $\mathcal{U}(\Theta)$ be the rendered polygon from the canonical template under $(\mathbf{p}, \alpha, s)$, and S the ground-truth polygon.

Reward. We use a scalar reward that trades off geometric coverage (IoU) against position error:

$$\mathcal{R}(\Theta) = \text{IoU}(\mathcal{U}(\Theta),\ S) - \lambda \cdot \frac{\|\hat{\mathbf{p}} - \mathbf{p}\|_2}{10}. \tag{2}$$

Here $\hat{\mathbf{p}}$ is the GT position, the canvas side length is 10 (hence the division by 10 for normalization), and $\lambda > 0$ is a small weight. *No* overlap penalty, edge-shape term, angle/scale regularizer, or global loss is used in our implementation.

Self-refinement loop mechanics (training-free). We run T iterations of self-refinement through ICL + feedback: (i) build k few-shot pairs *excluding* the current sample; (ii) query the VLM with a minimal JSON instruction, appending numeric feedback hints (e.g., "`previous IoU=$x.xx. Try a small correction` $(\Delta x, \Delta y)$`.`") from the second iteration onward; (iii) keep the

candidate with the highest $\mathcal{R}$ in Eq. equation 2, with early stop once $\text{IoU} \geq \tau$. If $\text{IoU} < \tau$ and the task involves position, we perform a tiny grid search around the current best (x, y) using a 3×3 neighborhood at step sizes $0.6 \to 0.3 \to 0.15$ (canvas units), accepting the first move that increases IoU. This *training-free* loop treats the VLM as a proposal generator with a geometry-based verifier that guides iterative improvement at test time. Visual examples are shown in Figure 2.

Setup. We focus on the *medium triangle* subset (single-piece) for refinement experiments, starting from the VLM's JSON output and running T self-refinement loop iterations with reward $\mathcal{R}$. Refinement is conducted on this subset only (rather than the full benchmark) due to its representativeness for single-piece tasks and computational tractability; full-benchmark refinement remains future work. We allow a tiny local search over $\mathbf{p}$ at the end if IoU remains low.

Table 3: Medium triangle IoU across different settings (baseline start = 0.65).

Setting Number	Description	ICL (k)	Loop	Threshold	Temp.	IoU (final)
1	VLM + ICL + Loop	15	6	0.9	0	**0.9320**
2	VLM + Loop	n/a	6	0.9	0	0.9320
3	VLM + ICL + Loop	20	6	0.9	0	0.9300
4	VLM + ICL	15	n/a	n/a	0	0.7950
5	VLM + ICL + temp	15	n/a	n/a	0.5	0.7690
6	VLM only	n/a	n/a	n/a	0	0.6500

Findings. Table 3 shows the refinement loop achieves IoU gains (0.65→0.93) **on the medium triangle subset**, but these improvements **fail to generalize** across diverse geometric configurations. Refinement saturates within 6 iterations (see Figure 3 in appendix), reaching a ceiling of ≈0.93 IoU, **falling short of human-level performance**, significantly low compared to human performance (Bohning & Althouse, 1997). **Boundary conditions**: refinement saturates regardless of iteration count, indicating systematic rather than correctable errors. For loop parameter sensitivity, performance saturates within 2–6 iterations. Lower thresholds (τ=0.5) cause steep IoU decline (0.72–0.84), while higher thresholds (τ=0.9) stabilize updates. Additional iterations offer diminishing returns, indicating accuracy depends mainly on threshold choice rather than loop depth.

A.5 Detailed Ablation Results

Table 4: Ablation on loop count and threshold (keep ICL = 15 fixed). Baseline IoU = 0.65.

Setting	Description	Loop	Threshold	IoU
1	ICL + Loop	6	0.9	**0.9320**
7	ICL + Loop	4	0.9	0.9287
8	ICL + Loop	2	0.9	0.9291
9	ICL + Loop	6	0.5	0.8410
10	ICL + Loop	4	0.5	0.8609
11	ICL + Loop	2	0.5	0.7200
12	ICL + Loop	6	0.8	0.9310
13	ICL + Loop	6	0.7	0.9233
14	ICL + Loop	6	0.6	0.9063
15	ICL + Loop	8	0.9	0.9345
16	ICL + Loop	10	0.9	0.9258
17	ICL + Loop	12	0.9	0.9323

Table 5: Ablation on ICL window size (k), keep loop and threshold constant.

ICL (k)	Loop	Threshold	IoU (final)
15	8	0.90	0.9345
20	8	0.90	0.9311
25	8	0.90	0.9310

B. THE USE OF LLMS

We use llm to check and correct grammar and spelling mistakes. In addition, we also use llm to polish the sentences in our paper to make them more fluent.