TangramPuzzle: Evaluating Multimodal Large Language Models with Compositional Spatial Reasoning

Anonymous ACL submission

Abstract

Multimodal Large Language Models (MLLMs) have achieved remarkable progress in visual recognition and semantic understanding. Nevertheless, their ability to perform precise compositional spatial reasoning remains largely unexplored. Existing benchmarks often involve relatively simple tasks and rely on semantic approximations or coarse relative positioning, while their evaluation metrics are typically limited and lack rigorous mathematical formulations. To bridge this gap, we introduce **TangramPuzzle** [1], a geometry-grounded benchmark designed to evaluate compositional spatial reasoning through the lens of the classic Tangram game. We propose the Tangram Construction Expression (TCE), a symbolic geometric framework that grounds tangram assemblies in exact, machine-verifiable coordinate specifications, to mitigate the ambiguity of visual approximation. We design two complementary tasks: Outline Prediction, which demands inferring global shapes from local components, and End-to-End Code Generation, which requires solving inverse geometric assembly problems. We conduct extensive evaluation experiments on advanced open-source and proprietary models, revealing an interesting insight: MLLMs tend to prioritize matching the target silhouette while neglecting geometric constraints, leading to distortions or deformations of the pieces.

1 Introduction

Multimodal Large Language Models (MLLMs) have achieved strong performance on a broad range of vision–language tasks, including visual understanding (Wang et al., 2025b; Fang et al., 2025; Ding et al., 2025), document question answering (Wu et al., 2025b; Huang et al., 2025a), OCR (Chen et al., 2025a; Huang et al., 2025b; Greif et al., 2025), and chart interpretation (Chen et al., 2025b; Zhao et al., 2025). Recently, both training and evaluation have been increasingly shifting from semantic recognition toward more demanding tasks that require precise spatial understanding and reasoning, such as visual grounding (Tang et al., 2025b; Xu et al., 2025c; Ma et al., 2025b), GUI agent (Cheng et al., 2025a; Xiao et al., 2025), and visual reasoning in complex scenes (Kuang et al., 2025; Li et al., 2024). Spatial reasoning is not merely about recognizing objects, but about constructing and validating structured geometric relationships that support reliable decision making.

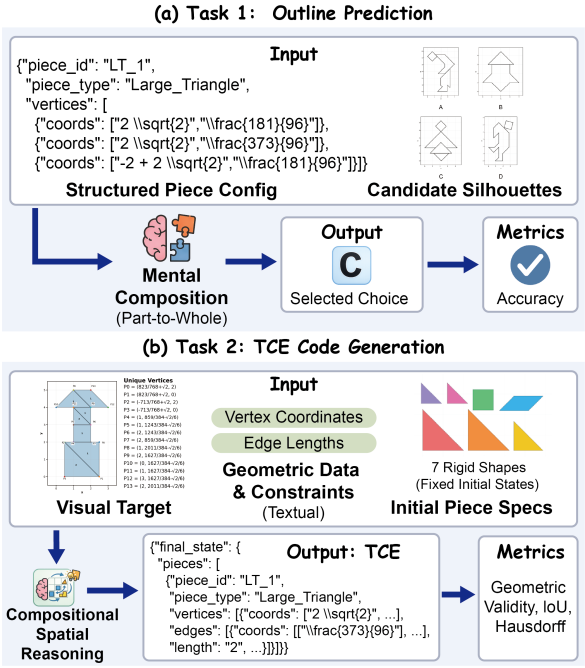

Figure 1: Illustration of the two spatial reasoning tasks. (a) Task 1 global shape inference; (b) Task 2 constrained shape decomposition.

Despite rapid progress, existing spatial reasoning tasks for MLLMs still exhibit several limitations. First, many tasks focus on coarse relational concepts (e.g., left/right, front/behind, adjacency) (Liu et al., 2025; Deng et al., 2025; Wu et al., 2025a),

[1] We will open-source the data after the paper is published.

which do not fully capture reasoning under strict geometric and physical constraints. Second, spatial problems are often specified using natural language descriptions or multiple-choice formats without a formal geometric representation (Rodionov et al., 2025; Ma et al., 2025a), relying more on linguistic reasoning ability than visual reasoning. In addition, evaluation commonly relies on text answer matching or human/LLM judgment, which is particularly problematic for constructive tasks where multiple valid solutions may exist. ***These issues make it difficult to precisely measure whether a model truly "compositionally reasons" in space or merely exploits superficial cues.***

In this work, we study a classic yet still underexplored problem of compositional spatial reasoning, for which tangram puzzles (Zhao et al., 2022; Bofferding and Aqazade, 2023; Bofferding and Zhu, 2023) provide a particularly suitable instantiation. The tangram serves as a minimalist yet highly expressive abstraction of real-world spatial assembly. This task requires the precise arrangement of seven fixed rigid polygonal pieces into a target silhouette through Euclidean transformations (translation, rotation, and optionally reflection), while satisfying strict constraints such as non-overlap, full coverage, and topological validity. These fundamental capabilities form the functional basis for solving critical real-world problems such as CAD-like mechanical assembly, robotic manipulation planning, and structural chart reconstruction. By mastering these atomic spatial operations and constraint-satisfaction principles within a controlled environment, we can effectively probe a model's underlying potential for more complex, embodied spatial planning in the physical world.

However, prior tangram-related resources and studies only focus on overall interpretation of shapes or perceptual approximations (Zhang et al., 2024; Ji et al., 2022), and typically lack (i) a standardized, machine-verifiable geometric representation, and (ii) a strict evaluation protocol that validates rigidity, physical feasibility, and solution correctness. Moreover, natural-language descriptions of complex spatial layouts are often ambiguous, reducing evaluation precision and reliability.

To address these gaps, we introduce TangramPuzzle, a benchmark designed to evaluate compositional spatial reasoning in MLLMs under rigorous geometric constraints. Central to our framework is the Tangram Construction Expression (TCE), a symbolic geometry schema that formalizes the assembly process. By transitioning from descriptive prose to an algebraic symbolic system, TCE represents each instance with exact coordinates and structural relations, effectively eliminating linguistic vagueness. Built upon TCE, the benchmark includes two complementary tasks, as shown in Figure 1: (1) Outline Prediction, where models infer the global silhouette from a given final arrangement of pieces; and (2) End-to-End Tangram Solution Generation, where models construct a complete assembly that exactly fills a given target outline. We evaluate outputs with a constraint-based verifier that checks syntactic validity, rigid-shape preservation, and physical feasibility (e.g., non-overlap and connectivity), and then measures silhouette fidelity using IoU and Hausdorff distance, avoiding ill-posed comparisons under multi-solution ambiguity. TangramPuzzle is constructed through a multi-stage pipeline involving source filtering, an interactive annotation tool with snapping, symbolic normalization into exact expressions, and human-in-the-loop validation. Extensive experiments across a wide range of MLLMs show that TangramPuzzle remains highly challenging, revealing systematic failure modes in both compositional outline reasoning and constraint-satisfying solution generation. Our main contributions are as follows:

- We introduce TangramPuzzle, a tangram-inspired benchmark for compositional spatial reasoning in MLLMs, with a rigorous and machine-verifiable math representation.
- We design the data construction pipeline for two tasks that jointly evaluate discriminative global-shape inference and constructive inverse assembly generation.
- We conduct comprehensive evaluation of advanced MLLMs, highlighting persistent limitations in complex spatial reasoning.

2 Related Work

2.1 General Multimodal Benchmarks

Standard benchmarks for Multimodal Large Language Models (MLLMs) have transitioned from foundational tasks like visual question answering (VQA) (Cheng et al., 2025b; Zhong et al., 2025; Liu et al., 2024; Chen et al., 2024a) and image captioning (Dong et al., 2024; Liu et al., 2022; Lu et al., 2025) to comprehensive assessments of emergent capabilities. Recent benchmarks target diverse

competencies, including reasoning ability (Guo et al., 2025; Yuan et al., 2025; Yue et al., 2025; Xiao et al., 2024), where challenging datasets like Humanity's Last Exam (Phan et al., 2025) and ZeroBench (Roberts et al., 2025) test the upper bounds of expert-level cognition; document and chart understanding (Wang et al., 2024; Fu et al., 2024a; Xia et al., 2025; Ouyang et al., 2025; Masry et al., 2025); as well as visual grounding (Paiss et al., 2023; Brazil et al., 2023), hallucination evaluation (Li et al., 2023; Guan et al., 2024; Leng et al., 2024; Chen et al., 2024b; Yang et al., 2025b), and multi-image comprehension (Cheng et al., 2025c; Fu et al., 2024b; Jiang et al., 2024; Wang et al., 2025a). Furthermore, the assessment of agentic capabilities (Li et al., 2025; Xie et al., 2025; Rawles et al., 2025; Zheng et al., 2025a) centers on GUI interpretation and precise element interaction.

2.2 Spatial Reasoning Benchmarks

Spatial reasoning is essential for MLLMs to perceive and interact with the physical world (Zheng et al., 2025b). Early research emphasized spatial understanding, as SpatialBench (Xu et al., 2025a) and RealWorldQA (xAI, 2024) evaluate relative positioning and depth in natural images. As the field progressed, benchmarks like OmniSpatial (Jia et al., 2025), MMSI-Bench (Yang et al., 2025a), and LEGO-Puzzles (Tang et al., 2025a) expanded this scope to dynamic scenarios, multi-view interactions, or the following of assembly instructions. However, most of these are formulated as visual question answering tasks, which may unintentionally encourage "shortcut" behaviors, such as exploiting language priors or superficial visual cues, rather than performing faithful geometric reasoning. Similarly, while ORIGAMISPACE (Xu et al., 2025b) emphasizes procedural reasoning through sequential folding steps, our benchmark focuses on a different aspect of spatial reasoning by requiring models to mentally assemble a complete structure from scratch under strict constraints, directly testing compositional spatial reasoning in an inverse assembly setting. In the tangram domain, Tangram (Zhang et al., 2024) targets basic element recognition and counting, while TANGAN (Yamada et al., 2025) treats assembly as an optimization problem rather than evaluating MLLMs. A detailed comparison with representative spatial reasoning benchmarks is provided in Appendix A.

3 Method

3.1 Task Definition

Tangram is a classic dissection puzzle in which seven rigid polygons must be assembled to form a target shape. Although simple in form, tangram assembly mirrors practical tasks such as assembling parts into a target structure, where each component must be placed with the correct orientation and position for the overall configuration to be valid, making it an ideal testbed for compositional spatial reasoning. TangramPuzzle consists of two challenging tasks designed to evaluate the spatial reasoning capabilities of MLLMs, ranging from passive spatial perception to active solution generation.

Geometry Description by TCE Formula Natural language descriptions are often ambiguous for geometric assembly, where a phrase like "place the triangle above the square" fails to specify whether the pieces touch or separate. Moreover, TangramPuzzle is governed by strict Euclidean and topological constraints, motivating a formal, coordinate-based representation that grounds visual inputs in precise mathematical formulations and supports unambiguous task definition, automatic verification, and fine-grained evaluation of spatial reasoning abilities. Hence, the Tangram Construction Expression (TCE) is introduced as a symbolic geometry description language for 2D tangram assembly. Each TCE instance consists of five core components: instance_id, target_outline, initial_state, final_state, and adjacency_graph. The representation of TCE is shown in Figure 2. Detailed TCE definitions can be found in Appendix B.

$$
\begin{aligned}
\textbf{TCE} &= \langle \texttt{id}, \Omega_{target}, \Sigma_{init}, \Sigma_{final}, G_{adj} \rangle \\
\texttt{id} &= \texttt{"swan_cfg_01"} \\
\Omega_{target} &= \{V : [(0,0), (4, 2\sqrt{2}), ...], E : [(v_0, v_1), ...]\} \\
\Sigma_{final} &= \{p_{T1} : \langle \mathbf{M} = \begin{bmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 2 \\ \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix}, V_{world} : [...] \rangle, ..\} \\
G_{adj} &= \{(T1, S1), (S1, T3), ...\}
\end{aligned}
$$

Figure 2: Compact representation of a TCE instance.

Task 1: Outline Prediction This setting tests whether MLLMs can synthesize a global visual shape from local component arrangements. Rather than performing conventional visual recognition, the model is required to mentally compose the union of seven discrete tangram pieces and reason about their resulting overall shape. The input

consists of the textual TCE specification of the seven tangram pieces (including piece types and exact vertex coordinates) and a candidate image displaying four distinct silhouette options (A–D). The model is required to select the correct matching silhouette from the candidate set. Accuracy and Invalid Rate are used as the evaluation metrics.

Task 2: End-to-End Tangram Solution Generation This task assesses the model's ability to solve a tangram puzzle under strict geometric constraints by generating a precise executable solution. Given a target silhouette image with annotated vertex coordinates, the model must decompose the outline into the seven fixed rigid pieces. The input also includes textual geometric data for the target outline, including vertex coordinates, edge relationships, and edge lengths, as well as the initial state of each tangram piece, including its type, vertices, and edges. The model is required to output a complete Tangram Configuration Expression (TCE) in JSON format, specifying the final state of all seven pieces so that they exactly fill the target outline. Evaluation checks both geometric validity, including rigid-body preservation and non-overlap, and shape fidelity, measured by Intersection over Union (IoU) and Hausdorff Distance (d_H).

3.2 Data Construction

To ensure high-quality geometric data, we implemented a multi-stage construction pipeline, as illustrated in Figure 3, comprising source filtering, interactive annotation, and symbolic refinement.

3.2.1 Raw Data Collection

We use visual patterns from the KiloGram (Ji et al., 2022) dataset as the foundational source. While KiloGram provides diverse tangram compositions, it primarily focuses on linguistic descriptions and semantic part labels (e.g., "a person wearing a robe"). In contrast, TangramPuzzle introduces TCE to ground each shape in exact LaTeX-based algebraic coordinates, requiring models to reason over precise spatial configurations rather than approximate semantic categories. Since some KiloGram instances violate the topological constraints needed for mathematical verification, we manually filter out invalid configurations, including patterns with holes, disconnected pieces, or incomplete assemblies. Only connected shapes without internal voids are retained for re-annotation and canonicalization.

3.2.2 Annotation Tool Development

Existing geometric tools are insufficient for efficient and precise tangram annotation, as GeoGebra[2] requires users to compute coordinates before drawing, while manipulation-friendly platforms like Polypad[3] cannot export precise vertex coordinates. We therefore developed a bespoke web-based annotation interface with vertex- and edge-snapping mechanisms to reduce human-induced alignment errors. The tool records all vertex coordinates in real time and exports complete annotations in structured jsonl format, substantially accelerating the labeling process. Detailed annotation procedures are provided in Appendix C.

3.2.3 Geometric Normalization

The raw annotations produced by the tool are stored in floating-point form, which is insufficient for exact geometric reasoning. To avoid ambiguity from finite-precision decimals, we post-process to normalize and convert all geometric values into exact symbolic forms. Specifically, each assembled configuration is translated such that its lower boundary lies on the x-axis and its left boundary lies on the y-axis, yielding a canonical placement. We then apply fixed rules to map all vertex coordinates to exact algebraic expressions in LaTeX. Based on these coordinates, we further extract the global outer contour, edge relations, and piece adjacency, and serialize the result into the unified TCE representation. Details on TCE construction and TCE-based generation of Task 1 multiple-choice options are provided in Appendix D.

3.2.4 Human Filtering

To ensure data correctness, we perform careful manual verification through visualization. Each annotated instance is rendered to display the assembled configuration, highlighting individual pieces and their boundaries. Instances exhibiting misaligned vertices, unintended gaps between adjacent edges, or overlaps between pieces are promptly flagged as invalid. Such cases are re-annotated using the tool, followed by repeated verification and conversion. This iterative process continues until all retained instances satisfy the geometric validity constraints required by TCE.

[2] https://www.geogebra.org/classic
[3] https://polypad.amplify.com/p#tangram

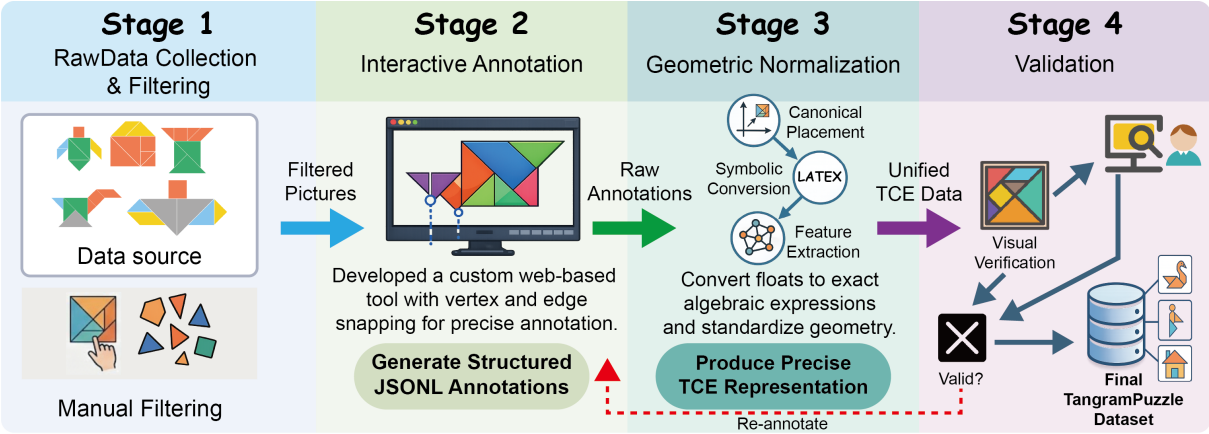

Figure 3: Overview of the TangramPuzzle data construction pipeline.

3.2.5 Data Statistics

TangramPuzzle exhibits both rich structural diversity and strict geometric rigor. The target silhouettes cover a wide range of semantic categories, including abstract shapes, everyday objects, and animals, spanning a broad and challenging spectrum of difficulty from simple, compact forms to complex, articulated silhouettes with intricate boundaries. In total, the dataset contains 668 unique tangram configurations. Each configuration is used to instantiate both Task 1 and Task 2, resulting in 1,336 problem instances overall.

3.3 Evaluation Metrics

Task 1: Outline Prediction To evaluate vision-grounded spatial perception, we measure the standard classification **Accuracy (Acc)**, which calculates the percentage of instances where the model selects the correct target outline. Additionally, to assess the model's instruction-following capability and robustness, we report the **Invalid Rate**, defined as the proportion of responses that fail to map to any valid candidate option. Invalid outputs may arise from formatting errors or from generating content unrelated to the candidate options, reflecting common failure modes in instruction execution, in which case no interpretable decision can be extracted. Detailed metric computations for Task 1 are provided in Appendix E.1.

Task 2: End-to-End Solution Generation Evaluating assembly generation is non-trivial because a single target silhouette may admit multiple valid internal arrangements, making direct comparison with a single reference solution unreasonable. Moreover, relying solely on a binary success metric is insufficient to understand model behavior: a prediction may fail to produce a fully valid assembly for many different reasons. To better diagnose these failure modes, we introduce a two-stage hierarchical constraint-based evaluation framework that not only measures whether the final objective is achieved, but also reveals where and how failures occur, while providing a smoother view of progress toward valid assembly generation. Detailed metric definitions and computation procedures for Task 2 are provided in Appendix E.2. Concretely, the framework comprises:

- **Stage 1: Constraint Validation.** We first check whether the generated assembly satisfies the required physical and geometric constraints by identifying three types of errors: **Syntax Error (TSE)** checks whether the output conforms to the TCE format and piece count; **Rigid Geometry Error (RGE)** verifies that the area and perimeter of each piece remain unchanged to prevent shape distortion; and **Physical Error (PE)** detects if any two pieces exhibit impermissible overlap or if the global union of all pieces fails to form a single connected component. The metric **Validation Pass Rate (VPR)** reports the percentage of samples that satisfy all these constraints.
- **Stage 2: Shape Similarity.** We further quantify how well the constructed shape matches the target silhouette using two metrics: **Intersection over Union (IoU)** measures the global area overlap between the predicted assembly and the target, while **Hausdorff Distance** (d_H) captures fine-grained boundary deviations to penalize shape outliers. Finally, **Success** reports the percentage of samples with a fully valid assembly that solve the puzzle.

4 Experiment

4.1 Experimental Setting

Baselines We evaluate TangramPuzzle across a diverse set of models. For open-source models, we evaluate Qwen3-VL-8B-Instruct, Qwen3-VL-32B-Instruct (Bai et al., 2025), InternVL3-78B (Zhu et al., 2025), DeepSeek-OCR (Wei et al., 2025), DeepSeek-VL2 (Wu et al., 2024), and GLM-4.6V (Z.ai, 2025). For commercial models, we include GPT-5.2 (OpenAI, 2025), Gemini3-Pro (Google DeepMind, 2025), and Claude-Sonnet-4.5 (Anthropic, 2025). Additional details are provided in Appendix F.

Details All experiments are conducted by calling the corresponding model APIs. We run the evaluation pipeline on a workstation equipped with an NVIDIA RTX 2080 Ti GPU, and use a unified implementation to send requests, parse outputs, and compute all metrics across models. Prompt templates are provided in Appendix G.

4.2 Main Results

The results for Task 1 are summarized in Table 1, revealing substantial performance differences across models. The task requires models to mentally compose a silhouette from disjointed parts and identify the correct visual match, making it difficult to solve through simple memorization or superficial cues alone. While most open-source models struggle to exceed 40% accuracy, Gemini3-Pro achieves the highest score among all models. Most models also maintain low invalid-output rates, although a few remain less stable. Detailed analysis for Task 1 is provided in Appendix H.1.

Table 2 presents the results for Task 2. Their generally low **TSE** scores indicate that models can usually produce outputs in the required TCE-style format, and a lightweight recovery mechanism is applied to avoid over-penalizing minor formatting errors. This suggests that the main bottleneck is not formatting, but compositional spatial reasoning under geometric constraints. In particular, high RGE and PE across most models indicate a failure to maintain rigid geometry and physical validity, despite achieving high IoU in silhouette similarity. Further inspection reveals that this apparent visual fidelity is often achieved through "cheating," namely by impermissibly distorting rigid pieces or allowing overlaps to force a visual match. Detailed analysis for Task 2 is provided in Appendix H.2.

Table 1: Performance comparison on Task 1.

Model	**Invalid** (↓)	**Acc** (↑)
Qwen3-VL-8B (Bai et al., 2025)	**00.00**	37.57
Qwen3-VL-32B (Bai et al., 2025)	00.15	73.05
InternVL3-78B (Zhu et al., 2025)	00.75	29.19
DeepSeek-OCR (Wei et al., 2025)	12.87	22.46
DeepSeek-VL2 (Wu et al., 2024)	10.18	25.45
GLM-4.6V (Z.ai, 2025)	**00.00**	30.84
GPT-5.2 (OpenAI, 2025)	**00.00**	77.54
Gemini3-Pro (Google DeepMind, 2025)	**00.00**	**98.65**
Claude-Sonnet-4.5 (Anthropic, 2025)	00.45	51.20

4.3 Detailed Analyses

Effect of In-context Learning (ICL) To examine whether the bottleneck in Task 2 stems from format unfamiliarity or deeper spatial reasoning limitations, we evaluate models under an In-Context Learning (ICL) setting by prepending three solved examples, as shown in Table 2. With ICL, some models exhibit higher TSE rates than in the zero-shot setting, suggesting that the longer symbolic context may introduce additional cognitive load and destabilize structured generation. Detailed analysis is provided in Appendix I.1.

Effect of Textual Geometry We further investigated the models' dependency on textual metadata through an ablation study in a Visual-Centric setting, as shown in Table 2. In this setup, we removed the textual description of the target outline (i.e., vertex and edge lists), providing only the outline image with annotated coordinates alongside the initial piece descriptions. Most MLLMs show a clear performance drop, accompanied by more hallucinations and syntax errors, indicating a strong dependence on textual "crutches" rather than precise geometric extraction from vision alone. Textual geometry analysis is detailed in Appendix I.2.

Effect of Chain-of-Thought (CoT) As shown in Table 2, CoT provides only limited and uneven benefits. While it slightly improves formatting or some intermediate metrics for certain models, it does not resolve the core difficulty of Task 2. In some cases, better silhouette matching even comes with increased geometric violations, suggesting continued reliance on "visual cheating." For smaller models, performance remains largely unchanged, indicating that simply adding step-by-step reasoning is insufficient without stronger spatial grounding. Detailed analysis is provided in Appendix I.3.

Table 2: Performance comparison on Task 2 under different settings.

Model	Constraint error (↓)			VPR (↑)	Silhouette Quality		Success (↑)
	TSE (↓)	RGE (↓)	PE (↓)		IoU (↑)	Hausdorff (↓)	
Qwen3-VL-8B-Instruct (Bai et al., 2025)	23.35	40.42	80.69	00.00	20.25	07.81	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	07.93	83.08	94.46	00.00	50.55	01.39	00.00
InternVL3-78B (Zhu et al., 2025)	06.59	91.77	92.96	00.00	36.15	03.18	00.00
GLM-4.6V (Z.ai, 2025)	06.29	91.02	93.56	00.15	43.79	01.37	00.15
GPT-5.2 (OpenAI, 2025)	07.19	90.12	92.81	00.00	58.49	00.84	00.00
Gemini3-Pro (Google DeepMind, 2025)	08.83	**37.57**	**65.57**	**22.60**	**85.93**	**00.37**	**21.56**
Claude-Sonnet-4.5 (Anthropic, 2025)	**00.60**	98.20	97.90	00.15	61.61	00.57	00.00
In-Context Learning (ICL)							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	33.98	57.04	66.17	00.15	28.48	03.51	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	**06.74**	87.28	93.71	00.00	48.65	01.49	00.00
GPT-5.2 (OpenAI, 2025)	09.43	86.98	90.72	00.00	60.30	00.74	00.00
Gemini3-Pro (Google DeepMind, 2025)	10.78	**34.28**	**61.53**	**26.20**	**87.36**	**00.35**	**24.85**
Visual-Centric Setting							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	15.42	17.07	86.23	00.00	11.39	12.28	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	**09.58**	74.70	93.26	00.15	43.69	02.78	00.15
DeepSeek-OCR (Wei et al., 2025)	100.0	**00.00**	**01.35**	00.00	11.72	07.81	00.00
DeepSeek-VL2 (Wu et al., 2024)	99.85	00.15	13.32	00.00	12.08	05.49	00.00
GPT-5.2 (OpenAI, 2025)	39.67	58.23	58.08	00.00	53.67	01.21	00.00
Gemini3-Pro (Google DeepMind, 2025)	16.77	32.34	56.89	**25.00**	**82.75**	**00.42**	**22.75**
Chain-of-Thought (CoT)							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	23.35	40.12	78.74	00.00	18.64	07.93	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	06.29	83.83	95.51	00.15	51.51	01.38	00.15
GPT-5.2 (OpenAI, 2025)	**04.04**	91.32	95.96	00.00	58.80	00.68	00.00
Gemini3-Pro (Google DeepMind, 2025)	24.40	**33.38**	**51.35**	**20.66**	**83.12**	**00.50**	**19.76**
Decomposition-Based Strategy							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	26.35	41.47	78.14	00.00	24.22	07.11	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	**06.89**	86.23	94.61	00.15	51.25	01.23	00.00
GPT-5.2 (OpenAI, 2025)	35.33	60.78	64.22	00.00	60.25	00.88	00.00
Gemini3-Pro (Google DeepMind, 2025)	25.45	**30.84**	**52.69**	**20.21**	**86.26**	**00.36**	**18.86**
Text-Only Setting							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	50.15	**26.35**	51.95	00.00	21.56	07.45	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	**13.92**	72.16	87.72	00.00	47.66	02.20	00.00
GPT-5.2 (OpenAI, 2025)	54.04	40.42	**43.86**	00.00	54.40	02.23	00.00
Gemini3-Pro (Google DeepMind, 2025)	28.14	33.68	56.74	**13.32**	**81.15**	**00.59**	**12.28**
TCE-Light Setting							
Qwen3-VL-8B-Instruct (Bai et al., 2025)	19.91	**00.00**	81.59	00.00	05.10	18.18	00.00
Qwen3-VL-32B-Instruct (Bai et al., 2025)	04.49	**00.00**	95.81	00.00	13.91	16.49	00.00
GPT-5.2 (OpenAI, 2025)	**01.95**	**00.00**	98.80	00.00	07.45	18.35	00.00
Gemini3-Pro (Google DeepMind, 2025)	14.82	**00.00**	**80.84**	**04.49**	**36.99**	**12.18**	**03.59**

Effect of Systematic Decomposition Motivated by the human expert analysis in Appendix J, we evaluate a decomposition-based prompting strategy for Task 2 that encourages models to represent both the target silhouette and tangram pieces using a shared unit—the small isosceles right triangle. As shown in Table 2, this approach yields modest improvements in silhouette quality and slightly reduces geometric constraint violations for some models. However, these gains rarely translate into higher final Success, and smaller models show little change. Overall, while decomposition provides a more structured reasoning prior, limited compositional spatial reasoning remains the primary obstacle. Detailed analysis is provided in Appendix I.4.

Effect of Removing Visual Input To examine whether Task 2 degenerates into a purely symbolic encoding–decoding problem once converted into TCE, we conduct a text-only ablation in which models receive only textual descriptions of the target outline vertices. As shown in Table 2, removing visual input leads to a clear performance drop, most notably for Gemini3-Pro. This suggests that the visual silhouette provides a crucial spatial anchor that cannot be fully replaced by symbolic coordinates alone. Overall, the degradation indicates that solving TangramPuzzle requires integrating both visual and symbolic modalities for spatial reasoning, rather than relying on simple symbolic mapping. Detailed analysis is presented in Appendix I.5.

Table 3: Human Performance on Tasks 1 and 2.

Human	Success (↑)	Complexity (↓)	Time usage (min) (↓)
Human performance on Task 1			
Human 1	**100.0**	02.10	03.10
Human 2	**100.0**	**01.50**	02.80
Human 3	**100.0**	01.70	**02.50**
Average	100.0	01.77	02.80
Human performance on Task 2			
Human 1	72.00	04.10	**06.30**
Human 2	**100.0**	**03.80**	06.60
Human 3	46.00	04.30	07.10
Average	72.67	04.07	06.67

Effect of TCE-Light Representation To examine whether Task 2 failures stem from the full TCE representation rather than spatial reasoning itself, we introduce a simplified *TCE-Light* setting, where models output only one rigid transformation per piece, including `piece_id`, rotation angle, optional reflection, and translation. The final vertices are then deterministically computed from the initial pieces, reducing symbolic coordinate generation and making RGE zero for successfully parsed outputs. As shown in Table 2, however, this simplification does not solve the task: models still exhibit high PE, low VPR, and near-zero Success, with silhouette quality often deteriorating. Detailed analysis is provided in Appendix I.6.

Relationship between Task 1 and Task 2 To better understand the relationship between the two tasks, we compute cross-model correlations between Task 1 Accuracy and the main Task 2 metrics using both Pearson and Spearman correlations. The results show that Task 1 Accuracy correlates strongly with Task 2 IoU, suggesting that global-shape composition helps approximate the target silhouette, but it does not reliably predict fully valid assemblies. Detailed analysis and the correlation table are provided in Appendix I.7.

Human Performance We evaluate human performance on both tasks using domain experts, as shown in Table 3. All participants achieve 100% accuracy on Task 1, while Task 2 remains more challenging but rarely leads to geometric-constraint violations; human failures are typically left unsolved rather than invalid. Detailed results and solving strategies (e.g., systematic triangle decomposition) are provided in Appendix J. We also conduct a complementary human evaluation of MLLM outputs on Task 2 to assess perceptual plausibility beyond strict automatic metrics, detailed in Appendix K.

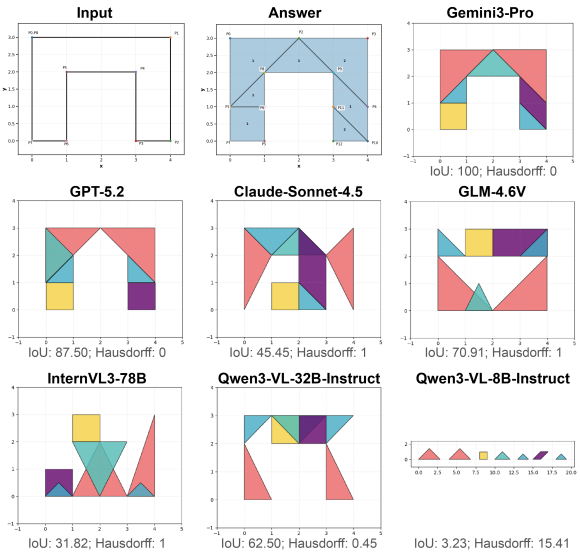

Figure 4: Visualization of a case from Task 2.

4.4 Case Study

Figure 4 presents a case study on a U-shaped target in Task 2. Gemini3-Pro is the only model that produces a fully valid assembly, showing strong geometric grounding and compositional reasoning. Other models tend to favor silhouette matching over geometric validity: Claude-Sonnet-4.5 deforms pieces, InternVL3-78B produces severe overlaps, and GPT-5.2 generates a plausible outline but with an incorrect piece inventory. Detailed analysis is presented in Appendix L.

5 Conclusion

We presented TangramPuzzle, a geometry-grounded benchmark specifically designed for evaluating MLLMs on compositional spatial reasoning. Addressing the limitations of existing semantic-focused evaluations, we introduced the Tangram Construction Expression (TCE), a symbolic geometric framework that enables the precise, machine-verifiable assessment of rigid body constraints and physical feasibility. The benchmark encompasses the Outline Prediction and End-to-End TCE Code Generation tasks, which jointly probe the spectrum of spatial intelligence. We conducted a comprehensive evaluation of open-source and closed-source MLLMs covering different scales, accompanied by in-depth exploratory analyses on in-context learning and modality dependency. By shifting the evaluation paradigm from approximate visual matching to exact constraint satisfaction, TangramPuzzle provides a standardized and challenging testbed for future research.

Limitations

While TangramPuzzle provides a rigorous framework for evaluating compositional spatial reasoning under exact geometric constraints, it is currently limited to 2D planar geometry with a fixed set of components, and thus does not fully capture real-world 3D complexities such as occlusion and viewpoint changes. Future work will extend this evaluation paradigm toward richer and more realistic spatial assembly settings; further discussion and future extensions are provided in Appendix M.

Ethical Considerations

Potential Risks The TangramPuzzle benchmark focuses on abstract geometric reasoning and is constructed entirely from publicly available sources. Due to the inherently abstract nature of these geometric shapes, the dataset is devoid of personally identifiable information (PII), sensitive biometric data, or offensive content. Consequently, there are no foreseeable risks related to safety, discrimination, or surveillance.

Ethical Statement Our data construction process strictly adhered to ethical research guidelines. All source images were derived from public domains, ensuring no copyright infringement. During the human-in-the-loop annotation and validation stages, all participants were informed of the nature of the task and compensated fairly in accordance with local labor standards. Furthermore, as the dataset comprises solely rigid geometric shapes, it is inherently free from biometric data, social biases, or other privacy concerns. Thus, the proposed research direction and tasks are ethically benign and socially harmless. All data and annotations will be released under a permissive open-source license upon acceptance to facilitate transparency and reproducibility.

LLMs Usage Statement We use large language models and multimodal models as baselines to generate predictions for evaluation and analysis. During the writing of this paper, we also use large language models to assist with grammar checking and improving text fluency. The authors have carefully reviewed and verified all suggested revisions provided by the large language models to ensure their correctness, and take full responsibility for the content of this paper.

References

Anthropic. 2025. Introducing claude sonnet 4.5. https://www.anthropic.com/news/claude-sonnet-4-5. Accessed: 2026-01-04.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Laura Bofferding and Mahtob Aqazade. 2023. "where does the square go?": reinterpreting shapes when solving a tangram puzzle. *Educational Studies in Mathematics*, 112(1):25–47.

Laura Bofferding and Yi Zhu. 2023. Composing tangram puzzles to support shape transformation. *Mathematics Teacher: Learning and Teaching PK-12*, 116(7):503–510.

Garrick Brazil, Abhinav Kumar, Julian Straub, Nikhila Ravi, Justin Johnson, and Georgia Gkioxari. 2023. Omni3d: A large benchmark and model for 3d object detection in the wild. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13154–13164.

Lin Chen, Jinsong Li, Xiaoyi Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and 1 others. 2024a. Are we on the right way for evaluating large vision-language models? *Advances in Neural Information Processing Systems*, 37:27056–27087.

Song Chen, Xinyu Guo, Yadong Li, Tao Zhang, Mingan Lin, Dongdong Kuang, Youwei Zhang, Lingfeng Ming, Fengyu Zhang, Yuran Wang, and 1 others. 2025a. Ocean-ocr: Towards general ocr application via a vision-language model. *arXiv preprint arXiv:2501.15558*.

Xiang Chen, Chenxi Wang, Yida Xue, Ningyu Zhang, Xiaoyan Yang, Qiang Li, Yue Shen, Lei Liang, Jinjie Gu, and Huajun Chen. 2024b. Unified hallucination detection for multimodal large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3235–3252, Bangkok, Thailand. Association for Computational Linguistics.

Zixin Chen, Sicheng Song, Kashun Shum, Yanna Lin, Rui Sheng, Weiqi Wang, and Huamin Qu. 2025b. Unmasking deceptive visuals: Benchmarking multimodal large language models on misleading chart question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 13767–13800.

Pengzhou Cheng, Zheng Wu, Zongru Wu, Tianjie Ju, Aston Zhang, Zhuosheng Zhang, and Gongshen Liu. 2025a. Os-kairos: Adaptive interaction for mllm-powered gui agents. In *Findings of the Association*

for Computational Linguistics: ACL 2025, pages 6701–6725.

Xianfu Cheng, Wei Zhang, Shiwei Zhang, Jian Yang, Xiangyuan Guan, Xianjie Wu, Xiang Li, Ge Zhang, Jiaheng Liu, Yuying Mai, and 1 others. 2025b. Simplevqa: Multimodal factuality evaluation for multimodal large language models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4637–4646.

Ziming Cheng, Binrui Xu, Lisheng Gong, Zuhe Song, Tianshuo Zhou, Shiqi Zhong, Siyu Ren, Mingxiang Chen, Xiangchao Meng, Yuxin Zhang, and 1 others. 2025c. Evaluating mllms with multimodal multi-image reasoning benchmark. *arXiv preprint arXiv:2506.04280*.

Nianchen Deng, Lixin Gu, Shenglong Ye, Yinan He, Zhe Chen, Songze Li, Haomin Wang, Xingguang Wei, Tianshuo Yang, Min Dou, and 1 others. 2025. Internspatial: A comprehensive dataset for spatial reasoning in vision-language models. *arXiv preprint arXiv:2506.18385*.

Yihao Ding, Siwen Luo, Yue Dai, Yanbei Jiang, Zechuan Li, Geoffrey Martin, and Yifan Peng. 2025. A survey on mllm-based visually rich document understanding: Methods, challenges, and emerging trends. *arXiv preprint arXiv:2507.09861*.

Hongyuan Dong, Jiawen Li, Bohong Wu, Jiacong Wang, Yuan Zhang, and Haoyuan Guo. 2024. Benchmarking and improving detail image caption. *arXiv preprint arXiv:2405.19092*.

Wenlong Fang, Qiaofeng Wu, Jing Chen, and Yun Xue. 2025. guided mllm reasoning: Enhancing mllm with knowledge and visual notes for visual question answering. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19597–19607.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, and 1 others. 2024a. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. 2024b. Blink: Multimodal large language models can see but not perceive. In *European Conference on Computer Vision*, pages 148–166. Springer.

Google DeepMind. 2025. Gemini 3 pro. https://deepmind.google/models/gemini/pro/. Accessed: 2026-01-04.

Gavin Greif, Niclas Griesshaber, and Robin Greif. 2025. Multimodal llms for ocr, ocr post-correction, and named entity recognition in historical documents. *arXiv preprint arXiv:2504.00414*.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, and 1 others. 2024. Hallusionbench: an advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14375–14385.

Meng-Hao Guo, Jiajun Xu, Yi Zhang, Jiaxi Song, Haoyang Peng, Yi-Xuan Deng, Xinzhi Dong, Kiyohiro Nakayama, Zhengyang Geng, Chen Wang, and 1 others. 2025. Rbench: Graduate-level multidisciplinary benchmarks for llm & mllm complex reasoning evaluation. In *Forty-second International Conference on Machine Learning*.

Jiaxi Huang, Dongxu Wu, Hanwei Zhu, Lingyu Zhu, Jun Xing, Xu Wang, and Baoliang Chen. 2025a. Q-doc: Benchmarking document image quality assessment capabilities in multi-modal large language models. *arXiv preprint arXiv:2511.11410*.

Mingxin Huang, Yongxin Shi, Dezhi Peng, Songxuan Lai, Zecheng Xie, and Lianwen Jin. 2025b. Ocr-reasoning benchmark: Unveiling the true capabilities of mllms in complex text-rich image reasoning. *arXiv preprint arXiv:2505.17163*.

Anya Ji, Noriyuki Kojima, Noah Rush, Alane Suhr, Wai Keen Vong, Robert Hawkins, and Yoav Artzi. 2022. Abstract visual reasoning with tangram shapes. In *Proceedings of the 2022 conference on empirical methods in natural language processing*, pages 582–601.

Mengdi Jia, Zekun Qi, Shaochen Zhang, Wenyao Zhang, Xinqiang Yu, Jiawei He, He Wang, and Li Yi. 2025. Omnispatial: Towards comprehensive spatial reasoning benchmark for vision language models. *arXiv preprint arXiv:2506.03135*.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max W.F. Ku, Qian Liu, and Wenhu Chen. 2024. Mantis: Interleaved multi-image instruction tuning. *Transactions on Machine Learning Research*, 2024.

Jiayi Kuang, Yinghui Li, Chen Wang, Haohao Luo, Ying Shen, and Wenhao Jiang. 2025. Express what you see: Can multimodal llms decode visual ciphers with intuitive semiosis comprehension? In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 12743–12774.

Sicong Leng, Yun Xing, Zesen Cheng, Yang Zhou, Hang Zhang, Xin Li, Deli Zhao, Shijian Lu, Chunyan Miao, and Lidong Bing. 2024. The curse of multi-modalities: Evaluating hallucinations of large multimodal models across language, visual, and audio. *arXiv preprint arXiv:2410.12787*.

Kaixin Li, Ziyang Meng, Hongzhan Lin, Ziyang Luo, Yuchen Tian, Jing Ma, Zhiyong Huang, and Tat-Seng Chua. 2025. Screenspot-pro: Gui grounding for professional high-resolution computer use. In *Proceedings of the 33rd ACM International Conference on Multimedia*, pages 8778–8786.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. 2023. Evaluating object hallucination in large vision-language models. In *The 2023 Conference on Empirical Methods in Natural Language Processing*.

Yinghui Li, Zishan Xu, Shaoshen Chen, Haojing Huang, Yangning Li, Shirong Ma, Yong Jiang, Zhongli Li, Qingyu Zhou, Hai-Tao Zheng, and 1 others. 2024. Towards real-world writing assistance: A chinese character checking benchmark with faked and misspelled characters. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8656–8668.

Jingping Liu, Ziyan Liu, Zhedong Cen, Yan Zhou, Yinan Zou, Weiyan Zhang, Haiyun Jiang, and Tong Ruan. 2025. Can multimodal large language models understand spatial relations? *arXiv preprint arXiv:2505.19015*.

Ruiyang Liu, Yinghui Li, Linmi Tao, Dun Liang, and Hai-Tao Zheng. 2022. Are we ready for a new paradigm shift? a survey on visual deep mlp. *Patterns*, 3(7).

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, and 1 others. 2024. Mmbench: Is your multi-modal model an all-around player? In *European conference on computer vision*, pages 216–233. Springer.

Fan Lu, Wei Wu, Kecheng Zheng, Shuailei Ma, Biao Gong, Jiawei Liu, Wei Zhai, Yang Cao, Yujun Shen, and Zheng-Jun Zha. 2025. Benchmarking large vision-language models via directed scene graph for comprehensive image captioning. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19618–19627.

Wufei Ma, Haoyu Chen, Guofeng Zhang, Yu-Cheng Chou, Jieneng Chen, Celso de Melo, and Alan Yuille. 2025a. 3dsrbench: A comprehensive 3d spatial reasoning benchmark. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6924–6934.

Xinyu Ma, Ziyang Ding, Zhicong Luo, Chi Chen, Zonghao Guo, Derek F Wong, Xiaoyi Feng, and Maosong Sun. 2025b. Deepperception: Advancing r1-like cognitive visual perception in mllms for knowledge-intensive visual grounding. *arXiv preprint arXiv:2503.12797*.

Ahmed Masry, Mohammed Saidul Islam, Mahir Ahmed, Aayush Bajaj, Firoz Kabir, Aaryaman Kartha, Md Tahmid Rahman Laskar, Mizanur Rahman, Shadikur Rahman, Mehrad Shahmohammadi, Megh Thakkar, Md Rizwan Parvez, Enamul Hoque, and Shafiq Joty. 2025. ChartQAPro: A more diverse and challenging benchmark for chart question answering. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19123–19151, Vienna, Austria. Association for Computational Linguistics.

OpenAI. 2025. Introducing gpt-5.2. https://openai.com/index/introducing-gpt-5-2/. Accessed: 2026-01-04.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.

Roni Paiss, Ariel Ephrat, Omer Tov, Shiran Zada, Inbar Mosseri, Michal Irani, and Tali Dekel. 2023. Teaching clip to count to ten. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3170–3180.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, and 1 others. 2025. Humanity's last exam. *arXiv preprint arXiv:2501.14249*.

Christopher Rawles, Sarah Clinckemaillie, Yifan Chang, Jonathan Waltz, Gabrielle Lau, Marybeth Fair, Alice Li, William E Bishop, Wei Li, Folawiyo Campbell-Ajala, and 1 others. 2025. Androidworld: A dynamic benchmarking environment for autonomous agents. In *The Thirteenth International Conference on Learning Representations*.

Jonathan Roberts, Mohammad Reza Taesiri, Ansh Sharma, Akash Gupta, Samuel Roberts, Ioana Croitoru, Simion-Vlad Bogolin, Jialu Tang, Florian Langer, Vyas Raina, and 1 others. 2025. Zerobench: An impossible visual benchmark for contemporary large multimodal models. *arXiv preprint arXiv:2502.09696*.

Fedor Rodionov, Abdelrahman Eldesokey, Michael Birsak, John Femiani, Bernard Ghanem, and Peter Wonka. 2025. Floorplanqa: A benchmark for spatial reasoning in llms using structured representations. *arXiv preprint arXiv:2507.07644*.

Kexian Tang, Junyao Gao, Yanhong Zeng, Haodong Duan, Yanan Sun, Zhening Xing, Wenran Liu, Kaifeng Lyu, and Kai Chen. 2025a. Lego-puzzles: How good are mllms at multi-step spatial reasoning? *arXiv preprint arXiv:2503.19990*.

Wei Tang, Yanpeng Sun, Qinying Gu, and Zechao Li. 2025b. Visual position prompt for mllm based visual grounding. *arXiv preprint arXiv:2503.15426*.

Fei Wang, Xingyu Fu, James Y Huang, Zekun Li, Qin Liu, Xiaogeng Liu, Mingyu Derek Ma, Nan Xu, Wenxuan Zhou, Kai Zhang, and 1 others. 2025a. Muirbench: A comprehensive benchmark for robust multi-image understanding. In *The Thirteenth International Conference on Learning Representations*.

Wei Wang, Zhaowei Li, Qi Xu, Linfeng Li, YiQing Cai, Botian Jiang, Hang Song, Xingcan Hu, Pengyu Wang, and Li Xiao. 2025b. Advancing fine-grained

visual understanding with multi-scale alignment in multi-modal models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14282–14301.

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, and 1 others. 2024. Charxiv: Charting gaps in realistic chart understanding in multimodal llms. *Advances in Neural Information Processing Systems*, 37:113569–113697.

Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. Deepseek-ocr: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.

Haoning Wu, Xiao Huang, Yaohui Chen, Ya Zhang, Yanfeng Wang, and Weidi Xie. 2025a. Spatialscore: Towards unified evaluation for multimodal spatial understanding. *arXiv preprint arXiv:2505.17012*.

Junda Wu, Yu Xia, Tong Yu, Xiang Chen, Sai Sree Harsha, Akash V Maharaj, Ruiyi Zhang, Victor Bursztyn, Sungchul Kim, Ryan A Rossi, and 1 others. 2025b. Doc-react: Multi-page heterogeneous document question-answering. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 67–78.

Zhiyu Wu, Xiaokang Chen, Zizheng Pan, Xingchao Liu, Wen Liu, Damai Dai, Huazuo Gao, Yiyang Ma, Chengyue Wu, Bingxuan Wang, and 1 others. 2024. Deepseek-vl2: Mixture-of-experts vision-language models for advanced multimodal understanding. *arXiv preprint arXiv:2412.10302*.

xAI. 2024. Realworldqa. https://huggingface.co/datasets/xai-org/RealworldQA. Accessed: 2025-12-30.

Renqiu Xia, Hancheng Ye, Xiangchao Yan, Qi Liu, Hongbin Zhou, Zijun Chen, Botian Shi, Junchi Yan, and Bo Zhang. 2025. Chartx & chartvlm: A versatile benchmark and foundation model for complicated chart reasoning. *IEEE Transactions on Image Processing*.

Han Xiao, Guozhi Wang, Yuxiang Chai, Zimu Lu, Weifeng Lin, Hao He, Lue Fan, Liuyang Bian, Rui Hu, Liang Liu, and 1 others. 2025. Ui-genie: A self-improving approach for iteratively boosting mllm-based mobile gui agents. *arXiv preprint arXiv:2505.21496*.

Yijia Xiao, Edward Sun, Tianyu Liu, and Wei Wang. 2024. Logicvista: Multimodal llm logical reasoning benchmark in visual contexts. *arXiv preprint arXiv:2407.04973*.

Tianbao Xie, Jiaqi Deng, Xiaochuan Li, Junlin Yang, Haoyuan Wu, Jixuan Chen, Wenjing Hu, Xinyuan Wang, Yuhui Xu, Zekun Wang, and 1 others. 2025. Scaling computer-use grounding via user interface decomposition and synthesis. *arXiv preprint arXiv:2505.13227*.

Peiran Xu, Sudong Wang, Yao Zhu, Jianing Li, and Yunjian Zhang. 2025a. Spatialbench: Benchmarking multimodal large language models for spatial cognition. *arXiv preprint arXiv:2511.21471*.

Rui Xu, Dakuan Lu, Zicheng Zhao, Xiaoyu Tan, Xintao Wang, Siyu Yuan, Jiangjie Chen, and Xu Yinghui. 2025b. Origamispace: Benchmarking multimodal llms in multi-step spatial reasoning with mathematical constraints. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yunqiu Xu, Linchao Zhu, and Yi Yang. 2025c. Mcbench: A benchmark for multi-context visual grounding in the era of mllms. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 17675–17687.

Fernanda Miyuki Yamada, Harlen Costa Batagelo, João Paulo Gois, and Hiroki Takahashi. 2025. Tangan: solving tangram puzzles using generative adversarial network. *Applied Intelligence*, 55(6):1–27.

Sihan Yang, Runsen Xu, Yiman Xie, Sizhe Yang, Mo Li, Jingli Lin, Chenming Zhu, Xiaochen Chen, Haodong Duan, Xiangyu Yue, and 1 others. 2025a. Mmsi-bench: A benchmark for multi-image spatial intelligence. *arXiv preprint arXiv:2505.23764*.

Songyuan Yang, Weijiang Yu, Wenjing Yang, Xinwang Liu, Huibin Tan, Long Lan, and Nong Xiao. 2025b. Wildvideo: Benchmarking lmms for understanding video-language interaction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Jiakang Yuan, Tianshuo Peng, Yilei Jiang, Yiting Lu, Renrui Zhang, Kaituo Feng, Chaoyou Fu, Tao Chen, Lei Bai, Bo Zhang, and 1 others. 2025. Mme-reasoning: A comprehensive benchmark for logical reasoning in mllms. *arXiv preprint arXiv:2505.21327*.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, and 1 others. 2025. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15134–15186.

Z.ai. 2025. Glm-4.6v: Open source multimodal models with native tool use. https://z.ai/blog/glm-4.6v. Accessed: 2025-01-04.

Chao Zhang, Jiamin Tang, and Jing Xiao. 2024. Tangram: Benchmark for evaluating geometric element recognition in large multimodal models. *arXiv preprint arXiv:2408.13854*.

Xuanle Zhao, Xuexin Liu, Haoyue Yang, Xianzhen Luo, Fanhu Zeng, Jianling Li, Qi Shi, and Chi Chen. 2025. Chartedit: How far are mllms from automating chart analysis? evaluating mllms' capability via chart editing. *arXiv preprint arXiv:2505.11935*.

Yizhou Zhao, Liang Qiu, Pan Lu, Feng Shi, Tian Han, and Song-Chun Zhu. 2022. Learning from the tangram to solve mini visual tasks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pages 3490–3498.

Longtao Zheng, Zhiyuan Huang, Zhenghai Xue, Xinrun Wang, Bo An, and Shuicheng YAN. 2025a. Agentstudio: A toolkit for building general virtual agents. In *The Thirteenth International Conference on Learning Representations*.

Xu Zheng, Zihao Dongfang, Lutao Jiang, Boyuan Zheng, Yulong Guo, Zhenquan Zhang, Giuliano Albanese, Runyi Yang, Mengjiao Ma, Zixin Zhang, and 1 others. 2025b. Multimodal spatial reasoning in the large model era: A survey and benchmarks. *arXiv preprint arXiv:2510.25760*.

Liangyu Zhong, Fabio Rosenthal, Joachim Sicking, Fabian Hüger, Thorsten Bagdonat, Hanno Gottschalk, and Leo Schwinn. 2025. Focus: Internal mllm representations for efficient fine-grained visual question answering. *arXiv preprint arXiv:2506.21710*.

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

A Comparison with Existing Spatial Benchmarks

To better contextualize our contributions, we provide a detailed comparison between TangramPuzzle and representative spatial reasoning benchmarks in Table 4. Compared to prior datasets, TangramPuzzle differs in four main aspects.

1. From discriminative VQA to inverse generation with verifiable structure: While recent benchmarks have significantly advanced spatial evaluation, they each emphasize different facets: OmniSpatial (Jia et al., 2025) focuses on broad coverage (spanning 4 major categories and 50 subcategories), MMSI-Bench (Yang et al., 2025a) highlights camera/object/region relations in multi-view real-world scenes, and LEGO-Puzzles (Tang et al., 2025a) targets 3D assembly sequences. Furthermore, prior Tangram datasets (Zhang et al., 2024) similarly focus on identifying and counting basic geometric elements (e.g., points, lines, circles, triangles). However, these benchmarks are predominantly formulated as Visual Question Answering (VQA) tasks. While valuable, VQA tasks can encourage shortcut behaviors (e.g., exploiting language priors) and do not require the model to construct a globally consistent structure. In contrast, our TCE framework provides a standardized, machine-verifiable symbolic bridge for compositional reasoning. By utilizing exact LaTeX algebraic expressions, we eliminate linguistic ambiguity and enable objective, automated verification.

2. Complementary focus to ORIGAMISPACE: Although ORIGAMISPACE (Xu et al., 2025b) is also explicitly generative, it fundamentally targets folding and deformation reasoning. It focuses on procedural manipulation by transforming a 2D crease pattern into a 3D folded shape through multi-step folding, and it relies heavily on an origami compiler. Additionally, these tasks often rely on task-specific answer formats or final-choice accuracy rather than exact symbolic representations of all component geometries. Conversely, TangramPuzzle requires a "global-to-local" decomposition where models must mentally simulate and inversely assemble a complete structure from scratch under strict geometric constraints. Thus, the two benchmarks evaluate complementary dimensions of spatial reasoning.

3. Fine-grained diagnosis to prevent "visual cheating": TangramPuzzle introduces TCE as a formal representation alongside a hierarchical verifier, which precisely categorizes failures into syntax errors, rigid-geometry errors, physical errors, and silhouette mismatches. This provides much deeper diagnostic value than a single accuracy score. For instance, our experiments reveal that models sometimes assemble visually plausible silhouettes by violating piece rigidity or allowing overlaps. This type of "visual cheating" is easily overlooked in benchmarks that only evaluate semantic answers or coarse visual similarities.

4. Controlled 2D environment to isolate core spatial reasoning: While 3D tasks add realism, they inevitably introduce confounds such as viewpoint variation, occlusion, depth ambiguity, and simulator assumptions. TangramPuzzle intentionally adopts a minimal yet expressive 2D abstraction governed by strict Euclidean and topological constraints. This design choice directly isolates the core challenge of compositional spatial reasoning, making model failures clear, reproducible, and automatically verifiable.

Table 4: Summary comparison between TangramPuzzle and representative spatial reasoning benchmarks.

Benchmark	Visual Domain	Task Type	Evaluated Capability	Evaluation Protocol	Geometric Formalism	Constraint Rigor
OmniSpatial (Jia et al., 2025)	2D / 3D scenes	Discriminative VQA	Dynamic, interaction, and perspective	LLM / text matching	Implicit natural language	Relative and qualitative
MMSI-Bench (Yang et al., 2025a)	3D multi-image scenes	Discriminative VQA	Cross-image spatial relations	Text matching / MCQ	Implicit natural language	Relative and qualitative
LEGO-Puzzles (Tang et al., 2025a)	3D synthetic scenes	Discriminative VQA	Sequential spatial reasoning	LLM / text matching	Implicit natural language	Relative and qualitative
Tangram (Zhang et al., 2024)	2D / 3D diagrams	Discriminative VQA	Geometric element counting	Text matching	Implicit natural language	Perceptual counting
ORIGAMISPACE (Xu et al., 2025b)	2D $\rightarrow$ 3D origami	Generative code	Multi-step procedural reasoning	MCQ / specialized compiler	Domain-specific rules	Local foldability
TangramPuzzle	**2D planar**	**Inverse generative**	**Compositional spatial reasoning**	**Deterministic math parsing**	**Explicit algebraic coordinates via TCE**	**Absolute global non-overlap**

B Detailed Definition of Tangram Construction Expression (TCE)

The Tangram Construction Expression (TCE) represents each tangram instance in a structured and machine-verifiable form. It consists of five core components: instance_id, target_outline, initial_state, final_state, and adjacency_graph. Their detailed definitions are given below.

The field **instance_id** uniquely identifies each puzzle instance and serves as a stable key for indexing, retrieval, and evaluation. Both **initial_state** and **final_state** describe the geometric states of the seven tangram pieces before and after assembly. In TCE, each piece is specified by its piece type, a set of vertex coordinates, explicit edge relations, and a center point. In addition, final_state includes a transform_matrix that parameterizes the rigid motion applied to each piece to reach the assembled configuration, including translation, rotation angle, and optional reflection. Because tangram geometries frequently involve irrational numbers, finite decimal representations induce precision loss. To ensure geometric rigor, TCE encodes all quantities as exact algebraic expressions in LaTeX. The **target_outline** specifies the global silhouette produced by the assembled tangram. It is represented by the outline's vertex coordinates together with its edge relations. The **adjacency_graph** captures local interactions between pieces in the final configuration, indicating which pieces are adjacent.

C Human Annotation Details

Recruitment and Payment. To ensure the high quality and rigorous geometric precision of the TangramPuzzle benchmark, we invited researchers with backgrounds in geometry and computer science to conduct the data annotation and verification. We strictly adhered to local labor standards regarding compensation. The payment rate was determined based on the estimated time required to complete the annotation for each puzzle, ensuring that the hourly rate met or exceeded the local minimum wage requirements.

Annotation Process. Annotators used our interactive tangram tool to assemble each target configuration by dragging, rotating, and flipping pieces. The annotation interface is shown in Figure 5. To avoid pixel-level misalignment, the tool incorporates a magnetic snapping mechanism that automatically aligns piece vertices to the global coordinate grid or to vertices of adjacent pieces, ensuring that spatial relationships are mathematically exact rather than visually approximate. After completing an instance, annotators exported the generated data, and we conducted additional visual inspection and re-annotation when necessary (e.g., minor misalignment, unintended overlaps, or disconnected components) to ensure geometric correctness.

Instructions and Consent. Prior to the data collection process, all annotators were provided with comprehensive instructions detailing the usage of the annotation interface, the definition of the Tangram Construction Expression (TCE), and the strict physical constraints (e.g., non-overlap, boundary

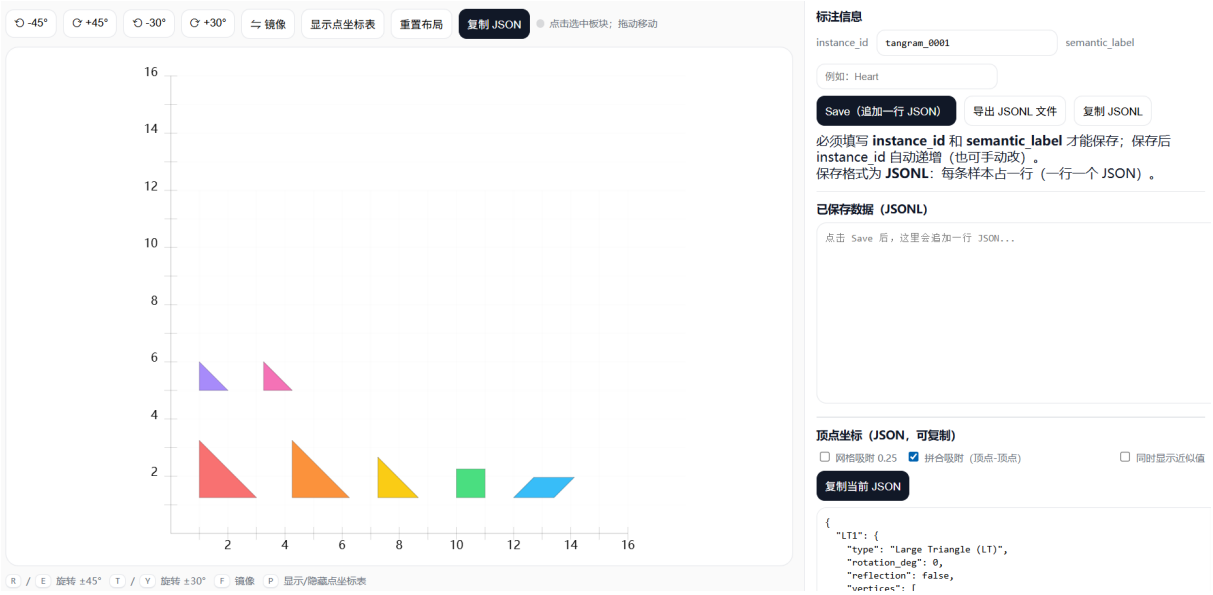

Figure 5: Annotation interface for TangramPuzzle.

adherence). We obtained informed consent from all participants, who were explicitly informed that the annotated data would be used for academic research and released to the public.

D Detailed Geometric Normalization

To ensure the resulting dataset is machine-verifiable and free from floating-point ambiguity, we implemented a rigorous post-processing pipeline to convert raw manual annotations into the symbolic TCE format. This process consists of three key steps:

1. Reference Alignment and Raw Capture. First, annotators utilize our custom web-based tool to manually reconstruct the tangram patterns from the source KiloGram dataset (Ji et al., 2022). By dragging, rotating, and snapping the seven virtual pieces, annotators align the pieces to visually match the reference image. Upon saving, the tool records the raw floating-point coordinates of all vertices for the seven constituent pieces in their current assembled state.

2. Canonical Placement. To standardize the coordinate system across all instances and eliminate arbitrary canvas positioning, we apply a translation operation to the raw assembly. Let $\mathcal{V} = \{(x_i, y_i)\}_{i=1}^{N}$ denote the set of all vertex coordinates in the raw annotation. We compute the global minima of the assembly:

$$x_{min} = \min_{(x,y)\in\mathcal{V}} x, \quad y_{min} = \min_{(x,y)\in\mathcal{V}} y \tag{1}$$

We then translate every vertex (x, y) to a new coordinate (x', y'):

$$x' = x - x_{min}, \quad y' = y - y_{min} \tag{2}$$

This operation ensures that the assembly's lower boundary lies strictly on the x-axis and its left boundary lies on the y-axis, resulting in a standardized, positive-quadrant layout.

3. Symbolic Conversion and TCE Generation. To support exact geometric reasoning and eliminate floating-point ambiguity, we convert the normalized coordinates into precise algebraic expressions represented in LaTeX format.

We employ a deterministic mapping algorithm to convert decimal approximations into exact rational or irrational numbers. For instance, a coordinate 0.5 is converted to `\frac{1}{2}`, and irrational numbers resulting from 45° rotations (e.g., $1.414\ldots$) are resolved to their symbolic forms (e.g., `\sqrt{2}`). This ensures exact geometric relations (e.g., edge equality).

With the exact coordinates of the seven pieces established (forming the `final_state`), we computationally extract the global geometry. This includes calculating the geometric union of the pieces to extract the `target_outline` (effectively dissolving internal edges) and generating an adjacency graph by identifying shared vertex pairs and overlapping edge segments. Finally, these components are serialized into the unified TCE JSON format, providing a mathematically perfect ground truth for model evaluation.

4. Task 1 Multiple-Choice Construction. To generate samples for Task 1, we leverage the geometric information encoded in TCE. For each tangram instance, we reconstruct the ground-truth silhouette from its TCE annotation. We then construct the multiple-choice options by randomly sampling three distinct silhouettes from the remaining dataset as distractors. These distractors are combined with the ground-truth silhouette, randomly shuffled, and rendered into a single candidate image labeled A through D. This automated pipeline ensures that the correct silhouette is geometrically consistent with the coordinate-based input while maintaining visual diversity among the options.

E Detailed Evaluation Protocols

This section provides the formal definitions and precise calculation details for the metrics used in our evaluation.

E.1 Task 1 Metrics

Accuracy (Acc) Performance is measured using classification accuracy:

$$\mathrm{Acc} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{a}_i = a_i)\,, \tag{3}$$

where $a_i \in \mathcal{A}$ denotes the ground-truth option for the i-th instance, $\hat{a}_i \in \mathcal{A}$ is the model's predicted option, $\mathcal{A}$ is the option set (e.g., $\{A, B, C, D\}$), and $\mathbb{I}(\cdot)$ is the indicator function.

Invalid Rate We further define the Invalid rate as the proportion of instances in which the model output does not yield any valid option that can be mapped to the candidate set. Invalid outputs may arise from formatting errors or from generating content unrelated to the candidate options. Formally:

$$\mathrm{Invalid} = \frac{1}{N}\sum_{i=1}^{N} \mathbb{I}(\hat{a}_i \notin \mathcal{A})\,. \tag{4}$$

E.2 Task 2 Metrics

Let $\mathcal{P} = \{P_1, \ldots, P_7\}$ denote the set of predicted piece polygons after assembly, and let $\mathcal{T}$ denote the ground-truth target silhouette. The evaluation pipeline consists of two stages: constraint validation and silhouette quality assessment.

Constraint Validation We consider three types of constraint errors:

- **TSE (Syntax Error)** checks whether the predicted output conforms to the required TCE format. A prediction is flagged as TSE if the JSON cannot be parsed, the number of pieces is incorrect, or piece types are invalid:

$$\mathrm{TSE} = \neg\mathrm{Valid}_{\mathrm{syn}}(\mathcal{P}), \tag{5}$$

  where $\mathrm{Valid}_{\mathrm{syn}}(\mathcal{P})$ denotes syntactic validity under the TCE schema.

- **RGE (Rigid Geometry Error)** verifies that each predicted piece preserves its original shape. For each piece, we compare its area and perimeter against the corresponding ground-truth specification. Let P_i be a simple polygon with ordered vertices $\{\mathbf{v}_1, \ldots, \mathbf{v}_n\}$. Its area is computed as:

$$\mathrm{Area}(P_i) = \frac{1}{2}\left|\sum_{k=1}^{n}(x_k y_{k+1} - x_{k+1} y_k)\right|, \tag{6}$$

  and its perimeter is given by

$$\mathrm{Perimeter}(P_i) = \sum_{k=1}^{n} \|\mathbf{v}_{k+1} - \mathbf{v}_k\|_2\,, \tag{7}$$

  where $\mathbf{v}_{n+1} = \mathbf{v}_1$. A rigid geometry error is triggered if either quantity deviates from that of the corresponding canonical tangram piece.

- **PE (Physical Error)** evaluates physical feasibility. A prediction is considered invalid if any two pieces overlap or if the union of all pieces is not a single connected shape.

Shape Similarity We adopt two complementary shape similarity metrics:

- **IoU (Intersection over Union)** measures the overlap between the predicted assembly U and the target silhouette $\mathcal{T}$:

$$\mathrm{IoU}(U, \mathcal{T}) = \frac{\mu(U \cap \mathcal{T})}{\mu(U \cup \mathcal{T})}, \tag{8}$$

  where $\mu(\cdot)$ denotes planar area.

- **Hausdorff Distance (d_H)** To capture boundary-level deviations, we compute the Hausdorff distance between the boundaries of U and $\mathcal{T}$:

$$d_H(\partial U, \partial \mathcal{T}) = \max \left\{ \begin{array}{l} \sup_{x \in \partial U} \inf_{y \in \partial \mathcal{T}} \|x - y\|_2, \\ \sup_{y \in \partial \mathcal{T}} \inf_{x \in \partial U} \|x - y\|_2 \end{array} \right\}. \quad (9)$$

Here ∂U and $\partial \mathcal{T}$ denote the boundaries of the predicted assembly and the target silhouette, respectively, and $\| \cdot \|_2$ represents the standard Euclidean distance.

F Details on Baselines

We utilize the official chat templates and default generation parameters for all models to ensure a fair comparison.

To contextualize the experimental results, we examine the released model cards and technical reports of the open-source baselines evaluated in this work. These models are generally trained on multimodal mixtures, including general image understanding, OCR, document question answering, chart interpretation, and conventional spatial-relation VQA. However, to the best of our knowledge, their released training descriptions report no training data in the same format as our TCE-style tangram inverse-assembly task, which requires generating exact algebraic coordinates under rigid-body, non-overlap, and connectivity constraints.

For closed-source models, such as Gemini3-Pro, GPT-5.2, and Claude-Sonnet-4.5, the exact training corpora are not publicly available, and we therefore cannot rule out possible exposure to tangram-like visual patterns. Nevertheless, our Tangram Construction Expression (TCE) is newly designed for this benchmark and represents tangram assemblies as strict, machine-verifiable JSON structures with exact symbolic coordinates. This makes direct memorization of TCE-formatted solutions unlikely, even if some models may have encountered standard tangram images or natural-language tangram descriptions during pre-training.

Therefore, the results should be interpreted not as a claim about training-data purity for every model, but as evidence that current MLLMs, especially those trained mainly on semantic recognition and coarse spatial relations, still struggle with constraint-based compositional spatial reasoning.

G Prompt Templates

To ensure a fair comparison, all prompts used in our experiments are setting-specific but model-agnostic. Within each experimental setting, all models are evaluated using exactly the same prompt template, with no model-specific prompt engineering or optimization. The placeholders in the following listings, such as `{target_outline_str_full}`, `{initial_pieces_str}`, `{target_str}`, `{pieces_str}`, and `{few_shot_section}`, are replaced by the corresponding instance-specific data at inference time. For multimodal settings, the target image is provided as the visual input together with the textual prompt. The detailed prompt templates for each experimental setting are presented as follows.

Listing 1: Base prompt template for the standard end-to-end Tangram solution generation setting.

```
You are a Tangram Solver Expert. Your task
     is to solve the puzzle by arranging the
     7 given tangram pieces to perfectly fill
      the target outline shown in the
     provided image.

Input Data:
1. Target Outline Geometry (Vertices & Edges
     ): {target_outline_str_full}
2. Initial Pieces Specifications: {
     initial_pieces_str}

Task Requirements:
Determine the final position for each of the
      7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
      are FIXED and cannot be modified (rigid
      transformation only).
The objective is to arrange them so that
     they completely fill the target area,
     forming a shape whose outer boundary is
     identical to the target outline.
There must be no overlaps between pieces and
      no gaps inside the shape.

Output Format:
Begin by providing your step-by-step spatial
      reasoning analysis of how you fit the
     pieces together. Then, at the very end
     of your response, output the final
     solution as a single JSON object wrapped
      in a json code block. The JSON
     structure must strictly follow this
     schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
      "ID", "piece_type": "Type", "
     transform_matrix": { "translation": {"dx
     ": 0.0, "dy": 0.0}, "rotation_deg": 0.0,
      "reflection": false }, "vertices": [ {"
     coords": [x, y], "symbolic": ["val_x", "
     val_y"]} ] } ] } }
```
```

Listing 2: Prompt template for the in-context learning setting. The solved examples are prepended before the current query while keeping the same task requirements and output schema.

```
You are a Tangram Solver Expert. Your task
    is to solve the puzzle by arranging the
    7 given tangram pieces to perfectly fill
     the target outline shown in the
    provided image.

Few-Shot Examples
{few_shot_section}

Current Task
Now, please solve the following specific
    task based on the image and data
    provided below:

Current Task Input Data:
1. Target Image Reference: <image>
2. Target Outline Vertices Coordinates and
    Edges: {target_str}
3. Initial Pieces Specifications: {
    pieces_str}

Task Requirements:
Determine the final position for each of the
     7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
     are fixed and cannot be modified (rigid
     transformation only).
The objective is to arrange them so that
    they completely fill the target area,
    forming a shape whose outer boundary is
    identical to the target outline.
There must be no overlaps between pieces and
     no gaps inside the shape.

Output Format:
Begin by providing your step-by-step spatial
     reasoning analysis of how you fit the
    pieces together. Then, at the very end
    of your response, output the final
    solution as a single JSON object wrapped
     in a json code block. The JSON
    structure must strictly follow this
    schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
     "ID", "piece_type": "Type", "vertices":
     [ {"coords": ["latex_x", "latex_y"]} ],
     "edges": [ {"coords": [["latex_x1", "
    latex_y1"], ["latex_x2", "latex_y2"]], "
    length": "latex_len"} ] } ] } }
```
```

Listing 3: Prompt template for the visual-centric setting, where the textual target outline geometry is removed.

```
You are a Tangram Solver Expert. Your task
    is to solve the puzzle by arranging the
    7 given tangram pieces to perfectly fill
     the target outline shown in the
    provided image.

Input Data:
1. Please carefully analyze the provided
    image, where the target outline and
    relevant vertex coordinates are
    explicitly displayed.
2. Initial Pieces Specifications: {
    pieces_str}

Task Requirements:
Determine the final position for each of the
     7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
     are fixed and cannot be modified (rigid
     transformation only).
The objective is to arrange them so that
    they completely fill the target area,
    forming a shape whose outer boundary is
    identical to the target outline.
There must be no overlaps between pieces and
     no gaps inside the shape.

Output Format:
Begin by providing your step-by-step spatial
     reasoning analysis of how you fit the
    pieces together. Then, at the very end
    of your response, output the final
    solution as a single JSON object wrapped
     in a json code block. The JSON
    structure must strictly follow this
    schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
     "ID", "piece_type": "Type", "vertices":
     [ {"coords": ["latex_x", "latex_y"]} ],
     "edges": [ {"coords": [["latex_x1", "
    latex_y1"], ["latex_x2", "latex_y2"]], "
    length": "latex_len"} ] } ] } }
```
```

Listing 4: Prompt template for the chain-of-thought setting, where models are explicitly instructed to reason through silhouette analysis, piece placement, detail filling, and constraint verification before producing the final JSON output.

```
You are a Tangram Solver Expert. Your task
    is to solve the puzzle by arranging the
    7 given tangram pieces to perfectly fill
     the target outline shown in the
    provided image.

Input Data:
1. Target Outline Vertices Coordinates and
    Edges: {target_str}
2. Initial Pieces Specifications: {
    pieces_str}

Task Requirements:
Determine the final position for each of the
     7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
     are fixed and cannot be modified (rigid
     transformation only).
The objective is to arrange them so that
    they completely fill the target area,
    forming a shape whose outer boundary is
    identical to the target outline.
There must be no overlaps between pieces and
```

```
      no gaps inside the shape.

Chain-of-Thought Reasoning:
Before generating the final coordinate JSON,
      you must solve this step-by-step:
1. Analyze the Silhouette: Identify key
     features of the target outline (e.g.,
     right angles, prominent corners, total
     width/height). Calculate or estimate the
      bounding box.
2. Strategize Placement: Decide where to
     place the two Large Triangles first, as
     they occupy the most space and usually
     define the main structure.
3. Fill the Details: Determine the positions
      of the Medium Triangle, Small Triangles
     , Square, and Parallelogram to fill the
     remaining gaps. Explain how their edges
     align with the target outline or other
     pieces.
4. Verify Constraints: Check if all 7 pieces
      are used, ensure there are no overlaps,
      and confirm the total shape matches the
      target outline coordinates.

Output Format:
Begin by providing your step-by-step spatial
      reasoning analysis of how you fit the
     pieces together. Then, at the very end
     of your response, output the final
     solution as a single JSON object wrapped
      in a json code block. The JSON
     structure must strictly follow this
     schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
      "ID", "piece_type": "Type", "vertices":
      [ {"coords": ["latex_x", "latex_y"]} ],
      "edges": [ {"coords": [["latex_x1", "
     latex_y1"], ["latex_x2", "latex_y2"]], "
     length": "latex_len"} ] } ] } }
```
```

Listing 5: Prompt template for the decomposition-based setting, where models reason using the small isosceles right triangle as the basic geometric unit.

```
You are a Tangram Solver Expert. Your task
     is to solve the puzzle by arranging the
     7 given tangram pieces to perfectly fill
      the target outline shown in the
     provided image.

Input Data:
1. Target Outline Vertices Coordinates and
     Edges: {target_str}
2. Initial Pieces Specifications: {
     pieces_str}

Task Requirements:
Determine the final position for each of the
      7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
      are fixed and cannot be modified (rigid
      transformation only).
The objective is to arrange them so that
     they completely fill the target area,
     forming a shape whose outer boundary is
     identical to the target outline.
There must be no overlaps between pieces and
      no gaps inside the shape.

Recommended Solving Strategy: Unit
     Decomposition
Consider solving this puzzle by breaking
     down the complex shapes into a common
     geometric basis:
1. Identify the Basic Unit: Interpret the
     puzzle using a single building block:
     the small isosceles right triangle (the
     'Basic Unit').
2. Quantify the Pieces: Note that every
     tangram piece is formed by combining
     these Basic Units:
   - Small Triangle = 1 Unit
   - Square, Medium Triangle, Parallelogram
     = 2 Units each
   - Large Triangle = 4 Units each
3. Decompose the Target: Mentally overlay a
     grid of these Basic Units onto the
     Target Outline. Try to partition the
     silhouette into regions that match the
     unit counts of the pieces (e.g., find a
     4-unit region for a Large Triangle).
4. Map and Assemble: Assign the pieces to
     these decomposed regions based on their
     unit size and shape.

Output Format:
Begin by providing your step-by-step spatial
      reasoning analysis of how you fit the
     pieces together. Then, at the very end
     of your response, output the final
     solution as a single JSON object wrapped
      in a json code block. The JSON
     structure must strictly follow this
     schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
      "ID", "piece_type": "Type", "vertices":
      [ {"coords": ["latex_x", "latex_y"]} ],
      "edges": [ {"coords": [["latex_x1", "
     latex_y1"], ["latex_x2", "latex_y2"]], "
     length": "latex_len"} ] } ] } }
```
```

Listing 6: Prompt template for the text-only setting, where the target image is removed.

```
You are a Tangram Solver Expert. Your task
     is to solve the puzzle by arranging the
     7 given tangram pieces to perfectly fill
      the specified target outline based on
     the coordinates provided below.

Input Data:
1. Target Outline Vertices Coordinates and
     Edges: {target_str}
2. Initial Pieces Specifications: {
     pieces_str}

Task Requirements:
Determine the final position for each of the
      7 pieces.
Important Constraints:
The geometric shape and size of the 7 pieces
      are fixed and cannot be modified (rigid
```

```
     transformation only).
The objective is to arrange them so that
     they completely fill the target area,
     forming a shape whose outer boundary is
     identical to the target outline.
There must be no overlaps between pieces and
      no gaps inside the shape.

Output Format:
Begin by providing your step-by-step spatial
      reasoning analysis of how you fit the
     pieces together. Then, at the very end
     of your response, output the final
     solution as a single JSON object wrapped
      in a json code block. The JSON
     structure must strictly follow this
     schema:
```json
{ "final_state": { "pieces": [ { "piece_id":
      "ID", "piece_type": "Type", "vertices":
      [ {"coords": ["latex_x", "latex_y"]} ],
      "edges": [ {"coords": [["latex_x1", "
     latex_y1"], ["latex_x2", "latex_y2"]], "
     length": "latex_len"} ] } ] } }
```
```

Listing 7: Prompt template for the TCE-light setting, where models output only the rigid transformation parameters rather than full vertices and edges.

```
You are a Tangram Solver Expert. Your task
     is to solve the puzzle by arranging the
     7 given tangram pieces to perfectly fill
      the target outline shown in the
     provided image.

Input Data:
1. Target Outline Vertices Coordinates and
     Edges: {target_str}
2. Initial Pieces Specifications: {
     pieces_str}

Task Requirements:
Determine one rigid transformation for each
     of the 7 pieces.

Important Constraints:
The geometric shape and size of each piece
     are fixed and cannot be modified.
Only rigid transformations are allowed:
     rotation, optional reflection, and
     translation.
The pieces must completely fill the target
     area, forming a shape whose outer
     boundary is identical to the target
     outline.
There must be no overlaps between pieces and
      no gaps inside the shape.

Transformation Convention:
For every vertex p=[x,y] of an initial piece,
      its final position is computed as:
p_final = R(theta) * S(reflection) *
     p_initial + translation.
Here R(theta) is a counter-clockwise
     rotation by rotation_degrees.
If reflection is true, S maps [x,y] to [-x,y
     ] before rotation; otherwise S is the
     identity transform.
translation is [tx, ty] and is added after
     reflection and rotation.
rotation_degrees should be given in degrees.
translation can be given as decimal numbers
     or LaTeX-style expressions.

Output Format:
Begin by providing your step-by-step spatial
      reasoning analysis of how you fit the
     pieces together. Then, at the very end
     of your response, output the final
     solution as a single JSON object wrapped
      in a json code block.
Do not output vertices or edges. Output only
      piece_id, rotation_degrees, reflection,
      and translation.
The JSON structure must strictly follow this
      schema:
```json
{
  "final_state": {
    "pieces": [
      {
        "piece_id": "ID",
        "rotation_degrees": 0,
        "reflection": false,
        "translation": ["latex_or_decimal_tx
     ", "latex_or_decimal_ty"]
      }
    ]
  }
}
```
```

H Detailed Experimental Results

H.1 Experimental Results for Task 1

The results for Task 1 are summarized in Table 1, revealing substantial performance differences across models. Task 1 serves as a lower-difficulty diagnostic that probes global-shape inference before the more demanding inverse assembly problem in Task 2. Specifically, this task assesses the model's ability to mentally compose a silhouette from disjointed parts and identify the correct visual match accurately. Notably, many models perform poorly on this task, suggesting that Task 1 cannot be reliably solved through simple memorization or superficial pattern matching; instead, it requires genuine compositional spatial reasoning based on part–whole structure. Among open-source baselines, Qwen3-VL-32B-Instruct stands out with an accuracy of 73.05%, approaching the performance of the closed-source model GPT-5.2, while smaller or earlier architectures struggle to exceed 40% accuracy. For commercial models, Gemini3-Pro achieves near-ceiling performance with an accuracy of 98.65%, underscoring its superior capacity for fine-grained visual perception and holistic mental shape composition regarding global silhouettes.

For most models, the rate of invalid outputs remains low, suggesting a strong ability to follow instructions and produce interpretable decisions. However, DeepSeek-OCR and DeepSeek-VL2 exhibit noticeably higher invalid rates, suggesting that under the stress of this complex spatial reasoning task, these models often fail to form a decisive conclusion or collapse into generating irrelevant content.

H.2 Experimental Results for Task 2

Table 2 presents the quantitative results for Task 2. Although **Success** is zero for most models, the intermediate metrics reveal why they fail. Most models achieve relatively low **TSE** scores, typically at the single-digit level, indicating that they can generally produce outputs in the required TCE-style JSON format. This suggests that the main bottleneck is not TCE formatting itself, but compositional spatial reasoning. To avoid over-penalizing minor formatting issues, our evaluation pipeline incorporates a lightweight recovery mechanism: we use `json_repair` to fix malformed JSON and regex-based heuristics to resolve common formatting defects in LaTeX-style mathematical expressions. This ensures that we extract and evaluate the model's true geometric intent.

By contrast, constraint-related errors remain widespread. In particular, **Rigid Geometry Errors (RGE)** and **Physical Errors (PE)** are highly prevalent, indicating that models struggle to comply with the fundamental geometric rules of the puzzle. One notable observation is the pronounced discrepancy between silhouette quality and the Success rate. Top-tier models like Claude-Sonnet-4.5 and GPT-5.2 achieve high IoU scores yet fail to produce a single valid solution (0% Success). Further inspection reveals that the accompanying high RGE and PE scores indicate that this apparent visual fidelity is achieved through "cheating," namely by impermissibly distorting rigid pieces or overlapping them to force a visual match. Meanwhile, Gemini3-Pro stands out as the only model demonstrating robust geometric reasoning, achieving the highest validation pass rate and success score while also maintaining superior silhouette quality.

These intermediate metrics are also useful for tracking incremental progress. Models show substantial variation in **IoU** and **Hausdorff Distance**, which remain informative even when **Success** is zero. Higher IoU corresponds to better global alignment with the target silhouette, while lower Hausdorff Distance captures more accurate boundary matching, providing graded signals below the strict validity threshold. More broadly, these metrics reveal a smoother progression toward successful assembly: improvements typically appear first in **IoU** (better outline alignment), then in reduced **RGE** and **PE** (better adherence to rigid and physical constraints), and finally in non-zero **Success**. This makes the benchmark suitable for tracking incremental advances, even when achieving full validity remains challenging.

To further validate that these failures reflect model limitations rather than issues in the benchmark construction or evaluation protocol, we also examine the ground-truth TCE solutions. Each TangramPuzzle instance already contains a ground-truth TCE solution. When this ground-truth TCE is evaluated, the seven polygonal pieces are instantiated according to their annotated final states, their union is computed, pairwise overlaps and global connectivity are checked, and the resulting outer boundary is compared with the target outline using the same IoU and Hausdorff Distance metrics as in the main evaluation. The ground-truth TCE achieves 100% **VPR** and 100% **Success**, confirming that every benchmark instance is constructively solvable and that our automated verifier correctly accepts valid solutions. Therefore, MLLM failures are not caused by invalid target shapes, missing solutions, or an overly strict verifier.

I Detailed Analytical Experiments

I.1 In-context Learning Details

We examine the effect of In-context Learning (ICL) by prepending three fully solved TCE examples to the prompt for a subset of models, including Qwen3-VL, GPT-5.2, and Gemini3-Pro. The experimental motivation is to disentangle format unfamiliarity from deeper spatial reasoning limitations. Specifically, the few-shot demonstrations explicitly show the required TCE-style output format, including the JSON structure, piece inventory, vertex and edge fields, and coordinate style. Therefore, if the main bottleneck were merely that models were unfamiliar with the TCE schema, providing solved examples should substantially reduce syntax errors and improve the rate of valid assemblies.

However, the results do not support a purely format-unfamiliarity explanation. Originally, Syntax Error (TSE) is already relatively low for most models versus the much higher Rigid Geometry Error (RGE) and Physical Error (PE), suggesting

that many models can follow the output format but still fail to satisfy geometric and physical constraints. With ICL, several models even exhibit increased TSE rates compared with the zero-shot setting. For instance, Qwen3-VL-8B-Instruct and Gemini3-Pro more frequently fail to generate valid JSON outputs, indicating that the longer symbolic context can impose an additional cognitive load and destabilize structured generation.

Despite this syntactic instability, ICL sometimes improves silhouette-level quality among successfully parsed responses. Models achieve better IoU or reduced Hausdorff distance in several cases, suggesting that demonstrations can help them better approximate the target shape. Nevertheless, these improvements rarely translate into higher final success, as models frequently violate rigid-body preservation or physical feasibility constraints. This implies that few-shot TCE demonstrations are insufficient to induce the compositional spatial reasoning needed to coordinate seven fixed rigid pieces under non-overlap, connectivity, and exact boundary-matching constraints. Thus, the ICL experiment weakens the hypothesis that Task 2 failures stem primarily from unfamiliarity with the TCE schema, instead highlighting constraint-satisfying spatial assembly as the central challenge.

I.2 Textual Geometry Ablation

To quantify the models' dependency on textual metadata, we evaluated a Visual-Centric setting where explicit textual descriptions of the target outline (vertices and edges) were removed, leaving only the image with annotated coordinates.

The elimination of explicit textual outline specifications significantly destabilized structured generation. Notably, GPT-5.2 experienced a dramatic surge in syntax errors, with TSE rising from 7.19% to 39.67%. Similarly, the Deepseek series frequently hallucinated irrelevant content, indicating a heavy reliance on text prompts to guide the generation process.

For the majority of models, the absence of textual data precipitated a marked deterioration in Silhouette Quality, exposing a critical dependency on semantic prompts. This confirms a widespread inability to precisely "read" geometric coordinates directly from raw visual inputs. However, Gemini3-Pro's robust performance in this setting serves as a notable exception, highlighting its advanced capability to perform rigorous reasoning solely based on visual grounding, independent of textual crutches.

I.3 Effect of Chain-of-Thought (CoT)

To investigate whether explicit step-by-step reasoning can alleviate the spatial logic bottleneck, we evaluate models using Chain-of-Thought (CoT) prompting. As shown in the "Chain-of-Thought (CoT)" results in Table 2, CoT provides only limited and highly uneven benefits. While it improves some intermediate metrics for certain models, it does not fundamentally resolve the core difficulty of the task: producing geometrically valid assemblies under rigid and physical constraints.

We observe that CoT can improve syntax compliance for certain models such as Qwen3-VL-32B and GPT-5.2, suggesting that explicit reasoning prompts may help organize outputs into a more structured form. However, this effect is inconsistent across the board. For example, Gemini3-pro becomes noticeably less stable in its output formatting under CoT, with its TSE rising significantly, indicating that longer reasoning traces can introduce additional formatting noise. This further supports our earlier conclusion that syntax is not the primary bottleneck in Task 2.

More importantly, CoT does not effectively enhance constraint satisfaction. Qwen3-VL-32B managed to achieve a non-zero Success rate (0.15%) under CoT, suggesting that explicit reasoning can occasionally help models stumble upon a valid configuration. Conversely, while silhouette matching slightly improved for GPT-5.2, we observed an exacerbation of geometric deformations, with both PE and RGE increasing. This reinforces our finding that models tend to prioritize "visual cheating" over rigorous physical rules.

For smaller-scale models such as Qwen3-VL-8B, performance remains largely stagnant despite CoT, with Success and VPR rates remaining at zero. This suggests that without basic spatial grounding, adding a thinking step is insufficient to solve complex inverse geometric problems.

I.4 Effect of Systematic Decomposition

Motivated by the human expert analysis in Appendix J, we explore whether a decomposition-based strategy can improve performance on Task 2. A key human strategy is to treat both the target silhouette and the seven tangram pieces as compositions of the same basic unit: the small isosceles right triangle. The solver can then reason about how groups of these basic units within the target outline correspond to individual tangram pieces,

making the puzzle easier to solve. We therefore introduce prompts that encourage models to perform this kind of decomposition before generating the final assembly.

As shown in the "Decomposition-Based Strategy" results in Table 2, this strategy leads to modest but consistent improvements in silhouette quality for several models. Qwen3-VL-8B, Qwen3-VL-32B, GPT-5.2, and Gemini3-Pro all achieve higher IoU scores under this setting, suggesting that explicit decomposition helps models align the overall assembly more closely with the target outline. In some cases, boundary precision also improves, suggesting that decomposition provides a more structured basis for global shape planning.

Proprietary models, particularly GPT-5.2 and Gemini3-Pro, show marked improvements in adhering to rigid-body and non-overlap constraints, as reflected by a decrease in RGE and PE scores. Such results underscore the role of a shared geometric basis in guiding models to reconcile how composite shapes populate the available interior space of the target outline. However, Gemini3-Pro's final Success rate experienced a slight decline compared to the original setting. For the Qwen3-VL series, the benefits of decomposition are more subtle; Qwen3-VL-32B showed a marginal improvement in VPR and silhouette quality, while the 8B model's performance remained largely stagnant, with Success and VPR staying at zero. Overall, while encouraging models to reason in terms of a shared geometric basis improves global shape planning and reduces constraint violations for some models, the lack of compositional spatial reasoning remains the primary cause of failure.

I.5 Effect of Removing Visual Input

To address the potential concern that the task might degenerate into a purely symbolic encoding–decoding problem once converted into TCE, we conducted a text-only ablation study. In this configuration, models receive only textual descriptions of the vertices of the target outline, with the visual image completely removed. This setup allows us to evaluate whether models can effectively solve the puzzle using symbolic coordinates alone, or whether visual grounding remains indispensable for compositional reasoning.

As illustrated by the "Text-Only Setting" results in Table 2, removing the visual modality leads to a substantial decline in overall performance. The strongest model, Gemini3-Pro, suffers a substantial drop in both VPR and Success, with both metrics falling by nearly half, while the other models remain unable to produce valid final assemblies. This result indicates that Task 2 is not merely a symbolic coordinate-translation problem: if the task could be solved by directly reading target vertices and converting them into TCE outputs, removing the image should not substantially hurt performance. Instead, the visual silhouette provides a critical global spatial anchor that helps models perceive topology, concavities, boundary structure, and part-to-whole organization, which are difficult to reconstruct from coordinate lists alone.

At the same time, Gemini3-Pro's sharp performance drop suggests that the model benefits substantially from visual heuristics. This observation is consistent with our broader finding that MLLMs can often produce visually plausible silhouettes while still violating rigid-body or physical constraints. Visual grounding helps the model approximate the global structure, but exact constraint satisfaction still requires reliable symbolic geometric planning, including rigid transformations, non-overlap, connectivity, and full boundary coverage.

The text-only setting does not produce a uniform pattern across intermediate metrics. For some models, certain constraint errors decrease, but these changes do not translate into better final outcomes. In other words, access to coordinates alone may help with parts of the formal representation, yet it remains insufficient for carrying out the full part-to-whole assembly process. The persistent gap in Success indicates that solving the task requires more than simply decoding symbolic input into TCE format.

I.6 Effect of TCE-Light Representation

To examine whether Task 2 failures are caused by the complexity of the full TCE representation rather than spatial reasoning itself, we introduce a simplified *TCE-Light* setting. Instead of requiring models to generate complete vertex and edge lists for every piece, TCE-Light asks them to output only one rigid transformation for each piece, including `piece_id`, rotation angle, optional reflection, and translation. The final vertices are then deterministically computed by applying the predicted transformations to the initial pieces. This design substantially reduces the burden of symbolic coordinate generation and enforces rigid-shape preservation by construction, making RGE zero for all successfully parsed outputs.

As shown in Table 2, however, simplifying the output format does not solve the task. Although TCE-Light removes rigid-geometry errors, models still exhibit high PE, low VPR, and near-zero Success. For example, Qwen3-VL-32B-Instruct and GPT-5.2 achieve 0.00% Success under TCE-Light, while Gemini3-Pro's success rate drops to 3.59%. Moreover, silhouette quality often drops substantially, with lower IoU and larger Hausdorff distance compared with the full TCE setting. These findings suggest that the primary barrier is not the ability to write detailed TCE coordinates. Even when the output is reduced to simple rigid transformations, MLLMs still struggle to determine physically feasible placements that jointly satisfy non-overlap, connectivity, and target-boundary constraints.

I.7 Relationship between Task 1 and Task 2

Task 1 and Task 2 are designed to probe complementary levels of compositional spatial reasoning. Serving as a foundational diagnostic, Task 1 uses a multiple-choice format for lightweight inference of part-to-whole global shapes, which enables clean and objective automatic evaluation. In contrast, Task 2, our primary constructive evaluation, requires models to independently generate a complete assembly under strict rigid-body, non-overlap, connectivity, and boundary-matching constraints.

To better understand the relationship between the two tasks, we compute cross-model correlations between Task 1 Accuracy and the main Task 2 metrics using the models evaluated on both tasks. We report both Pearson and Spearman correlations, where Pearson correlation measures linear association between the scores while Spearman correlation assesses ranking consistency across tasks. As shown in Table 5, Task 1 Accuracy correlates strongly with Task 2 IoU, suggesting that Task 1 captures a capability that is useful for Task 2: composing local pieces into a plausible global silhouette. Models that perform better on Task 1 tend to achieve better silhouette overlap in Task 2.

However, Task 1 does not reliably predict whether a model can produce a fully valid final assembly. Although Pearson correlations with VPR and Success are moderately positive, the corresponding Spearman correlations are much weaker, indicating that the relationship is not robust in rank order and is affected by the sparsity of successful Task 2 solutions. For example, models such as GPT-5.2 and Qwen3-VL-32B-Instruct achieve strong Task 1 accuracy but still obtain 0% Success in Task 2 under the standard setting. This indicates that Task 2 requires additional capabilities beyond global-shape recognition, including preserving rigid piece geometry, avoiding overlaps, maintaining connectivity, and exactly matching the target boundary. Overall, the two tasks form a hierarchical evaluation: Task 1 measures the perceptual prerequisite of part-to-whole silhouette composition, while Task 2 measures inverse generative assembly under strict geometric constraints.

Table 5: Cross-model correlation between Task 1 Accuracy and Task 2 metrics.

Correlation	RGE	PE	VPR	IoU	Hausdorff	Success
Pearson	-0.36	-0.51	0.69	0.82	-0.53	0.69
Spearman	-0.57	-0.32	0.30	0.79	-0.68	0.27

J Detailed Human Performance Analysis

J.1 Human Performance on Task 1

We evaluate human performance on Task 1 using three domain experts, each of whom completed a random subset of 50 questions. As shown in Table 3, all three participants achieved 100% accuracy. This indicates that Task 1 is straightforward for humans: once the seven tangram pieces are composed, the correct outer silhouette is typically easy to recognize. Participants also reported low complexity and short time usage, suggesting that the task can be solved efficiently and with little difficulty by human experts.

The human results also help contextualize the model performance in Table 1. Although Gemini3-Pro achieves near-ceiling accuracy and GPT-5.2 also performs strongly, most other models remain substantially below the human level, with many open-source models still far from reliable recognition. In particular, model accuracies span a wide range, from below 30% to nearly 99%, revealing a clear gap between proprietary and many open-source models. This spread indicates that Task 1 is far from saturated: despite being easy for humans, it remains discriminative for models and provides a meaningful test of visual perception and mental shape composition.

J.2 Human Performance on Task 2

We evaluate human performance on Task 2 by involving three domain experts who independently solved the task sets under the same conditions as the models.

Performance Patterns. As shown in our results, human solvers rarely violate fundamental geometric constraints such as piece shape preservation or piece count correctness. Consequently, human performance manifests as a binary outcome: a solution is either strictly valid or the participant fails to find a solution within a reasonable time. This stands in sharp contrast to model predictions, which often achieve partial visual similarity through impermissible physical violations (e.g., overlapping pieces or shape distortion).

Solving Strategies. We observe substantial variance in success rates across the strategies employed. One expert achieved perfect performance by adopting a *systematic decomposition strategy*. Specifically, all seven tangram pieces were interpreted as compositions of a common geometric primitive: the small isosceles right triangle. Under this view, each piece corresponds to a fixed number and arrangement of these primitive units, and the target silhouette can likewise be mentally decomposed into the same unit-triangle basis. Once the silhouette is represented in this way, the puzzle becomes a structured assignment problem: the solver searches for regions within the decomposed outline whose unit counts and geometric patterns match those of the seven tangram pieces, and then assembles the solution by mapping pieces to these regions. In contrast, the other two experts primarily relied on *iterative trial-and-error*, repeatedly adjusting piece positions until a feasible configuration emerged. This heuristic approach resulted in lower success rates and significantly longer solving times.

Complexity Assessment. Participants were asked to rate task difficulty on a scale from 1 to 5. The high average complexity score of 4.1 confirms that TangramPuzzle is non-trivial even for humans. On average, each instance required approximately 6.7 minutes to solve, reflecting the substantial cognitive effort required for precise geometric reasoning and assembly planning.

K Human Evaluation on MLLMs

Automatic metrics primarily capture strict geometric validity, often overlooking perceptual plausibility. To address this, we conduct a human evaluation on Task 2 using three criteria rated on a 5-point scale: Perceptual Silhouette Match (PSM), which assesses the visual similarity of the assembly to the target; Piece Plausibility & Canonicality (PPC), which evaluates whether piece placements appear natural rather than visually strained; and Near-Miss Severity (NMS), which measures the proximity of an incorrect prediction to a successful solution. Table 6 summarizes the human evaluation results, where the average scores are reported as mean $\pm$ sample standard deviation across the three experts. Gemini3-Pro consistently receives high scores, indicating strong perceptual alignment with the target silhouette; even when automatic success is not achieved, many solutions are judged to be close to correct. In contrast, GPT-5.2 receives substantially lower scores, with frequent perceptual mismatches, implausible constructions, and failures that are judged to be far from correct. The relatively low inter-rater variance, ranging from 0.15 to 0.36 across all reported criteria, further supports the reliability and consistency of the human evaluation process.

Table 6: Human evaluation of MLLM on Task 2.

Model	Expert	PSM (↑)	PPC (↑)	NMS (↑)
Gemini3-Pro	Expert 1	04.20	04.30	04.60
	Expert 2	04.50	04.60	04.10
	Expert 3	04.20	04.50	04.00
	Average	$\mathbf{04.30 \pm 0.17}$	$\mathbf{04.47 \pm 0.15}$	$\mathbf{04.23 \pm 0.32}$
GPT-5.2	Expert 1	01.60	01.50	01.80
	Expert 2	01.20	01.10	01.10
	Expert 3	01.50	01.20	01.60
	Average	$\underline{01.43 \pm 0.21}$	$\underline{01.27 \pm 0.21}$	$\underline{01.50 \pm 0.36}$

L Detailed Case Study

L.1 Visual Examples

Figure 4 visualizes the generation results for a U-shaped target (Task 2). Gemini3-Pro achieves a perfect assembly, demonstrating its ability to effectively ground coordinate constraints to solve complex compositional problems. A critical failure mode observed across other models is the tendency to prioritize visual silhouette matching at the expense of geometric constraints. For instance, Claude-Sonnet-4.5 attempts to force an alignment with the target contour by non-rigidly elongating the edges of the left and right triangles, whereas InternVL3-78B generates severe overlaps to maximize area coverage. Meanwhile, GPT-5.2 produces a visually plausible silhouette but fails to preserve component fidelity, utilizing an incorrect piece inventory (e.g., hallucinating an extra square while omitting the parallelogram). We further observe that smaller models such as Qwen3-VL-8B-Instruct often arrange pieces in a simple linear sequence, indicating an inability to map symbolic coordinate

Table 7: Fine-grained breakdown of Physical Errors.

Model	OE ↓	CE ↓	PE ↓
Qwen3-VL-8B (Bai et al., 2025)	36.23	79.19	80.69
Qwen3-VL-32B (Bai et al., 2025)	79.64	88.17	94.46
InternVL3-78B (Zhu et al., 2025)	86.38	85.63	92.96
GLM-4.6V (Z.ai, 2025)	81.59	88.62	93.56
GPT-5.2 (OpenAI, 2025)	81.74	82.93	92.81
Gemini3-Pro (Google DeepMind, 2025)	**29.19**	**62.87**	**65.57**
Claude-Sonnet-4.5 (Anthropic, 2025)	70.21	90.72	97.90

specifications onto a 2D planar workspace.

L.2 Fine-grained Insights into Physical Errors

To further understand why models fail to produce physically valid tangram assemblies, we decompose the original Physical Error (PE) metric into two independent sub-metrics: **Overlap Error (OE)** and **Connectivity Error (CE)**. OE measures the rate of predictions containing impermissible area overlap between any pair of pieces, while CE measures the rate of predictions whose piece union fails to form a single connected component. The original PE metric is retained as an aggregate physical-validity indicator:

$$\mathrm{PE} = \mathrm{OE} \vee \mathrm{CE}. \tag{10}$$

Table 7 reports the fine-grained breakdown. The results reveal that connectivity is a particularly challenging constraint for current MLLMs. Across most models, CE is higher than OE, indicating that models are not only prone to local collision errors, but are even more likely to scatter pieces across the canvas or form multiple disconnected groups instead of assembling a single cohesive tangram object. For example, Gemini3-Pro achieves the lowest OE among all evaluated models, with only 29.19% overlap errors, but its CE remains high at 62.87%. This suggests that even when a model can partially avoid local overlaps, it still struggles to enforce global connectedness.

The decomposition also shows that physical invalidity often arises from multiple interacting failure modes. For several models, both OE and CE are high, meaning that their predictions simultaneously contain local overlaps and global disconnections. This supports our broader finding that current MLLMs struggle not only with local geometric placement, but also with global compositional organization under strict physical constraints.

M Discussion

M.1 Generalizability to Real-World Spatial Scenarios

Although TangramPuzzle is instantiated in a highly controlled puzzle domain, the benchmark targets a set of core capabilities that extend beyond tangrams themselves and are broadly relevant to real-world spatial reasoning. Fundamentally, solving a tangram requires preserving the identity of rigid parts under Euclidean transformations, composing multiple local pieces into a globally valid configuration, and satisfying hard geometric constraints such as non-overlap, connectivity, and exact boundary alignment. These requirements mirror the structural demands of practical spatial tasks, where success depends not only on recognizing a plausible overall shape, but also on maintaining the validity of each constituent part throughout construction.

This is particularly relevant to applications such as CAD-style assembly and robotic manipulation. In CAD assembly, parts have fixed geometry and must be placed through rigid transformations while satisfying contact, fit, and compatibility constraints; a visually plausible arrangement is insufficient if dimensions, orientations, or interfaces are inconsistent. The Tangram Construction Expression (TCE) explicitly reflects the parametric representations used in CAD software. Similarly, robotic manipulation requires translating visual goals into precise, collision-free spatial plans for rotating, translating, and composing objects into a target structure. Our deterministic evaluation engine mirrors these requirements through programmatic checks for geometric validity. TangramPuzzle abstracts them into a minimal 2D setting, enabling evaluation of whether a model can preserve rigid-body structure and compose parts coherently, rather than merely relying on coarse silhouette matching. These capabilities are essential for integrating AI into reliable design and assembly workflows.

The controlled nature of the benchmark is therefore a deliberate design choice. Real-world spatial tasks typically involve additional sources of difficulty, including perception noise, occlusion, uncertain correspondences, and underspecified goals. While these factors are important, they can also obscure the underlying reasoning problem and make evaluation ambiguous. By abstracting away such confounds, TangramPuzzle isolates the compositional geometric core of the problem and enables exact, machine-verifiable assessment. This

is aligned with the motivation of our symbolic TCE representation and constraint-based verifier, which were introduced precisely to move evaluation beyond approximate perceptual similarity toward rigorous geometric correctness. In this sense, the benchmark should be viewed as a diagnostic testbed for compositional spatial reasoning abilities that real-world systems also rely upon.

M.2 Unique Challenges of TangramPuzzle

TangramPuzzle poses a form of spatial reasoning that is structurally different from many existing benchmarks. Rather than asking the model to recognize a local spatial relation, answer a geometry-related question, or reason over an already observed scene, it requires the model to construct a globally valid configuration from multiple rigid parts under strict constraints.

Diagnostic interpretability. One advantage of TangramPuzzle is that it does not collapse evaluation into a single accuracy or success score. Because the benchmark separates failures into syntax errors, rigid-geometry violations, physical-validity violations, and silhouette mismatch, it reveals not only whether a model fails, but also *how* it fails. This makes the benchmark more informative for diagnosing weaknesses and tracking progress.

Rigid-body preservation. A distinctive challenge of TangramPuzzle is that every piece is a rigid geometric object whose shape must remain unchanged. Our experiments reveal a systematic failure mode in which MLLMs appear to "cheat" by implicitly deforming pieces to better match the target silhouette. This type of error is easy to miss when evaluation relies mainly on answer correctness or approximate outline similarity, but TangramPuzzle makes it explicit by checking whether each piece preserves its exact geometry.

Inverse assembly. TangramPuzzle is fundamentally an inverse assembly problem. The model must search over a large transformation space—including translations, rotations, and reflections—to find a configuration that satisfies both the outer boundary and the internal non-overlap constraints. This is more demanding than simply interpreting a given spatial scene, because the model must reason from the desired global shape back to the placement of individual parts.

Mathematically grounded validity. Rather than evaluating spatial reasoning through natural-language answers or coarse perceptual judgments, TangramPuzzle grounds the task in explicit geometry via TCE, where each solution is specified by exact coordinates and structural relations, making the output exactly verifiable. As a result, the benchmark can distinguish outputs that merely *look* correct from those that are truly geometrically valid: a predicted assembly may closely match the target silhouette while still violating rigidity or non-overlap constraints, and thus be invalid.

M.3 Future Work

While the current benchmark utilizes the classic seven-piece set, the TCE framework inherently represents pieces by types, vertices, edges, and transformations. This representation can be directly extended to include more than seven polygonal components, variable piece inventories, or irregular dissection puzzles. The existing TCE schema can seamlessly record these new shapes, maintaining a consistent evaluation protocol while significantly increasing combinatorial difficulty.

A direction for future work is to bridge the current controlled setting with more realistic spatial reasoning scenarios. The coordinate-based TCE can be naturally upgraded to a 3D variant, where 2D polygons become polyhedra or voxel-based components, and transformations expand to 6-Degrees-of-Freedom (3D translations and rotations). Consequently, the verifier would shift from 2D polygon intersection to 3D collision detection, volumetric coverage, and surface alignment checks. This progression would allow TangramPuzzle-style evaluation to extend to polycube or CAD-style assembly tasks, while preserving the core principle of exact, machine-verifiable geometry.

To improve model performance on TangramPuzzle, future work could explore inference-time reasoning systems that couple generation with deterministic verification. In particular, test-time computation methods such as verifier-guided search, Monte Carlo Tree Search, and iterative self-refinement could exploit the deterministic TCE verifier to explore and repair candidate assemblies. Code-Interpreter-based verification loops could allow models to iteratively validate and correct physical overlap errors after generating an initial TCE. Multi-agent formulations may also be effective: different agents could specialize in proposing placements, checking topological overlaps, verifying connectivity, evaluating global silhouette matching, and revising invalid configurations.