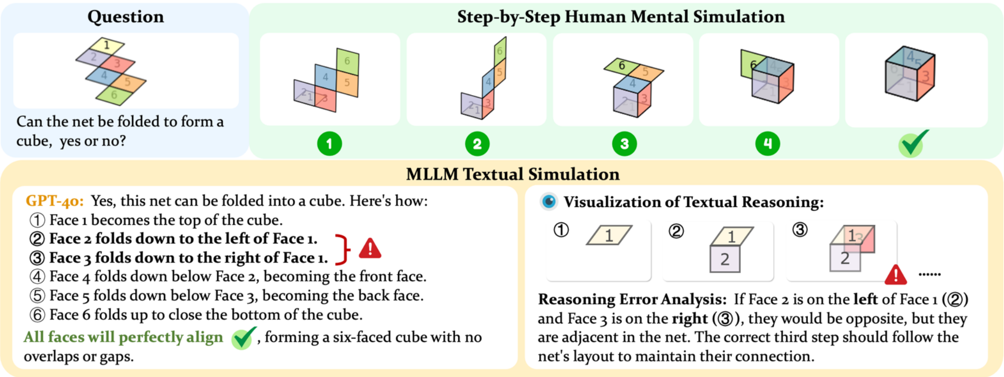
Unfolding Spatial Cognition: Evaluating Multimodal Models on Visual Simulations

Anonymous authors
Paper under double-blind review


Question
Step-by-Step Human Mental Simulation
1
2
3
4
5
6
❶
❷
❸
❹
✓
MLLM Textual Simulation
Visualization of Textual Reasoning:
①
②
③
1
2
......

Can the net be folded to form a cube, yes or no?

GPT-4o: Yes, this net can be folded into a cube. Here's how:
① Face 1 becomes the top of the cube.
② Face 2 folds down to the left of Face 1.
③ Face 3 folds down to the right of Face 1.
④ Face 4 folds down below Face 2, becoming the front face.
⑤ Face 5 folds down below Face 3, becoming the back face.
⑥ Face 6 folds up to close the bottom of the cube.
All faces will perfectly align ✓, forming a six-faced cube with no overlaps or gaps.

Reasoning Error Analysis: If Face 2 is on the **left** of Face 1 (②) and Face 3 is on the **right** (③), they would be opposite, but they are adjacent in the net. The correct third step should follow the net's layout to maintain their connection.

Figure 1: Visual simulations play a crucial role in real-world tasks, from assembling complex structures to interpreting mechanical diagrams and predicting spatial interactions. Different from how humans would approach a cube net folding problem, existing multimodal models rely heavily on textual simulation, which is not sufficient for reaching human-level spatial cognition. The above example shows how textual simulations of GPT-4o make obvious errors when we simulate the steps in 3D space.

Abstract

Spatial cognition is essential for human intelligence, enabling problem-solving through visual simulations rather than solely relying on verbal reasoning. However, existing AI benchmarks primarily assess verbal reasoning, neglecting the complexities of non-verbal, multi-step visual simulation. We introduce **STARE (Spatial Transformations and Reasoning Evaluation)**, a benchmark designed to rigorously evaluate multimodal large language models on tasks better solved through multi-step visual simulation. STARE features ∼4K tasks spanning foundational geometric transformations (2D and 3D), integrated spatial reasoning (cube net folding and tangram puzzles), and real-world spatial reasoning (perspective and temporal reasoning), reflecting practical cognitive challenges like object assembly, mechanical diagram interpretation, and everyday spatial navigation. Our evaluations show that models excel at reasoning over simpler 2D transformations, but perform close to random chance on more complex tasks like 3D cube net folding and tangram puzzles that require multi-step visual simulations. Humans achieve near-perfect accuracy but take considerable time (up to 28.0s) on complex tasks, significantly speeding up (down by 7.5 seconds on average) with intermediate visual simulations. In contrast, models exhibit inconsistent performance gains from visual simulations, improving on most tasks but declining in specific cases like tangram puzzles (GPT-4o, o1) and cube net folding (Claude-3.5, Gemini-2.0 Flash), indicating that models may not know how to effectively leverage intermediate visual information.

1 Introduction

Spatial reasoning is not merely a subset of human cognitive abilities but rather the fundamental underpinnings of intellectual processes (Tversky & Suwa, 2009). Reasoning with space enables individuals to solve complex tasks through visually simulating transformations of objects in the mind, anticipating how their actions would physically manipulate other artifacts. Cognitive psychologists have found ample evidence that humans simulate 2D and 3D transformations to reason about spatial problems (Mitko & Fischer, 2020; Duan et al., 2022; Wai et al., 2009; Battaglia et al.). Shepard & Metzler (1971) found that the time taken by a subject to recognize two perspective drawings as the same 3D shape increases linearly with their angular difference in orientation, suggesting an analog *mental rotation* process. Hegarty (1992) found that humans employ *mental animation*, incrementally simulating the movement of parts to understand mechanical diagrams. Such abilities enable everyday tasks like assembling furniture, reading maps or instructional diagrams, navigating new environments, and are strongly correlated with success in STEM disciplines (Judd & Klingberg, 2021; Christensen & Schunn, 2009; Hegarty, 2004b).

Despite their prevalence in real-world applications—from arranging furniture in a house to molecular docking for drug discovery—*dynamic* visual simulations are still under-represented when evaluating multimodal large language models (MLLMs). Existing datasets largely target static recognition or problems that can be re-phrased as linguistic reasoning (Johnson, 2017; Zhang, 2019; Ji, 2022; Duan et al., 2021; Chollet, 2019; Ramakrishnan et al., 2024). In contrast, humans frequently solve spatial challenges—such as folding a 2D net into a 3D object, assembling a tangram, or taking another visual perspective—by running internal, step-wise *visual simulations* (Fig. 1), which have a long pedigree in the cognitive science studying human spatial reasoning (Huttenlocher & Presson, 1973; Gunalp et al., 2019; Shepard & Feng, 1972; Preuss et al., 2024; Ayaz et al., 2012).

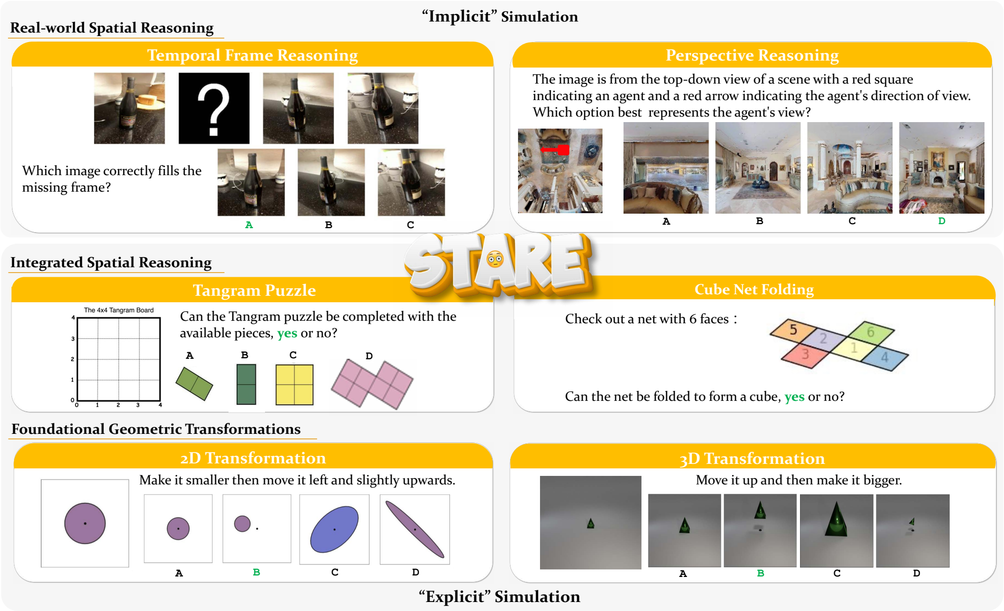
To bridge this gap, we introduce **STARE (Spatial Transformations and Reasoning Evaluation)**, a benchmark focused on spatial reasoning tasks that can be better solved through multi-step visual simulations. STARE evaluates whether MLLMs can perform complex visual reasoning akin to the visual simulations humans perform. It spans a spectrum of spatial cognition challenges (Fig. 2), structured in increasing complexity:

- **Foundational geometric transformations**: Tasks involving basic planar (2D) and volumetric (3D) transformations, such as rotations, translations, and reflections.
- **Integrated spatial reasoning**: Cube net folding, requiring understanding how 2D patterns fold into 3D objects, and tangram puzzles, assessing sequential assembly and spatial positioning.
- **Real-world spatial reasoning**: Tasks demanding reasoning about perspective changes and temporal frame sequences, simulating realistic spatial cognition scenarios encountered in daily life.

In the first two categories, each transformation or operation (e.g., folding a face) can be explicitly visualized step by step, and indeed humans often draw or imagine intermediate states when solving them. The last category demands higher-level visual simulation skills without always having clear intermediate visual cues (e.g., perspective reasoning) (Bass et al., 2022; Chen et al., 2023). We carefully curate $\sim$4K total instances across these categories, controlling difficulty via distractor similarity and number of simulation steps, to push models beyond superficial pattern-matching.

Our experiments show that models find reasoning over simple 2D transformations relatively easy but struggle with 3D cube net folding and tangram puzzles, performing near random chance due to the need for multi-step simulations. Humans, though nearly perfect in accuracy, took significantly longer—up to 28.0 seconds—to solve some tasks but sped up considerably (down by 7.5 seconds on average) when given intermediate steps. Meanwhile, when models receive intermediate visual steps, their performance varies: *e.g.,* GPT-4o, Gemini-2.0 Flash Thinking and o1 improve while Gemini-2.0 Flash and Claude worsen on cube net folding, suggesting that not all models effectively utilize visual guidance. In general, models (even o3) lag behind human performance significantly. To better understand these gaps, we conduct detailed error analyses, pinpointing specific reasons for model failures, such as difficulties in accurately interpreting 3D spatial relationships, inadequate of “imagining in space”, and struggles with extended visual contexts even when providing explicit visual simulations. Fundamentally, models cannot effectively perform visual simulation.

Overall, STARE aims to comprehensively test MLLMs’ ability to perform sequential visual simulations as opposed to pure textual reasoning. By evaluating models on tasks grounded in cognitive phenomena like mental imagery, we aim to reveal whether current MLLMs can approach the flexible spatial problem-solving of humans.



Which image correctly fills the missing frame?

The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view. Which option best represents the agent's view?

Can the Tangram puzzle be completed with the available pieces, yes or no?

Check out a net with 6 faces :

Can the net be folded to form a cube, yes or no?

Make it smaller then move it left and slightly upwards.

Move it up and then make it bigger.

Figure 2: Overview of STARE. STARE consists of 3 levels of tasks, 2D Transformation and 3D Transformation for foundational spatial reasoning skills, tangram puzzle and cube net folding for integrated spatial reasoning, temporal frame inference and perspective reasoning to mimic real-world scenarios. The intermediate steps for completing tasks in the first two levels can be explicitly simulated, while the more real-word spatial reasoning tasks requires more abstract and implicit mental simulations.

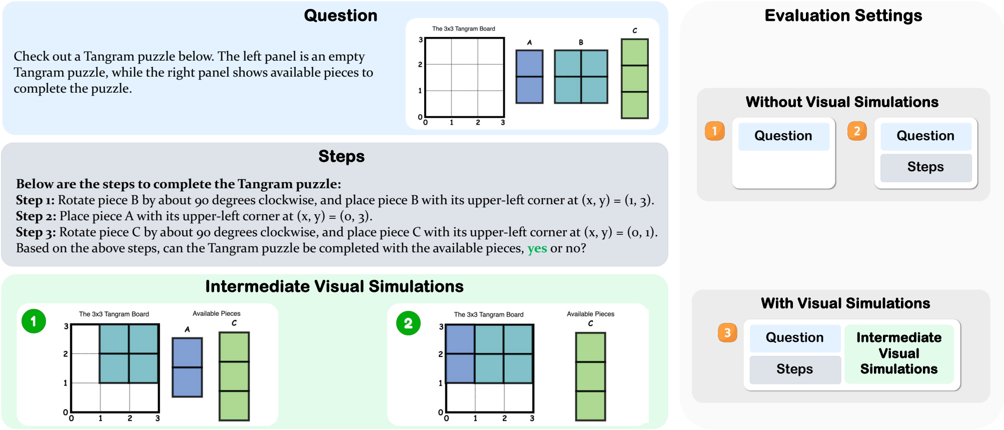
2 THE STARE BENCHMARK

STARE is designed to evaluate multimodal models' abilities in spatial cognition and visual reasoning, focusing specifically on tasks that humans solve non-linguistically, through visual simulation. Current perception-focused multimodal benchmarks still rely heavily on linguistic reasoning (Fu* et al., 2024; Lu et al., 2021; Li et al., 2024a) or static visual recognition (Tong et al., 2024; Wu & Xie, 2023; Fu et al., 2024), failing to measure models' abilities in sequential visual problem-solving. Parallel work in spatial cognition (Yiu et al., 2024; Zhang, 2019; Hu et al., 2021; Ramakrishnan et al., 2024; Rismanchian et al., 2024) probes analogy making and pattern induction, yet simulation is optional and intermediate visual states are seldom provided because of annotation cost. VSI-Bench (Yang et al., 2024) underscores the role of mental imagery in spatial reasoning, but focuses on spatial memory and estimation from video rather than explicit step-by-step simulation. STARE closes the gap by testing multimodal models across diverse spatial tasks that require step-by-step visual simulations with or without explicit linguistic guidance. We describe the overall design of STARE (§2.1), highlighting key differences compared to existing benchmarks. We then provide detailed descriptions of each task, discussing how the data was curated (§2.2).

2.1 OVERVIEW OF STARE

STARE is structured to comprehensively cover spatial reasoning at multiple complexity levels, from basic geometric transformations (2D and 3D) to more integrated tasks (cube net folding and tangram puzzles) and real-world spatial reasoning scenarios (temporal frame and perspective reasoning). Each task is presented as a multiple-choice or yes/no question using carefully designed visual and textual prompts. In total, the dataset contains ∼4K instances across different evaluation setups (Fig. 3). Detailed statistics of STARE are provided in Appendix Fig. 8.

STARE separates tasks that can be visually simulated, i.e., where each transformation step is visually observable, from tasks demanding more abstract and implicit mental simulations, such as perspective reasoning. To support more fine-grained evaluation, we synthesize the tasks that humans can mentally picture or even explicitly draw the intermediate steps, including 2D transformations, 3D transformations, cube net folding and tangram puzzle. Additionally, STARE tasks are intentionally crafted to closely reflect real-world scenarios such as assembling objects (e.g., tangram puzzles),



Check out a Tangram puzzle below. The left panel is an empty Tangram puzzle, while the right panel shows available pieces to complete the puzzle.

Below are the steps to complete the Tangram puzzle:
Step 1: Rotate piece B by about 90 degrees clockwise, and place piece B with its upper-left corner at (x, y) = (1, 3).
Step 2: Place piece A with its upper-left corner at (x, y) = (0, 3).
Step 3: Rotate piece C by about 90 degrees clockwise, and place piece C with its upper-left corner at (x, y) = (0, 1).
Based on the above steps, can the Tangram puzzle be completed with the available pieces, yes or no?

Figure 3: The different variants in the Tangram Puzzle task. We provide visualizations of the complete interleaved inputs for all three types in Appendix G.2.

interpreting mechanical diagrams (e.g., cube net folding) and navigating environments (e.g., perspective reasoning). These scenarios can potentially shed lights on models' abilities in practical, everyday spatial cognition, providing meaningful assessments aligned with common human challenges. A detailed discussion about related works in human visual reasoning and MLLM benchmarks are provided in Appendix D.

2.2 DATA CURATION

2D transformations: We design two types of tasks assessing spatial reasoning through two-dimensional shape transformations: visual analogy, and instruction-based tasks. In visual analogy tasks, a shape A is shown to transform visually into shape A', after which a shape B is provided with candidate shapes for applying the same transformation sequence to B. Instruction-based tasks explicitly describe transformations (e.g., "Rotate 90 degrees clockwise, then make it bigger") and require selecting the correctly transformed shape from 4 answer choices. Transformations include rotations, translations, uniform scaling, reflection and shearing, with clearly defined parameters. Each task is created with three difficulty levels: easy (with two distractors out of three clearly different in appearance), medium (one obvious distractor), and hard (all distractors visually similar, forcing the model to pay attention to the transformation itself). In addition, we synthesize samples with 1/2/3 transformation steps to facilitate evaluations in multi-turn visual transformations. We programmatically generate all shapes and their transformed version using Matplotlib (Matplotlib, 2012). Visualization of different variants of 2D transformation samples is shown in Appendix Fig. 9, 10.

We develop two experimental setups: (1) **question + transformation steps**, where the transformation steps are shown either verbally (for instruction-based tasks) or visually (for visual analogy tasks); and (2) **question + transformation steps + intermediate visual simulations**, showing all intermediate visualizations of shape B, excluding the final step. We synthesize a total of $\sim$1000 instances, $\sim$600 of which are without intermediate visual simulations.

3D transformations: We extend the 2D transformation tasks to three dimensions, creating similar tasks using 3D shapes. Reflection is omitted in 3D because the mirror plane isn't obviously recognizable to human evaluators. The transformations include rotations around arbitrary axes, translations in 3D space, scaling, and shearing. Tasks, difficulty levels, and experimental setups mirror those of the 2D tasks, with a total of $\sim$1000 instances. Following (Johnson et al., 2017), we create abstract 3D shapes as detailed meshes and use Blender (Blender) to render realistic and consistent visuals.

Tangram puzzles: Tangram puzzles test spatial reasoning about how individual pieces fit together to form a complete shape. Each puzzle provides a target grid and pieces, and the task is to determine whether the pieces can exactly fill the grid. Valid puzzles were generated by randomly dividing small grids (3x3 or 4x4) into rectangular or square shapes, then randomly rotated. Irregular variants were also created by merging adjacent rectangles. Invalid puzzles were constructed by adding or removing pieces, altering piece sizes, or giving incorrect placement instructions.

We create three setups for evaluation: (1) **question-only**, which presents the initial puzzle configuration with a query about solvability; (2) **question + assembly steps**, adding descriptive instructions of each assembly step without visual aids; and (3) **question + assembly steps + intermediate visual simulations**, providing both descriptive annotations and intermediate visualizations of the assembly process, excluding the final visualization indicating success or failure. This task comprises $\sim$800 puzzles, evenly divided into solvable and unsolvable instances.

Cube net folding: This task evaluates the model's capacity to mentally fold flat 2D patterns into 3D cubes. We provide examples comprising both valid nets (correctly folding into a cube) and invalid nets (leading to overlapping or disconnected faces). Each cube net has explicitly labeled faces. To generate these examples, we implement a step-by-step algorithm that simulates the folding process by designating a stationary base face and sequentially folding the connected faces. During each folding step, we detect and annotate errors, such as overlaps or disconnected faces, and generate corresponding visualizations using Matplotlib, clearly delineating face boundaries. Similar to tangram puzzles, we evaluate models in three setups, including (1) question-only, (2) question + folding steps, and (3) question + folding steps + intermediate visual simulations. The final cube net folding task contains $\sim$320 samples, balanced between valid and invalid configurations.

Temporal frame reasoning: This task evaluates a model's ability to infer missing sequential visual information. Each example consists of four consecutive frames from a video, with one frame hidden. The model must identify the missing frame from a set of three options, relying on temporal consistency and logical scene progression.

We construct 471 examples from the Objectron (Ahmadyan et al., 2021) dataset, which contains short, object-centric videos with camera pose annotations. To create meaningful sequences, we extract the longest continuous segment where the camera moves only in one direction (left or right), divide it into four equal intervals, and select a frame from the central portion of each interval. One of these frames is hidden, and the model must identify it from three choices: the correct missing frame and two distractor frames sampled from different, non-overlapping parts of the video.

Perspective reasoning: This task assesses a model's ability to understand how scenes appear from different viewpoints. Each example consists of a top-down map that indicates an agent's position and orientation, represented by an arrow showing the agent's viewing direction. The model must then select the correct first-person view from four choices, emphasizing spatial perspective reasoning and spatial relationships in various indoor environments.

We construct 250 samples using the HM3D dataset (Ramakrishnan et al., 2021), a large collection of 3D indoor spaces derived from real-world environments. To generate each example, we use the Habitat simulator (Savva et al., 2019; Szot et al., 2021; Puig et al., 2023) to place an agent at a random position on the floor while ensuring the surrounding scene contains enough visual cues, such as objects and structures, rather than just walls. A top-down view of the agent's position is then captured, and a random viewing direction is assigned (forward, right, left, or backward). The four answer choices correspond to these fixed 90-degree viewpoints, ensuring clear distinctions between them. To improve dataset quality, we manually remove ambiguous cases and low-resolution images.

3 EXPERIMENTS

In this section, we describe our experimental setup in detail, present comprehensive results, and provide an in-depth analysis of common model errors and limitations.

3.1 EXPERIMENTAL SETUP

For synthetic tasks involving explicit simulations (2D transformations, 3D transformations, cube net folding, tangram puzzles), we explore two evaluation settings:

- *Without Visual Simulations:* Models receive only an initial image with or without step-by-step textual instructions and had to mentally infer the subsequent transformations without visual guidance, thereby testing their internal mental simulation capabilities.

- *With Visual Simulations:* Models were provided with step-by-step visualizations clearly illustrating each transformation step before the final result, enabling explicit visual reasoning. Instead of collating the complex step-by-step visualizations into a single image, we provide the model with interleaved image and text query for evaluation.

Model	2D Trans.		3D Trans.		Cube Net		Tangram		Temp-oral	Pers-pective	Overall
	✗VSim	✓VSim	✗VSim	✓VSim	✗VSim	✓VSim	✗VSim	✓VSim			
Random	25.0	25.0	25.0	25.0	50.0	50.0	50.0	50.0	33.3	25.0	34.8
Closed-source Models											
GPT-4o	71.2	82.7 (↑ 11.5)	65.5	68.4 (↑ 2.9)	50.3	52.2 (↑ 1.9)	52.5	51.5 (↓ 1.0)	39.0	38.7	53.9
Claude-3.5 Sonnet	65.9	71.4 (↑ 5.5)	51.5	57.8 (↑ 6.3)	52.3	51.6 (↓ 0.7)	59.0	67.6 (↑ 8.6)	**54.0**	26.1	53.1
Gemini-2.0 Flash	69.5	75.2 (↑ 5.7)	56.1	59.3 (↑ 1.6)	37.7	35.6 (↓ 2.1)	65.0	65.5 (↑ 0.5)	38.6	37.2	51.3
Gemini-2.0 Flash Think	60.6	62.8 (↑ 2.2)	49.5	56.1 (↑ 6.6)	48.3	50.7 (↑ 2.4)	39.8	62.8 (↑ 23.0)	45.0	32.7	48.8
o1	81.8	87.7 (↑ 5.9)	67.9	71.6 (↑ 3.7)	51.3	53.4 (↑ 2.1)	55.3	53.2 (↓ 2.1)	45.0	36.8	57.2
o3	**87.5**	**89.3** (↑ 1.8)	**75.2**	**78.4** (↑ 3.2)	**68.4**	**79.4** (↑ 11.0)	**68.6**	**82.1** (↑ 13.5)	51.4	**42.8**	**68.1**
Open-source Models											
LLaVA-OneVision-72B	32.9	32.2 (↓ 0.7)	27.0	30.6 (↑ 3.6)	28.5	34.2 (↑ 3.7)	30.3	39.8 (↑ 9.5)	35.7	24.8	31.4
InternVL2.5-78B	47.5	50.1 (↑ 2.6)	38.1	36.5 (↓ 1.6)	37.1	37.3 (↑ 0.2)	60.7	48.2 (↓ 12.5)	31.4	26.0	39.2
Qwen2.5-VL-3B	16.6	20.0 (↑ 3.4)	29.1	31.4 (↑ 2.3)	43.5	41.0 (↓ 2.5)	50.1	42.7 (↓ 7.4)	33.3	23.3	32.3
Qwen2.5-VL-7B	35.4	32.4 (↓ 3.0)	28.8	31.7 (↑ 2.9)	40.7	44.9 (↑ 4.2)	54.5	52.9 (↓ 1.6)	36.5	23.2	36.7
Qwen2.5-VL-72B	45.2	48.5 (↑ 3.2)	43.0	49.1 (↑ 6.1)	35.2	53.4 (↑ 18.2)	61.2	56.9 (↓ 4.3)	31.4	26.0	42.3
Human Performance											
Accuracy	96.8	98.6 (↑ 1.8)	94.6	96.9 (↑ 2.3)	98.3	98.9 (↑ 0.6)	91.5	95.8 (↑ 4.3)	99.0	98.1	97.1
Response Time (s)	14.2	11.0 (↓ 3.2)	17.1	12.5 (↓ 4.6)	13.7	5.2 (↓ 8.5)	28.0	10.1 (↓ 17.9)	16.2	18.4	-
Δ(Best Model, Human)	-9.3	-9.3	-19.4	-18.5	-29.9	-19.5	-22.9	-13.7	-45.0	-55.3	-29.0

Table 1: Model Performance With or Without Visual Simulation (VSim) Across Tasks in STARE. Even the top performer, o1, achieves just under 60% accuracy. Humans, in contrast, get near perfect scores. Green (Red) arrows indicate performance improvements (degradations) with visual simulation.

For real-world reasoning tasks, including temporal frame and perspective reasoning, we evaluate models under the standard single image setting without providing explicit intermediate visual steps.

Evaluation Metrics. We report accuracy for multiple-choice questions in 2D/3D transformations, temporal frame, and perspective reasoning tasks. For cube net folding and tangram puzzles, which involve binary yes/no questions, we report the F1 score. We report macro-average performance across tasks as the overall evaluation metric.

Models. We consider the following models: **(1) Closed-source models**: GPT-4o (OpenAI), Claude-3.5 Sonnet (Anthropic), Gemini2.0 Flash (Deepmind, a), and the reasoning-focused Gemini2.0 Flash Thinking (Deepmind, b), o1 (OpenAI et al., 2024) and o3 (OpenAI, 2025). **(2) Open-source models**: InternVL2.5-78B (Chen et al., 2024), LLaVA-OneVision-72B (Li et al., 2024b), Qwen2.5-VL-3B, Qwen2.5-VL-7B, and Qwen2.5-VL-72B (Bai et al., 2025).

Additionally, we invite 5 undergraduate students to complete the same tasks as the models. The averaged performance and response time are recorded to benchmark model capabilities against human-level spatial reasoning. Additional human eval results are provided in Appendix H.

3.2 MAIN RESULTS

The results present in Tab. 1 show notable variations in model performance across different spatial reasoning tasks in the STARE benchmark. Models achieve the highest accuracy (up to 89.3%) on simpler 2D transformation tasks, significantly surpassing random chance (25%). Accuracy decreases by roughly 5% on average for more complex 3D transformations. Tasks involving intricate multi-step reasoning, such as cube net folding and tangram puzzles, resulted in even worse model performance without visual simulation. Additionally, temporal frame reasoning and perspective reasoning, which require interpreting sequential visual contexts and viewpoint changes, posed considerable difficulties, with most models performing similarly to random chance.

The use of visual simulations (VisSim) enhances model performance in most cases, but not all. GPT-4o exhibits a notable improvement of 11.5% accuracy on 2D transformations with visual simulations, and Claude-3.5 Sonnet shows significant gains (+8.6%) on tangram puzzles. However, visual simulations did not uniformly benefit model performance; certain models like Gemini-2.0 Flash experienced slight performance declines (e.g., a 2.1% decrease on F1 for cube net tasks), indicating that models can not always effectively leverage intermediate visual information. The latest reasoning-focused o3 model outperforms all other models with visual simulations. Overall, it improves over GPT-4o by 14.2% on average, but still lag behind human performance. Notably, o3 seems to be better at leveraging visual simulations. However, humans show relatively small performance gaps between conditions with and without visual simulation, indicating they are able to mentally simulate transformations effectively when explicit visual simulations are absent.

Open-source models generally exhibit lower accuracy compared to closed-source counterparts, highlighting a significant performance gap. Larger models like InternVL2.5-78B and Qwen2.5-VL-72B performe relatively better, suggesting benefits from scale, but their results with visual simulations also varied. For instance, InternVL2.5-78B's performance decreases significantly in tangram tasks (-12.5%), whereas Qwen2.5-VL-72B improves notably (18.2%) in cube net tasks.

Human performance consistently surpasses that of models, achieving high accuracy across all STARE tasks, and further improved by intermediate visual simulations. However, these tasks were cognitively demanding even for humans, reflected by relatively long response times (e.g., 28.0 seconds on tangram puzzles without visual simulations). Although intermediate visual simulations significantly reduces cognitive load and response time, humans still require more than 5 seconds to mentally manipulate and reason through these problems and complete the last step. Thus, STARE tasks clearly involve complex, multi-step spatial reasoning beyond simple recognition tasks solvable at a glance (Fu et al., 2024). These findings underscore humans' superior spatial reasoning capabilities, particularly when aided by visual simulations.

Moreover, to study whether gains on abstract, synthetic spatial tasks translate to real-world tasks, we computed model-level correlations between the two domains. Concretely, for each model, we average its performance across with or without visual simulation on the 4 synthetic tasks and contrast that with its mean accuracy on the two real-world tasks. This yields a strong overall Pearson correlation ($r \approx 0.88$, $p \approx 5e^{-4}$) across all 11 models. Counting in human performance, further increase the correlation to ($r \approx 0.97$, $p \approx 1e^{-7}$).

3.3 DETAILED ANALYSIS

To gain deeper insights into model limitations and identify specific reasoning challenges, we structure our detailed analysis around several targeted questions. We focus our discussion below on the GPT-4o model, given that it achieves the best performance among the non-reasoning models. Analysis on other models can be found in Appendix H.

Q1: How well do models understand individual transformation types in 2D and 3D? We evaluate model accuracy on individual transformation operations—rotation, translation, scaling, reflection, and shearing—for both 2D and 3D tasks, comparing performance with and without visual simulation (Fig. 4). For 2D tasks, scaling achieves the highest accuracy (approximately 90% without visual simulation), improving further with visual simulation. Shearing was the most challenging in 2D (around 54%), showing minimal improvement from visual aids. Reflection, rotation, and translation significantly benefits from visual simulation, improving roughly 10 percentage points each. In 3D tasks, translation had the highest accuracy (about 76% without visual simulation), although it slightly declines with visual simulation. However, shearing, scaling, and rotation notably improve with visual simulation by about 3–8 percentage points. Overall, visual simulation substantially enhances performance for complex transformations, especially in 2D, though the added complexity of 3D transformations continues to present significant challenges.

Q2: How does model accuracy change as task complexity increases? (1) Performance vs. Difficulty-level: The left sub-Fig. in Fig. 6 shows model accuracy decreased as tasks became harder. For 2D tasks, GPT-4o performed best on easy tasks ($\sim$86% with visual simulation), with accuracy declining notably for medium and hard tasks, especially without visual simulation (dropping to $\sim$66% for hard tasks). For 3D tasks, overall accuracy was lower, decreasing from easy tasks ($\sim$72% without visual simulation) to hard tasks ($\sim$60% without). Visual simulation generally improved accuracy but was less effective or even slightly detrimental for the hardest 3D tasks (60.5% without, 57.4% with). **(2) Performance vs. Number of Turns**: The right sub-Fig. in Fig. 6 shows that how model performance varies with the number of transformation steps (N = 1, 2, 3). Without visual simulation, accuracy for both 2D and 3D tasks initially increases from N = 1 to N = 2, and then decreases at $N = 3$. The observed peak at N = 2 likely occurs because two-step transformations combine simpler transformations (e.g., scaling) with more challenging ones (e.g., shearing), allowing models to leverage the simpler transformations to determine the correct answer. In contrast, one-step transformations are evenly distributed across all transformation types, while at N = 3, the increased complexity from multiple transformations compounds cognitive demands, reducing over-

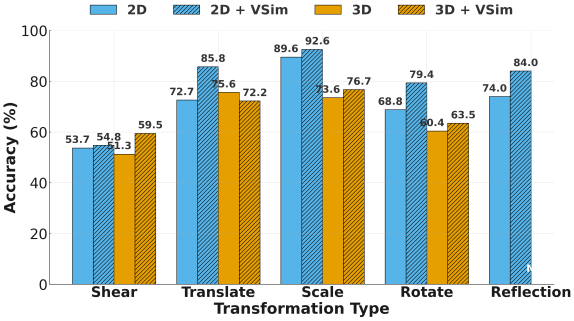
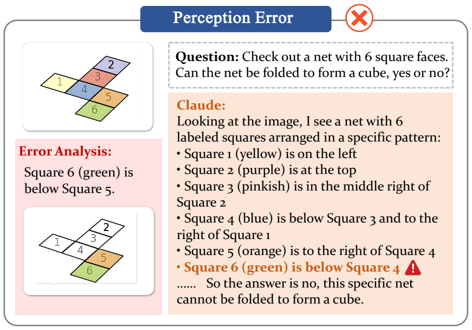


Figure 4: GPT-4o performance on individual 2D/3D transformation types, with and without Visual Simulation (VSim).



Square 6 (green) is below Square 5.

Question: Check out a net with 6 square faces. Can the net be folded to form a cube, yes or no?

Claude:
Looking at the image, I see a net with 6 labeled squares arranged in a specific pattern:
- Square 1 (yellow) is on the left
- Square 2 (purple) is at the top
- Square 3 (pinkish) is in the middle right of Square 2
- Square 4 (blue) is below Square 3 and to the right of Square 1
- Square 5 (orange) is to the right of Square 4
- **Square 6 (green) is below Square 4** ⚠

...... So the answer is no, this specific net cannot be folded to form a cube.

Figure 5: A perception error from Claude-3.5 Sonnet. See Appendix G.5 for more case study.

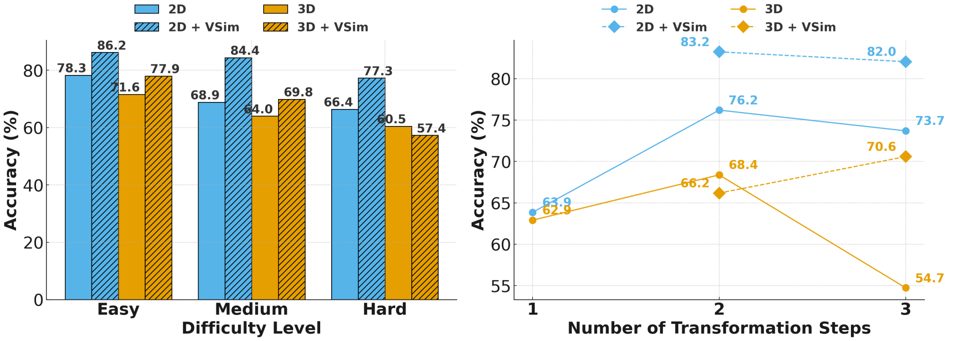


Figure 6: GPT-4o performance vs. task complexity (left: difficulty levels and right: number of transformation steps) with or without Visual Simulation (VSim).

all model accuracy. With visual simulation, accuracy remains consistently high across 2 and 3 steps in 2D tasks and shows stable or slightly improved performance at $N = 3$ in 3D tasks. Performance at N = 1 with visual simulation is not shown because there is no intermediate step.

Q3: Do model failures originate from basic visual perception errors? To determine if model failures originate from fundamental visual perception rather than higher-level reasoning limitations, we design a straightforward probing experiment. Specifically, we simplify the task by directly presenting the model with the final, fully simulated outcomes, reducing the problem to visually matching these outcomes to the correct candidate answers. Under these conditions, accuracy improves by 4.2% (from 82.7% to 86.9%) on 2D transformations and 2.8% (from 68.4% to 71.2%) on 3D transformations, indicating only a modest improvement when eliminating intermediate steps. However, for more structured tasks like cube net folding and tangram puzzles, providing the fully completed final form drastically raises accuracy to 100% and 91.6%, respectively, highlighting that models can solve these tasks when the perceptual complexity is minimized. To further isolate the nature of perceptual errors in cube net folding, we create targeted tasks to test both 2D perception (color recognition and face connectivity) and 3D perception (identifying if a face has been folded). Results from these tasks (Tab. 2) reveal perfect color recognition but a notable decrease in accuracy for face connectivity (94.1%) and particularly low accuracy in correctly identifying folded faces (57.4%). Fig. 5 illustrates an example of perception error on connectivity misalignments from Claude-3.5 Sonnet. Moreover, these specific perceptual errors in folding explain the limited benefits from visual simulations observed in Tab. 1 for GPT-4o. Overall, while some errors indeed stem from basic visual perception deficits, particularly in more complex 3D scenarios, the results suggest higher-level reasoning also plays a large role in overall model failures. Refer to Appendix for more quantitative (Appendix E) and qualitative (Appendix G.5) error analysis.

Q4: How well do models reason spatially in text? To evaluate how well models reason spatially from text alone, we translate each visual task into clear, concise descriptions. For *2D and 3D transformation tasks*, each object is described by stating its shape, color, position, size and etc.—for instance, *"a red square at position (3,4) with size 2"*. In the *cube-net folding task*, the unfolded cube is represented by numbering each face and arranging these numbers in a grid matching the cube net's visual layout. For example, "`123456`" represents all six faces in a single row. Lastly, for the *tangram puzzle task*, each piece is labeled (e.g., "Piece A") and represented by a compact grid

Model	2D Perception		3D Perception
	Color	Connectivity	Folded?
GPT-4o	100.0	94.1	57.4

Table 2: 2D and 3D perception accuracy in cube-net folding.

Input	2D Trans.	3D Trans.	Cube Nets	Tangram
Text-only	87.5	64.7	57.0	**72.6**
Image-only	75.1	67.7	56.0	62.5
Image+Text	**90.8**	**70.0**	**62.1**	–

Table 3: GPT-4o performance without visual simulation under different input representations.

Input	Cube Nets	Tangram
Question-only	50.2	**62.4**
Question+Steps	**50.4**	34.7

Table 4: GPT-4o performance with question-only vs. explicit reasoning steps.

Simulation State	2D Trans.	3D Trans.	Cube Nets	Tangram
Partial	86.8	**72.1**	51.3	43.5
All	82.7	68.4	**52.2**	**51.5**
Last	**89.4**	68.4	35.2	43.4

Table 5: GPT-4o performance with different intermediate visual-simulation states.

indicating occupied cells marked by "`1`". For instance, a square piece might be shown as two rows of "`11`". Examples of text representations of each task are provided in Appendix G.4.

As shown in Tab. 3, providing the model with a text representation removes much of the perception challenge, yet accuracy remains well below human performance—about 57% on cube-net folding, 65% on 3D transformations, and roughly 73% on tangram puzzles, suggesting that the model still lacks the ability to mentally simulate the steps to solve each task. Text helps most on 2D spatial reasoning: accuracy on 2D transformations rises from 75% with images alone to 87% with text, and tangram performance climb from 63% to 73%. For tasks involving 3D spatial reasoning, however, text gives little benefit, partly because the simple text description about shape, color, material, center, and size, cannot capture all the depth and adjacency cues in 3D spatial reasoning.

Q5: How well do models verbally simulate without visual simulation? We evaluate how effectively models verbally simulate spatial reasoning without intermediate visual simulations by comparing performance when provided only the question (Question-only) versus explicit verbal reasoning steps (Question+Steps). Tab. 4 shows minimal improvement in cube net folding (50.2% to 50.4%), indicating limited benefit from verbal reasoning alone. Conversely, tangram performance notably decreases (62.4% to 34.7%), suggesting models adopt shortcuts like summing piece areas rather than genuine spatial simulation. This result partially reflects a bias in our question-only set: models can achieve $\sim$75% accuracy by checking the total areas of available pieces.

Q6: How well do models integrate textual context with isolated visual simulations? We compared accuracy when presenting models with complete visual sequences versus only the final or most relevant visual state (Tab. 5). Easier tasks like 2D and 3D transformations showed improved or comparable accuracy when presented only the final state (e.g., 82.7% for complete vs. 89.4% for last), suggesting that for these tasks, the final visual state closely resembles the initial state, reducing cognitive load. However, in complex tasks such as cube net folding (52.2% complete vs. 35.2% last) and tangram puzzles (51.5% complete vs. 43.4% last), the final state becomes more disconnected from the initial configuration, requiring deeper understanding of preceding verbal steps. This disconnection introduces significant challenges for models, aligning with earlier findings (Q4) and underscoring their difficulties in integrating complex visual sequences during multi-step reasoning. Refer to Appendix E for additional experimental results on the impact of # visual simulations.

4 CONCLUSION

In this paper, we introduced STARE, a novel benchmark specifically designed to evaluate multimodal models on diverse spatial cognition tasks involving complex visual reasoning and mental simulations. STARE uniquely assesses model capabilities across foundational geometric transformations, integrated spatial reasoning tasks, and real-world scenarios requiring temporal and perspective reasoning. Our extensive experiments reveal significant performance variations among multimodal models, highlighting substantial challenges, especially in complex, multi-step reasoning scenarios. Visual simulations notably enhance performance on simpler tasks but yield mixed results for more sophisticated tasks. The substantial gap in performance between closed-source and open-source models further emphasizes the necessity for advancements in multimodal reasoning. Overall, STARE sets a critical benchmark to guide future research towards human-level spatial reasoning capabilities in AI.

5 ETHICS STATEMENT

STARE provides a standardized way to measure AI capabilities in spatial reasoning tasks, potentially guiding research toward AI systems that can better support robotics, autonomous driving, augmented reality, and education. However, improved spatial reasoning could also lead to negative societal impacts if misused, such as enhanced surveillance or military applications. Additionally, the synthetic nature of STARE may introduce biases toward simplified or artificial scenarios, limiting direct applicability to real-world conditions. Future versions should aim to include more realistic, diverse datasets and consider ethical guidelines to minimize risks and ensure fair, positive societal outcomes.

6 REPRODUCIBILITY STATEMENT

We have taken substantial steps to ensure the reproducibility of our results. All experimental settings are described in detail in Appendix G. We provide complete documentation of the statistics of STARE and the design spaces for all synthetic tasks, including 2D transformations, 3D transformations, cube net folding, and tangram puzzles in Appendix F. Data curation and evaluation code is included in the supplementary material to facilitate verification and reuse. Together, these resources enable the community to reproduce our experiments and extend our findings.

REFERENCES

Adel Ahmadyan, Liangkai Zhang, Artsiom Ablavatski, Jianing Wei, and Matthias Grundmann. Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`.

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. VQA: Visual question answering. In *ICCV*, 2015.

Hasan Ayaz, Patricia A Shewokis, Meltem İzzetoğlu, Murat P Çakır, and Banu Onaral. Tangram solved? prefrontal cortex activation analysis during geometric problem solving. In *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 4724–4727. IEEE, 2012.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report, 2025. URL `https://arxiv.org/abs/2502.13923`.

L.W. Barsalou. Grounded cognition. *Annual Review of Psychology*, 59:617–645, 2008.

Ilona Bass, Kevin A. Smith, Elizabeth Bonawitz, and Tomer D. Ullman. Partial mental simulation explains fallacies in physical reasoning. *Cognitive Neuropsychology*, 2022.

Peter W Battaglia, Jessica B Hamrick, and Joshua B Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332.

P.W. Battaglia, J.B. Hamrick, and J.B. Tenenbaum. Simulation as an engine of physical scene understanding. *Proceedings of the National Academy of Sciences*, 110(45):18327–18332, 2013.

Daniel M. Bear, Elias Wang, Damian Mrowca, Felix J. Binder, Hsiao-Yu F. Tung, R. T. Pramod, Cameron Holdaway, Sirui Tao, Kevin A. Smith, Fan-Yun Sun, Li Fei-Fei, Nancy Kanwisher, Joshua B. Tenenbaum, Daniel L. K. Yamins, and Judith E. Fan. Physion: Evaluating physical prediction from vision in humans and machines. *arXiv preprint arXiv:2106.08261*, 2022.

Mahtab Bigverdi, Zelun Luo, Cheng-Yu Hsieh, Ethan Shen, Dongping Chen, Linda G Shapiro, and Ranjay Krishna. Perception tokens enhance visual reasoning in multimodal language models. *arXiv preprint arXiv:2412.03548*, 2024.

Blender. Blender is free software. `https://www.blender.org/`.

P.A. Carpenter, M.A. Just, and P. Shell. What one intelligence test measures: A theoretical account of the processing in the raven progressive matrices test. *Psychological Review*, 97(3):404–431, 1990.

Tony Chen, Kelsey R. Allen, Samuel J. Cheyette, Joshua B. Tenenbaum, and Kevin A. Smith. Ïust in timer̈epresentations for mental simulation in intuitive physics. In *Proceedings of the 45th Annual Meeting of the Cognitive Science Society (CogSci)*, 2023.

Zhe Chen, Weiyun Wang, Yue Cao, Yangzhou Liu, Zhangwei Gao, Erfei Cui, Jinguo Zhu, Shenglong Ye, Hao Tian, Zhaoyang Liu, et al. Expanding performance boundaries of open-source multimodal models with model, data, and test-time scaling. *arXiv preprint arXiv:2412.05271*, 2024.

François Chollet. On the measure of intelligence. *arXiv preprint arXiv:1911.01547*, 2019.

Bo T Christensen and Christian D Schunn. The role and impact of mental simulation in design. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition*, 23(3):327–344, 2009.

Google Deepmind. Introducing gemini 2.0: our new ai model for the agentic era. `https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024//`, a.

Google Deepmind. Gemini 2.0 flash thinking mode. `https://ai.google.dev/gemini-api/docs/thinking-mode`, b.

Jiafei Duan, Samson Yu, and Cheston Tan. Space: A simulator for physical interactions and causal learning in 3d environments. In *Proceedings of the ieee/cvf international conference on computer vision*, pp. 2058–2063, 2021.

Jiafei Duan, Samson Yu, Soujanya Poria, Bihan Wen, and Cheston Tan. Pip: Physical interaction prediction via mental simulation with span selection. In *European Conference on Computer Vision*, pp. 405–421. Springer, 2022.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, Yunsheng Wu, and Rongrong Ji. MME: A comprehensive evaluation benchmark for multimodal large language models. In *arXiv preprint arXiv:2306.13394*, 2023.

Deqing Fu*, Ghazal Khalighinejad*, Ollie Liu*, Bhuwan Dhingra, Dani Yogatama, Robin Jia, and Willie Neiswanger. IsoBench: Benchmarking multimodal foundation models on isomorphic representations, 2024.

Xingyu Fu, Yushi Hu, Bangzheng Li, Yu Feng, Haoyu Wang, Xudong Lin, Dan Roth, Noah A Smith, Wei-Chiu Ma, and Ranjay Krishna. Blink: Multimodal large language models can see but not perceive. *arXiv preprint arXiv:2404.12390*, 2024.

Dedre Gentner. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2): 155–170, 1983.

Peri Gunalp, Tara Moossaian, and Mary Hegarty. Spatial perspective taking: Effects of social, directional, and interactive cues. *Memory & cognition*, 47:1031–1043, 2019.

Wenyu Han, Siyuan Xiang, Chenhui Liu, Ruoyu Wang, and Chen Feng. Spare3d: A dataset for spatial reasoning on three-view line drawings. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14690–14699, 2020.

M. Hegarty. Mechanical reasoning by mental simulation. *Trends in Cognitive Sciences*, 8(6):280–285, 2004a.

Mary Hegarty. Mental animation: Inferring motion from static displays of mechanical systems. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 18(5):1084–1102, 1992.

Mary Hegarty. Mechanical reasoning by mental simulation. *Trends in Cognitive Sciences*, 8(6):280–285, 2004b. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2004.04.001. URL `https://www.sciencedirect.com/science/article/pii/S1364661304001007`.

Sheng Hu, Yuqing Ma, Xianglong Liu, Yanlu Wei, and Shihao Bai. Stratified rule-aware network for abstract visual reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 1567–1574, 2021.

Janellen Huttenlocher and Clark C Presson. Mental rotation and the perspective problem. *Cognitive Psychology*, 4(2):277–299, 1973. ISSN 0010-0285. doi: https://doi.org/10.1016/0010-0285(73)90015-7. URL `https://www.sciencedirect.com/science/article/pii/0010028573900157`.

N. Ichien, Q. Liu, S. Fu, K.J. Holyoak, A. Yuille, and H. Lu. Two computational approaches to visual analogy: Task-specific models versus domain-general mapping. *Cognitive Science*, 47(4): e13347, 2023.

Nicholas Ichien, Qing Liu, Shuhao Fu, Keith J. Holyoak, Alan Yuille, and Hongjing Lu. Visual analogy: Deep learning versus compositional models. In *Proceedings of the 43rd Annual Meeting of the Cognitive Science Society (CogSci)*, 2021.

et al. Ji, W. Abstract visual reasoning with tangram shapes. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2350–2360, 2022.

et al. Johnson, Justin. Clevr: A diagnostic dataset for compositional language and elementary visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2901–2910, 2017.

Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. CLEVR: A diagnostic dataset for compositional language and elementary visual reasoning. In *CVPR*, 2017.

Nicholas Judd and Torkel Klingberg. Training spatial cognition enhances mathematical learning in a randomized study of 17,000 children. *Nature Human Behaviour*, 5(11):1548–1554, 2021.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Clarence Lee, M Ganesh Kumar, and Cheston Tan. Determinet: A large-scale diagnostic dataset for complex visually-grounded referencing using determiners. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 20019–20028, 2023.

Baiqi Li, Zhiqiu Lin, Wenxuan Peng, Jean de Dieu Nyandwi, Daniel Jiang, Zixian Ma, Simran Khanuja, Ranjay Krishna, Graham Neubig, and Deva Ramanan. Naturalbench: Evaluating vision-language models on natural adversarial samples. *European Conference on Computer Vision*, 2024a.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer. *arXiv preprint arXiv:2408.03326*, 2024b.

Xiongkun Linghu, Jiangyong Huang, Xuesong Niu, Xiaojian Shawn Ma, Baoxiong Jia, and Siyuan Huang. Multi-modal situated reasoning in 3d scenes. *Advances in Neural Information Processing Systems*, 37:140903–140936, 2024.

Fangyu Liu, Guy Emerson, and Nigel Collier. Visual spatial reasoning. *Transactions of the Association for Computational Linguistics*, 11:635–651, 2023.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. MMBench: Is your multi-modal model an all-around player? In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2024.

A. Lovett and K. Forbus. Modeling visual problem solving as analogical reasoning. *Psychological Review*, 124(1):60–90, 2017.

Pan Lu, Ran Gong, Shibiao Jiang, Liang Qiu, Siyuan Huang, Xiaodan Liang, and Song-Chun Zhu. Inter-gps: Interpretable geometry problem solving with formal language and symbolic reasoning. In *The 59th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2021.

Shixian Luo, Zezhou Zhu, Yu Yuan, Yuncheng Yang, Lianlei Shan, and Yong Wu. Geogrambench: Benchmarking the geometric program reasoning in modern llms. *arXiv preprint arXiv:2505.17653*, 2025.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. Sqa3d: Situated question answering in 3d scenes. *arXiv preprint arXiv:2210.07474*, 2022.

Matplotlib. Matplotlib: Visualization with python. `https://matplotlib.org/`, 2012.

Alex Mitko and Jason Fischer. When it all falls down: The relationship between intuitive physics and spatial cognition. *Cognitive research: principles and implications*, 5:1–13, 2020.

OpenAI. Hello gpt-4o. `https://openai.com/index/hello-gpt-4o/`.

OpenAI. Openai o3 and o4-mini system card, 2025.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024. URL `https://arxiv.org/abs/2412.16720`.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Richard Ren, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Tung Nguyen, Daron Anderson, Imad Ali Shah, Mikhail Doroshenko, Alun Cennyth Stokes, Mobeen Mahmood, Jaeho Lee, Oleksandr Pokutnyi, Oleg Iskra, Jessica P. Wang, Robert Gerbicz, John-Clark Levin, Serguei Popov, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Mstyslav Kazakov, Geoff Galgon, Johannes Schmitt, Alvaro Sanchez, Yongki Lee, Will Yeadon, Scott Sauers, Marc Roth, Chidozie Agu, Søren Riis, Fabian Giska, Saiteja Utpala, Antrell Cheatom, Zachary Giboney, Gashaw M. Goshu, Sarah-Jane Crowson, Mohinder Maheshbhai Naiya, Noah Burns, Lennart Finke, Zerui Cheng, Hyunwoo Park, Francesco Fournier-Facio, Jennifer Zampese, John Wydallis, John B. Wydallis, Ryan G. Hoerr, Mark Nandor, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Jungbae Nam, Edwin Taylor, Jun Jin, Gautier Abou Loume, Hangrui Cao, Alexis C Garretson, Damien Sileo, Qiuyu Ren, Doru Cojoc, Pavel Arkhipov, Usman Qazi, Aras Bacho, Lianghui Li, Sumeet Motwani, Christian Schroeder de Witt, Alexei Kopylov, Johannes Veith, Eric Singer, Paolo Rissone, Jaehyeok Jin, Jack Wei Lun Shi, Chris G. Willcocks, Ameya Prabhu, Longke Tang, Kevin Zhou, Emily de Oliveira Santos, Andrey Pupasov Maksimov, Edward Vendrow, Kengo Zenitani, Joshua Robinson, Aleksandar Mikov, Julien Guillod, Yuqi Li, Ben Pageler, Joshua Vendrow, Vladyslav Kuchkin, Pierre Marion, Denis Efremov, Jayson Lynch, Kaiqu Liang, Andrew Gritsevskiy, Dakotah Martinez, Nick Crispino, Dimitri Zvonkine, Natanael Wildner Fraga, Saeed Soori, Ori Press, Henry Tang, Julian Salazar, Sean R. Green, Lina Brüssel, Moon Twayana, Aymeric Dieuleveut, T. Ryan Rogers, Wenjin Zhang, Ross Finocchio, Bikun Li, Jinzhou Yang, Arun Rao, Gabriel Loiseau, Mikhail Kalinin, Marco Lukas, Ciprian Manolescu, Nate Stambaugh, Subrata Mishra, Ariel Ghislain Kemogne Kamdoum, Tad Hogg, Alvin Jin, Carlo Bosio, Gongbo Sun, Brian P Coppola, Haline Heidinger, Rafael Sayous, Stefan Ivanov, Joseph M Cavanagh, Jiawei Shen, Joseph Marvin Imperial, Philippe Schwaller, Shaipranesh Senthilkuma, Andres M Bran, Andres Algaba, Brecht Verbeken, Kelsey Van den Houte, Lynn Van Der Sypt, David Noever, Lisa Schut, Ilia Sucholutsky, Evgenii Zheltonozhskii, Qiaochu Yuan, Derek Lim, Richard Stanley, Shankar Sivarajan, Tong Yang, John Maar, Julian Wykowski, Martí Oller, Jennifer Sandlin, Anmol Sahu, Cesare Giulio Ardito, Yuzheng Hu, Felipe Meneguitti Dias, Tobias Kreiman, Kaivalya Rawal, Tobias Garcia Vilchis, Yuexuan Zu, Martin Lackner, James Koppel, Jeremy Nguyen, Daniil S. Antonenko, Steffi Chern, Bingchen Zhao, Pierrot Arsene, Sergey Ivanov, Rafał Poświata, Chenguang Wang, Daofeng Li, Donato Crisostomi, Ali Dehghan, Andrea Achilleos, John Arnold Ambay, Benjamin Myklebust, Archan Sen, David Perrella, Nurdin Kaparov, Mark H Inlow, Allen Zang, Kalyan Ramakrishnan, Daniil Orel, Vladislav Poritski, Shalev Ben-David, Zachary Berger, Parker Whitfill, Michael Foster, Daniel Munro, Linh Ho, Dan Bar Hava, Aleksey Kuchkin, Robert Lauff, David Holmes, Frank Sommerhage, Anji Zhang, Richard Moat, Keith Schneider, Daniel Pyda, Zakayo Kazibwe, Mukhwinder Singh, Don Clarke, Dae Hyun Kim, Sara Fish, Veit Elser, Victor Efren Guadarrama Vilchis, Immo Klose, Christoph Demian, Ujjwala Anantheswaran, Adam Zweiger, Guglielmo Albani, Jeffery Li, Nicolas Daans, Maksim Radionov, Václav Rozhoň, Vincent Ginis, Ziqiao Ma, Christian Stump, Jacob Platnick, Volodymyr Nevirkovets, Luke Basler, Marco Piccardo, Niv Cohen, Virendra Singh, Josef Tkadlec, Paul Rosu, Alan Goldfarb, Piotr Padlewski, Stanislaw Barzowski, Kyle Montgomery, Aline Menezes, Arkil Patel, Zixuan Wang, Jamie Tucker-Foltz, Jack Stade, Declan Grabb, Tom Goertzen, Fereshteh Kazemi, Jeremiah Milbauer, Abhishek Shukla, Hossam Elgnainy, Yan Carlos Leyva Labrador, Hao He, Ling Zhang, Alan Givré, Hew Wolff, Gözdenur Demir, Muhammad Fayez Aziz, Younesse Kaddar, Ivar Ängquist, Yanxu Chen, Elliott Thornley, Robin Zhang, Jiayi Pan, Antonio Terpin, Niklas Muennighoff, Hailey Schoelkopf, Eric Zheng, Avishy Carmi, Jainam Shah, Ethan D. L. Brown, Kelin Zhu, Max Bartolo, Richard Wheeler, Andrew Ho, Shaul Barkan, Jiaqi Wang, Martin Stehberger, Egor Kretov, Peter Bradshaw, JP Heimonen, Kaustubh Sridhar, Zaki Hossain, Ido Akov, Yury Makarychev, Joanna Tam, Hieu Hoang, David M. Cunningham, Vladimir Goryachev, Demosthenes Patramanis, Michael Krause, Andrew Redenti, David Aldous, Jesyin Lai, Shannon Coleman, Jiangnan Xu, Sangwon Lee, Ilias Magoulas, Sandy Zhao, Ning Tang, Michael K. Cohen, Micah Carroll, Orr Paradise, Jan Hendrik Kirchner, Stefan Steinerberger, Maksym Ovchynnikov, Jason O. Matos, Adithya Shenoy, Michael Wang, Yuzhou Nie, Paolo Giordano, Philipp Petersen, Anna Sztyber-Betley, Paolo Faraboschi, Robin Riblet, Jonathan Crozier, Shiv Halasyamani, Antonella Pinto, Shreyas Verma, Prashant Joshi, Eli Meril, Zheng-Xin Yong, Allison Tee, Jérémy Andréoletti, Orion Weller, Raghav Singhal, Gang Zhang, Alexander Ivanov, Seri Khoury, Nils Gustafsson, Hamid Mostaghimi, Kunvar Thaman, Qijia Chen, Tran Quoc Khánh, Jacob Loader, Stefano Cavalleri, Hannah Szlyk, Zachary Brown, Himanshu Narayan, Jonathan Roberts, William Alley, Kunyang Sun, Ryan Stendall, Max Lamparth, Anka Reuel, Ting Wang, Hanmeng Xu, Pablo Hernández-Cámara, Freddie Martin, Thomas Preu, Tomek Korbak, Marcus Abramovitch, Dominic Williamson, Ida Bosio, Ziye Chen, Biró Bálint, Eve J. Y. Lo, Maria Inês S. Nunes, Yibo Jiang, M Saiful Bari, Peyman Kassani, Zihao Wang, Behzad Ansarinejad, Yewen Sun, Stephane Durand, Guillaume Douville, Daniel Tordera, George Balabanian, Earth Anderson, Lynna Kvistad, Alejandro José Moyano, Hsiaoyun Milliron, Ahmad Sakor, Murat Eron, Isaac C. McAlister, Andrew Favre D. O., Shailesh Shah, Xiaoxiang Zhou, Firuz Kamalov, Ronald Clark, Sherwin Abdoli, Tim Santens, Harrison K Wang, Evan Chen, Alessandro Tomasiello, G. Bruno De Luca, Shi-Zhuo Looi, Vinh-Kha Le, Noam Kolt, Niels Mündler, Avi Semler, Emma Rodman, Jacob Drori, Carl J Fossum, Luk Gloor, Milind Jagota, Ronak Pradeep, Honglu Fan, Tej Shah, Jonathan Eicher, Michael Chen, Kushal Thaman, William Merrill, Moritz Firsching, Carter Harris, Stefan Ciobâcă, Jason Gross, Rohan Pandey, Ilya Gusev, Adam Jones, Shashank Agnihotri, Pavel Zhelnov, Siranut Usawasutsakorn, Mohammadreza Mofayezi, Alexander Piperski, Marc Carauleanu, David K. Zhang, Kostiantyn Dobarskyi, Dylan Ler, Roman Leventov, Ignat Soroko, Thorben Jansen, Scott Creighton, Pascal Lauer, Joshua Duersch, Vage Taamazyan, Dario Bezzi, Wiktor Morak, Wenjie Ma, William Held, Tran uc Huy, Ruicheng Xian, Armel Randy Zebaze, Mohanad Mohamed, Julian Noah Leser, Michelle X Yuan, Laila Yacar, Johannes Lengler, Katarzyna Olszewska, Hossein Shahrtash, Edson Oliveira, Joseph W. Jackson, Daniel Espinosa Gonzalez, Andy Zou, Muthu Chidambaram, Timothy Manik, Hector Haffenden, Dashiell Stander, Ali Dasouqi, Alexander Shen, Emilien Duc, Bita Golshani, David Stap, Mikalai Uzhou, Alina Borisovna Zhidkovskaya, Lukas Lewark, Miguel Orbegozo Rodriguez, Mátyás Vincze, Dustin Wehr, Colin Tang, Shaun Phillips, Fortuna Samuele, Jiang Muzhen, Fredrik Ekström, Angela Hammon, Oam Patel, Faraz Farhidi, George Medley, Forough Mohammadzadeh, Madellene Peñaflor, Haile Kassahun, Alena Friedrich, Claire Sparrow, Rayner Hernandez Perez, Taom Sakal, Omkar Dhamane, Ali Khajegili Mirabadi, Eric Hallman, Kenchi Okutsu, Mike Battaglia, Mohammad Maghsoudimehrabani, Alon Amit, Dave Hulbert, Roberto Pereira, Simon Weber, Handoko, Anton Peristyy, Stephen Malina, Samuel Albanie, Will Cai, Mustafa Mehkary, Rami Aly, Frank Reidegeld, Anna-Katharina Dick, Cary Friday, Jasdeep Sidhu, Hassan Shapourian, Wanyoung Kim, Mariana Costa, Hubeyb Gurdogan, Brian Weber, Harsh Kumar, Tong Jiang, Arunim Agarwal, Chiara Ceconello, Warren S. Vaz, Chao Zhuang, Haon Park, Andrew R. Tawfeek, Daattavya Aggarwal, Michael Kirchhof, Linjie Dai, Evan Kim, Johan Ferret, Yuzhou Wang, Minghao Yan, Krzysztof Burdzy, Lixin Zhang, Antonio Franca, Diana T. Pham, Kang Yong Loh, Joshua Robinson, Abram Jackson, Shreen Gul, Gunjan Chhablani, Zhehang Du, Adrian Cosma, Jesus Colino, Colin White, Jacob Votava, Vladimir Vinnikov, Ethan Delaney, Petr Spelda, Vit Stritecky, Syed M. Shahid, Jean-Christophe Mourrat, Lavr Vetoshkin, Koen Sponselee, Renas Bacho, Florencia de la Rosa, Xiuyu Li, Guillaume Malod, Leon Lang, Julien Laurendeau, Dmitry Kazakov, Fatimah Adesanya, Julien Portier, Lawrence Hollom, Victor Souza, Yuchen Anna Zhou, Julien Degorre, Yiğit Yalın, Gbenga Daniel Obikoya, Luca Arnaboldi, Rai, Filippo Bigi, M. C. Boscá, Oleg Shumar, Kaniuar Bacho, Pierre Clavier, Gabriel Recchia, Mara Popescu, Nikita Shulga, Ngefor Mildred Tanwie, Denis Peskoff, Thomas C. H. Lux, Ben Rank, Colin Ni, Matthew Brooks, Alesia Yakimchyk, Huanxu, Liu, Olle Häggström, Emil Verkama, Hans Gundlach, Leonor Brito-Santana, Brian Amaro, Vivek Vajipey, Rynaa Grover, Yiyang Fan, Gabriel Poesia Reis e Silva, Linwei Xin, Yosi Kratish, Jakub Łucki, Wen-Ding Li, Sivakanth Gopi, Andrea Caciolai, Justin Xu, Kevin Joseph Scaria, Freddie Vargus, Farzad Habibi, Long, Lian, Emanuele Rodolà, Jules Robins, Vincent Cheng, Tony Fruhauff, Brad Raynor, Hao Qi, Xi Jiang, Ben Segev, Jingxuan Fan, Sarah Martinson, Erik Y. Wang, Kaylie Hausknecht, Michael P. Brenner, Mao Mao, Xinyu Zhang, David Avagian, Eshawn Jessica Scipio, Alon Ragoler, Justin Tan, Blake Sims, Rebeka Plecnik, Aaron Kirtland, Omer Faruk Bodur, D. P. Shinde, Zahra Adoul, Mohamed Zekry, Ali Karakoc, Tania C. B. Santos, Samir Shamseldeen, Loukmane Karim, Anna Liakhovitskaia, Nate Resman, Nicholas Farina, Juan Carlos Gonzalez, Gabe Maayan, Sarah Hoback, Rodrigo De Oliveira Pena, Glen Sherman, Elizabeth Kelley, Hodjat Mariji, Rasoul Pouriamanesh, Wentao Wu, Sandra Mendoza, Ismail Alarab, Joshua Cole, Danyelle Ferreira, Bryan Johnson, Mohammad Safdari, Liangti Dai, Siriphan Arthornthurasuk, Alexey Pronin, Jing Fan, Angel Ramirez-Trinidad, Ashley Cartwright, Daphiny Pottmaier, Omid Taheri, David Outevsky, Stanley Stepanic, Samuel Perry, Luke Askew, Raúl Adrián Huerta Rodríguez, Ali M. R. Minissi, Sam Ali, Ricardo Lorena, Krishnamurthy Iyer, Arshad Anil Fasiludeen, Sk Md Salauddin, Murat Islam, Juan Gonzalez, Josh Ducey, Maja Somrak, Vasilios Mavroudis, Eric Vergo, Juehang Qin, Benjámin Borbás, Eric Chu, Jack Lindsey, Anil Radhakrishnan, Antoine Jallon, I. M. J. McInnis, Pawan Kumar, Laxman Prasad Goswami, Daniel Bugas, Nasser Heydari, Ferenc Jeanplong, Archimedes Apronti, Abdallah Galal, Ng Ze-An, Ankit Singh, Joan of Arc Xavier, Kanu Priya Agarwal, Mohammed Berkani, Benedito Alves de Oliveira Junior, Dmitry Malishev, Nicolas Remy, Taylor D. Hartman, Tim Tarver, Stephen Mensah, Javier Gimenez, Roselynn Grace Montecillo, Russell Campbell, Asankhaya Sharma, Khalida Meer, Xavier Alapont, Deepakkumar Patil, Rajat Maheshwari, Abdelkader Dendane, Priti Shukla, Sergei Bogdanov, Sören Möller, Muhammad Rehan Siddiqi, Prajvi Saxena, Himanshu Gupta, Innocent Enyekwe, Ragavendran P V, Zienab EL-Wasif, Aleksandr Maksapetyan, Vivien Rossbach, Chris Harjadi, Mohsen Bahaloohoreh, Song Bian, John Lai, Justine Leon Uro, Greg Bateman, Mohamed Sayed, Ahmed Menshawy, Darling Duclosel, Yashaswini Jain, Ashley Aaron, Murat Tiryakioglu, Sheeshram Siddh, Keith Krenek, Alex Hoover, Joseph McGowan, Tejal Patwardhan, Summer Yue, Alexandr Wang, and Dan Hendrycks. Humanity's last exam, 2025. URL `https://arxiv.org/abs/2501.14249`.

Luis S. Piloto, Ari Weinstein, Peter Battaglia, and Matthew Botvinick. Intuitive physics learning in a deep-learning model inspired by developmental psychology. *Nature Human Behaviour*, 6(9): 1257–1267, 2022.

Kai Preuss, Christopher Hilton, Klaus Gramann, and Nele Russwinkel. Identifying cognitive processes and neural substrates of spatial transformation in a mental folding task with cognitive modeling. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 46, 2024.

Xavi Puig, Eric Undersander, Andrew Szot, Mikael Dallaire Cote, Ruslan Partsey, Jimmy Yang, Ruta Desai, Alexander William Clegg, Michal Hlavac, Tiffany Min, Theo Gervet, Vladimír Vondruš, Vincent-Pierre Berges, John Turner, Oleksandr Maksymets, Zsolt Kira, Mrinal Kalakrishnan, Jitendra Malik, Devendra Singh Chaplot, Unnat Jain, Dhruv Batra, Akshara Rai, and Roozbeh Mottaghi. Habitat 3.0: A co-habitat for humans, avatars and robots, 2023.

S. K. Ramakrishnan, E. Wijmans, P. Krahenbuhl, and V. Koltun. Does spatial cognition emerge in frontier models? *arXiv preprint arXiv:2410.06468*, 2024.

Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. URL `https://arxiv.org/abs/2109.08238`.

Sina Rismanchian, Yasaman Razeghi, Sameer Singh, and Shayan Doroudi. Turtlebench: A visual programming benchmark in turtle geometry. *arXiv preprint arXiv:2411.00264*, 2024.

William Rudman, Michal Golovanevsky, Amir Bar, Vedant Palit, Yann LeCun, Carsten Eickhoff, and Ritambhara Singh. Forgotten polygons: Multimodal large language models are shape-blind. *arXiv preprint arXiv:2502.15969*, 2025.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, Devi Parikh, and Dhruv Batra. Habitat: A Platform for Embodied AI Research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

Roger N. Shepard and Christine Feng. A chronometric study of mental paper folding. *Cognitive Psychology*, 3(2):228–243, 1972. ISSN 0010-0285. doi: https://doi.org/10.1016/0010-0285(72)90005-9. URL `https://www.sciencedirect.com/science/article/pii/0010028572900059`.

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

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

Andrew Szot, Alex Clegg, Eric Undersander, Erik Wijmans, Yili Zhao, John Turner, Noah Maestre, Mustafa Mukadam, Devendra Chaplot, Oleksandr Maksymets, Aaron Gokaslan, Vladimir Vondrus, Sameer Dharur, Franziska Meier, Wojciech Galuba, Angel Chang, Zsolt Kira, Vladlen Koltun, Jitendra Malik, Manolis Savva, and Dhruv Batra. Habitat 2.0: Training home assistants to rearrange their habitat. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

J.B. Tenenbaum, T.L. Griffiths, and C. Kemp. Theory-based bayesian models of inductive learning and reasoning. *Trends in Cognitive Sciences*, 10(7):309–318, 2006.

Shengbang Tong, Zhuang Liu, Yuexiang Zhai, Yi Ma, Yann LeCun, and Saining Xie. Eyes wide shut? exploring the visual shortcomings of multimodal llms. *arXiv preprint arXiv:2401.06209*, 2024.

Barbara Tversky and Masaki Suwa. Thinking with sketches. 2009.

T.D. Ullman, E.S. Spelke, P. Battaglia, and J.B. Tenenbaum. Mind games: Game engines as an architecture for intuitive physics. *Trends in Cognitive Sciences*, 21(9):649–665, 2017.

Jonathan Wai, David Lubinski, and Camilla P Benbow. Spatial ability for stem domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology*, 101(4):817, 2009.

Taylor W. Webb, Shuhao Fu, Trevor Bihl, Keith J. Holyoak, and Hongjing Lu. Zero-shot visual reasoning through probabilistic analogical mapping. *arXiv preprint arXiv:2209.15087*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

Penghao Wu and Saining Xie. V*: Guided visual search as a core mechanism in multimodal llms. *ArXiv*, abs/2312.14135, 2023. URL `https://api.semanticscholar.org/CorpusID:266436019`.

Weiye Xu, Jiahao Wang, Weiyun Wang, Zhe Chen, Wengang Zhou, Aijun Yang, Lewei Lu, Houqiang Li, Xiaohua Wang, Xizhou Zhu, et al. Visulogic: A benchmark for evaluating visual reasoning in multi-modal large language models. *arXiv preprint arXiv:2504.15279*, 2025.

Jihan Yang, Shusheng Yang, Anjali Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. Thinking in Space: How Multimodal Large Language Models See, Remember and Recall Spaces. *arXiv preprint arXiv:2412.14171*, 2024.

Lingxiao Yang, Hongzhi You, Zonglei Zhen, Dahui Wang, Xiaohong Wan, Xiaohua Xie, and Ru-Yuan Zhang. Neural prediction errors enable analogical visual reasoning in human standard intelligence tests. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, 2023.

Eunice Yiu, Maan Qraitem, Charlie Wong, Anisa Noor Majhi, Yutong Bai, Shiry Ginosar, Alison Gopnik, and Kate Saenko. Kiva: Kid-inspired visual analogies for testing large multimodal models. *arXiv preprint arXiv:2407.17773*, 2024.

Xiang Yue, Tianyu Zheng, Yuansheng Ni, Yubo Wang, Kai Zhang, Shengbang Tong, Yuxuan Sun, Botao Yu, Ge Zhang, Huan Sun, et al. Mmmu-pro: A more robust multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2409.02813*, 2024.

Xiyao Yue, Yifan Ni, Kai Zhang, Tao Zheng, Ruixuan Liu, Wen Chen, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. *arXiv preprint arXiv:2311.16502*, 2023.

C. et al. Zhang. Raven: A dataset for relational and analogical visual reasoning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5317–5327, 2019.

Wenxuan Zhang, Sharifah M. Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks*, 2023.

A OVERVIEW OF THE APPENDIX

This Appendix is organized as follows:

- Section B discusses the use of LLMs.
- Section C discusses the limitations of STARE.
- Section D presents an extended discussion about related works.
- Section F details the statistics of STARE and the design spaces for all synthetic tasks, including 2D transformations, 3D transformations, cube net folding, and tangram puzzles.
- Section E provides additional analysis complementary to Section 3, including detailed error analysis, impact of # visual simulations and reasoning efforts, and additional results on human evaluation.
- Section G describes the experimental setup, covering the prompts used, model configurations, hyperparameter settings, and presents full visualizations of different experimental settings (e.g., evaluation settings with or without visual simulations, perception probing questions).
- Section H provides results on additional models for analysis conducted in Section 3.

B THE USE OF LARGE LANGUAGE MODELS

We used large language models (LLMs) as auxiliary tools during manuscript preparation, but only for surface-level editing such as grammar correction, minor rephrasing, and stylistic refinements to improve readability. AI-assisted coding was employed in curating synthetic data, but under strict human supervision and review. In addition, LLMs served as judges in our detailed error analysis; however, we manually reviewed a subset of their outputs to verify accuracy. All research ideas, methodologies, experiments, and conclusions were conceived and executed exclusively by the authors.

C LIMITATIONS

Although STARE provides valuable insights, it still has several limitations. First, it uses simplified synthetic images that do not fully represent real-world complexity; future versions could include realistic or dynamic scenes with clutter and occlusion. Second, it focuses only on rigid shape transformations; adding tasks involving flexible shapes, articulated objects, or additional sensory cues (such as audio or depth) would cover a wider range of spatial reasoning skills. Lastly, multiple-choice scoring hides intermediate reasoning steps; extending evaluations with explanations, step-by-step checks, or open-ended responses would give more detailed insights, which we briefly explore in Appendix E.

Still, STARE 's current design has clear strengths. The simplified images isolate spatial reasoning from general object recognition tasks. Its structured variety of tasks helps pinpoint specific model difficulties. Automatic scoring ensures consistent and easy-to-scale evaluations. Modular task presentations (image-only, text-only, image+text prompts) let researchers analyze individual modality contributions. Additionally, synthetic data makes STARE easily reproducible, accessible, and extensible. Overall, STARE is a strong first step toward measuring multimodal spatial reasoning, with clear pathways toward more realistic and comprehensive future benchmarks.

D RELATED WORK

Human visual reasoning. Human visual reasoning relies on two complementary faculties: *relational analogy*—mapping abstract structures across scenes—and *mental simulation*—predicting future states through incremental transformations. Structure–Mapping Theory (Gentner, 1983) and analyses of Raven's Progressive Matrices (Carpenter et al., 1990) first showed that success in visual problem-solving hinges on aligning relations rather than surface features. Computational accounts echo this claim: explicit relational models reproduce human-like performance (Lovett & Forbus,

Dataset	VSim	2D/3D	Synth/Real	Multi-step	Train/Eval	Size	Focus
STARE	✔	2D & 3D	Both	✔	Eval	∼4K	Multi-step spatial simulations
VSI-Bench	✘	3D (video)	Real	✘	Eval	5K	Spatial memory & layout recall from egocentric videos
KiVA	✘	2D	Synthetic	✔	Eval	4.3K	Visual analogical reasoning inspired by child cognition
TurtleBench	✘	2D	Synthetic	✔	Eval	260	Reproduce geometric programs via turtle graphics
SPARE3D	✘	2D & 3D	Synthetic	✔	Both	220K+	Spatial reasoning with 3-view CAD drawings
VSR	✘	2D	Real	✘	Eval	10K	Spatial relation understanding in caption-image pairs
DetermiNet	✘	2D	Synthetic	✘	Both	250K	Referring expression understanding with quantifiers and determiners
Forgotten Polygons	✘	2D	Synthetic	✘	Eval	2K	Shape identification and counting under visual ambiguity
GeoGramBench	✘	2D	Synthetic	✔	Eval	500	Symbolic geometric reasoning from procedural programs
VisuLogic	✘	2D	Synthetic	✘	Eval	1K	General visual logic across diverse reasoning types

Table 6: Comparison of existing visual/spatial reasoning datasets versus STARE.

2017), whereas modern deep networks still struggle with visual analogy tasks (Ichien et al., 2021; Webb et al., 2022; Ichien et al., 2023).

Mental simulation complements analogy-making. Classic work on mental rotation (Shepard & Metzler, 1971) and mechanical reasoning (Hegarty, 2004a) demonstrates that people mentally "run" transformations, consistent with grounded-cognition theories (Barsalou, 2008). Intuitive-physics studies cast the mind as a noisy physics engine that combines object-centric structure with probabilistic dynamics (Battaglia et al., 2013; Tenenbaum et al., 2006; Ullman et al., 2017). Object-based predictive-coding models such as PLATO extend these ideas, achieving human-like physical prediction and developmental trajectories (Yang et al., 2023; Piloto et al., 2022). Simulations are also *selective*: people allocate attention "just in time," focusing on the most diagnostic elements instead of exhaustively modeling the entire scene (Bass et al., 2022; Bear et al., 2022; Chen et al., 2023).

Together, these findings suggest that effective problem-solving hinges on the ability to carry out step-by-step visual simulations; our benchmark therefore probes whether multimodal models can effectively leverage or even produce such simulations and exhibit *human-like visual reasoning* on sequential, compositional tasks.

Multimodal evaluation benchmarks. Recent advances in evaluating multimodal large language models have led to the development of benchmarks targeting diverse aspects of visual reasoning. Early benchmarks such as VQA (Antol et al., 2015) and CLEVR (Johnson, 2017) focus on compositional reasoning and general visual question answering. However, more challenging benchmarks, such as MMMU (Yue et al., 2023) and Humanity's Last Exam (HLE) (Phan et al., 2025), assess expert-level, domain-specific reasoning using complex multimodal inputs, where state-of-the-art models achieve only around 60% on MMMU-pro (Yue et al., 2024) and below 20% on HLE.

In response to the growing demand for robust evaluation, several new benchmarks (Fu* et al., 2024; Lu et al., 2021; Li et al., 2024a; Tong et al., 2024; Wu & Xie, 2023) have been introduced. For example, M3Exam repurposes multilingual professional-license questions (Zhang et al., 2023). MME (Fu et al., 2023) and MMBench (Liu et al., 2024) separate low-level perception from higher-level cognition. BLINK (Fu et al., 2024) departs from pure linguistic reasoning tasks to include tasks grounded in core computer vision capabilities, including relative depth estimation, semantic correspondence, visual similarity assessment, inpainting, etc. Improvements on BLINK require the use of perception tokens (Bigverdi et al., 2024), which generate latent intrinsic images to reason, demonstrating for the first time, that reasoning doesn't have to be linguistic. In this work, we build upon this finding, targeting primarily visual reasoning that can be better solved with visual cues.

The most relevant benchmarks to ours are KiVA (Yiu et al., 2024), RAVEN/I-RAVEN (Zhang, 2019; Hu et al., 2021), SPACE (Ramakrishnan et al., 2024), and TurtleBench (Rismanchian et al., 2024), which primarily evaluate static analogy or pattern induction, where intermediate visual simulations are optional and often infeasible to curate. VSI-Bench (Yang et al., 2024) emphasizes mental imagery in spatial reasoning but centers on spatial memory and distance estimation from video. Other recent efforts—such as Forgotten Polygons (Rudman et al., 2025), GeoGramBench (Luo et al., 2025), and VisuLogic (Xu et al., 2025)—target more isolated failures in visual reasoning, including shape recognition, symbolic geometry, or visual logic puzzles. In contrast, STARE introduces programmatically generated puzzles—2D/3D transformations, cube-net folding, and tangram assembly—that isolate a model's capacity to benefit from *explicit* visual simulations, and further extends to perspective-taking and temporal reasoning tasks that mirror real-world scenarios.

Table 6 compares STARE with other spatial reasoning datasets: RoboSpatial (Song et al., 2025) provides large-scale real 2D/3D images with annotated spatial relations; MSR3D (MSQA) (Linghu et al., 2024) and SQA3D (Ma et al., 2022) support situated QA in 3D scenes but focus on single-step queries; the Visual Spatial Reasoning benchmark (Liu et al., 2023) probes basic positional relations; SPARE3D (Han et al., 2020) presents synthetic 2D to 3D consistency puzzles; and DetermiNet (Lee et al., 2023) emphasizes logical spatial tasks without multi-step simulation. As shown, STARE is the only benchmark that offers a **diverse suite of multi-step visual simulation tasks** across both **2D and 3D domains**, uniquely combining **procedural puzzles**, **geometric transformations**, and **realistic inference** (perspective and temporal reasoning). Its synthetic design allows fine-grained control over step difficulty and granularity, enabling analyses of visual reasoning beyond what existing datasets support.

E ADDITIONAL EXPERIMENTAL RESULTS

Fine-grained Reasoning Evaluation Because each synthetic task in STARE includes ground-truth metadata for every intermediate simulation step, we can scrutinize a model's entire reasoning chain—something impossible on benchmarks that provide only final answers.

We have conducted a deeper error analysis of GPT-4o and Claude-3.5 predictions on all synthetic tasks.

- We first examined representative case studies (Appendix G.5) and identified four recurring categories of reasoning failure:

	Error Category	**Description**	**Representative Example**
A	Misperception	The model misreads shapes, color, adjacency, or face layout.	Misidentifies cube-net face positions (Fig. 21)
B	Flawed Spatial Simulation	The model forms an incorrect mental model of how shapes transform.	Claims rotated hexagon is unchanged after 30° rotation (Fig. 19-right)
C	Heuristic Over-Use	The model falls back on shallow heuristics (e.g., area counting).	Sums tangram piece areas and misjudges solvability (Fig. 22-left)
D	Logic Inconsistency	The reasoning process contradicts itself or the final answer.	Correctly identifies two valid answers but chooses the wrong one arbitrarily (Fig. 20-right)

Table 7: Representative error categories (A–D) observed in model predictions, with descriptions and examples.

- We implemented an automated LLM-based judgment pipeline. Given a model's chain-of-thought, its prediction, and full ground-truth metadata (e.g., initial shape, transformation sequence, correct answer, and options), the judge model categorizes each incorrect prediction into one of the four failure types above. Here, we use o3 as the judge model.

  The table below summarizes the distribution of error types across 100 randomly sampled incorrect predictions per task, per model:

Model	Task	A	B	C	D
GPT-4o	2D	75.8%	21.2%	0.0%	3.0%
	3D	31.3%	67.7%	0.0%	1.0%
	Cube Net	12.7%	87.3%	0.0%	0.0%
	Tangram	68.1%	12.1%	6.6%	13.2%
Claude-3.5 Sonnet	2D Transform	60.0%	35.8%	0.8%	3.3%
	3D	34.0%	64.1%	0.0%	2.0%
	Cube Net	22.2%	74.7%	3.0%	0.0%
	Tangram	65.9%	17.1%	9.8%	7.3%

Table 8: Distribution of error categories (A–D) across tasks for GPT-4o and Claude-3.5 Sonnet.

Different tasks show distinct failure modes: 2D and tangram errors stem from misperception, 3D and cube nets from simulation gaps, with occasional heuristic over-use and logic inconsistency in chain-of-thought.

- We use an LLM-as-judge with a structured rubric to score how well model reasoning traces align with ground-truth metadata (e.g., shape interpretation, transformation accuracy). We then sorted model responses into quartiles based on alignment score and evaluated accuracy within each group.

Model	Task	Q1 (alignment)	Q2 (alignment)	Q3 (alignment)	Q4 (alignment)
GPT-4o	2D	61.1% (0.664)	100.0% (0.917)	100.0% (0.950)	100.0% (0.952)
	3D	33.3% (0.325)	100.0% (0.875)	100.0% (0.927)	100.0% (0.973)
	Cube Net	0.0% (0.004)	0.0% (0.096)	0.0% (0.118)	100.0% (0.606)
	Tangram	0.0% (0.000)	50.0% (0.228)	100.0% (0.819)	100.0% (0.998)
Claude-3.5 Sonnet	2D	0.0% (0.214)	83.3% (0.768)	100.0% (0.931)	100.0% (0.959)
	3D	0.0% (0.168)	90.9% (0.727)	100.0% (0.909)	100.0% (0.981)
	Cube Net	0.0% (0.041)	0.0% (0.100)	0.0% (0.150)	100.0% (0.714)
	Tangram	0.0% (0.000)	42.9% (0.161)	100.0% (0.725)	100.0% (0.991)

Table 9: Task accuracy by alignment quartile for GPT-4o and Claude-3.5 Sonnet.

This analysis reveals a strong correlation between reasoning quality and final task accuracy. The prompt we used to scoring model reasoning is shown below.

Reasoning Error Analysis Rubrics

You are an expert spatial-reasoning judge.
Given the information blocks below, score the model's reasoning using the rubric.

Question: {`question_text`}

Ground Truth

- Start shape: {`initial_shape_desc`}
- {`transformation_step_desc`} {`transformation_outcome_desc`}
- {`choice_shape_desc`}
- Correct answer: {`gt_answer`}

Model Response: {`model_response`}
Model Final Answer: {`model_pred`}

Rubric — assign 0 or 1 to each item

1. Shape interpretation
2. Transformation comprehension
3. Spatial simulation accuracy
4. Answer justification (choice evaluation)
5. Logical consistency

Return JSON

```
{
  "shape_interpretation": {"score": 0|1, "details": ""},
  "transformation_comprehension": {"score": 0|1, "details": ""},
  "spatial_simulation_accuracy": {"score": 0|1, "details": ""},
  "answer_justification": {"score": 0|1, "details": ""},
  "logical_consistency": {"score": 0|1, "details": ""},
  "primary_errors": [],
  "overall_assessment": ""
}
```

Number of visual simulations vs. Performance, We investigated this question briefly in Table 5. The key insight is that the benefit of visual simulations depends on task complexity and where in the sequence the decisive information appears. We also conducted a stricter ablation that incrementally reveals 0 to 3 simulation frames. The takeaway is straightforward: visual simulations help only when the model can integrate them; otherwise they add noise.

Because examples have different total chain lengths, the same column mixes complete simulations for short chains (e.g., all 2–step tasks are already complete at # simulations = 1) and partial simulations for longer ones (e.g., only one of two frames visible at # simulations = 1 for 3–step tasks). If a model could perfectly integrate every extra frame, scores would rise monotonically. However, the results above suggest otherwise, especially for more complex tasks. These results again highlight a core limitation: models lack the capacity to mentally simulate and reason over visual sequences, a skill that humans perform reliably.

Model	**Task**	# of simulations = 0	1	2	3
GPT-4o	2D	71.2	78.0	**85.6**	—
	3D	65.5	67.2	**69.6**	—
	Cube Net	50.3	**50.5**	49.2	49.1
	Tangram	52.5	51.7	46.6	**54.7**
Claude-3.5	2D	65.9	67.7	**75.3**	—
	3D	51.5	**60.8**	53.9	—
	Cube Net	**52.3**	51.3	51.7	50.0
	Tangram	59.0	60.1	62.2	**69.0**

Table 10: Ablation with 0–3 intermediate simulations. Bold indicates best performance per row.

Reasoning Efforts vs. Performance. In addition to the results of o3 reported in Table 1, we further report o3 performance across different reasoning efforts (low, auto, high), as it is specifically optimized to "think with images" and perform extended reasoning.

Task (w/o vs. w/ VSim)	**o3-Low**	**o3-Auto**	**o3-High**	**Human**
2D	88.1 / **92.3**	87.5 / 89.3	**89.7** / 89.5	96.8/ 98.6
3D	73.7 / 75.3	**75.2 / 78.4**	73.9 / 76.5	94.6 / 97.0
Cube Net	65.3 / 71.7	**68.4 / 72.5**	66.3 / 71.1	98.3 / 99.0
Tangram	**68.6** / 76.5	**68.6** / 82.1	66.4 / **82.8**	91.5 / 95.8
Video-temporal	**55.8**	51.4	54.3	99.0
Perspective	43.6	42.8	**44.0**	98.1
Overall	**67.5**	**67.5**	68.8	97.1

Table 11: Performance of o3 model across different reasoning efforts (Low, Auto, High) compared to human annotators on STARE tasks. For tasks with visual simulation (VSim), accuracy is reported as "w/o / w/ VSim".

While o3 outperforms earlier models such as o1 and GPT-4o, its performance on STARE remains significantly below human-level. Notably, o3 leverages visual simulations more effectively than prior models. However, humans show only small performance gaps between conditions with and without visual simulation, reflecting their ability to mentally simulate transformations. In contrast, o3 exhibits substantial drops without external visuals, highlighting a key limitation: the inability to perform internal, structured visual simulation–a core component of human spatial reasoning.

In addition, while o3 has reported to benefit from extended reasoning, simply increasing the reasoning efforts from low to high does not guarantee better spatial reasoning.

Additional Results on Human Evaluation. In Table 1, we reported the average human performance and average response time for each task across 5 participants. In Tab. 12, wer further report the standard deviation. Mean accuracy across five annotators is 97.07% with a standard deviation of 0.47, indicating that the questions in STARE are well-defined.

Task	Accuracy (%)	Time (s)
2D	96.75 ± 1.30	14.23 ± 1.99
2D + VSim	98.56 ± 1.10	10.95 ± 1.51
3D	94.61 ± 1.44	17.06 ± 5.08
3D + VSim	96.98 ± 0.80	12.53 ± 0.65
Cube Net	98.29 ± 1.65	13.67 ± 3.66
Cube Net + VSim	98.86 ± 1.46	5.16 ± 0.49
Tangram	91.53 ± 2.56	27.98 ± 5.85
Tangram + VSim	95.78 ± 1.07	10.08 ± 4.20
Temporal	99.03 ± 0.98	16.19 ± 3.96
Perspective	98.10 ± 0.17	18.04 ± 4.46
Overall	**97.07 ± 0.47**	–

Table 12: Task accuracy and response time across STARE benchmark tasks, with and without visual simulation (VSim).

Correlation Analysis between Synthetic tasks and Real tasks. In Section 3.2, we briefly discussed the correlation between averaged model performance on synthetic tasks (including 2D transformation, 3D transformation, cube net folding and tangram puzzle) and that on real-world tasks (including temporal frame reasoning and perspective reasoning). Fig. 7 shows the averaged model performance on synthetic and real-world tasks across 11 models and the fitted line with correlation coefficient $r \approx 0.88$.

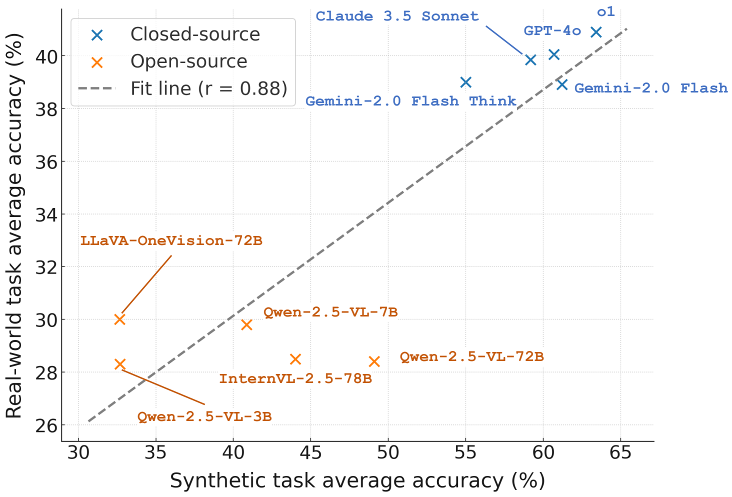


Figure 7: Correlation between model performance on synthetic tasks and that on real-world tasks.

Note that for open-source models, the real-world task performance is close to random guessing (29%). Removing the open-source models, the correlation coefficient decreased to $r \approx 0.58$, still showing a weak but positive correlation between synthetic task performance and real-world task performance.

F DATA CURATION DETAILS

Fig. 8 presents the overall composition of STARE. Tab. 13 details the number of instances for each task in STARE, further broken down by whether the input contains an explicit intermediate visual simulations.

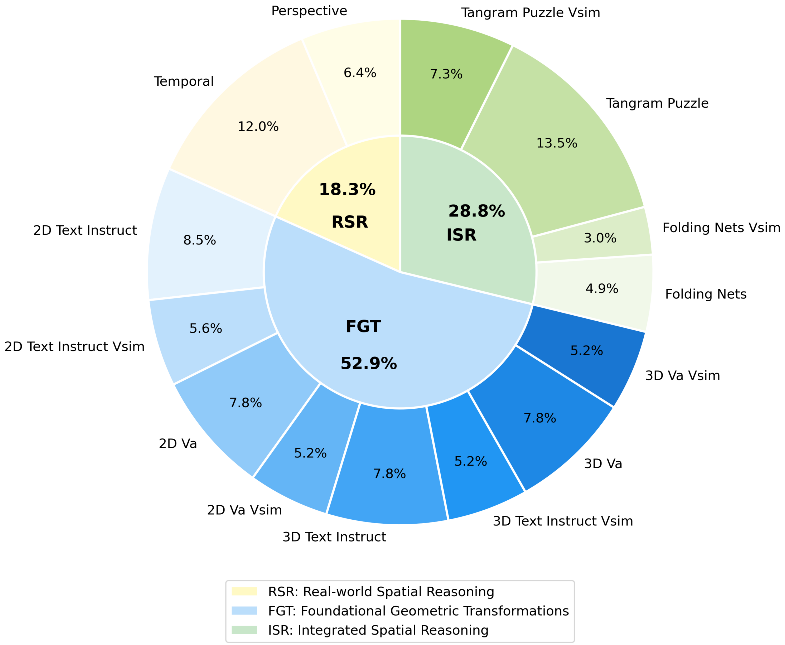


Figure 8: Data Statistics of STARE.

Task category	Without visual simulation	With visual simulation	Total
Foundational Geometric Transformations			
2D transformations	639	423	1,062
3D transformations	612	408	1,020
Integrated Spatial Reasoning			
Cube net folding	193	120	313
Tangram puzzle	532	289	821
Real-world Spatial Reasoning			
Perspective reasoning	250	–	250
Temporal frame reasoning	471	–	471
Total	2,697	1,240	3,937

Table 13: Dataset statistics grouped by task category and by the presence of full intermediate visual simulation.

Below, we summarize the design space of data curation for synthetic tasks, including (1) 2D Transformations (§F.1); (2) 3D Transformations (§F.2); (3) Cube Net Folding (§F.3); and (4) Tangram Puzzles (§F.4);

F.1 2D TRANSFORMATIONS

Shape generation. Shapes are selected from a fixed set and assigned properties as follows:

- **Types:** Circle, Square, Rectangle, Triangle, Ellipse, Hexagon, Pentagon.
- **Colors:** Face color is a random RGB tuple ($r, g, b \in [0, 1]$); edge color is fixed (black).
- **Center & Size:** All shapes are centered at $(0, 0)$. For circles, squares, triangles, hexagons, and pentagons, size is a scalar drawn from $[30, 35]$; for rectangles and ellipses, size is a tuple (width in $[30, 35]$, height in $[20, 25]$).

Transformations. A sequence of randomly sampled operations is applied to the shapes:

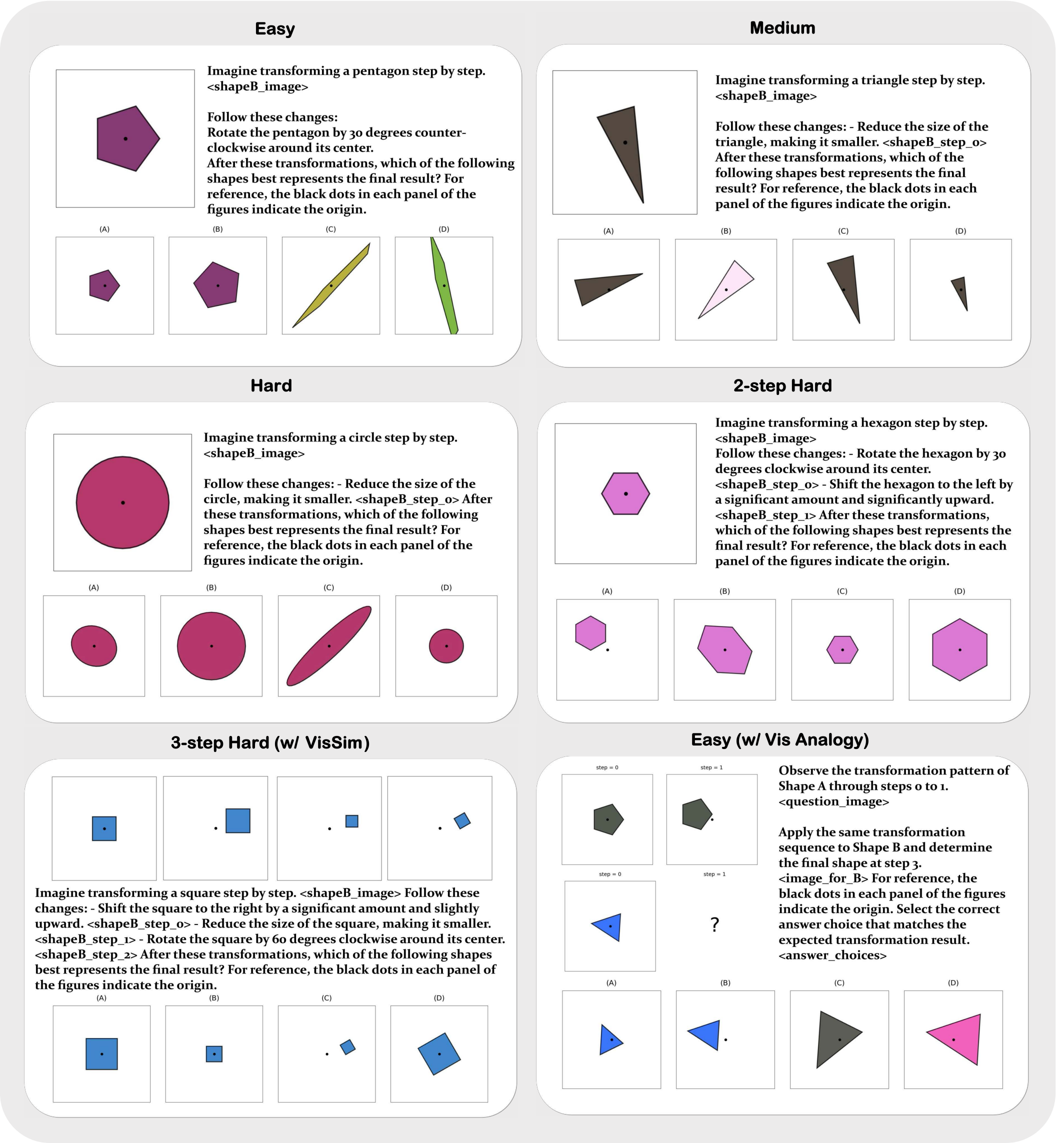
- **Rotate:**
  - *Squares:* $\pm 30°$, $\pm 60°$ (avoiding $90°$).

  - *Hexagons:* ±30°, ±90°.
  - *Others:* ±30°, ±60°, or ±90°.

  Rotation is applied w.r.t the shape's center.

- **Flip:** Horizontal (about $y = 0$) or vertical (about $x = 0$); not applied when the shape is centered at $(0, 0)$ for symmetric shapes such as square, circle and etc.
- **Translate:** (dx, dy) with $dx, dy \in \{-30, -10, 0, 10, 30\}$ with constraints to ensure a nonzero translation.
- **Scale:** Factors chosen from $\{0.5, 2.0\}$, ensuring the resultant size is within roughly $[10, 40]$.
- **Shear:** Parameters $(\text{shear}_x, \text{shear}_y)$ are drawn from approximately $[-1, 1]$, with constraints to ensure a perceptible skew. Shear is excluded for 2D text instructed transformation tasks, as human participants find it hard to describe the degree of shear such that they can differentiate among the answer candidates.

Number of Transformation Steps. The final dataset contains instances with 1, 2, or 3 transformation steps.



Imagine transforming a pentagon step by step. <shapeB_image>

Follow these changes:
Rotate the pentagon by 30 degrees counter-clockwise around its center.
After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Imagine transforming a triangle step by step. <shapeB_image>

Follow these changes: - Reduce the size of the triangle, making it smaller. <shapeB_step_0> After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Imagine transforming a circle step by step. <shapeB_image>

Follow these changes: - Reduce the size of the circle, making it smaller. <shapeB_step_0> After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

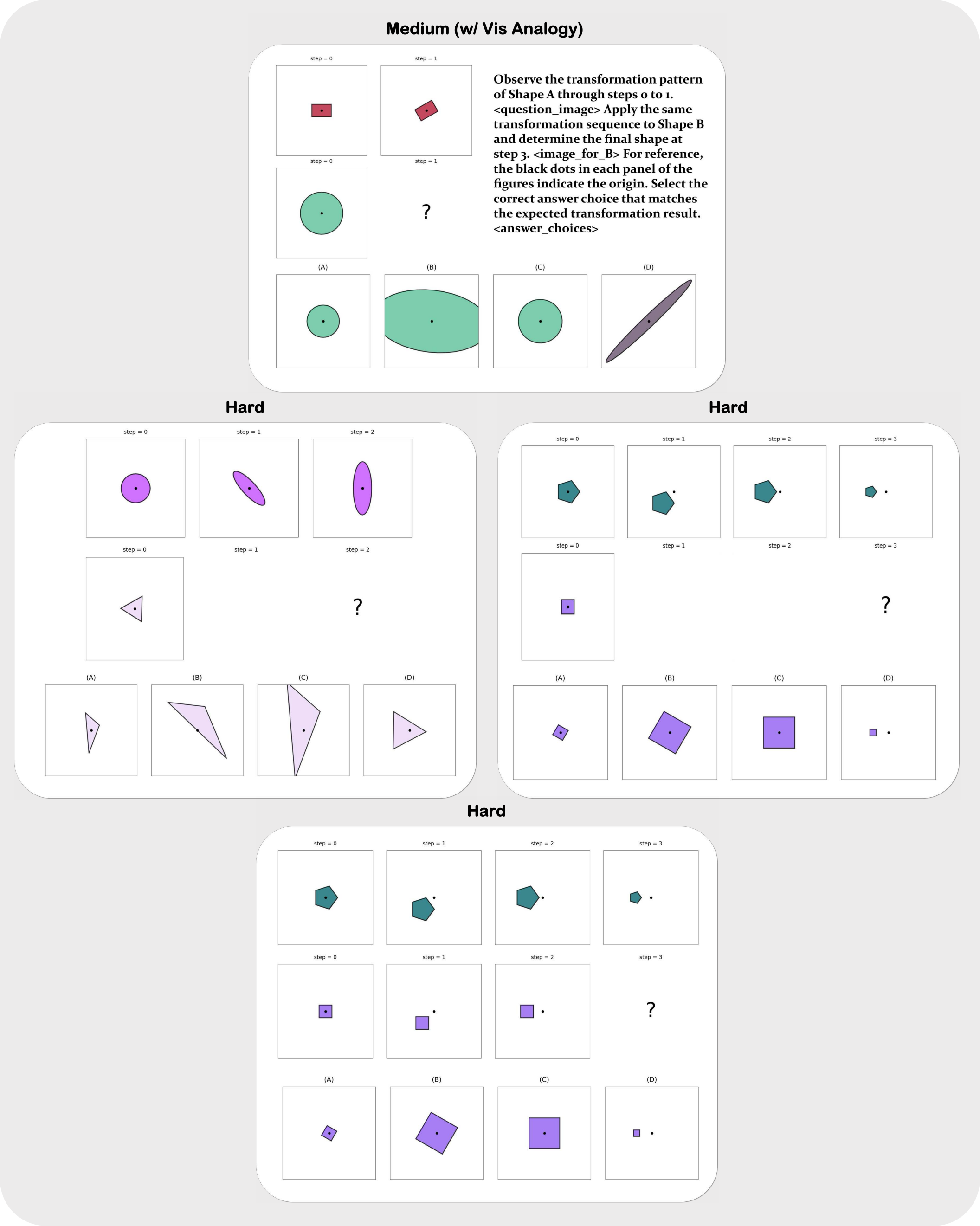
Imagine transforming a hexagon step by step. <shapeB_image>
Follow these changes: - Rotate the hexagon by 30 degrees clockwise around its center. <shapeB_step_0> - Shift the hexagon to the left by a significant amount and significantly upward. <shapeB_step_1> After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Imagine transforming a square step by step. <shapeB_image> Follow these changes: - Shift the square to the right by a significant amount and slightly upward. <shapeB_step_0> - Reduce the size of the square, making it smaller. <shapeB_step_1> - Rotate the square by 60 degrees clockwise around its center. <shapeB_step_2> After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Observe the transformation pattern of Shape A through steps 0 to 1. <question_image>

Apply the same transformation sequence to Shape B and determine the final shape at step 3. <image_for_B> For reference, the black dots in each panel of the figures indicate the origin. Select the correct answer choice that matches the expected transformation result. <answer_choices>

Figure 9: Design space of 2D Transformations (1).



Observe the transformation pattern of Shape A through steps 0 to 1. <question_image> Apply the same transformation sequence to Shape B and determine the final shape at step 3. <image_for_B> For reference, the black dots in each panel of the figures indicate the origin. Select the correct answer choice that matches the expected transformation result. <answer_choices>

Figure 10: Design space of 2D Transformations (2).

F.2 3D TRANSFORMATIONS

Shape generation. 3D objects are loaded from external blend files and instantiated with random properties defined in a JSON file. Their attributes include:

- **Types:** Various 3D models such as cube, sphere, cone, cylinder, torus, pyramid, etc.
- **Colors & Materials:** Colors are sampled from a predefined set, and materials are selected from external files.
- **Size & Location:** Objects are assigned a size scalar (from the JSON-specified values) and an initial 3D location (typically near the origin), with adjustments to ensure they remain above the ground plane.

Transformations. A sequence of randomly sampled operations is applied to the objects in 3D space:

- **Translate:**
  - *Axis selection:* Randomly choose one or more axes from *x*, *y*, and *z* (e.g., “x”, “xy”, “xz”, “yz”).
  - *Displacement:* Translations are applied with discrete displacements: along *x* and *y* by ± 2 units and along *z* by ± 1 unit, with constraints to keep the object above the ground ($z \geq 0$).
- **Rotate:**
  - *Axis:* A single rotation axis is chosen randomly from *x*, *y*, or *z*.
  - *Angle:* The rotation angle is drawn from a discrete set (typically $\pm 30°$, $\pm 60°$, or $\pm 90°$), with the range sometimes adjusted for specific shapes (e.g., cubes or pyramids).
  - Rotation is applied about the object’s center.
- **Shear:**
  - *Plane:* The shear operation is applied along one of three directional pairs: x_y, x_z, or y_z.
  - *Factors:* Two shear factors are sampled uniformly from the interval $[0.2, 1.0]$, with an enforced minimum difference (approximately 0.4) to ensure a perceptible skew.
- **Scale:**
  - *Factor:* A uniform scaling factor is chosen from $0.5, 2.0$, either reducing or enlarging the object while keeping its final size within acceptable bounds.
- **Flip:**
  - *Direction:* The object is reflected along a principal axis—flipped horizontally (reflection across the *x*-axis) or vertically (reflection across the *y*-axis).

All transformation operations are applied sequentially, updating the object’s 3D coordinates (including its bounding box and center) to reflect the cumulative effects.

Number of Transformation Steps. Instances are generated with transformation sequences comprising 1, 2, or 3 steps, where each step randomly selects one of the available operations. This multi-step approach enables a diverse design space of 3D transformations, as the operations can compound in various orders and combinations.

F.3 CUBE NET FOLDING

Net Representation. Cube nets are represented as collections of faces, where each face is defined by its vertices in 3D space. Additional attributes include:

- **Face Geometry:** Each face is a polygon (typically a quadrilateral) with vertex coordinates stored as NumPy arrays.
- **Connectivity:** A mapping of face connections identifies which faces share common edges, serving as potential hinges.
- **Visual Attributes:** Faces are rendered with colors (sampled from a colormap) and labeled with their keys for easy identification.

Folding Operations. The folding process simulates converting a 2D cube net into a 3D cube via a sequence of rotation operations:

- **Shared Edge Detection:** The algorithm locates the common edge between a candidate face and an already folded face. A tolerance is used to robustly identify two shared vertices.
- **Rotation Calculation:** Using the shared edge as a hinge, a rotation is computed with a fixed magnitude of 90° (i.e. $\pm\pi/2$ radians). The sign of the angle is chosen by comparing the candidate face’s center (projected onto the hinge’s perpendicular plane) with the desired direction toward the cube’s center, which is derived from the base face.

- **Recursive Propagation:** The rotation is applied not only to the candidate face but also recursively to all connected faces that have not been folded yet, ensuring that the entire net adjusts consistently.

Folding Sequence and Visualization. The design space supports iterative, step-by-step folding, with each step comprising:

- **Candidate Selection:** Among the faces not yet folded, the algorithm picks one that is connected to an already folded face.
- **Folding Parameters:** It computes the rotation axis (the shared edge) and the appropriate 90° rotation (with correct sign) to fold the face into its 3D position.
- **Instruction Generation:** Each fold is described in natural language (e.g., "Fold face 2 upwards towards face 3") based on changes in the face's center relative to the cube's base.
- **3D Rendering:** After each step, the current state of the net is visualized using a 3D plot (with Poly3DCollection) and saved as an image.

Perturbation and Validity. To enrich the design space and introduce challenge:

- **Perturbations:** Selected folding steps can be intentionally altered by inverting the rotation angle or modifying the rotation axis. This simulates errors or variations, yielding nets that might fold incorrectly.
- **Validity Checks:** Functions are provided to verify that folded faces do not overlap, that shared edges are consistently maintained, and that face connections remain intact. These checks ensure that the final folded cube is geometrically valid.

Dataset Generation and Perception Tasks. Beyond simulating the folding process, the design space incorporates mechanisms to create annotated datasets:

- **Instructional Sequences:** Detailed, step-by-step folding instructions (with corresponding images) are generated, supporting tasks that require understanding the folding procedure.
- **Perception Variants:** Additional tasks query the observer's perception—such as verifying if a particular face has been folded or determining the connectivity between faces—using intermediate folding images.

Randomness and Parameter Control. Stochastic elements pervade the folding simulation:

- Random seeds govern the selection of candidate faces, the decision to perturb a folding step, and the choice of rotation adjustments.
- This randomness ensures that a diverse range of cube nets and folding sequences are produced, which is crucial for generating robust datasets and for studying perception and reasoning in 3D folding tasks.

F.4 TANGRAM PUZZLE

Segmentation. The puzzle begins with an iterative segmentation algorithm that splits a full rectangular board into smaller pieces. The process is governed by a minimum piece size and a maximum number of pieces. At each segmentation step, the algorithm:

- Selects a splittable rectangle based on its area.
- Chooses a split direction (horizontal if the height is greater or vertical otherwise) and a split line ensuring both resulting pieces exceed the minimum size.
- Records each split as an action with details (original rectangle, split line, and direction) that form the basis for later textual instructions.

Piece Generation & Attributes. Each tangram piece is defined by its board coordinates (e.g., `(r0, r1, c0, c1)`) and derived properties such as area and dimensions. Additionally:

- **Colors:** Pieces are assigned unique, randomly generated colors.

- **Visualization:** Grid lines and labels are overlaid on each piece to indicate its boundaries and area, facilitating clear visualization during reassembly.

Scrambling and Transformation. Once segmented, pieces are scrambled to increase puzzle complexity. This involves applying a series of random transformation operations:

- **Rotation:** Each piece is rotated by a discrete angle chosen from $0^\circ, 30^\circ, 60^\circ, 90^\circ$.
- **Translation:** Pieces are repositioned into non-overlapping cells on a larger canvas.
- **Flip:** In some reassembly variants, horizontal or vertical flips are applied to further randomize the piece orientations.

G EXPERIMENTAL DETAILS

G.1 MODELS AND SETTINGS

To expedite response generation, we use the vLLM (Kwon et al., 2023) library, an open-source tool for fast LLM inference and serving. For all other cases, we load models directly using the Transformers (Wolf et al., 2020) library. All model sources are official and listed in Tab. 14. When evaluating different models, we use default hyperparameter values unless otherwise specified, with detailed parameter settings provided in Tab. 14. For all models, we explicitly prompt it with `Think step-by-step, and then put your final answer in \"\\boxed{}\".` to encourage chain-of-thought reasoning and for easier answer parsing.

Model	Parameter Setting	Source	URL
GPT-4o	temperature = 0.0	chatgpt-4o-latest	`https://platform.openai.com`
Claude 3.5 Sonnet	temperature = 0.0	claude-3-5-sonnet	`https://www.anthropic.com/`
Gemini 2.0 Flash	temperature = 0.0	gemini-2.0-flash-exp	`https://ai.google.dev/`
Gemini 2.0 Flash Thinking	temperature = 0.0	gemini-2.0-flash-thinking-exp-1219	`https://ai.google.dev/`
OpenAI o1	temperature = 0.0	o1-2024-12-17	`https://platform.openai.com`
OpenAI o3	reasoning-efforts=auto	o3-2025-04-16	`https://platform.openai.com`

Qwen2.5-VL-3B	do_sample=True,
	temperature = 0.7
		local checkpoint	`https://huggingface.co/Qwen/Qwen2.5-VL-3B-Instruct`

Qwen2.5-VL-7B	do_sample=True,
	temperature = 0.7
		local checkpoint	`https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct`

Qwen2.5-VL-72B	do_sample=True,
	temperature = 0.7
		local checkpoint	`https://huggingface.co/Qwen/Qwen2.5-VL-72B-Instruct`

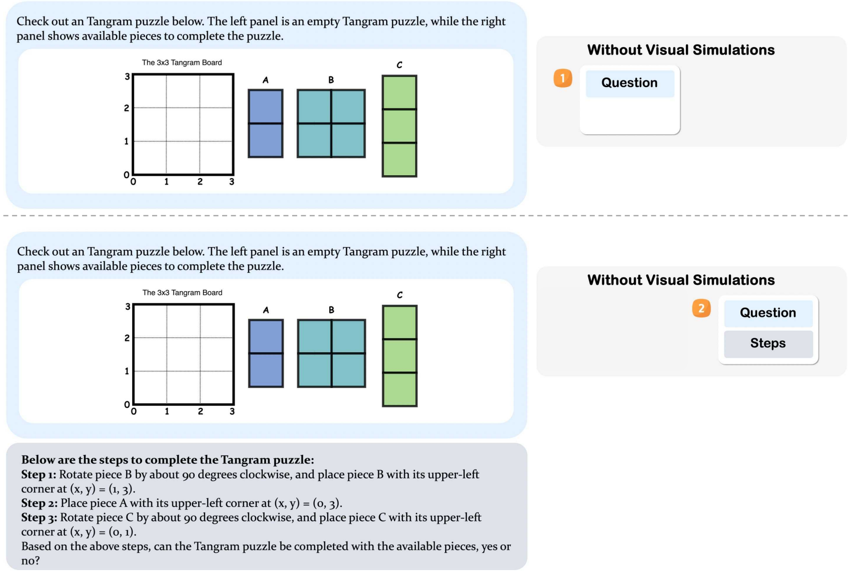
LLaVA-Onevision-72B	do_sample=True,
	temperature = 0.7
		local checkpoint	`https://huggingface.co/llava-hf/llava-onevision-qwen2-72b-ov-hf`

InternVL2.5-78B	do_sample=True,
	temperature = 0.7
		local checkpoint	`https://huggingface.co/OpenGVLab/InternVL2_5-78B`

Table 14: The sources of models used in the experiments and the hyperparameters configuration.

G.2 VISUALIZATION OF EVALUATION SETTINGS

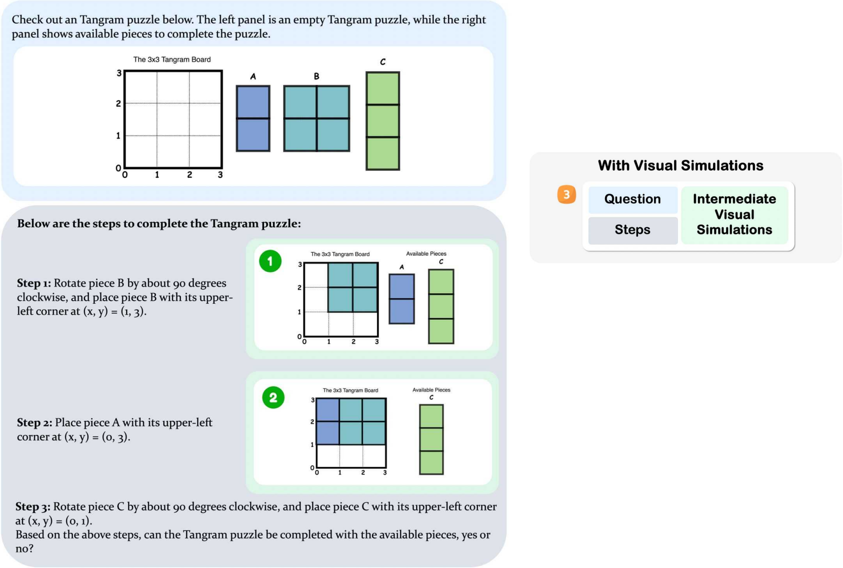
Fig.s 11–12 provide full visualizations of evaluation settings illustrated in Fig. 3. In addition, we show an example of how real-world spatial reasoning task – temporal frame reasoning is evaluated without visual simulation in Fig. 13.


Check out an Tangram puzzle below. The left panel is an empty Tangram puzzle, while the right panel shows available pieces to complete the puzzle.

Check out an Tangram puzzle below. The left panel is an empty Tangram puzzle, while the right panel shows available pieces to complete the puzzle.

Below are the steps to complete the Tangram puzzle:
Step 1: Rotate piece B by about 90 degrees clockwise, and place piece B with its upper-left corner at (x, y) = (1, 3).
Step 2: Place piece A with its upper-left corner at (x, y) = (0, 3).
Step 3: Rotate piece C by about 90 degrees clockwise, and place piece C with its upper-left corner at (x, y) = (0, 1).
Based on the above steps, can the Tangram puzzle be completed with the available pieces, yes or no?

Figure 11: Examples of Tangram Puzzle under "without Visual Simulations" Evaluation Setting (top: question-only, bottom: question+assembly steps).


Check out an Tangram puzzle below. The left panel is an empty Tangram puzzle, while the right panel shows available pieces to complete the puzzle.

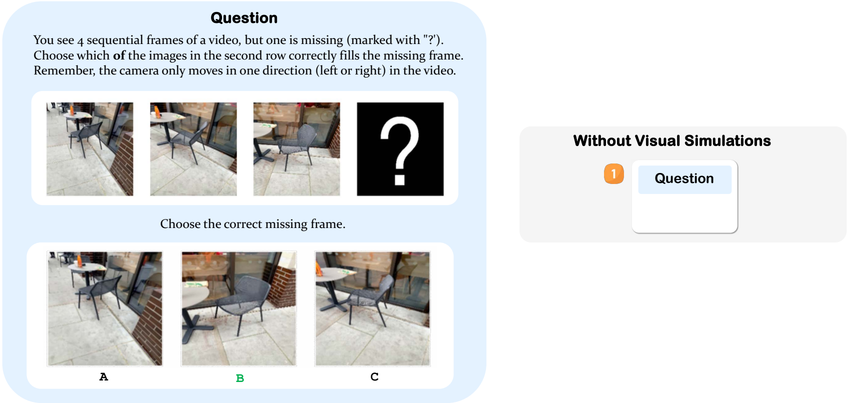
Below are the steps to complete the Tangram puzzle:

Step 1: Rotate piece B by about 90 degrees clockwise, and place piece B with its upper-left corner at (x, y) = (1, 3).

Step 2: Place piece A with its upper-left corner at (x, y) = (0, 3).

Step 3: Rotate piece C by about 90 degrees clockwise, and place piece C with its upper-left corner at (x, y) = (0, 1).
Based on the above steps, can the Tangram puzzle be completed with the available pieces, yes or no?

Figure 12: Example of Tangram Puzzle under "with Visual Simulations" Evaluation Setting.


Question

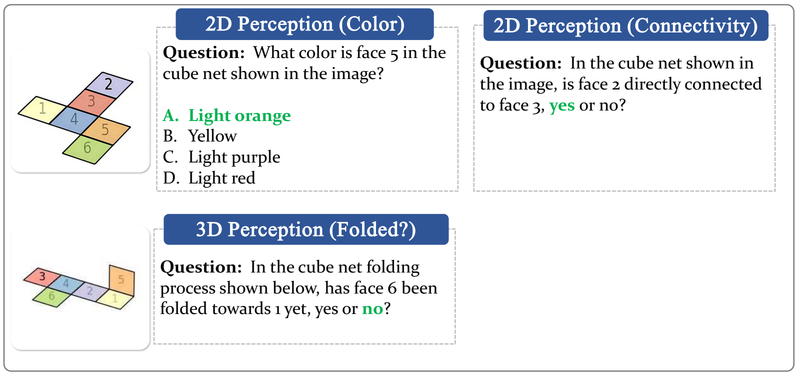
You see 4 sequential frames of a video, but one is missing (marked with "?"). Choose which **of** the images in the second row correctly fills the missing frame. Remember, the camera only moves in one direction (left or right) in the video.

Choose the correct missing frame.

Figure 13: Examples of Temporal Frame Reasoning under "without Visual Simulations" Evaluation Setting.

G.3 VISUALIZATIONS OF PERCEPTION PROBING QUESTIONS

In Fig. 5, Claude demonstrates a perceptual error: while it correctly identifies all face colors, it incorrectly perceives face 6 to be positioned beneath face 4, when it is actually located beneath face 5. Such errors prompt an important question regarding task performance: for challenging tasks like cube net folding, to what extent does the low performance stem from perceptual inaccuracies rather than deficiencies in simulation capabilities or an inability to correctly interpret simulation outcomes? We design probing questions to evaluate model performance 2D and 3D perception on cube nets (Fig. 14), which reveals that model fail substantially on 3D perception (Tab. 2), which may be the main bottleneck in understanding intermediate visualizations in cube net folding (Tab. 1).



2D Perception (Color)

Question: What color is face 5 in the cube net shown in the image?

A. **Light orange**
B. Yellow
C. Light purple
D. Light red

2D Perception (Connectivity)

Question: In the cube net shown in the image, is face 2 directly connected to face 3, **yes** or no?

3D Perception (Folded?)

Question: In the cube net folding process shown below, has face 6 been folded towards 1 yet, yes or **no**?

Figure 14: Exemplary questions on cube nets to probe model performance on 2D and 3D perception.

G.4 VISUALIZATIONS OF STARE TASK IN DIFFERENT REPRESENTATIONS

Fig.s 15–18 provide concrete examples of the input modalities evaluated in STARE. For every task family we visualize the image-only variant (the original format in STARE), the text-only variant (compact symbolic description that can be consumed without vision), and—where applicable—the combined image+text variant that concatenates the two.

- **2D and 3D transformations**. In the text-only panels, each object is serialized as `<shape>`, `<color>`, `<x,y>`, `<size>`, with attributes separated by commas (e.g., "`square, red, (3, 4), 2`"). The image+text panels place the same textual description beneath the image, so that language and vision can be attended to jointly.

- **Cube-net folding**. We flatten the cube into a 2D grid and enumerate its faces from 1 to 6. The text-only representation thus becomes a short digit string (e.g., "`123456`") or a block array that mirrors the spatial arrangement of the net.

- **Tangram puzzle**. Because rotations in the image cannot be expressed succinctly in the image+text setting, we show only image-only and text-only variants. Each piece is labeled alphabetically and encoded by a binary occupancy grid—rows of "`1`" indicate filled cells, yielding a representation that is both human-readable and unambiguous for MLLMs.

Together, these examples clarify the correspondence between the natural visual stimuli and the stripped-down symbolic forms used in our text-only experiments, as introduced in Section 3.3.

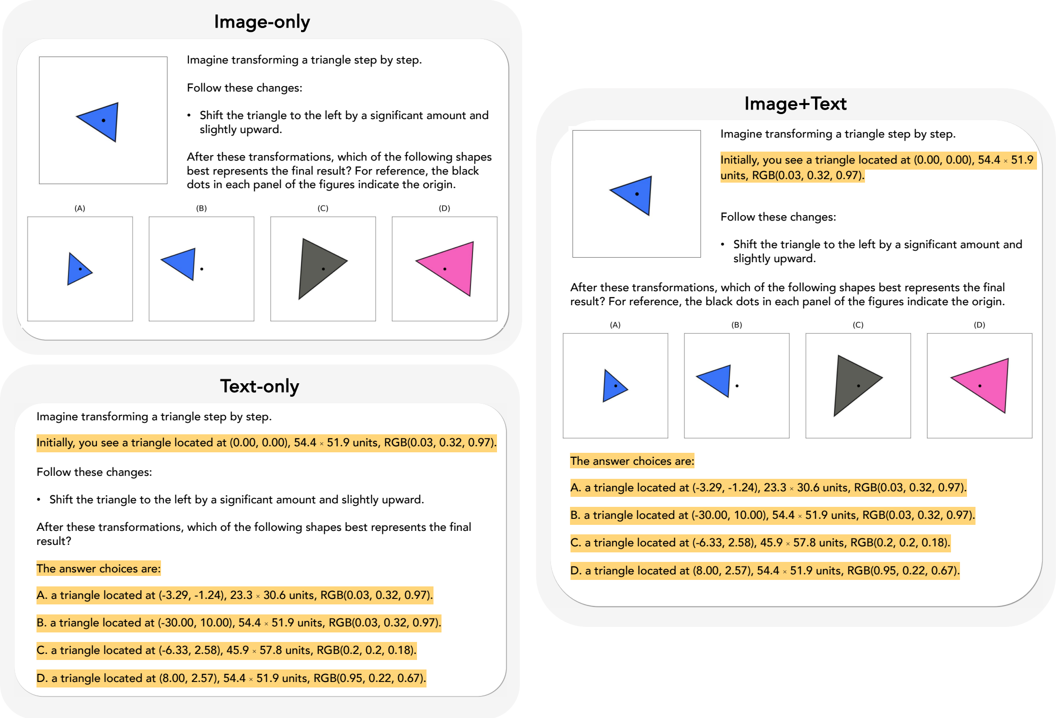


Image-only

Imagine transforming a triangle step by step.

Follow these changes:

- Shift the triangle to the left by a significant amount and slightly upward.

After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Text-only

Imagine transforming a triangle step by step.

Initially, you see a triangle located at (0.00, 0.00), 54.4 × 51.9 units, RGB(0.03, 0.32, 0.97).

Follow these changes:

- Shift the triangle to the left by a significant amount and slightly upward.

After these transformations, which of the following shapes best represents the final result?

The answer choices are:

A. a triangle located at (-3.29, -1.24), 23.3 × 30.6 units, RGB(0.03, 0.32, 0.97).

B. a triangle located at (-30.00, 10.00), 54.4 × 51.9 units, RGB(0.03, 0.32, 0.97).

C. a triangle located at (-6.33, 2.58), 45.9 × 57.8 units, RGB(0.2, 0.2, 0.18).

D. a triangle located at (8.00, 2.57), 54.4 × 51.9 units, RGB(0.95, 0.22, 0.67).

Image+Text

Imagine transforming a triangle step by step.

Initially, you see a triangle located at (0.00, 0.00), 54.4 × 51.9 units, RGB(0.03, 0.32, 0.97).

Follow these changes:

- Shift the triangle to the left by a significant amount and slightly upward.

After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

The answer choices are:

A. a triangle located at (-3.29, -1.24), 23.3 × 30.6 units, RGB(0.03, 0.32, 0.97).

B. a triangle located at (-30.00, 10.00), 54.4 × 51.9 units, RGB(0.03, 0.32, 0.97).

C. a triangle located at (-6.33, 2.58), 45.9 × 57.8 units, RGB(0.2, 0.2, 0.18).

D. a triangle located at (8.00, 2.57), 54.4 × 51.9 units, RGB(0.95, 0.22, 0.67).

Figure 15: Visualizations of 2D transformations (w/ text instructions) in different representations (upper left: image-only, lower left: text-only, right: image+text).

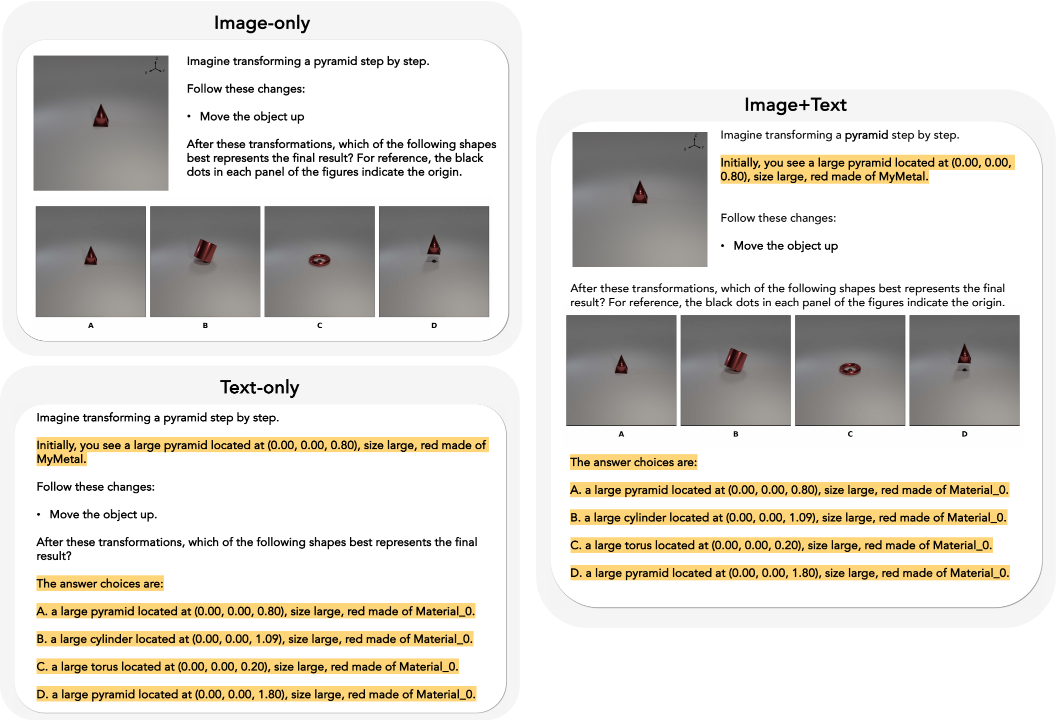


Image-only

Imagine transforming a pyramid step by step.

Follow these changes:

- Move the object up

After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

Text-only

Imagine transforming a pyramid step by step.

Initially, you see a large pyramid located at (0.00, 0.00, 0.80), size large, red made of MyMetal.

Follow these changes:

- Move the object up.

After these transformations, which of the following shapes best represents the final result?

The answer choices are:

A. a large pyramid located at (0.00, 0.00, 0.80), size large, red made of Material_0.

B. a large cylinder located at (0.00, 0.00, 1.09), size large, red made of Material_0.

C. a large torus located at (0.00, 0.00, 0.20), size large, red made of Material_0.

D. a large pyramid located at (0.00, 0.00, 1.80), size large, red made of Material_0.

Image+Text

Imagine transforming a **pyramid** step by step.

Initially, you see a large pyramid located at (0.00, 0.00, 0.80), size large, red made of MyMetal.

Follow these changes:

- Move the object up

After these transformations, which of the following shapes best represents the final result? For reference, the black dots in each panel of the figures indicate the origin.

The answer choices are:

A. a large pyramid located at (0.00, 0.00, 0.80), size large, red made of Material_0.

B. a large cylinder located at (0.00, 0.00, 1.09), size large, red made of Material_0.

C. a large torus located at (0.00, 0.00, 0.20), size large, red made of Material_0.

D. a large pyramid located at (0.00, 0.00, 1.80), size large, red made of Material_0.

Figure 16: Visualizations of 3D transformations (w/ text instructions) in different representations (upper left: image-only, lower left: text-only, right: image+text).

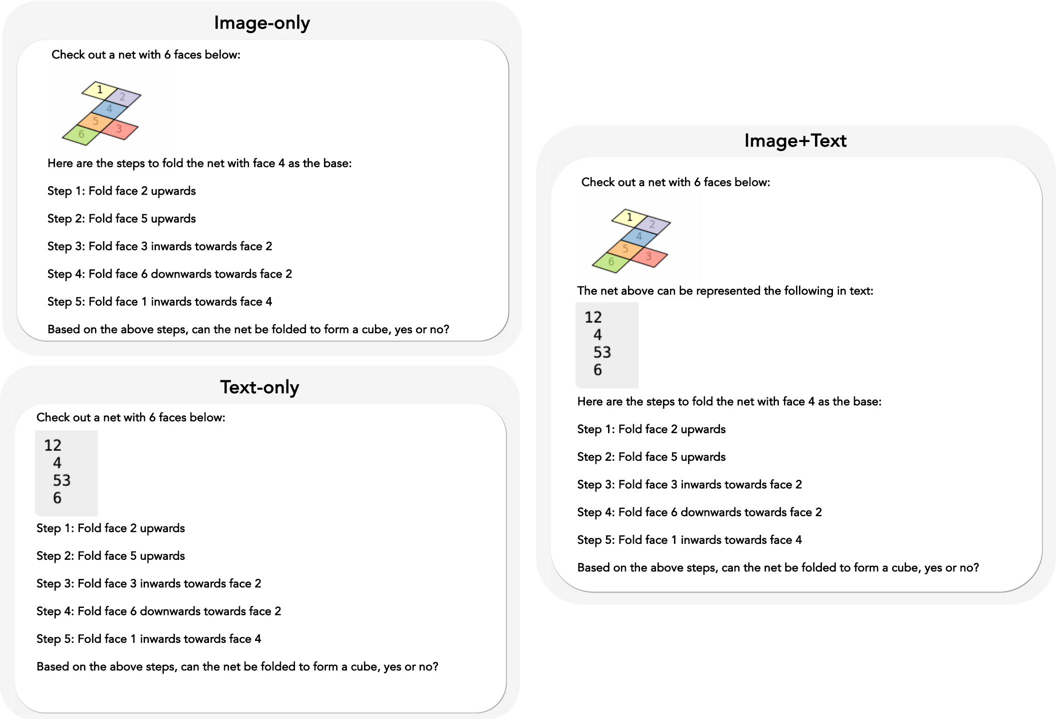


Image-only

Check out a net with 6 faces below:

Here are the steps to fold the net with face 4 as the base:

Step 1: Fold face 2 upwards

Step 2: Fold face 5 upwards

Step 3: Fold face 3 inwards towards face 2

Step 4: Fold face 6 downwards towards face 2

Step 5: Fold face 1 inwards towards face 4

Based on the above steps, can the net be folded to form a cube, yes or no?

Text-only

Check out a net with 6 faces below:

```
12
 4
 53
 6
```

Step 1: Fold face 2 upwards

Step 2: Fold face 5 upwards

Step 3: Fold face 3 inwards towards face 2

Step 4: Fold face 6 downwards towards face 2

Step 5: Fold face 1 inwards towards face 4

Based on the above steps, can the net be folded to form a cube, yes or no?

Image+Text

Check out a net with 6 faces below:

The net above can be represented the following in text:

```
12
 4
 53
 6
```

Here are the steps to fold the net with face 4 as the base:

Step 1: Fold face 2 upwards

Step 2: Fold face 5 upwards

Step 3: Fold face 3 inwards towards face 2

Step 4: Fold face 6 downwards towards face 2

Step 5: Fold face 1 inwards towards face 4

Based on the above steps, can the net be folded to form a cube, yes or no?

Figure 17: Visualizations of cube net folding in different representations (upper left: image-only, lower left: text-only, right: image+text).

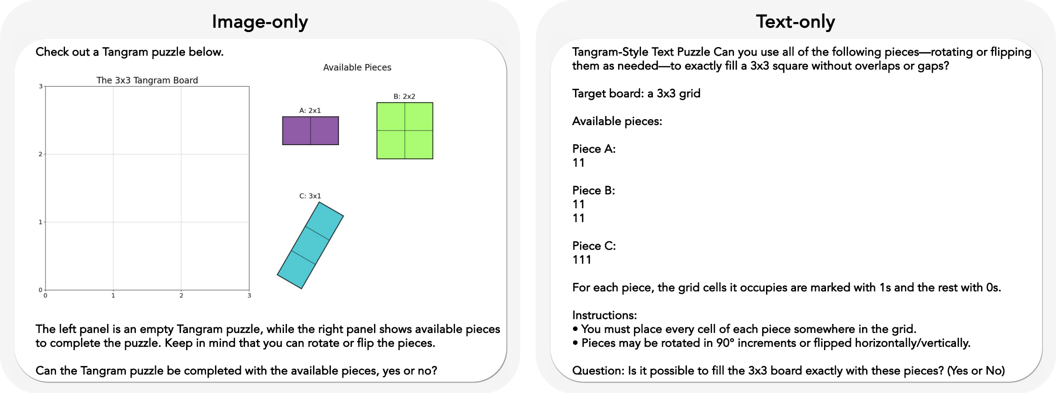

The 3x3 Tangram Board
0
1
2
3
Available Pieces
A: 2x1
B: 2x2
C: 3x1

Image-only

Check out a Tangram puzzle below.

The left panel is an empty Tangram puzzle, while the right panel shows available pieces to complete the puzzle. Keep in mind that you can rotate or flip the pieces.

Can the Tangram puzzle be completed with the available pieces, yes or no?

Text-only

Tangram-Style Text Puzzle Can you use all of the following pieces—rotating or flipping them as needed—to exactly fill a 3x3 square without overlaps or gaps?

Target board: a 3x3 grid

Available pieces:

Piece A:
11

Piece B:
11
11

Piece C:
111

For each piece, the grid cells it occupies are marked with 1s and the rest with 0s.

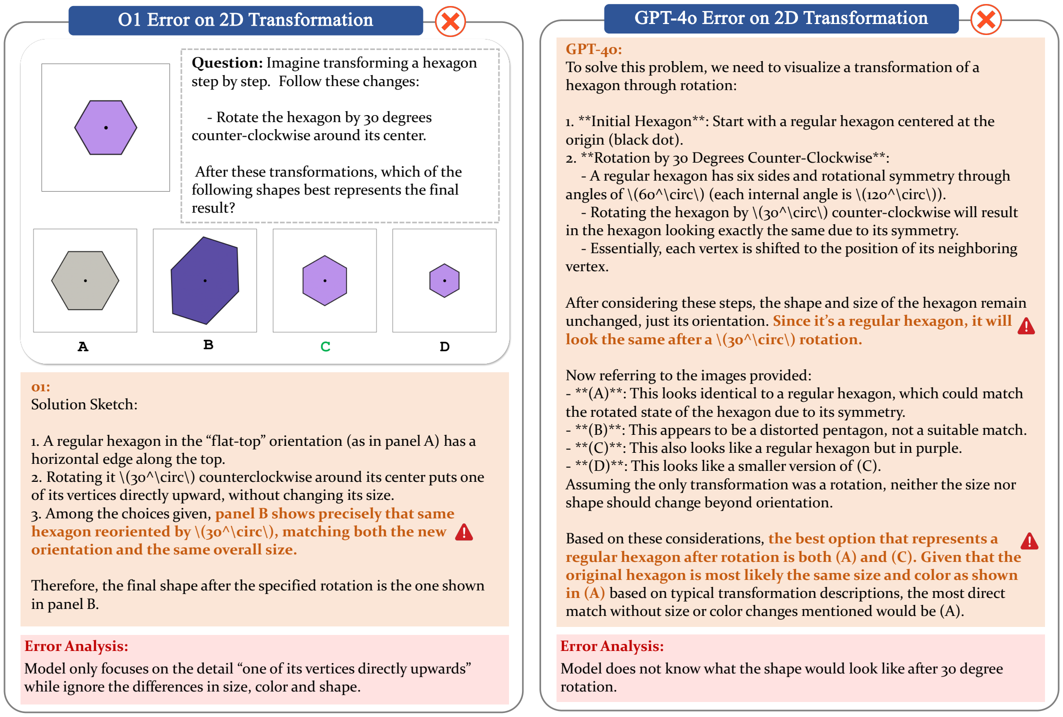
Instructions:
- You must place every cell of each piece somewhere in the grid.
- Pieces may be rotated in 90° increments or flipped horizontally/vertically.

Question: Is it possible to fill the 3x3 board exactly with these pieces? (Yes or No)

Figure 18: Visualizations of tangram in different representations (left: image-only, right: text-only).

G.5 CASE STUDY

Fig. 19 presents error cases on 2D transformation tasks from o1 and GPT-4o. The o1 explanation correctly identifies that a regular hexagon rotated 30° counter-clockwise will have a vertex pointing straight up, so it chooses panel B, but its justification overlooks potential differences in size, color, or shape variants among the answer choices. GPT-4o, meanwhile, mistakenly claims the hexagon looks "exactly the same" after a 30° rotation (confusing 60° symmetry with 30°) and vacillates between panels A and C, revealing it doesn't fully grasp the visual outcome of the specified rotation.



O1 Error on 2D Transformation

Question: Imagine transforming a hexagon step by step. Follow these changes:

- Rotate the hexagon by 30 degrees counter-clockwise around its center.

After these transformations, which of the following shapes best represents the final result?

o1:
Solution Sketch:

1. A regular hexagon in the "flat-top" orientation (as in panel A) has a horizontal edge along the top.
2. Rotating it \(30^\circ\) counterclockwise around its center puts one of its vertices directly upward, without changing its size.
3. Among the choices given, **panel B shows precisely that same hexagon reoriented by \(30^\circ\), matching both the new orientation and the same overall size.**

Therefore, the final shape after the specified rotation is the one shown in panel B.

Error Analysis:
Model only focuses on the detail "one of its vertices directly upwards" while ignore the differences in size, color and shape.

GPT-4o Error on 2D Transformation

GPT-4o:
To solve this problem, we need to visualize a transformation of a hexagon through rotation:

1. **Initial Hexagon**: Start with a regular hexagon centered at the origin (black dot).
2. **Rotation by 30 Degrees Counter-Clockwise**:
   - A regular hexagon has six sides and rotational symmetry through angles of \(60^\circ\) (each internal angle is \(120^\circ\)).
   - Rotating the hexagon by \(30^\circ\) counter-clockwise will result in the hexagon looking exactly the same due to its symmetry.
   - Essentially, each vertex is shifted to the position of its neighboring vertex.

After considering these steps, the shape and size of the hexagon remain unchanged, just its orientation. **Since it's a regular hexagon, it will look the same after a \(30^\circ\) rotation.**

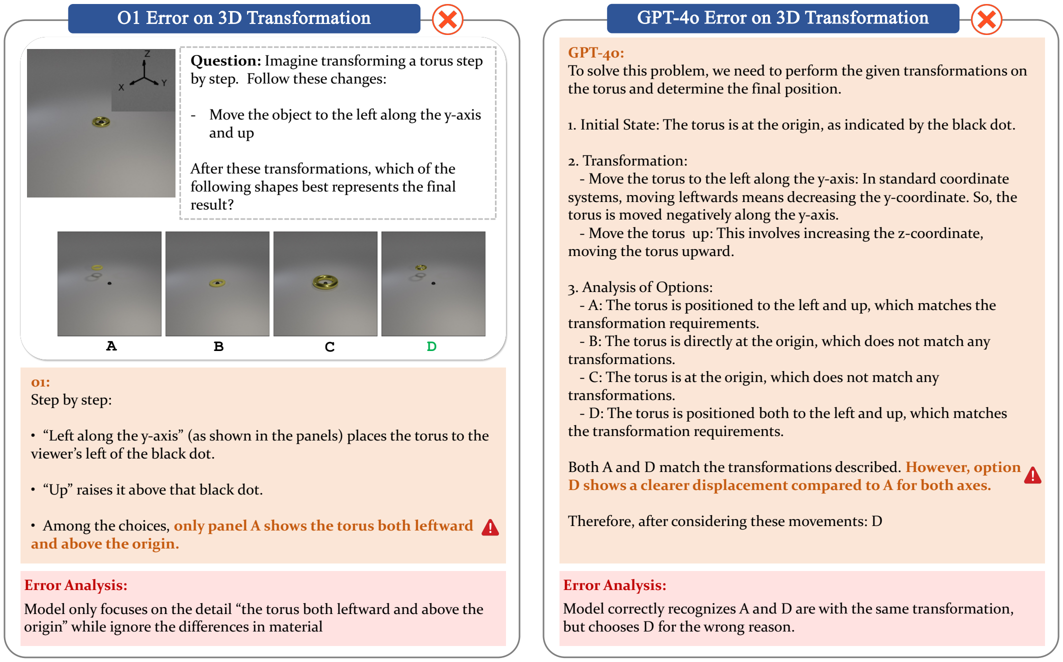
Now referring to the images provided:
- **(A)**: This looks identical to a regular hexagon, which could match the rotated state of the hexagon due to its symmetry.
- **(B)**: This appears to be a distorted pentagon, not a suitable match.
- **(C)**: This also looks like a regular hexagon but in purple.
- **(D)**: This looks like a smaller version of (C).
Assuming the only transformation was a rotation, neither the size nor shape should change beyond orientation.

Based on these considerations, **the best option that represents a regular hexagon after rotation is both (A) and (C). Given that the original hexagon is most likely the same size and color as shown in (A)** based on typical transformation descriptions, the most direct match without size or color changes mentioned would be (A).

Error Analysis:
Model does not know what the shape would look like after 30 degree rotation.

Figure 19: Error cases of o1 and GPT-4o on 2D transformation.

Fig. 20 presents error cases on 3D transformation tasks from o1 and GPT-4o. The o1 response accurately places the torus up and left of the origin and selects panel A, but it overlooks that material/appearance differences could also rule out that choice. GPT-4o correctly sees that both panels A and D share the required displacement, yet it arbitrarily favors D, giving a justification ("clearer displacement") unrelated to the stated transformation criteria, so its final selection is wrong despite partially sound reasoning.



O1 Error on 3D Transformation

Question: Imagine transforming a torus step by step. Follow these changes:

- Move the object to the left along the y-axis and up

After these transformations, which of the following shapes best represents the final result?

o1:
Step by step:

- "Left along the y-axis" (as shown in the panels) places the torus to the viewer's left of the black dot.
- "Up" raises it above that black dot.
- Among the choices, **only panel A shows the torus both leftward and above the origin.**

Error Analysis:
Model only focuses on the detail "the torus both leftward and above the origin" while ignore the differences in material

GPT-4o Error on 3D Transformation

GPT-4o:
To solve this problem, we need to perform the given transformations on the torus and determine the final position.

1. Initial State: The torus is at the origin, as indicated by the black dot.

2. Transformation:
   - Move the torus to the left along the y-axis: In standard coordinate systems, moving leftwards means decreasing the y-coordinate. So, the torus is moved negatively along the y-axis.
   - Move the torus up: This involves increasing the z-coordinate, moving the torus upward.

3. Analysis of Options:
   - A: The torus is positioned to the left and up, which matches the transformation requirements.
   - B: The torus is directly at the origin, which does not match any transformations.
   - C: The torus is at the origin, which does not match any transformations.
   - D: The torus is positioned both to the left and up, which matches the transformation requirements.

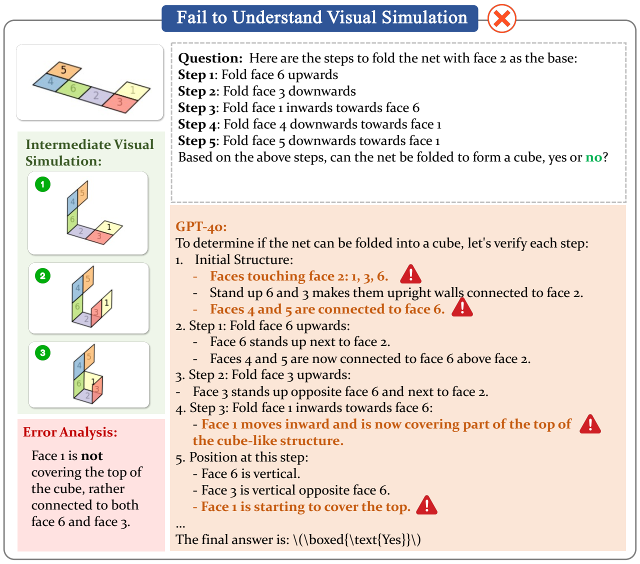
Both A and D match the transformations described. **However, option D shows a clearer displacement compared to A for both axes.**

Therefore, after considering these movements: D

Error Analysis:
Model correctly recognizes A and D are with the same transformation, but chooses D for the wrong reason.

Figure 20: Error cases of o1 and GPT-4o on 3D transformation.

On cube net folding task, besides the perception error from Claude in Fig. 5 and the text simulation error from GPT-4o in Fig. 1, when provided with intermediate visual simulation, models like GPT-4o still struggles with understanding the intermediate visual cues. In Fig. 21, GPT-4o asserts that face 1 becomes the cube's top surface, whereas face 1 is actually enclosed between faces 6 and 3. This misrepresentation of face adjacency highlights GPT-4o's difficulty in maintaining accurate 3D spactial cognition, which led to the wrong final result. This observation aligns with the conclusion from the perception probing test on cube nets in Section 3.3, where GPT-4o fails substantially on 3D perception than 2D perception.



Fail to Understand Visual Simulation ☒

Question: Here are the steps to fold the net with face 2 as the base:
Step 1: Fold face 6 upwards
Step 2: Fold face 3 downwards
Step 3: Fold face 1 inwards towards face 6
Step 4: Fold face 4 downwards towards face 1
Step 5: Fold face 5 downwards towards face 1
Based on the above steps, can the net be folded to form a cube, yes or **no**?

Intermediate Visual Simulation:

GPT-4o:
To determine if the net can be folded into a cube, let's verify each step:
1. Initial Structure:
    - **Faces touching face 2: 1, 3, 6.** ⚠
    - Stand up 6 and 3 makes them upright walls connected to face 2.
    - **Faces 4 and 5 are connected to face 6.** ⚠
2. Step 1: Fold face 6 upwards:
    - Face 6 stands up next to face 2.
    - Faces 4 and 5 are now connected to face 6 above face 2.
3. Step 2: Fold face 3 upwards:
- Face 3 stands up opposite face 6 and next to face 2.
4. Step 3: Fold face 1 inwards towards face 6:
    - **Face 1 moves inward and is now covering part of the top of the cube-like structure.** ⚠
5. Position at this step:
    - Face 6 is vertical.
    - Face 3 is vertical opposite face 6.
    - **Face 1 is starting to cover the top.** ⚠

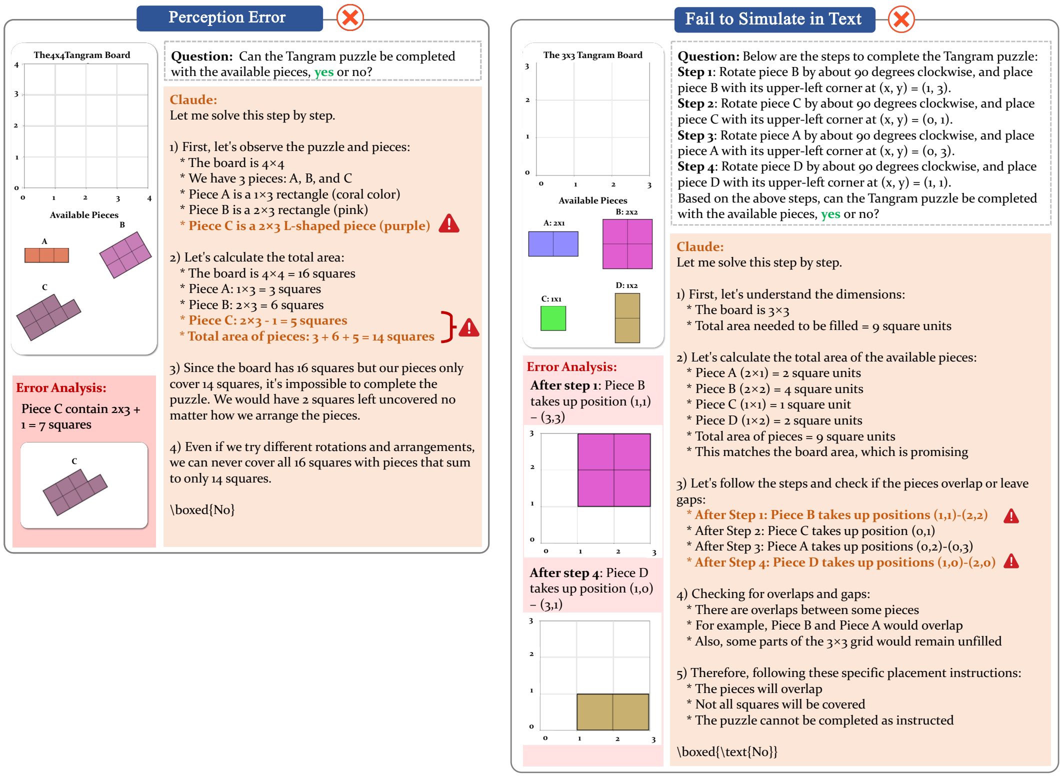
...
The final answer is: \(\boxed{\text{Yes}}\)

Error Analysis:

Face 1 is **not** covering the top of the cube, rather connected to both face 6 and face 3.

Figure 21: A error case from GPT-4o on cube net folding, where the model fails to understand the intermediate visual simulation.

For tangram puzzles, the left example of Fig. 22 shows how Claude takes a shortcut by conducting an area-based feasibility check, correctly noting that the 4×4 board contains 16 unit squares, but it erroneously counts the L-shaped piece C as 5 rather than its actual 7 squares. This miscalculation reduces the summed piece area to 14 instead of the correct 16, leading to conclude—incorrectly—that the puzzle cannot be completed. The right example of Fig. 22 shows how Claude fails to reason about the piece positions when given clear step-by-step instructions. Claude correctly interprets the coordinate instructions and board dimensions, yet its spatial reasoning falters when projecting how the rotated pieces occupy the grid. It underestimates the extents of pieces B and D, inventing overlaps and gaps that never arise, and therefore erroneously concludes the puzzle cannot be completed. This error stems from an inaccurate mental simulation of spatial relationships.



Perception Error

Question: Can the Tangram puzzle be completed with the available pieces, yes or no?

Claude:
Let me solve this step by step.

1) First, let's observe the puzzle and pieces:
   * The board is 4×4
   * We have 3 pieces: A, B, and C
   * Piece A is a 1×3 rectangle (coral color)
   * Piece B is a 2×3 rectangle (pink)
   *** Piece C is a 2×3 L-shaped piece (purple)**

2) Let's calculate the total area:
   * The board is 4×4 = 16 squares
   * Piece A: 1×3 = 3 squares
   * Piece B: 2×3 = 6 squares
   *** Piece C: 2×3 - 1 = 5 squares**
   *** Total area of pieces: 3 + 6 + 5 = 14 squares**

3) Since the board has 16 squares but our pieces only cover 14 squares, it's impossible to complete the puzzle. We would have 2 squares left uncovered no matter how we arrange the pieces.

4) Even if we try different rotations and arrangements, we can never cover all 16 squares with pieces that sum to only 14 squares.

\boxed{No}

Error Analysis:

Piece C contain 2x3 + 1 = 7 squares

Fail to Simulate in Text

Question: Below are the steps to complete the Tangram puzzle:
Step 1: Rotate piece B by about 90 degrees clockwise, and place piece B with its upper-left corner at (x, y) = (1, 3).
Step 2: Rotate piece C by about 90 degrees clockwise, and place piece C with its upper-left corner at (x, y) = (0, 1).
Step 3: Rotate piece A by about 90 degrees clockwise, and place piece A with its upper-left corner at (x, y) = (0, 3).
Step 4: Rotate piece D by about 90 degrees clockwise, and place piece D with its upper-left corner at (x, y) = (1, 1).
Based on the above steps, can the Tangram puzzle be completed with the available pieces, yes or no?

Claude:
Let me solve this step by step.

1) First, let's understand the dimensions:
   * The board is 3×3
   * Total area needed to be filled = 9 square units

2) Let's calculate the total area of the available pieces:
   * Piece A (2×1) = 2 square units
   * Piece B (2×2) = 4 square units
   * Piece C (1×1) = 1 square unit
   * Piece D (1×2) = 2 square units
   * Total area of pieces = 9 square units
   * This matches the board area, which is promising

3) Let's follow the steps and check if the pieces overlap or leave gaps:
   *** After Step 1: Piece B takes up positions (1,1)-(2,2)**
   * After Step 2: Piece C takes up position (0,1)
   * After Step 3: Piece A takes up positions (0,2)-(0,3)
   *** After Step 4: Piece D takes up positions (1,0)-(2,0)**

4) Checking for overlaps and gaps:
   * There are overlaps between some pieces
   * For example, Piece B and Piece A would overlap
   * Also, some parts of the 3×3 grid would remain unfilled

5) Therefore, following these specific placement instructions:
   * The pieces will overlap
   * Not all squares will be covered
   * The puzzle cannot be completed as instructed

\boxed{\text{No}}

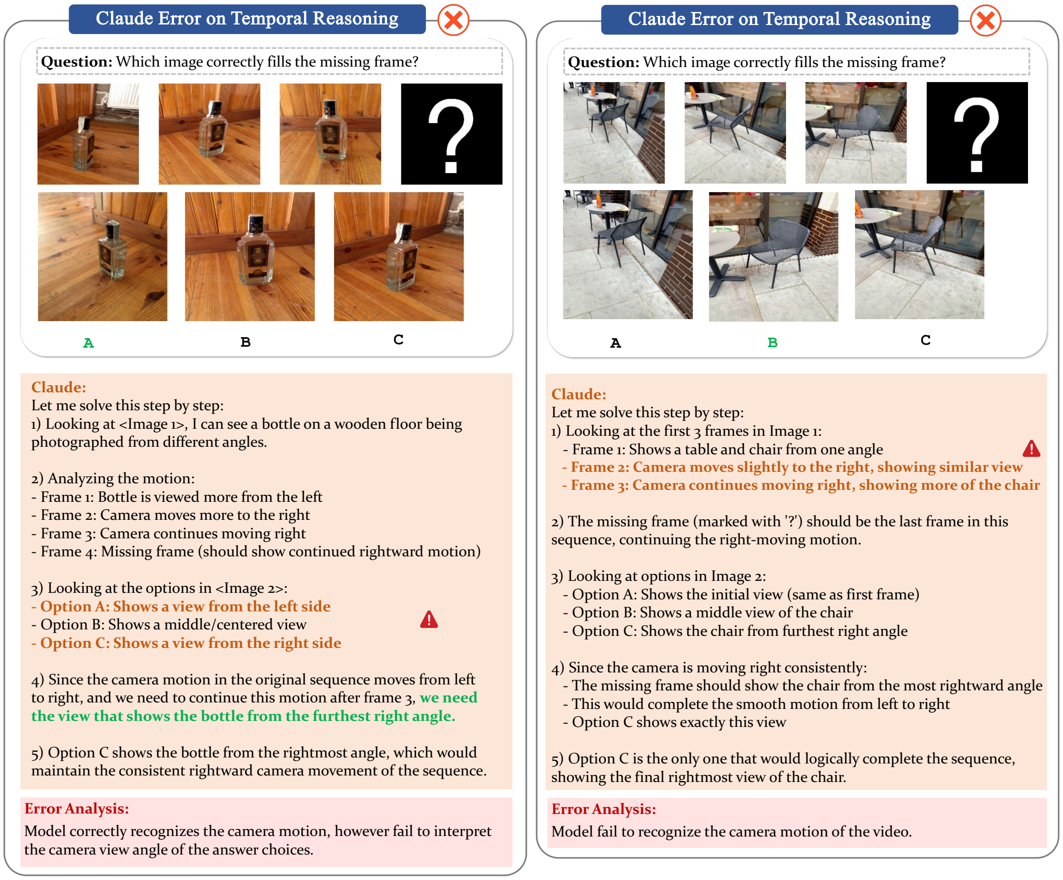
Error Analysis:

After step 1: Piece B takes up position (1,1) – (3,3)

After step 4: Piece D takes up position (1,0) – (3,1)

Figure 22: Left: A perception error case from Claude on tangram puzzle. Right: An error case from Claude on tangram puzzle, which failed to simulate the intermediate steps even when step-by-step instructions are given.

Fig. 23 presents two error cases from Claude on temporal frame reasoning. In the left example, Claude correctly inferred the camera's left-to-right movement across the given frames, yet it misevaluated the viewpoints depicted in the answer choices and consequently selected the wrong completion frame. In the right example, the model erred even earlier, misconstruing the direction of camera motion itself; this foundational misinterpretation then led to an incorrect choice despite a seemingly systematic rationale.



Claude Error on Temporal Reasoning

Question: Which image correctly fills the missing frame?

Claude:
Let me solve this step by step:
1) Looking at <Image 1>, I can see a bottle on a wooden floor being photographed from different angles.

2) Analyzing the motion:
- Frame 1: Bottle is viewed more from the left
- Frame 2: Camera moves more to the right
- Frame 3: Camera continues moving right
- Frame 4: Missing frame (should show continued rightward motion)

3) Looking at the options in <Image 2>:
- **Option A: Shows a view from the left side**
- Option B: Shows a middle/centered view
- **Option C: Shows a view from the right side**

4) Since the camera motion in the original sequence moves from left to right, and we need to continue this motion after frame 3, **we need the view that shows the bottle from the furthest right angle.**

5) Option C shows the bottle from the rightmost angle, which would maintain the consistent rightward camera movement of the sequence.

Error Analysis:
Model correctly recognizes the camera motion, however fail to interpret the camera view angle of the answer choices.

Claude Error on Temporal Reasoning

Question: Which image correctly fills the missing frame?

Claude:
Let me solve this step by step:
1) Looking at the first 3 frames in Image 1:
- Frame 1: Shows a table and chair from one angle
- **Frame 2: Camera moves slightly to the right, showing similar view**
- **Frame 3: Camera continues moving right, showing more of the chair**

2) The missing frame (marked with '?') should be the last frame in this sequence, continuing the right-moving motion.

3) Looking at options in Image 2:
- Option A: Shows the initial view (same as first frame)
- Option B: Shows a middle view of the chair
- Option C: Shows the chair from furthest right angle

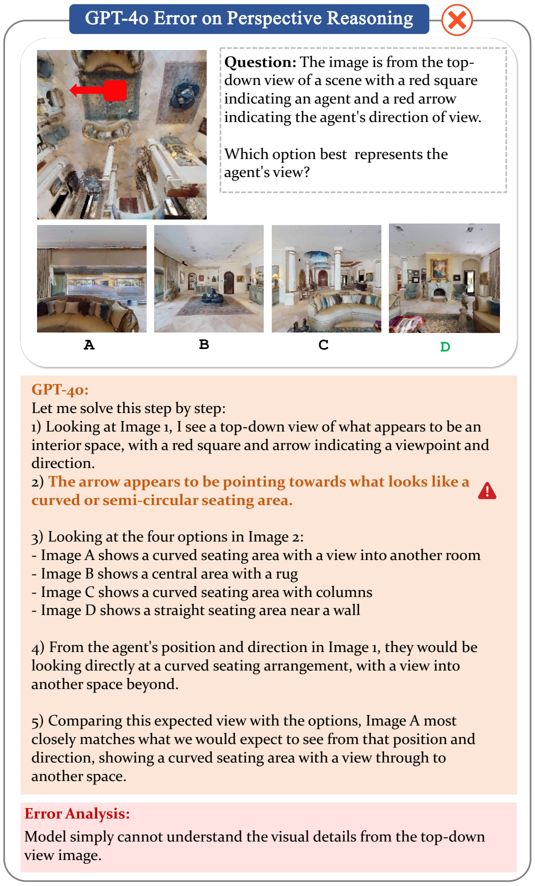
4) Since the camera is moving right consistently:
- The missing frame should show the chair from the most rightward angle
- This would complete the smooth motion from left to right
- Option C shows exactly this view

5) Option C is the only one that would logically complete the sequence, showing the final rightmost view of the chair.

Error Analysis:
Model fail to recognize the camera motion of the video.

Figure 23: Error case on temporal frame reasoning.

Fig. 24 presents an error for perspective reasoning from GPT-4o. GPT-4o misprojects the agent's top-down pose into egocentric space: it assumes the arrowed direction is pointing towards a curved seating area and therefore selects option A. A correct geometric mapping should conclude that the agent would instead stand next to the curved seating area. This error underscores the model's difficulty in reason about perspective changing in 3D space.



Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

GPT-4o:
Let me solve this step by step:
1) Looking at Image 1, I see a top-down view of what appears to be an interior space, with a red square and arrow indicating a viewpoint and direction.
2) **The arrow appears to be pointing towards what looks like a curved or semi-circular seating area.**

3) Looking at the four options in Image 2:
- Image A shows a curved seating area with a view into another room
- Image B shows a central area with a rug
- Image C shows a curved seating area with columns
- Image D shows a straight seating area near a wall

4) From the agent's position and direction in Image 1, they would be looking directly at a curved seating arrangement, with a view into another space beyond.

5) Comparing this expected view with the options, Image A most closely matches what we would expect to see from that position and direction, showing a curved seating area with a view through to another space.

Error Analysis:
Model simply cannot understand the visual details from the top-down view image.

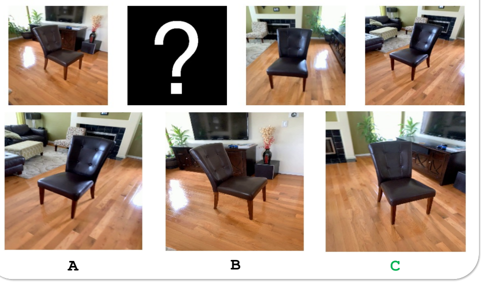
Figure 24: Error case on perspective reasoning.

Figure 25-27 presents error cases from Gemini-2.0-Flash(-Thinking), o1 and GPT-4o. on temporal frame reasoning.

Figure 30-29 presents errors for perspective reasoning from Gemini-2.0-Flash(-Thinking), o1 and Claude.

GPT-4o Error on Temporal Reasoning ⊗

Question: Which image correctly fills the missing frame?



GPT-4o:
To solve which image fits in the missing sequence, we need to observe the movement of the camera and orientation of the chair in the images from <Image 1> and match it with the options in <Image 2>.

1. Orientation and Position Analysis:
  - In the first frame of <Image 1>, the chair is on the left side of the TV stand.
  - In the third frame, the chair is further to the right but still not in front of the couch.
  - In the fourth frame, the chair has moved next to the couch, with more of the couch visible.

2. Direction of Movement:
  - Observing the transition of the chair from the first to the fourth frame, the camera appears to move from left to right.

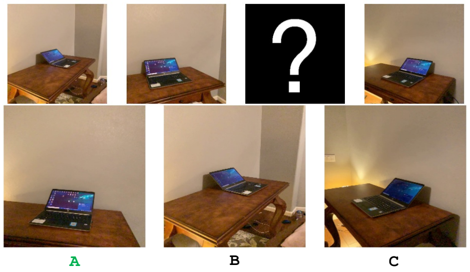
3. Identifying Correct Frame:
  - For the second frame (the missing one), the chair should be positioned between the locations seen in the first and third frames.
  - **Option B in <Image 2> shows this position. The camera is between the starting position and the subsequent move.** ⚠

Therefore, the correct image to fill in the missing frame is \(\boxed{B}\).

Error Analysis:
Model fails to identify that Option B is really the same as the first frame. And Option A is the same as the last frame.

GPT-4o Error on Temporal Reasoning ⊗

Question: Which image correctly fills the missing frame?



GPT-4o:
To solve this problem, we need to determine which image correctly fills the missing frame in the sequence given in Image 1. We have images labeled A, B, and C to choose from in Image 2.

Analysis of Image 1
1. First Frame: Shows the laptop on a desk with the wall visible to the right.
2. Second Frame: **The camera has moved to the left** as more of the wall on the right side is visible. ⚠
3. Third Frame: Missing frame.
4. Fourth Frame**: The camera seems to have moved further left, with a lamp visible to the right.** ⚠

Choices in Image 2
- A: Shows the laptop with a large portion of the right wall visible.
- B: Shows the laptop with a lot of wall visible behind it, similar to a position further left than the first frame.
- C: Shows the laptop with the lamp close to it, **which suggests it's a frame further left than all previous.** ⚠

Determine the Order and Direction:
- The movement seems to be progressing from right to left as more of the left side (the lamp) becomes visible in subsequent frames.

Comparison:
- From Right to Left:
  - 1st Frame: Most of the right wall visible.
  - 2nd Frame: Wall is more to the left, but space still visible on the right.
  - 4th Frame: Lamp visible, implying maximum left position.

- Missing Frame Analysis:
  - Given that the camera moves from right to left:
    - A: Seems to show more of the right wall (first frame position).
    - B: Fits between the first two frames—it's between mostly right wall visibility and the lamp.
    - C: Shows the lamp, indicating a further left position than the final frame.

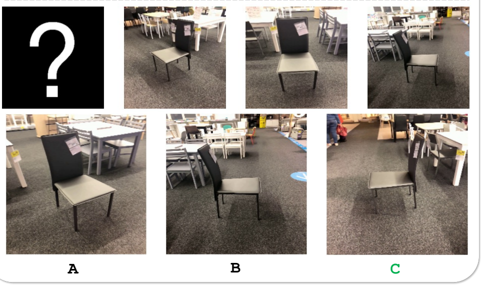
The best fit for the missing frame, which should logically be between the current second and fourth frames (moving left), is B.

Error Analysis:
Model fail to recognize the camera motion of the video, and the relative camera angle change between the answer choices and the provided frames.

Figure 25: Error cases from GPT-4o on temporal frame reasoning.

Gemini Error on Temporal Reasoning ❌

Question: Which image correctly fills the missing frame?

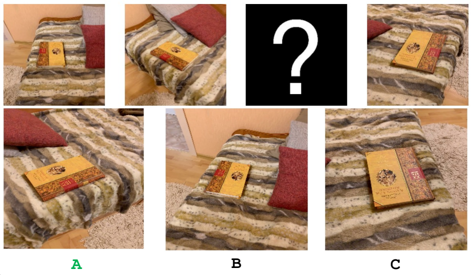


Gemini-2.0-Flash-Thinking:
Step-by-step analysis:
1. Analyze the given frames (Image 1, frames 2, 3, 4): Observe the position of the chair in each frame.
   - Frame 2: Chair is positioned slightly to the right and viewed from the right-front.
   - Frame 3: Chair is more centered and viewed more from the front.
   - Frame 4: Chair is positioned slightly to the left and viewed from the left-front.
2. Deduce camera movement direction: The change in chair position from right to center to left as we go from Frame 2 to Frame 3 to Frame 4 suggests a leftward camera movement.
3. Determine the expected view for the missing frame (Frame 1): If the camera is moving left, Frame 1, which precedes Frame 2, should be taken from a position further to the right than Frame 2. This means Frame 1 should show the chair from a more right-side perspective compared to Frame 2.
4. Examine the options (Image 2, A, B, C): Compare each option with the expected view for Frame 1.
   - **Option A: Shows the chair from a right-front perspective, similar to Frame 2, but potentially slightly more from the right.** ⚠ This is consistent with being a frame taken from a position further to the right than Frame 2 in a leftward camera movement.
   - Option B: Shows a side view of the chair. This is significantly different from the views in Frames 2, 3, and 4 and is unlikely to fit in a sequence with them resulting from a simple left or right camera movement.
   - Option C: Shows the chair from a right-front perspective, but it appears to be slightly more frontal than Option A and Frame 2. If we are moving left, we expect to progress from a more right view to a less right view. Option C being more frontal than Frame 2 makes it less likely to be a frame taken from a position further to the right than Frame 2.
5. Compare Option A with Frame 2: Option A shows the chair from a slightly more right angle than Frame 2. This is consistent with a leftward camera movement, where Frame 1 (Option A) is taken from a position to the right of Frame 2.
6. Conclusion: Option A is the most plausible choice for the missing frame as it logically precedes Frame 2 in a sequence of frames taken with a leftward moving camera.

Error Analysis:
Model correctly recognizes the camera motion, however fail to interpret the camera view angle of the answer choices.

Gemini Error on Temporal Reasoning ❌

Question: Which image correctly fills the missing frame?

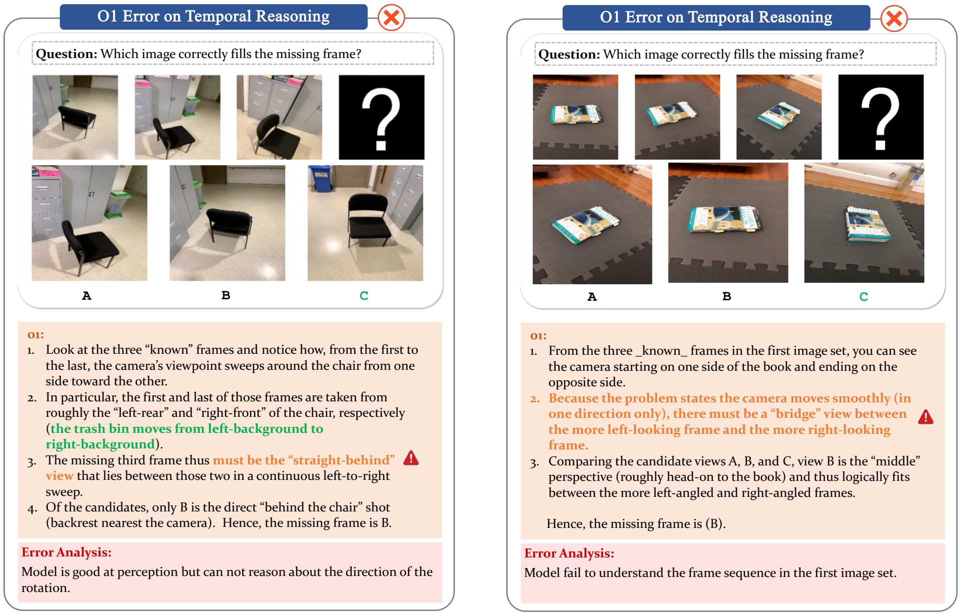


Gemini-2.0-Flash:
The given image <Image 1> has four frames, with the third frame missing and denoted by a '?'. The possible options for filling the frame are A, B and C from <Image 2>.
The frames show a bed with a book lying on it. From <Image 1>, the first frame shows the book on the bed with a red cushion next to the book.
The second frame shows the book, and part of the cushion moved upwards.
We need to find a frame which fits between the second and the fourth images.
The fourth frame from <Image 1> shows the book has moved rightwards along the bed and the cushion is visible to the left of the book.
From the second image, we can see that:
A: Shows a book on the bed and a cushion. The orientation is a bit different to the first image in the first image set.
B: Shows the book on the bed, with the cushion moved to the right, **and it appears to be showing a position similar to the missing third frame.**
C: **The book is on the bed, but it appears to be in the fourth position.** This does not correctly fill the gap. ⚠

So the missing frame should be the one showing the book slightly to the right from the second frame. Therefore, the option B fills the gap.

Error Analysis:
Model does well on perception, correctly describes the given frames, and the provided answer choices, but fail to reason which one is the missing frame.

Figure 26: Error cases from Gemini models on temporal frame reasoning.



O1 Error on Temporal Reasoning

Question: Which image correctly fills the missing frame?

o1:

1. Look at the three “known” frames and notice how, from the first to the last, the camera’s viewpoint sweeps around the chair from one side toward the other.
2. In particular, the first and last of those frames are taken from roughly the “left-rear” and “right-front” of the chair, respectively (**the trash bin moves from left-background to right-background**).
3. The missing third frame thus **must be the “straight-behind” view** that lies between those two in a continuous left-to-right sweep. ⚠
4. Of the candidates, only B is the direct “behind the chair” shot (backrest nearest the camera). Hence, the missing frame is B.

Error Analysis:

Model is good at perception but can not reason about the direction of the rotation.

O1 Error on Temporal Reasoning

Question: Which image correctly fills the missing frame?

o1:

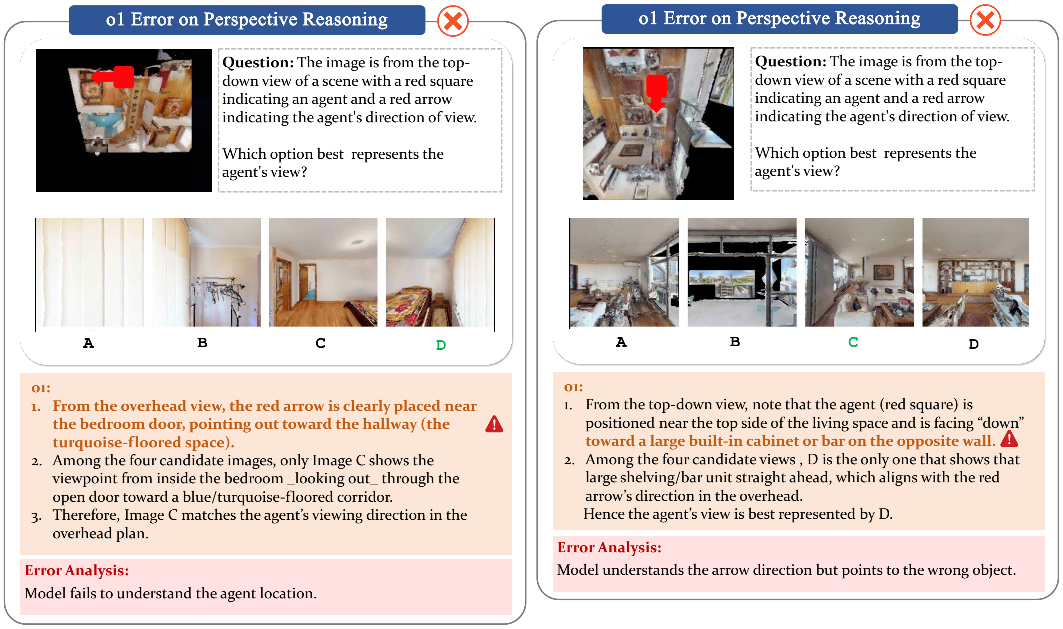
1. From the three _known_ frames in the first image set, you can see the camera starting on one side of the book and ending on the opposite side.
2. **Because the problem states the camera moves smoothly (in one direction only), there must be a “bridge” view between the more left-looking frame and the more right-looking frame.** ⚠
3. Comparing the candidate views A, B, and C, view B is the “middle” perspective (roughly head-on to the book) and thus logically fits between the more left-angled and right-angled frames.

Hence, the missing frame is (B).

Error Analysis:

Model fail to understand the frame sequence in the first image set.

Figure 27: Error cases from o1 on temporal frame reasoning.



o1 Error on Perspective Reasoning

Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

o1:

1. **From the overhead view, the red arrow is clearly placed near the bedroom door, pointing out toward the hallway (the turquoise-floored space).** ⚠
2. Among the four candidate images, only Image C shows the viewpoint from inside the bedroom _looking out_ through the open door toward a blue/turquoise-floored corridor.
3. Therefore, Image C matches the agent’s viewing direction in the overhead plan.

Error Analysis:

Model fails to understand the agent location.

o1 Error on Perspective Reasoning

Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

o1:

1. From the top-down view, note that the agent (red square) is positioned near the top side of the living space and is facing “down” **toward a large built-in cabinet or bar on the opposite wall.** ⚠
2. Among the four candidate views , D is the only one that shows that large shelving/bar unit straight ahead, which aligns with the red arrow’s direction in the overhead.

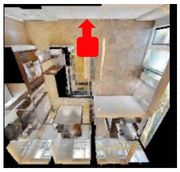
Hence the agent’s view is best represented by D.

Error Analysis:

Model understands the arrow direction but points to the wrong object.

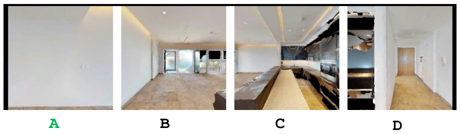
Figure 28: Error case from o1 on perspective reasoning.

o1 Error on Perspective Reasoning



Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

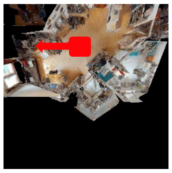


o1:

1. In the top-down view, the red square (agent) is near the upper hallway.
2. The red arrow indicates the agent is facing “up” along that hallway.
3. **Among the four candidate photos (A, B, C, D), only option D shows a hallway view with a door straight ahead, matching the agent's position and orientation in the top-down image.** ⚠

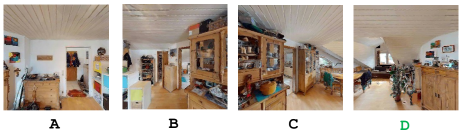
Error Analysis:
Model understand the agent location but hallucinates about the objects.

Claude Error on Perspective Reasoning



Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?



Claude:
Let me solve this step by step:

1) First, let's understand what we see in Image 1:
- It's a top-down view of what appears to be a room or space
- There's a red square (representing an agent's position)
- There's a red arrow showing which direction the agent is looking

2) The arrow in Image 1 is pointing roughly towards what appears to be a longer wall/section of the room.

3) Looking at the four images (A, B, C, D) in Image 2:
- All images show what appears to be an attic or room with wooden furniture and storage
- They all have white/light colored sloped ceilings
- They're taken from different angles/positions

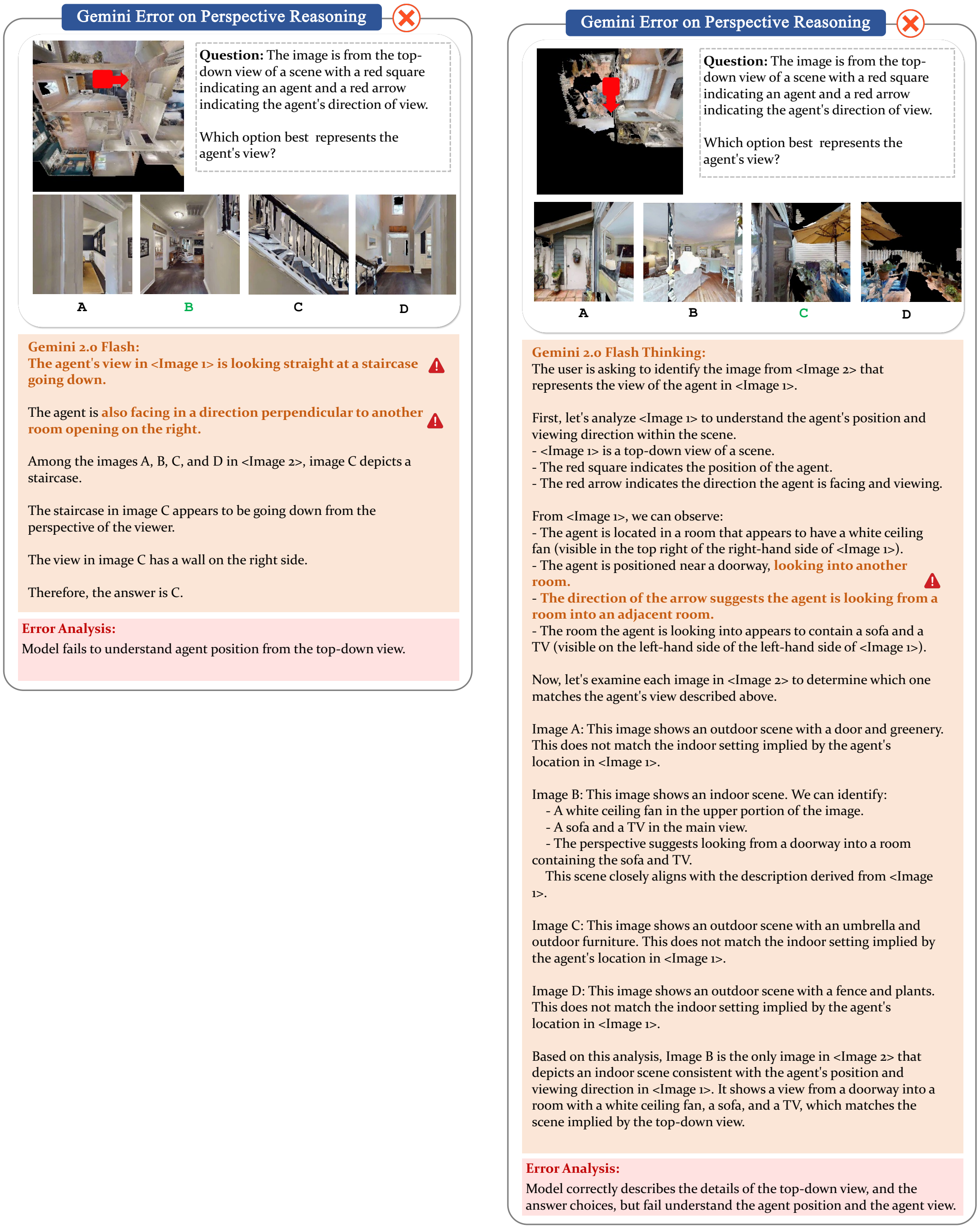
4) Analyzing the perspective of each image:
- Image A shows a view down a space with furniture on the left
- Image B shows more cluttered storage area
- Image C shows a similar view but from opposite direction
- Image D shows what appears to be the end of the room

5) Comparing the direction of view in Image 1 with these perspectives:
- **The red arrow's direction in Image 1 aligns with the perspective shown in Image A** ⚠
- **Image A shows a straight view down the room, which matches the arrow's direction**

Therefore, the image that best represents the agent's view is Image A.

Error Analysis:
Model correctly describes the details of the top-down view, the agent position, and the answer choices, but fail to reason (or does not reason) what the first-person view of the agent looks like

Figure 29: Error case from o1 and Claude on perspective reasoning.



Gemini Error on Perspective Reasoning

Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

Gemini 2.0 Flash:
The agent's view in <Image 1> is looking straight at a staircase going down.

The agent is **also facing in a direction perpendicular to another room opening on the right.**

Among the images A, B, C, and D in <Image 2>, image C depicts a staircase.

The staircase in image C appears to be going down from the perspective of the viewer.

The view in image C has a wall on the right side.

Therefore, the answer is C.

Error Analysis:
Model fails to understand agent position from the top-down view.

Gemini Error on Perspective Reasoning

Question: The image is from the top-down view of a scene with a red square indicating an agent and a red arrow indicating the agent's direction of view.

Which option best represents the agent's view?

Gemini 2.0 Flash Thinking:
The user is asking to identify the image from <Image 2> that represents the view of the agent in <Image 1>.

First, let's analyze <Image 1> to understand the agent's position and viewing direction within the scene.
- <Image 1> is a top-down view of a scene.
- The red square indicates the position of the agent.
- The red arrow indicates the direction the agent is facing and viewing.

From <Image 1>, we can observe:
- The agent is located in a room that appears to have a white ceiling fan (visible in the top right of the right-hand side of <Image 1>).
- The agent is positioned near a doorway, **looking into another room.**
- **The direction of the arrow suggests the agent is looking from a room into an adjacent room.**
- The room the agent is looking into appears to contain a sofa and a TV (visible on the left-hand side of the left-hand side of <Image 1>).

Now, let's examine each image in <Image 2> to determine which one matches the agent's view described above.

Image A: This image shows an outdoor scene with a door and greenery. This does not match the indoor setting implied by the agent's location in <Image 1>.

Image B: This image shows an indoor scene. We can identify:
- A white ceiling fan in the upper portion of the image.
- A sofa and a TV in the main view.
- The perspective suggests looking from a doorway into a room containing the sofa and TV.

This scene closely aligns with the description derived from <Image 1>.

Image C: This image shows an outdoor scene with an umbrella and outdoor furniture. This does not match the indoor setting implied by the agent's location in <Image 1>.

Image D: This image shows an outdoor scene with a fence and plants. This does not match the indoor setting implied by the agent's location in <Image 1>.

Based on this analysis, Image B is the only image in <Image 2> that depicts an indoor scene consistent with the agent's position and viewing direction in <Image 1>. It shows a view from a doorway into a room with a white ceiling fan, a sofa, and a TV, which matches the scene implied by the top-down view.

Error Analysis:
Model correctly describes the details of the top-down view, and the answer choices, but fail understand the agent position and the agent view.

Figure 30: Error case from Gemini models on perspective reasoning.

H COMPLETE ANALYSIS RESULTS ON OTHER MODELS

Model Performance on 2D/3D Individual Transformation Types. Tab. 15 presents model accuracy across 2D visual analogy and text instruction tasks. Across the nine subtasks, adding visual simulation lifted accuracy for every model except in a few narrow cases, and the size of the gain correlates strongly with baseline capability. Closed-source leaders that were already solid on the raw pixel tasks—o1 ($\sim$ +3 points overall) and GPT-4o ($\sim$ +8 points)—were pushed into the mid-80 s and low-90 s, effectively reaching ceiling on the text-instruction variants, where gains were biggest (e.g., GPT-4o jumps +25 points on "Reflection" and +18 points on both "Rotation" and "Translation"). Mid-tier proprietary models such as Gemini 2.0 Flash ($\sim$ +5 points) and its "Flash Thinking" mode ($\sim$ +5.5 points) benefited even more on instructions than on analogies, narrowing the gap to

GPT-4-class systems. Open-source vision-language models lag a full generation behind—the best of them (InternVL 2.5-78B) still sits below 55% on average after simulation—but they, too, record healthy boosts of 6–12 points, chiefly on the analogy side. The lone regression is GPT-4o's –5 pt dip on "Reflection" analogies, suggesting that simulation may occasionally overwrite a correct latent heuristic. Overall, the pattern indicates that visual simulation chiefly helps models convert verbal transformation instructions into precise spatial operations, while stronger base perception/reasoning models harvest the largest absolute improvements and approach human-like proficiency.

Tab. 16 presents model accuracy across 3D visual analogy and text instruction tasks. Visual simulation gives 3D spatial reasoning a measurable—but more uneven—boost than in 2D: averaged over all eight subtasks, every proprietary model gains between $\sim$+2 points (GPT-4o, o1) and +6 points (Claude-3.5 Sonnet, Gemini-Flash Thinking), while the open-source field improves by $\sim$ +4–7 points—except InternVL, which slips a point. Gains concentrate in the conceptually harder operations: across models, Shearing (both analogy +6.6 points and instruction +6.6 points) and Rotation-instruction (+6.4 points) see the largest lifts, whereas Translation under visual analogy actually falls slightly (–0.9 points), echoing a smaller 2D reflection dip. Even after simulation, closed-source leaders plateau in the high-60s to mid-70s on most 3D subtasks—roughly 15 points below their 2D ceilings—indicating that depth-aware transformations remain a major bottleneck. Open-source VL models still trail a full generation ($\leq$45% average), but their sharper relative gains suggest they, too, leverage synthetic roll-outs to bridge language and geometry.

Model	**2D Transformations w/ Visual Analogy**					**2D Transformations w/ Text Instruction**			
	Reflection	Rotation	Shearing	Scaling	Translation	Reflection	Rotation	Scaling	Translation
Without Visual Simulation									
GPT-4o	82.1	69.8	53.7	88.5	72.0	65.8	67.8	90.6	73.3
Claude-3.5 Sonnet	75.0	60.9	55.8	87.4	71.2	63.8	58.9	85.9	66.5
Gemini2.0 Flash	85.7	63.8	51.0	84.4	71.4	65.8	62.3	88.4	70.3
Gemini2.0 Flash Thinking	52.4	48.9	46.9	71.9	55.1	63.2	60.6	83.0	67.8
o1	92.9	70.7	59.2	83.3	84.0	89.5	78.1	92.2	92.2
LLaVA-OneVision	7.1	25.9	32.7	24.4	25.4	31.7	33.1	51.0	34.6
Qwen2.5-VL-72B	57.1	38.8	34.7	64.4	42.3	29.3	49.6	62.5	38.8
InternVL2.5-78B	35.7	41.4	34.7	45.6	36.6	41.5	51.1	75.0	51.9
With Visual Simulation									
GPT-4o	76.9	72.8	54.8	91.9	80.0	91.2	86.0	93.2	91.5
Claude-3.5 Sonnet	73.1	70.9	50.0	85.5	73.9	55.9	72.9	83.8	73.9
Gemini2.0 Flash	73.1	70.9	59.5	85.5	74.5	79.4	74.8	90.5	78.2
Gemini2.0 Flash Thinking	61.5	68.2	40.5	71.0	56.4	70.6	68.2	89.2	73.9
o1	80.8	80.6	54.8	87.1	84.2	100	93.5	94.6	97.6
LLaVA-OneVision	15.4	30.1	31.0	30.6	24.8	20.6	41.1	48.6	33.9
Qwen2.5-VL-72B	65.4	56.3	35.7	71.0	57.0	41.2	40.2	60.8	39.4
InternVL2.5-78B	69.2	43.7	33.3	59.7	47.3	50.0	53.3	73.0	53.9

Table 15: Model Performance With or Without Visual Simulation across 2D Transformation types in Visual Analogy and Text Instruction Tasks.

Model	3D Transformations w/ Visual Analogy				3D Transformations w/ Text Instruction			
	Rotation	Shearing	Scaling	Translation	Rotation	Shearing	Scaling	Translation
Without Visual Simulation								
GPT-4o	60.7	55.7	76.0	80.1	60.1	46.9	71.1	71.2
Claude-3.5 Sonnet	50.0	46.2	63.3	62.6	45.9	40.4	55.6	53.4
Gemini2.0 Flash	54.2	53.9	63.3	73.0	55.86	44.44	61.90	51.63
Gemini2.0 Flash Thinking	42.4	43.6	61.5	63.8	37.8	32.7	52.5	55.7
o1	65.6	58.1	76.7	85.6	61.3	46.3	70.5	73.9
LLaVA-OneVision	18.8	29.1	28.9	25.3	27.0	19.4	41.0	30.7
Qwen2.5-VL-72B	36.5	40.2	61.1	46.6	36.9	33.3	47.6	45.1
InternVL2.5-78B	31.2	30.8	51.1	37.4	37.8	32.4	60.0	40.5
With Visual Simulation								
GPT-4o	64.3	64.3	78.2	76.0	62.6	54.7	75.3	68.5
Claude-3.5 Sonnet	51.2	59.5	69.2	59.7	55.6	48.0	64.5	59.3
Gemini2.0 Flash	46.4	64.3	62.8	68.2	60.9	49.5	64.9	56.1
Gemini2.0 Flash Thinking	50.0	47.6	60.3	66.7	48.5	46.7	59.1	59.3
o1	63.1	63.1	76.9	79.8	69.7	50.7	79.6	74.1
LLaVA-OneVision	27.4	28.6	32.1	29.5	27.3	26.7	45.2	35.2
Qwen2.5-VL-72B	46.4	54.8	69.2	55.0	45.5	34.7	48.4	46.3
InternVL2.5-78B	31.0	28.6	43.6	32.6	43.4	37.3	57.0	40.7

Table 16: Model Performance With or Without Visual Simulation across 3D Transformation types in Visual Analogy and Text Instruction Tasks.

Task complexity vs. performance. Tab. 18 presents model performance across different task difficulties for 2D and 3D transformations. Adding visual simulation helps most when tasks get tougher, but the effect differs by setting. For 2D text instructions tasks, we observe big boost – closed-source models jump about 10-20 points on medium and hard tasks, often hitting 90%+. For 2D visual analogy tasks, we observe smaller lift—several points on easy, up to 1̃0 points on medium/hard. For 3D tasks, only a few-point gain, and some models slip on the hardest visual analogy tasks, showing 3D reasoning is still hard. Open-source MLLMs stay well behind; their scores move up and down unpredictably, meaning they haven't yet learned to use the simulated views well.

Tab. 17 presents model performance across different number of transformation steps for 2D and 3D transformations. Models struggle more as the number of transformation steps grows, and visual simulation mainly fixes that. Without simulation, accuracy often peaks at one or two steps and drops at three—especially in 3D visual-analogy, where GPT-4o falls from 73% (N = 2) to 49% (N = 3). When simulation is added, scores for the multi-step cases (N = 2–3) jump 10–15 points for the top proprietary systems and a few points for open-source ones, erasing most of the earlier decline in 2D tasks and cutting the 3D drop roughly in half. Single-step problems were already easy for the best models and see little change. Overall, simulation is most useful for longer, instruction-driven chains of transforms, while depth-heavy 3D sequences remain the hardest setting.

Model	**2D Visual Analogy**			**2D Text Instruction**			**3D Visual Analogy**			**3D Text Instruction**		
	N=1	N=2	N=3	N=1	N=2	N=3	N=1	N=2	N=3	N=1	N=2	N=3
Without Visual Simulation												
GPT-4o	60.46	74.84	73.86	67.27	77.56	73.55	62.75	73.37	48.69	63.07	63.40	60.78
Claude-3.5 Sonnet	63.73	75.82	65.69	65.17	65.02	60.61	45.10	57.35	57.35	50.98	55.23	45.75
Gemini2.0 Flash	64.71	73.53	68.53	63.96	76.24	70.25	61.76	60.78	63.73	46.08	56.86	56.86
Gemini2.0 Flash Thinking	54.58	52.94	55.56	61.71	67.33	71.07	47.71	53.92	57.19	45.59	50.00	20.59
o1	66.7	81.4	82.4	82.0	89.1	89.3	66.67	72.55	77.45	61.76	66.67	62.75
LLaVA-OneVision	30.39	26.47	24.51	49.57	33.70	31.53	25.49	28.43	22.55	30.39	30.39	24.51
InternVL2.5-78B	43.14	34.31	42.16	61.74	52.17	50.45	40.2	29.41	36.27	34.31	48.04	40.2
Qwen2.5-VL-72B	50.00	45.10	41.18	55.65	36.96	40.54	48.04	42.16	45.10	38.24	43.14	41.18
With Visual Simulation												
GPT-4o	-	78.43	73.53	-	88.04	90.57	-	70.59	72.55	-	61.76	68.63
Claude-3.5 Sonnet	-	70.59	70.59	-	71.74	72.64	-	56.86	57.84	-	65.69	50.98
Gemini2.0 Flash	-	69.6	73.5	-	80.43	77.40	-	61.76	59.80	-	61.76	53.92
Gemini2.0 Flash Thinking	-	46.08	58.82	-	79.35	67.92	-	55.88	60.78	-	53.92	53.92
o1	-	73.4	85.3	-	94.6	97.2	-	70.6	75.5	-	70.6	69.6
LLaVA-OneVision	-	30.39	25.49	-	38.04	34.91	-	28.43	28.43	-	36.27	29.41
InternVL2.5-78B	-	39.22	51.96	-	56.52	52.83	-	25.49	35.29	-	46.08	39.22
Qwen2.5-VL-72B	-	51.96	58.82	-	43.48	41.51	-	49.02	58.82	-	47.06	43.14

Table 17: Model Performance With or Without Visual Simulation across number of transformation steps (N) in 2D/3D Visual Analogy and Text Instruction Tasks.

Model	**2D Visual Analogy**			**2D Text Instruction**			**3D Visual Analogy**			**3D Text Instruction**		
	easy	medium	hard	easy	medium	hard	easy	medium	hard	easy	medium	hard
Without Visual Simulation												
GPT-4o	80.4	67.3	61.4	76.2	70.4	71.3	74.2	64.9	65.7	69.0	63.1	55.2
Claude-3.5 Sonnet	76.5	66.7	62.1	68.7	61.8	59.4	54.9	54.4	50.5	55.6	52.0	44.4
Gemini 2.0 Flash	78.4	63.7	64.7	75.0	67.2	67.3	67.6	59.8	58.8	56.9	52.9	50.0
Gemini 2.0 Flash Think	66.3	52.3	44.4	65.5	69.4	65.4	54.6	53.9	50.3	48.0	44.7	46.1
o1	83.3	77.5	69.6	90.6	81.1	89.1	78.4	70.6	67.7	69.6	64.7	56.9
LLaVA-OneVision	22.6	32.4	26.5	39.5	46.3	29.2	25.5	20.6	30.4	31.4	28.4	25.5
InternVL 2.5-78B	45.1	40.2	34.3	63.2	50.9	50.0	32.4	34.3	39.2	48.0	37.3	37.3
Qwen 2.5-VL-72B	57.8	40.2	38.2	50.9	41.7	41.7	55.9	40.2	39.2	42.2	38.2	42.2
With Visual Simulation												
GPT-4o	80.9	79.4	67.7	91.6	89.4	86.9	80.9	75.0	58.8	75.0	64.7	55.9
Claude-3.5 Sonnet	76.5	72.1	63.2	78.9	65.2	72.1	67.7	52.9	51.5	66.2	57.4	51.5
Gemini 2.0 Flash	79.4	72.1	63.2	81.7	86.4	67.2	64.7	58.8	58.8	57.4	55.9	60.3
Gemini 2.0 Flash Think	54.4	55.9	47.1	76.1	74.2	68.9	72.1	54.4	48.5	63.2	48.5	50.0
o1	80.9	82.4	75.0	94.4	98.5	95.1	85.3	69.1	64.7	73.5	75.0	61.8
LLaVA-OneVision	36.8	19.1	27.9	39.4	34.9	34.4	26.5	20.6	38.2	45.6	25.0	27.9
InternVL 2.5-78B	57.4	44.1	35.3	64.8	48.5	49.2	23.5	27.9	39.7	55.9	27.9	44.1
Qwen 2.5-VL-72B	72.1	50.0	44.1	59.2	30.3	36.1	63.2	50.0	48.5	47.1	44.1	44.1

Table 18: Model Performance With or Without Visual Simulation across different difficulty levels in 2D/3D Visual Analogy and Text Instruction Tasks.

2D and 3D Perception Probing with Cube Nets. Tab. 19 presents model performance on 2D and 3D perception probing questions about cube nets, in comparison to the success rate on cube net folding task. The results show that success on cube-net folding is driven by a model's 3D perception, not its 2D eyesight. All closed-source systems (and several open-source ones) already read colors and 2D face connectivity at or near ceiling, yet their cube-net scores diverge sharply. When we compare cube accuracy (✗VSim column) with each perceptual measure, the strongest linear relationship is with the 3D "Folded?" test (Pearson $r \approx 0.89$), while 2D connectivity ($r \approx 0.68$) and color ($r \approx 0.72$) are weaker. Gemini Flash illustrates the pattern: it pairs the top 3D perception score (69%) with the best cube-net performance (65%), whereas GPT-4o and InternVL match its

2D vision but lag 10-20 points on both 3D perception and cube folding. In short, being able to judge how faces come together in depth—rather than recognizing colors or flat adjacencies—largely determines how well a model can reason about folded cubes.

Model	**2D Perception**		**3D Perception**	**Cube Net Performance**	
	Color	Connectivity	Folded?	✗VSim	✓Vsim
Random	25.0	50.0	50.0	50.5	50.5
		Closed-source Models			
GPT-4o	**100.0**	**94.1**	57.4	52.5	49.1
Gemini-2.0-Flash	**100.0**	84.9	68.8	65.0	**65.5**
Gemini-2.0-Flash-Thinking	99.0	49.4	54.3	39.8	62.8
		Open-source Models			
LLaVA-OneVision	88.0	10.0	22.0	28.5	34.2
InternVL 2.5-78B	92.0	86.0	40.2	43.5	41.0
Qwen 2.5-VL-72B	96.0	81.7	42.1	35.2	53.4

Table 19: 2D and 3D perception performance in cube net folding.

Question-only vs. Question+Steps As shown in Tab. 20, adding explicit reasoning steps ("Q + Steps") has opposite effects on cube-net tasks for the two model groups: open-source models gain, while closed-source ones do not. The three open-source VL models jump a mean + 20 points on cube nets (driven by LLaVA's + 40 pts), whereas the five proprietary models average a small decline (-1 pt, with mixed signs). On tangram puzzles, however, the pattern converges: every model—open or closed—drops sharply once reasoning steps are included, with average losses of about -24 pts for closed-source and -19 pts for open-source models. Again, the trivial solution on tangram puzzles would be comparing the total areas of all available pieces and the grid area, which can easily lead to 75% performance. This result suggest that the models cannot leverage explicit text reasoning steps.

Model	**Cube Nets**			**Tangram Puzzles**		
	Q-only	Q+Steps	Δ	Q-only	Q+Steps	Δ
			Closed-source Models			
GPT-4o	50.2	50.4	**+0.2**	62.4	34.7	**-27.7**
Claude-3.5 Sonnet	51.5	46.4	**-5.1**	71.1	41.9	**-29.2**
Gemini-2.0 Flash	47.4	51.5	**+4.1**	72.8	59.8	**-13.0**
Gemini-2.0 Flash Thinking	47.2	49.6	**+2.4**	42.9	35.3	**-7.6**
o1	56.0	47.0	**-7.0**	73.5	29.6	**-43.9**
			Open-source Models			
LLaVA-OneVision	0.0	40.5	**+40.5**	30.3	14.6	**-15.7**
InternVL 2.5-78B	33.2	41.4	**+8.2**	69.5	51.7	**-17.8**
Qwen 2.5-VL-72B	29.0	41.6	**+12.6**	72.3	47.7	**-24.6**

Table 20: Model performance on question-only prompts versus prompts that include explicit reasoning steps (Q+Steps). Δ values are Q+Steps performance - Q-only performance.

Intermediate Visual Simulation States vs. Performance

Tab. 21 summarizes extended results on varying the slice of intermediate visual simulation presented to the model across different tasks. Across models, which slice of the simulation you show matters, and the "best slice" shifts with task type. For 2D transformations, most closed-source models and the stronger open-source one (InternVL) peak when they see only the last intermediate state, gaining 2–6 points over the full roll-out; showing every intermediate frame ("all") often drags accuracy down a few points. For 3D transformations, the pattern flips—accuracy is usually highest with "all" states ($\approx$ +2–4 points over "partial"), while the last-only view tends to erase that gain, especially for GPT-4o, Gemini Flash, and o1. For cube nets, no single view helps every model. Scores barely change with "all" frames, and last-only often hurts closed-source models (-8 points on average) yet uniquely rescues LLaVA (+11 points). For Tangram puzzles, seeing "all" steps is consistently best: every model but LLaVA jumps 7–24 points versus the partial view, whereas last-only falls back to—or below—the partial baseline. Overall, for more complex tasks, models struggle to leverage intermediate visual states effectively.

Model	**2D Transformation**			**3D Transformation**			**Cube Nets**			**Tangram Puzzles**		
	Partial	All	Last	Partial	All	Last	Partial	All	Last	Partial	All	Last
Closed-source Models												
GPT-4o	86.8	82.8	89.4	72.1	68.4	68.4	51.3	52.2	35.2	43.5	51.5	43.4
Claude-3.5-Sonnet	67.8	71.4	70.7	54.9	57.8	55.9	58.7	51.6	46.8	43.5	67.6	43.3
Gemini-2.0-Flash	75.4	75.2	79.3	61.0	59.3	57.8	40.5	35.6	41.5	63.8	65.5	58.2
o1	89.3	87.7	93.4	70.1	71.6	65.2	54.4	53.4	45.4	34.8	53.2	46.0
Open-source Models												
LLaVA-OneVision	28.3	32.2	31.8	25.5	30.6	29.4	40.2	34.2	45.6	44.9	40.2	39.8
InternVL 2.5-78B	48.3	54.5	56.6	32.3	36.5	40.2	34.7	37.3	37.8	54.3	48.2	41.8
Qwen 2.5-VL-72B	44.4	48.5	44.4	48.7	49.1	43.6	41.9	53.4	42.3	49.0	56.7	44.3

Table 21: Model performance with partial, all, and last intermediate visual simulations.