TANGRAMSR: A BENCHMARK FOR RECURSIVE SELF-IMPROVEMENT 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

Vision–Language Models (VLMs) have achieved remarkable success on discrete multimodal benchmarks, yet struggle with continuous geometric reasoning tasks that require precise spatial alignment. This paper addresses a fundamental challenge in **self-improving AI**: how can models iteratively refine their predictions *at test time* without parameter updates? We introduce a **test-time self-refinement framework** that combines in-context learning with reward-guided feedback loops to enable VLMs to improve geometric alignment. Our approach operates on Tangram puzzle assembly, a mathematically rigorous, NP-hard (Yamada et al., 2024) shape arrangement task requiring precise estimation of position, rotation, and scale. We establish a **continuous-space evaluation benchmark** that decomposes geometric reasoning into factorized subtasks (position, angle, size) and measures performance using ℓ_2 distance and polygonal intersection-over-union (IoU). Comprehensive experiments across five representative VLMs reveal systematic performance gaps (average IoU 0.41 on single-piece tasks, dropping to 0.23 on two-piece composition). Our **training-free verifier–refiner agent** applies **recursive refinement loops** that iteratively self-refine predictions based on geometric consistency feedback under oracle access to ground-truth polygons, achieving IoU improvements from 0.63 to 0.932 on medium-triangle cases *without any model retraining*. This demonstrates that **oracle-guided test-time self-refinement** can substantially enhance geometric reasoning in VLMs, moving self-improving AI from promise to practice in continuous spatial domains. Our work is available at this anonymous link https://anonymous.4open.science/r/TangramVLM-F582/.

1 INTRODUCTION

When arranging pieces of a Tangram puzzle, even a small positional or angular deviation can disrupt the target configuration. The task requires reasoning in *continuous geometric space*, where success depends on accurately aligning orientations, adjusting fine-scale positions, and composing parts into a coherent whole (Shepard & Metzler, 1971). Despite their impressive performance on vision and language benchmarks (Radford et al., 2021), current Vision–Language Models (VLMs) still struggle in such settings.

This gap between discrete testing and continuous reasoning motivates our study. Building on the idea that geometry is inherently continuous, we design a benchmark where Tangram pieces must be precisely arranged to cover a target silhouette without overlap. Tangrams offer a controlled yet computationally challenging testbed (Shepard & Metzler, 1971; Bohning & Althouse, 1997; Yamada & Batagelo, 2017): the task is not only cognitively natural but also *NP-hard* (Yamada et al., 2024), requiring reasoning over combinatorial configurations in continuous space. It bridges perception and reasoning: models must not only recognize objects, but also understand how they fit together geometrically.

*Corresponding author

Test-time self-improvement as a path forward. Traditional approaches to improving VLM performance rely on retraining or fine-tuning, requiring computational resources and training data that may not always be available. However, recent advances in *self-improving AI* have demonstrated that models can enhance their capabilities *at test time* through iterative refinement, feedback loops, and reward-guided corrections. In continuous geometric reasoning, small errors in position, angle, or scale can compound to produce large failures, yet these errors are often correctable through feedback-driven adjustments. This suggests a natural role for **test-time self-refinement**: enabling models to iteratively improve their geometric predictions based on geometric consistency feedback, without requiring parameter updates or retraining. **While model parameters remain unchanged, changes occur in the ICL context and prompt content** as the refinement loop iteratively updates the in-context examples and feedback signals based on previous iterations, enhancing the model's geometric reasoning capabilities (tools and skills) through feedback-driven refinement at test time.

Our framework establishes a **test-time self-improvement pipeline** for VLMs operating in continuous geometric space, with applications in **scientific discovery** and robotics. First, it isolates fundamental geometric skills, estimating position, angle, and size using mathematically grounded metrics (ℓ_2 distance and IoU). Then, it escalates to two-piece arrangement, a compositional task that probes whether models can reason jointly about multiple parts. Through this lens, we uncover systematic failure modes across diverse models, Qwen (Bai et al., 2023), GPT-4o (Hurst et al., 2024; Islam & Moushi, 2025), LLaMA (Gao et al., 2023), Gemini, and Claude, showing that even frontier systems perform badly when geometry becomes continuous. **Crucially**, we demonstrate that a **recursive refinement loop** can iteratively correct these errors, progressively improving geometric alignment through feedback-guided iterations and achieving substantial IoU improvements without any model training. While symbolic approaches such as AlphaGeometry (Chervonyi et al., 2025) have reached Olympiad-level theorem proving, they operate in a discrete symbolic space of axioms and proofs. Our work, by contrast, situates reasoning directly in the continuous geometric world, where correctness is measured not by logic alone but by spatial alignment. We find that the gap between discrete success and continuous failure reveals a fundamental limitation of current VLMs: they can recognize geometric concepts, but struggle to compute and reason in a continuous, quantitative manner.

Contributions (1) We introduce a *math-grounded* evaluation benchmark for **VLM spatial reasoning** in continuous space, with explicit geometric metrics, and **release it as an open resource for the community**. (2) We conduct a *systematic evaluation* across leading VLMs and find consistent and severe geometric failures: **average IoU on single-piece tasks is only 0.41**, while **two-piece composition drops to 0.23**, revealing that even frontier systems cannot maintain geometric consistency in continuous space. (3) We propose a lightweight, **oracle-guided test-time self-refinement framework** that applies reward-guided refinement loops via ICL and geometric feedback to progressively enhance geometric alignment *without parameter updates*, achieving substantial IoU improvements (from 0.63 to 0.93 on medium-triangle tasks) at test time.

2 Related Work

Test-time self-improvement and refinement. Recent work on self-improving AI has explored various mechanisms for enhancing model performance without retraining, including test-time adaptation (Sun et al., 2020), in-context learning with feedback loops (Madaan et al., 2023), and reward-guided refinement. Methods that leverage iterative refinement, such as self-consistency decoding (Wang et al., 2022) and verifier-guided generation (Cobbe et al., 2021), demonstrate that models can improve their outputs through multiple rounds of generation and verification. In the context of reasoning tasks, approaches like ReAct (Yao et al., 2022) and Reflexion (Shinn et al., 2023) show that language models can refine their reasoning traces based on feedback, while methods in computer vision adapt representations at inference time. Our work extends this paradigm to *continuous geometric reasoning*, where we apply reward-based feedback (IoU) to iteratively refine spatial predictions through verifier–refiner loops at test time. Unlike prior work that focuses on discrete or symbolic reasoning, we demonstrate test-time self-improvement in a continuous metric space, where **our geometry verifier serves as instrumentation** to measure geometric consistency and provide feedback for iterative refinement.

Evaluation and benchmarks for self-improving systems. Evaluation frameworks play a crucial role in assessing self-improvement capabilities. Prior work has examined spatial reasoning in mul-

timodal systems through tasks such as mental rotation (Shepard & Metzler, 1971), spatial relation matching (Johnson et al., 2017), and compositionality (Wu et al., 2023; Hesham et al., 2025). Large-scale evaluations report degradation in long-context or complex spatial settings (Ma et al., 2024; Stogiannidis et al., 2025). Benchmarks like Winoground (Thrush et al., 2022), *Unfolding Spatial Cognition* (Li et al., 2025), and *SpatialVLM* (Chen et al., 2024) probe spatial reasoning capabilities, while RoboSpatial (Song et al., 2025) and GIQ (Michalkiewicz et al., 2025) evaluate geometric understanding for robotics and 3D reasoning. However, these protocols largely reduce spatial reasoning to symbolic or discrete judgments (e.g., multiple choice, caption matching), without measuring *continuous* geometric errors such as rotation angle drift or sub-pixel position offsets. In contrast, we introduce a *continuity-space evaluation benchmark* that explicitly measures self-improvement in continuous geometric reasoning, providing a testbed for evaluating refinement mechanisms.

Continuous geometric reasoning applications. Tangram is a classical tool for studying spatial visualization and compositional reasoning (Bohning & Althouse, 1997), with computational assembly being NP-hard (Yamada & Batagelo, 2017; Yamada et al., 2019). Recent AI work employs Tangram for abstract visual reasoning (Zhao et al., 2025; Lin et al., 2023; Li et al., 2026). (Li et al., 2026) formulate Tangram assembly as a video generation task, but pixel-based generation suffers from geometric inconsistencies. Unlike prior efforts emphasizing symbolic inference or implicit visual dynamics, our evaluation operates in *continuity space* with explicit coordinate-based optimization, explicitly quantifying errors in position, angle, and size. Modern VLMs such as Qwen (Bai et al., 2023), GPT-4o (Islam & Moushi, 2025), and LLaMA variants (Gao et al., 2023) show impressive language–vision capabilities, yet fundamental geometric invariances (rotation, scale) remain undertested, with models exhibiting brittle behavior under geometric transformations (Chen et al., 2024; Michalkiewicz et al., 2025; Huang et al., 2025). Our continuity-space evaluation tailored to Tangram shapes (i) isolates the estimation of position, angle, and size with geometry-aware metrics, and (ii) escalates to two-piece arrangement where error accumulation becomes salient, motivating robust spatial competence for embodied and robotic applications (Song et al., 2025; Duan et al., 2022).

3 METHODOLOGY

3.1 METRICS AND INFERENCE PROTOCOL

We report the rasterized intersection-over-union (IoU) between predicted and ground-truth shapes on a 512×512 canvas, using 1–2 px dilation for morphology-tolerant evaluation. All silhouettes are rendered from continuous, parameterized polygons, so raster IoU serves as a close approximation to polygon-level overlap in continuous space. For two-piece assembly, IoU is computed over the union of both pieces to capture overlap penalties. 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 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 loop is summarized in Algorithm 3 in the Appendix.

3.2 DATASET AND TASKS (CONTINUITY-SPACE PROTOCOL)

Each dataset sample is annotated in JSON format with fields `type`, `pos`$= [x, y]$, `angle` (in degrees), and `size` ($s > 0$). All shapes are rendered from canonical templates rather than segmented images, eliminating boundary noise and enabling exact polygon-level IoU computation. We build a Tangram benchmark with two splits: **Single-piece**: each image contains a *single* canonical piece placed with GT $(\mathbf{p}, \alpha, s)$. This supports pos/angle/size decoupling. **Two-piece**: each image contains two pieces whose relative arrangement matters (mutual non-overlap coverage). We design four tasks that progressively increase geometric difficulty: **pos-only**: fix GT (α, s), predict $\hat{\mathbf{p}}$; **angle-only**: fix GT $(\mathbf{p}, s)$, predict $\hat{\alpha}$; **size-only**: fix GT $(\mathbf{p}, \alpha)$, predict $\hat{s}$; **two-piece arrangement**: fix each piece's (α, s), predict two positions $(\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2)$, and assess composition. We also include a **joint** setting that predicts all three fields simultaneously to expose compounding errors. The complete dataset construction pipeline is detailed in Algorithm 2 (see Appendix). Overall, the benchmark consists of

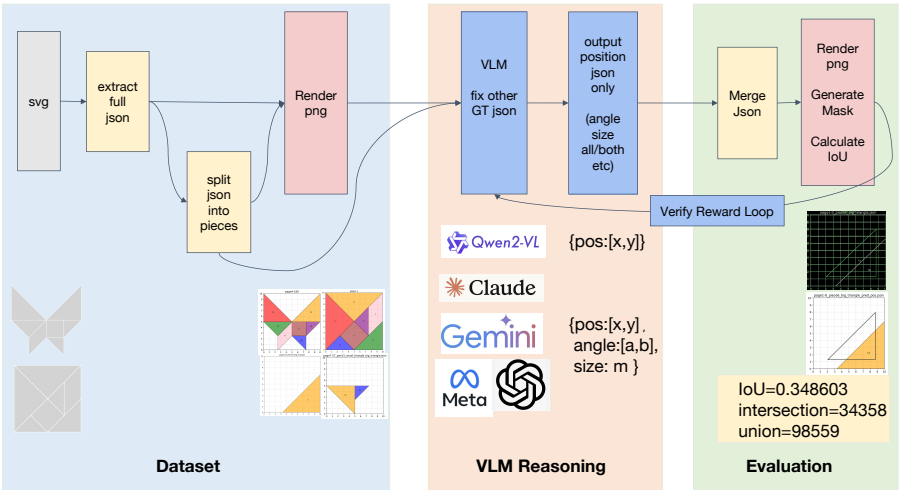


Figure 1: Overall dataset construction pipeline from SVG → JSON → PNG → split tasks. The diagram shows how raw SVG tangram silhouettes are parsed into JSON annotations (type, position, angle, size), rendered into training/evaluation images, and split into single-piece, two-piece, or full-tangram subsets.

synthetic single-piece and two-piece samples over canonical Tangram shapes, with positions sampled in a normalized $[0, 1]^2$ frame, angles spanning $[0, 360)$, and sizes drawn from a fixed range; full per-task counts and splits are documented in the released code. This moderate-scale, synthetic design is deliberate: it trades raw dataset size for a clean, mathematically controlled diagnostic of continuous-space error modes and refinement dynamics.

3.3 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 (what we actually optimize). We use a scalar reward that trades off geometric coverage (IoU) against position error only:

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

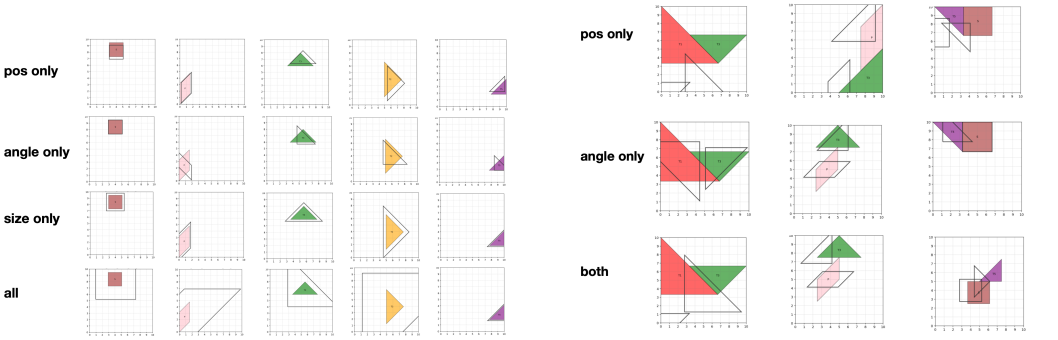
Here $\hat{\mathbf{p}}$ is the GT position, the canvas side length is 10 (hence the division by 10 for normalization), and $\lambda > 0$ is a small weight. *No* overlap penalty, edge-shape term, angle/scale regularizer, or global loss is used in our implementation. This reward explicitly depends on access to the ground-truth polygon S, so the refinement loop is *oracle-guided* in our benchmark setting. In principle, the same mechanism could instead use a learned critic or environment-derived feedback when ground-truth shapes are unavailable.

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 for the desired field(s), and from the second iteration onward append a numeric feedback hint of the form "`previous IoU=$x.xx. Try a small correction` $(\Delta x, \Delta y)$`.`"; (iii) keep the candidate with the highest $\mathcal{R}$ in Eq. equation 1, with early stop once $\mathrm{IoU} \geq \tau$.

Deterministic local refinement. If $\mathrm{IoU} < \tau$ and the task involves position (`pos` or `all`), 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). We accept the first move that increases IoU and update the best candidate; finally we recompute the reward $\mathcal{R} = \mathrm{IoU} - \lambda \cdot (\mathrm{L2}/10)$.

Algorithm 1 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 $\leftarrow (\text{iou} = 0,\ J = \varnothing)$
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) 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.

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).

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.

Table 1: Unified results for VLM one-piece spatial reasoning (↑ higher is better). VLM performance is relatively low compared to human performance (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

Table 2: Unified results for VLM two-piece spatial reasoning (↑ higher is better). VLM performance is relatively low compared to human performance (Bohning & Althouse, 1997).

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. Arrangement is *significantly* harder than single-piece: IoU drops $\sim$0.3 even when single-piece IoU exceeds 0.7. Typical failure modes: mutual collision, near-miss adjacency, and mirrored angles producing plausible shapes but wrong unions, as shown on figure 1.

4.3 Part III: Test-Time Self-Improvement via Reward-Guided Refinement

Setup. We focus on the *medium triangle* subset (single-piece), start from the VLM's JSON output, run T self-refinement loop iterations with reward $\mathcal{R}$. 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 compares six configurations (Settings 1–6), illustrating how ICL, loop refinement, and temperature jointly affect performance. Figure 3 in Appendix A.4 visualizes mean IoU across these ablations on the medium triangle.

- **VLM only vs. ICL (Setting 6 vs. Setting 4).** The plain VLM baseline (0.65) improves to 0.795 with *ICL15*, yielding an absolute gain of **+0.145**. However, raising the temperature to 0.5 (Setting 5) slightly degrades IoU to 0.769 (**–0.026**), suggesting that while ICL stabilizes generation, excessive sampling diversity introduces numerical noise that harms geometric consistency.
- **VLM only vs. Loop only (Setting 6 vs. Setting 2).** Running the geometry-based **Loop** (6 iterations, threshold 0.9) elevates IoU from 0.65 to **0.932** (**+0.282**), far exceeding the ICL-only gain. This demonstrates that the *test-time self-refinement loop* effectively corrects small positional and angular errors that token-based generation cannot fix.
- **ICL + Loop comparison (Settings 1–3, 15–17).**

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

Setting Number	Description	Loop	Threshold	IoU (final)
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

Combining ICL with the loop (Setting 1, 0.932) achieves nearly the same performance as Loop only (Setting 2, 0.932), while a larger ICL window (k=20, Setting 3) yields slightly lower IoU (0.930). When extending the loop length further (Settings 15–17, with 8–12 iterations), we observe that IoU improvements become marginal: 0.9345, 0.9258, and 0.9323 respectively. This indicates that the refinement process quickly saturates, as most geometric errors are already corrected within six iterations. In our analysis, we consider an IoU increase above 0.01 as statistically significant; thus, we treat 0.932 as the effective performance ceiling under the current framework.

- **General trend.** Comparing Settings 1–6 collectively, the **Loop mechanism contributes the majority of improvement**, while ICL provides consistent but smaller gains. Overly high temperature or large ICL windows introduce instability, confirming that precision feedback, not randomness, is key to continuous geometric refinement. Furthermore, as shown in Table 5, when the loop count and threshold are held constant, increasing the ICL window size (k) from 15 to 25 does not yield further improvement; instead, IoU slightly decreases from 0.9345 to 0.9310. This suggests that excessively large ICL contexts may introduce redundancy or noise, and that moderate exemplars (k=15) strike a more effective balance between stability and precision.

Takeaways. (1) The **self-refinement loop** contributes more than ICL overall, acting as a geometry-aware *test-time improvement mechanism* correcting small continuous errors; (2) **ICL+Loop** reaches the same ceiling as *Loop only*, while improving initialization and convergence stability; (3) **Excessive ICL** (larger k or higher temperature) introduces noise, causing slight degradation; (4) The recommended setting is **ICL** k=15 **+ Loop** =6 **+** τ=0.9 **+ temperature** =0, balancing stability and accuracy for effective test-time self-improvement.

Loop Parameter Sensitivity (Settings 7–14). Table 4 compares different loop counts and acceptance thresholds under constant ICL (k=15). By contrasting Settings 8 (2 loops, 0.9291), 7 (4 loops, 0.9287), and 1 (6 loops, 0.9320), we observe that performance quickly saturates after only a few refinement iterations, indicating that most geometric errors are corrected early in the process. Varying the acceptance threshold further highlights its decisive role in stability: comparing Settings 1 (τ=0.9, 0.9320) with 9 (τ=0.5, 0.8410) and 11 (τ=0.5, 0.7200), a steep IoU decline appears as the gating becomes more lenient. This degradation shows that low thresholds admit noisy or suboptimal corrections, breaking geometric consistency, whereas higher thresholds act as a filter that stabilizes

updates. Across Settings 12–14 (6 loops, τ decreasing from 0.8 to 0.6), the same trend persists: tightening the gate consistently improves stability while additional iterations beyond six offer diminishing returns. Overall, these results indicate that accuracy depends mainly on the threshold choice, while increasing loop depth yields limited additional benefits.

5 DISCUSSION

Results summary. Our evaluation across single-piece and two-piece Tangram assembly reveals that most VLMs achieve moderate IoU (0.2–0.45) on single-piece tasks, dropping to 0.23 on two-piece composition due to error compounding. **Oracle-guided test-time self-refinement** consistently improves alignment under access to ground-truth polygons, with loops converging within 1–2 iterations and achieving IoU gains of over 40% without parameter updates. The refinement is **recursive** because each iteration updates the ICL context and feedback signals based on previous outputs, forming a closed-loop process. Our reward-guided framework generalizes beyond Tangram to any task requiring continuous spatial alignment, including robotics manipulation and navigation (Song et al., 2025; Duan et al., 2022), where geometric precision is critical.

Despite strong performance on semantic benchmarks, VLMs perform poorly in continuous-space evaluation when operating in single-pass mode. We attribute this to three factors: (1) **Training distribution mismatch**: VLMs are trained on discrete semantic tasks, rarely encountering precise coordinate prediction; (2) **Output mismatch**: autoregressive decoders force continuous quantities into discrete tokens, causing rounding and numerical drift; (3) **Absence of geometry-aware feedback**: models lack geometric consistency supervision during training.

Oracle-guided refinement vs autonomous self-improvement. Our loop assumes access to ground-truth polygons in order to compute the reward in Eq. 1, so it is best viewed as test-time optimization with an oracle verifier in a controlled benchmark, rather than a fully autonomous self-improving agent. In more realistic settings, the same mechanism could instead rely on a learned critic or environment-derived feedback, replacing the oracle while preserving the refinement structure.

Continuous geometry beyond pixels. Although the reward is implemented as rasterized IoU on a 512^2 canvas, all shapes are generated from continuous, parameterized polygons, and we also report explicit errors in position, angle, and scale. The benchmark therefore probes continuous spatial reasoning rather than mere pixel-level matching. Our synthetic Tangram dataset is intentionally moderate in scale and complexity: this simplicity yields a clean, mathematically controlled diagnostic of geometric error modes and refinement dynamics, rather than a large-scale, noisy computer vision benchmark.

Connection to long-term self-improvement and evolutionary methods. While our framework focuses on test-time refinement, it can be extended to long-term self-improvement by accumulating successful strategies across episodes and adapting reward schedules based on performance. This connects to evolutionary approaches where successful patterns are preserved and refined over time, though operating in continuous rather than discrete space.

6 CONCLUSION AND FUTURE WORK

We presented a **test-time self-improvement framework** for vision–language models (VLMs) operating in continuous geometric space. By introducing reward-guided refinement loops, we demonstrate that VLMs can substantially enhance their geometric reasoning capabilities *without retraining*, achieving significant IoU improvements and moving self-improving AI from promise to practice in continuous spatial domains. Our continuous-space evaluation benchmark explicitly measures spatial reasoning through geometric metrics, revealing systematic weaknesses in current VLMs and providing a testbed for evaluating self-improvement mechanisms. Looking forward, we plan to extend our framework to more complex geometric reasoning tasks, explore adaptive reward schedules, investigate stability and alignment properties of refinement loops, and conduct human with VLM comparison studies. Our work contributes to the broader vision of self-improving AI systems that can adapt and enhance their capabilities through feedback-driven refinement, particularly in domains where continuous precision matters.

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.

Yuri Chervonyi, Trieu H Trinh, Miroslav Olšák, Xiaomeng Yang, Hoang Nguyen, Marcelo Menegali, Junehyuk Jung, Vikas Verma, Quoc V Le, and Thang Luong. Gold-medalist performance in solving olympiad geometry with alphageometry2. *arXiv preprint arXiv:2502.03544*, 2025.

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.

Jiafei Duan, Samson Yu, Hui Li Tan, Hongyuan Zhu, and Cheston Tan. A survey of embodied ai: From simulators to research tasks. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 6(2):230–244, 2022.

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.

Syed Ariff Syed Hesham, Yun Liu, Guolei Sun, Henghui Ding, Jing Yang, Ender Konukoglu, Xue Geng, and Xudong Jiang. Exploiting temporal state space sharing for video semantic segmentation. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24211–24221, 2025.

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.

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.

Justin Johnson, Bharath Hariharan, Laurens Van Der Maaten, Li Fei-Fei, C Lawrence Zitnick, and Ross Girshick. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2901–2910, 2017.

Chengzu Li, Zanyi Wang, Jiaang Li, Yi Xu, Han Zhou, Huanyu Zhang, Ruichuan An, Dengyang Jiang, Zhaochong An, Ivan Vulić, et al. Thinking in frames: How visual context and test-time scaling empower video reasoning. *arXiv preprint arXiv:2601.21037*, 2026.

Linjie Li, Mahtab Bigverdi, Jiawei Gu, Zixian Ma, Yinuo Yang, Ziang Li, Yejin Choi, and Ranjay Krishna. Unfolding spatial cognition: Evaluating multimodal models on visual simulations. *arXiv preprint arXiv:2506.04633*, 2025.

Zijun Lin, Haidi Azaman, M Ganesh Kumar, and Cheston Tan. Compositional learning of visually-grounded concepts using reinforcement. *arXiv preprint arXiv:2309.04504*, 2023.

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.

Mateusz Michalkiewicz, Anekha Sokhal, Tadeusz Michalkiewicz, Piotr Pawlikowski, Mahsa Baktashmotlagh, Varun Jampani, and Guha Balakrishnan. Giq: Benchmarking 3d geometric reasoning of vision foundation models with simulated and real polyhedra. *arXiv preprint arXiv:2506.08194*, 2025.

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

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.

Chan Hee Song, Valts Blukis, Jonathan Tremblay, Stephen Tyree, Yu Su, and Stan Birchfield. Robospatial: Teaching spatial understanding to 2d and 3d vision-language models for robotics. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 15768–15780, 2025.

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.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

Chenwei Wu, Li Erran Li, Stefano Ermon, Patrick Haffner, Rong Ge, and Zaiwei Zhang. The role of linguistic priors in measuring compositional generalization of vision-language models. In *Proceedings on*, pp. 118–126. PMLR, 2023.

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.

Fernanda Miyuki Yamada, Joao Paulo Gois, and Harlen Costa Batagelo. Solving tangram puzzles using raster-based mathematical morphology. In *2019 32nd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)*, pp. 116–123. IEEE, 2019.

Fernanda Miyuki Yamada, Harlen Costa Batagelo, João Paulo Gois, and Hiroki Takahashi. Generative approaches for solving tangram puzzles. *Discover Artificial Intelligence*, 2024.

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.

Chao Zhao, Chunli Jiang, Lifan Luo, Guanlan Zhang, Hongyu Yu, Michael Yu Wang, and Qifeng Chen. Master rules from chaos: Learning to reason, plan, and interact from chaos for tangram assembly. *arXiv preprint arXiv:2505.11818*, 2025.

A Appendix

A.1 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.

A.2 Dataset Construction Pipeline

Algorithm 2 Tangram Dataset Pipeline (SVG $\rightarrow$ JSON $\rightarrow$ PNG)

Input: Directory of SVG files (`IN_SVG_DIR`)
Output: JSON annotations and optional rendered PNGs
for all `svg_path` $\in$ `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 Evaluation Pipeline

Algorithm 3 Evaluation Pipeline

Input: images I, GT JSONs G, model M, mode
for all (I, G) **do**
    Predict JSON with M under `mode`
    Compute L2/angle/size/IoU from prediction vs. G
    Optionally render GT / prediction / overlay
end for

A.4 Improvement Visualization

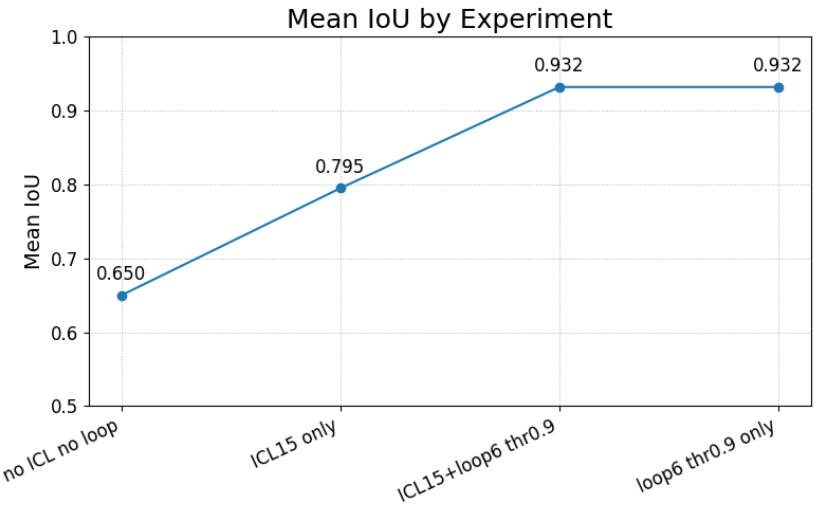


Figure 3: Mean IoU across ablations on the *medium triangle*. The test-time self-refinement loop (ICL + reward) yields the largest gain.