RoTA-Bench: Evaluating 3D Spatial Reasoning through Interactive Projection Alignment

Anonymous authors
Paper under double-blind review

Abstract

Can vision-language models turn 2D observations into 3D understanding that supports precise action? This requires inferring object geometry, anticipating the effects of rotation, and refining actions through visual feedback. We introduce RoTA-Bench to evaluate these abilities through interactive 3D-to-2D projection alignment. Given a target silhouette, a model observes a rotated object and chooses successive rotations to recover the target projection within a limited action budget. Our automated pipeline constructs silhouette-preserving 3D objects from 2D icons and uses reversible initialization to guarantee solvability, yielding 674 tasks and a stratified 100-task test set. Empirically, we find that (i) a substantial human-model gap persists: GPT-5.6-Sol succeeds on only 17% of tasks, whereas 20 undergraduate participants collectively complete the test set four times with a 100% success rate; (ii) image-space and pose-space progress can diverge: at matched states, the model more often improves silhouette overlap while worsening pose alignment than humans (19.4% vs. 10.9%); and (iii) fine-grained control is sensitive to interaction history: removing it leaves 96.8% of the model's actions at a coarse 30-degree step size and sharply reduces observed task success. These findings highlight the challenge of converting visual evidence into precise 3D control, beyond recognizing or matching individual images. See the online demo and source code.

1 Introduction

Psychological and cognitive-science research establishes spatial perception as a foundational domain of cognitive ability (Carroll, 1993), essential for maintaining object constancy across viewpoints (Shepard & Metzler, 1971) and recovering three-dimensional geometry from two-dimensional projections (Richards et al., 1987) through the mental representation, manipulation, and inference of shape and orientation. As vision-language models are increasingly applied in visual question answering, interactive agents, and embodied systems (Chen et al., 2024; Wang et al., 2026), agents in interactive settings such as embodied exploration and geometry-aware navigation must translate spatial understanding into physical actions (Zhu et al., 2025). However, current models continue to face challenges in reasoning about three-dimensional positions, orientations, and spatial relationships (Ma et al., 2025). Therefore, addressing these challenges requires evaluation tools that assess whether models can turn 2D observations into 3D understanding that supports precise action.

However, current visual and spatial benchmarks fall short of this requirement. Static benchmarks evaluate spatial reasoning through passive question answering on fixed images (Johnson et al., 2017; Hudson & Manning, 2019; Yue et al., 2024; Lu et al., 2024), lacking action execution and closed-loop feedback entirely. Conversely, existing interactive benchmarks spanning software interfaces, visual mazes, and sliding puzzles (Zhou et al., 2024; Xie et al., 2024; Rodriguez Salgado, 2026; Mayer et al., 2025) focus on high-level task planning, tool invocation, or spatial pathfinding in complex environments. In these settings, task success largely depends on coding ability, reasoning, and planning, rather than perceiving three-dimensional orientation from visual cues and translating it into action.

To bridge this gap, inspired by classical insights on inferring three-dimensional shape from two-dimensional projections and silhouettes (Richards et al., 1987), we ask whether vision-language models can turn 2D observations into 3D understanding that supports precise action. Therefore, we

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

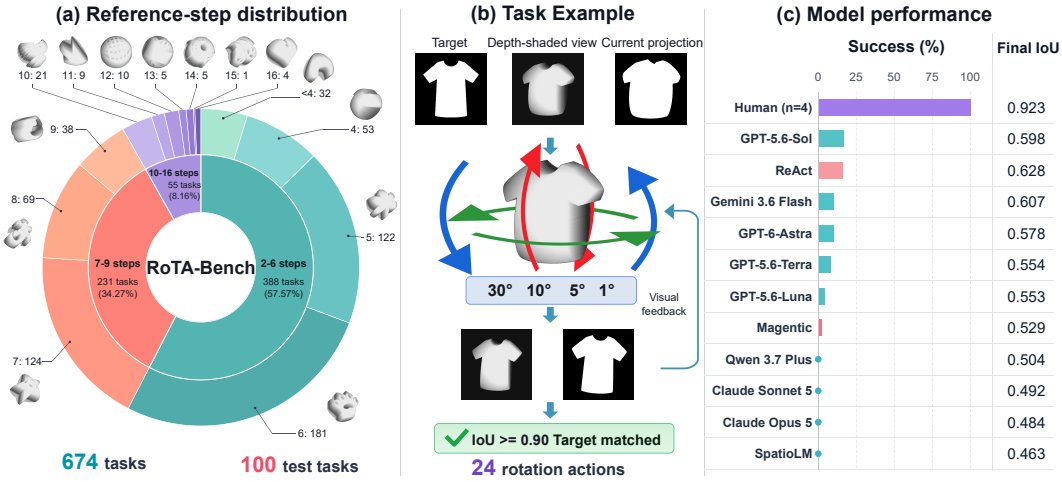


Figure 1: RoTA-Bench overview. The benchmark contains 674 canonical tasks, with 100 test tasks selected by stratified sampling over reference path lengths. In the T-shirt example, the solver chooses discrete rotations from visual observations and receives updated views until projection IoU reaches 0.90 or another stopping rule applies. The right panel shows task success rates and final IoU on the test set; full results are reported in Table 1.

introduce RoTA-Bench, an automatically constructed benchmark of interactive 3D-to-2D projection alignment. Our pipeline constructs three-dimensional objects from two-dimensional icons, generates initial orientations through reversible rotations, and guarantees solvability automatically. Given a target silhouette and visual observations of the rotated object, the model selects a rotation from 24 discrete actions at each step, receives an updated observation at the next step, and iteratively refines alignment within a limited action budget. The benchmark contains 674 tasks, with a stratified 100-task test set.

Through evaluations of nine vision-language models, two agent configurations, and 20 undergraduate participants, each completing a consecutive 20-task block from the 100-task test set, we find: First, a substantial human–model gap persists: GPT-5.6-Sol succeeds on only 17% of tasks, whereas all 400 human task runs succeed. Second, image-space and pose-space progress can diverge: at matched states, GPT-5.6-Sol more often improves silhouette overlap while worsening pose alignment than humans (19.4% vs. 10.9%). Third, fine-grained control is sensitive to interaction history: removing it leaves 96.8% of the model's actions at a coarse 30° step size and sharply reduces observed task success.

2 Related Work

2.1 Multimodal Large Language Models and Spatial Reasoning

Early vision-language models established contrastive image–text alignment (Radford et al., 2021), while subsequent architectures bridged visual encoders with language models for open-ended generation (Alayrac et al., 2022; Li et al., 2023) and visual instruction following (Liu et al., 2023), scaled further by frontier models for advanced visual reasoning (OpenAI, 2026a; Anthropic, 2026b;a; DeepMind, 2026; Qwen Team, 2026). To capture 3D geometry, specialized architectures incorporate explicit 3D representations (Hong et al., 2023; Xu et al., 2024), whereas spatial VLMs infuse coordinate reference frames (Song et al., 2025), metric relations (Chen et al., 2024), 3D object orientations (Ma et al., 2025), and physical spatial intelligence (Wu et al., 2026). Embodied and vision-language-action architectures further connect visual reasoning to physical planning (Huang et al., 2023; Rivera et al., 2025) and robotic execution (Driess et al., 2023; Zhao et al., 2025). Nonetheless, current multimodal models struggle with camera-pose estimation (Deng et al., 2026), orthographic projection (Niu et al., 2026), and shape recognition (Eppel et al., 2026). Unlike explicit 3D approaches, we examine whether models can turn 2D observations into 3D understanding that supports precise action.

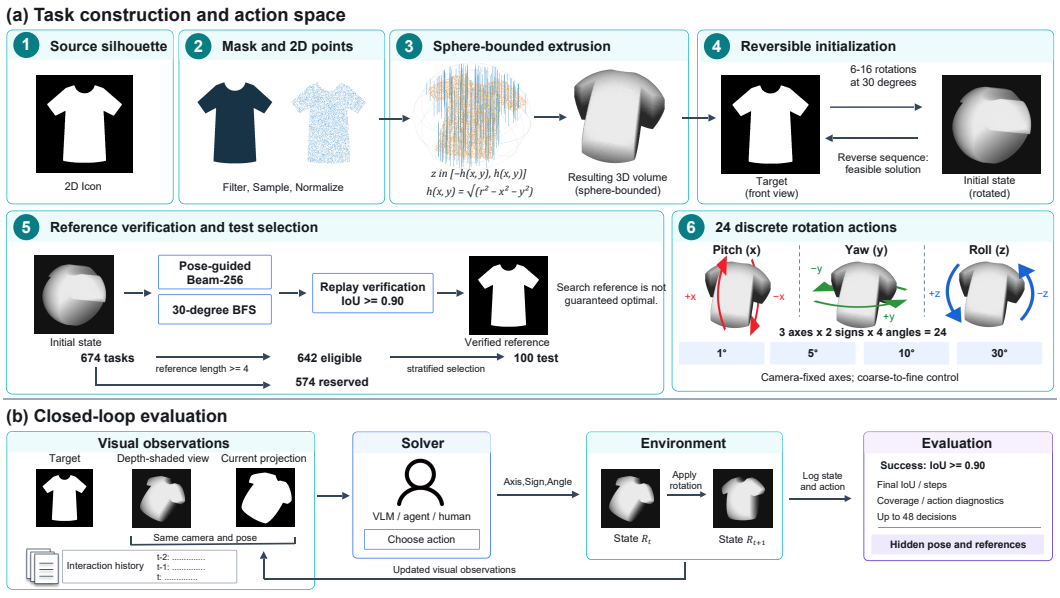


Figure 2: RoTA-Bench construction and evaluation pipeline. **(a)** **(1)** Source 2D icons are **(2)** filtered, sampled, and normalized, then **(3)** converted into 3D objects through sphere-bounded extrusion. **(4)** Reversible initialization generates an initial pose with a feasible solution. **(5)** Pose-guided beam search and 30°-restricted breadth-first search provide verified reference paths, which are used to select 100 test tasks from 674 canonical tasks by stratified sampling over reference path lengths. **(6)** The action space contains 24 discrete rotations across three axes, two directions, and four angles. **(b)** In closed-loop evaluation, the solver selects rotations from visual observations and interaction history, then receives updated observations. Success requires projection IoU ≥ 0.90.

2.2 Static Benchmarks for Visual and Spatial Reasoning

Static benchmarks evaluate visual and spatial capabilities on fixed inputs. Foundational suites probe spatial relation queries on synthetic 3D scenes (Johnson et al., 2017) and natural scene graphs (Hudson & Manning, 2019), alongside multidisciplinary visual diagrams (Yue et al., 2024) and geometric problem-solving (Lu et al., 2024). Recent diagnostic benchmarks target 6D object pose (Wang et al., 2025) and multi-step action preconditions under complex indoor scenes (Xie et al., 2026). While A2RBench (Ma et al., 2026) procedurally generates formally verifiable, multidimensional abstract reasoning tasks, it focuses on symbolic rules rather than grounded 3D spatial perception. Concurrently, offline spatial analyses demonstrate that high performance on static visual mazes often stems from serial search in token space rather than genuine visual planning (Rodriguez Salgado, 2026). Crucially, these benchmarks are strictly non-interactive, evaluating spatial reasoning in a single turn without any action or visual feedback.

2.3 Interactive Benchmarks for Visual and Spatial Reasoning

Interactive benchmarks evaluate visual and spatial reasoning through multi-turn closed-loop interaction. In embodied AI, interactive benchmarks and agent architectures evaluate spatial reasoning through embodied navigation, long-horizon task execution, and scene exploration (Shridhar et al., 2020; Savva et al., 2019; Driess et al., 2023; Rana et al., 2023; Wang et al., 2023; Zhu et al., 2025); however, they predominantly emphasize macroscopic agent locomotion and high-level semantic planning. Across digital domains, benchmarks test 2D layout grounding and interface parsing for web browsing and operating system control (Zhou et al., 2024; Koh et al., 2024; Xie et al., 2024), yet they operate strictly within 2D planar graphical displays. Closer to geometric reasoning, interactive benchmarks evaluate spatial search via textual coordinates (Einarsson, 2026), open-world 3D labyrinth navigation with Sokoban-style box-pushing puzzles (Pappas & Pappas, 2026), and visual sliding-tile puzzles (Mayer et al., 2025); nevertheless, their objects are axis-aligned grid entities without rotational pose variation, reducing spatial reasoning to combinatorial pathfinding over dis-

crete states. In contrast, RoTA-Bench evaluates whether models can turn 2D observations into 3D understanding that supports precise action through closed-loop projection alignment.

3 RoTA-Bench

RoTA-Bench evaluates whether vision-language models can turn 2D observations into 3D understanding that supports precise action through interactive projection alignment (Figure 1). Figure 2 summarizes task construction and closed-loop evaluation. Below, we detail the benchmark design, task formulation, automated task construction, and evaluation protocol.

3.1 Motivation and Design Rationale

Classical studies in cognitive vision and psychology demonstrate that underlying 3D geometry and orientations can be inferred from 2D visual contours (Richards et al., 1987) and compared through mental rotation (Shepard & Metzler, 1971). Inspired by these foundational insights, we formulate 3D orientation alignment as a projection-matching task: rotating a 3D object until its 2D silhouette matches a target image. Restricting physical interaction to 3D rotation provides two key methodological advantages. First, out-of-plane rotations induce nonlinear silhouette deformations, aspect changes, and self-occlusion, creating an inherently visual challenge that resists textual representation. Second, pure rotation cleanly decouples spatial orientation reasoning from extraneous physical dynamics—such as translational drift, grasp affordances, and contact friction—allowing us to examine whether models can turn 2D observations into 3D understanding that supports precise action.

To reduce reliance on semantic shortcuts and encourage visual-spatial reasoning, RoTA-Bench synthesizes 3D objects from 2D vector icons rather than drawing directly from common 3D CAD repositories, where familiar shapes may trigger pre-trained priors. By extruding 2D icons within a spherical volume, this automated generation produces abstract, novel 3D geometries while mathematically guaranteeing that the canonical front view reproduces the original icon, preserving its fine-grained details (Appendix E). Finally, generating initial poses via sequences of reversible discrete rotations guarantees by construction that every task is solvable.

3.2 Task Definition

Following the design rationale in Section 3.1, we formulate the interactive projection-matching task as follows. Each task instance is defined by a rigid 3D object $G \subset \mathbb{R}^3$ centered at the coordinate origin, an initial pose $R_0 \in \mathrm{SO}(3)$, and a canonical target pose $R^\star = I_3$. Let $\Pi(\cdot)$ denote the orthographic projection operator that renders the binary silhouette mask of object G at a given pose. At step t, the environment provides an observation triplet:

$$O_t = (I^\star,\, V_t,\, I_t), \quad I^\star = \Pi(G, R^\star), \quad I_t = \Pi(G, R_t), \tag{1}$$

where $I^\star \in \{0,1\}^{H\times W}$ represents the invariant target silhouette, and $I_t \in \{0,1\}^{H\times W}$ is the current projection silhouette. To complement the binary silhouette with missing depth and occlusion cues, the agent also receives a depth-shaded auxiliary view $V_t \in [0,255]^{H\times W}$, rendered from the identical viewpoint as I_t.

Conditioned on this visual feedback, the agent interacts with the object through a discrete action space. At each step t, the agent selects an action a_t from a vocabulary $\mathcal{A}$ consisting of 24 camera-fixed rotation actions and a terminal STOP command. These 24 actions span six directional senses across three principal camera axes—pitch (up/down), yaw (left/right), and in-plane roll (clockwise/counterclockwise)—paired with four step sizes $\Delta\theta \in \{1^\circ, 5^\circ, 10^\circ, 30^\circ\}$ to enable both coarse and fine pose adjustments. Letting $Q(a_t) \in \mathrm{SO}(3)$ denote the rotation matrix for action a_t, the object pose updates via:

$$R_{t+1} = Q(a_t) R_t. \tag{2}$$

Executing a_t updates the 3D state and re-renders observations I_{t+1} and V_{t+1}.

To evaluate projection alignment, we compute the projection Intersection over Union (projection IoU) between the current projection I_t and the target silhouette $I^\star$:

$$s_t = \mathrm{IoU}(I_t, I^\star) = \frac{|I_t \cap I^\star|}{|I_t \cup I^\star|}. \tag{3}$$

An episode succeeds whenever $s_t \geq \tau$, where we set $\tau = 0.90$. This threshold enforces high visual alignment fidelity while providing a necessary tolerance margin for the discrete action space, where exact alignment is difficult to achieve in finite steps. Furthermore, because geometric symmetries and orthographic self-overlap can produce identical silhouettes from distinct 3D poses, the task naturally admits multiple equivalent target poses. We formalize this goal set as:

$$\mathcal{R}_{\text{goal}}(G, I^\star) = \{R \in \text{SO}(3) \mid \text{IoU}(\Pi(G, R), I^\star) \geq \tau\}. \tag{4}$$

Finally, an episode terminates upon task success ($s_t \geq \tau$), voluntary halting via STOP, reaching a maximum action budget, or stagnation after consecutive steps without improving the best IoU.

3.3 Dataset Construction

Our automated pipeline constructs the benchmark across three sequential phases: task generation, reference search, and dataset construction, as summarized in Figure 2.

First, in the task generation phase, following the design rationale in Section 3.1, we curate 2D vector icons from the open-source Game-icons library (Appendix B), rasterizing them into high-resolution binary masks after geometric filtering on connectivity, occupancy, aspect ratio, and foreground point count (Appendix G). To construct a 3D object from each 2D icon while strictly bounding its spatial profile under arbitrary rotations, the planar point set $P \subset \mathbb{R}^2$ is scaled within a disk of radius r and symmetrically extruded along the depth axis under a spherical envelope:

$$h(x, y) = \sqrt{\max(r^2 - x^2 - y^2,\, 0)}, \quad z \in [-h(x, y),\, h(x, y)]. \tag{5}$$

The resulting solid $G = \{(x, y, z) \mid (x, y) \in P,\, |z| \leq h(x, y)\}$ is discretized as a dense 3D point cloud (Appendix G). Moreover, because 3D rotations preserve Euclidean norms, any rotated pose satisfies $x^2 + y^2 + z^2 \leq r^2$. This spherical invariance prevents viewport clipping and fixes the camera scale globally across all tasks. To provide monocular depth cues and bridge discretization gaps between sampled points, visual observations are rendered via orthographic projection using depth-dependent shading and Gaussian smoothing (Appendix G).

To preserve the original 2D icon's fine contour details (Appendix E), the canonical target pose is fixed to $R^\star = I_3$, and initial poses $R_0 = Q(b_L) \cdots Q(b_1) R^\star$ are generated via $L \sim \text{Uniform}\{6, \ldots, 16\}$ discrete rotations. At each step, $b_k \in \mathcal{A}_{30^\circ}$ rotates by 30°, with its direction sampled uniformly at random from the six candidate camera directions (each with probability $1/6$), forbidding the immediate inverse ($b_k \neq b_{k-1}^{-1}$) to avoid cancellations and guarantee genuine displacement. Initial poses already projection-aligned ($s_0 \geq \tau$) are discarded. Crucially, because every rotation in $\text{SO}(3)$ is invertible, executing the reversed sequence $Q(b_1)^{-1} \cdots Q(b_L)^{-1}$ restores $R^\star$, ensuring 100% reachability and task solvability by construction. In total, this generation pipeline yields 674 canonical tasks with unique source icons.

Second, in the reference search phase, we compute reference paths using pose-guided beam search and 30°-restricted breadth-first search. These discover significantly shorter trajectories than raw reversed generation paths (Appendix H).

Finally, in the dataset construction phase, a stratified test set of 100 tasks was frozen across reference path lengths (≥ 4 steps) using proportional stratified sampling via Hamilton's method and deterministic hash sorting, reserving the remaining 574 tasks for future training and research (Appendix I).

3.4 Evaluation Metrics

Task performance metrics. Task Success Rate is the fraction of episodes reaching IoU $s_t \geq 0.90$. Mean Final IoU and mean rotation steps average the final IoU and executed rotations over all tasks, including failures. An episode fails if it ends through STOP, the 48-decision limit, or stagnation before reaching the threshold.

Geometric diagnostic metrics. We compute 3D pose error from ground-truth pose data, which are not provided to the model:

$$e_t = \min_{S \in \mathcal{G}} d_{\text{SO}(3)}(R_t,\, R^\star S) = \min_{S \in \mathcal{G}} \arccos\left(\frac{\text{Tr}(R_t (R^\star S)^\top) - 1}{2}\right). \tag{6}$$

Here, $\mathcal{G}$ contains the object's estimated symmetries (Appendix D). Pose error is the smallest rotation needed to reach an equivalent target pose. Equivalent target poses produce exact alignment, whereas task success requires only a projection IoU of at least 0.90. Thus, a successful pose need not have zero pose error. Pose error is used as a geometric diagnostic rather than as the task objective; a lower pose error does not necessarily imply a larger immediate IoU improvement. Surface coverage is the fraction of the object's surface observed during an episode.

A direction (axis and sign) is effective if at least one legal angle reduces pose error: $\min_{\theta \in \{1^\circ, 5^\circ, 10^\circ, 30^\circ\}} e(R_t, d, \theta) < e_t$. Effective Direction Rate is the fraction of rotations that choose such a direction; the chosen angle may still increase error. For effective directions, let $\Theta_t^\star = \arg\min_\theta e(R_t, d, \theta)$ be the best legal angles. Best Legal Angle Rate is the fraction choosing one of these angles. Angle Gap is $\min_{\theta \in \Theta_t^\star} |\theta_{\text{selected}} - \theta|$. Angle Loss is the extra pose error caused by the chosen angle: $e(R_t, d, \theta_{\text{selected}}) - \min_\theta e(R_t, d, \theta)$.

We also compare changes in IoU and pose error, and correlate rotation angles with both values before each action. Appendix K gives the definitions and confidence intervals.

4 Experimental Setup

We evaluate nine vision-language models—GPT-5.6-Sol, GPT-5.6-Terra, GPT-5.6-Luna (OpenAI, 2026a), Gemini 3.6 Flash (DeepMind, 2026), Claude Opus 5, Claude Sonnet 5 (Anthropic, 2026a;b), Qwen 3.7 Plus (Qwen Team, 2026), GPT-6-Astra (OpenAI, 2026b), and SpatioLM-Understanding-InternVL3.5 (Wu et al., 2026)—as well as two GPT-5.6-Sol-based agent frameworks, ReAct (Yao et al., 2022) and Magentic (Fourney et al., 2024), 20 undergraduate participants (Human; Appendix C), two stochastic baselines (Uniform random and Random with IoU rollback), and two greedy baselines. All evaluations follow a common task protocol, action space, decision budget, success criterion, and termination rules. All models are evaluated at temperature 0; closed-source models are queried via APIs (total cost: $1,482.07), while SpatioLM is deployed locally on a server equipped with eight NVIDIA A100 (40GB) GPUs. Complete prompts, hyperparameters, agent tools, and evaluation settings are detailed in Appendices L and M.

5 Results

In this section, we present main results, conduct rotation direction and angle analysis, perform an ablation of auxiliary views and interaction history, analyze action association with projection and pose discrepancies, and discuss limitations and future work.

5.1 Main Results

Table 1 presents alignment performance across 100 test tasks, with success counts and confidence intervals in Appendix K.1. Frontier models struggle markedly on interactive 3D projection alignment: GPT-5.6-Sol achieves the highest success rate at only 17.0%, followed by Gemini 3.6 Flash at 10.0%, while ReAct attains the highest final IoU of 0.628 (16.0% success). Claude Sonnet 5, Claude Opus 5, and Qwen 3.7 Plus fail entirely (0.0% success), with final IoU values (0.484–0.504) barely exceeding uniform random exploration (0.473). Notably, SpatioLM-Understanding-InternVL3.5 suffers a catastrophic policy collapse, where 91.12% of its actions degenerate into `YAW_POS 10°`, yielding 0.0% success and the lowest final IoU of 0.463 (below random actions). In stark contrast, human participants achieve a 100.0% success rate, reaching an average final IoU of 0.923 in 16.22 steps. As shown in Figure 3, human trajectories exhibit overall IoU gains surpassing the 0.90 threshold, whereas model trajectories plateau early between 0.50 and 0.63. The two greedy baselines sharpen this distinction: at each step, each enumerates all 24 legal rotation successors using the ground-truth simulator, then selects the action that maximizes successor projection IoU or minimizes true 3D pose error, respectively. The IoU-greedy baseline succeeds on only 39.0% of tasks (final IoU 0.725), showing that greedily maximizing immediate projection overlap is insufficient on many tasks; by contrast, the pose-greedy baseline solves every task (100.0%, final IoU 0.924 in 8.32 steps), showing that reliable access to the true 3D pose error is sufficient to complete all of them. Overall, the success rates and IoU trajectories show a substantial human–model gap on RoTA-Bench.

Table 1: Task alignment performance and action decomposition on 100 test tasks, sorted by Final IoU. Metrics include Task Success Rate (Succ.), final projection IoU, executed steps, visual surface coverage (Cov.), Effective Direction Rate (Eff. dir.), and for effective directions: Best Legal Angle Rate (Best ang.), Angle gap, and Angle loss (geometric penalty from sub-optimal step size). SpatioLM abbreviates SpatioLM-Understanding-InternVL3.5.

	Task Alignment Performance				**Pose Action Decomposition**			
Method	Succ. (%)	Final IoU	Steps	Cov. (%)	Eff. dir. (%)	Best ang. (%)	Gap (°)	Loss (°)
Pose greedy	**100.0**	**0.924**	**8.32**	–	**100.0**	**100.0**	**0.00**	**0.00**
IoU greedy	39.0	0.725	16.49	–	63.86	40.46	15.45	8.00
Human	**100.0**	0.923	16.22	83.07	83.91	71.82	2.99	0.96
ReAct	16.0	**0.628**	24.62	79.08	59.59	58.01	7.44	2.81
Gemini 3.6 Flash	10.0	0.607	24.68	74.72	61.87	61.69	5.47	1.99
GPT-5.6-Sol	**17.0**	0.598	23.66	79.58	58.75	61.80	6.49	2.25
GPT-6-Astra	10.0	0.578	23.18	78.93	55.78	67.36	5.85	2.14
GPT-5.6-Terra	8.0	0.554	22.91	73.13	56.31	53.80	8.50	3.58
GPT-5.6-Luna	4.0	0.553	24.29	71.56	57.64	50.86	9.15	4.03
Random + IoU rollback	0.7	0.545	35.57	–	57.52	30.69	14.53	6.48
Magentic	2.0	0.529	16.00	59.71	58.31	49.95	9.77	4.12
Qwen 3.7 Plus	0.0	0.504	21.03	52.02	52.40	47.46	9.96	4.90
Claude Sonnet 5	0.0	0.492	19.39	58.66	53.38	70.53	6.10	2.15
Claude Opus 5	0.0	0.484	22.09	62.40	53.28	65.68	6.85	2.29
Uniform random	0.0	0.473	22.23	–	53.78	23.45	17.05	7.67
SpatioLM-InternVL3.5	0.0	0.463	18.81	66.85	46.15	13.36	16.55	8.61

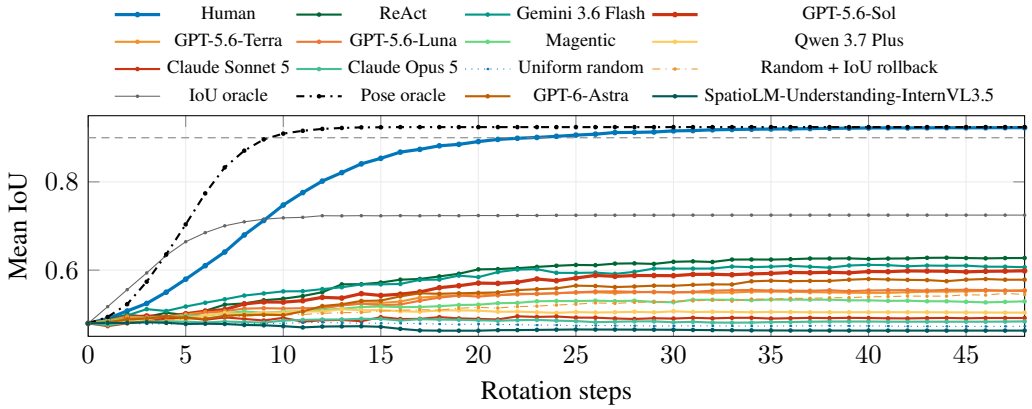


Figure 3: Mean IoU versus executed rotations for all methods with exported step trajectories on the same 100 tasks, including the 400 Human task runs and both extension runs. Each marker is the mean at one integer step (0–48), joined without smoothing; terminal IoU is retained after termination through step 48. The dashed line marks the success threshold.

5.2 Diagnosing Model Failures

We first evaluate whether model failure stems from inadequate visual exploration. As reported in Table 1, GPT-5.6-Sol achieves a cumulative surface coverage of 79.58%, closely trailing human participants (83.07%), while ReAct (79.08%) and GPT-6-Astra (78.93%) maintain comparable exposure. Total coverage is similar, but the regions viewed and their use may differ.

To diagnose model action failures, we decouple each rotation into direction (rotation axis and sign) and angle (step size), which correspond to estimating the direction and magnitude of the 3D pose difference between the current pose and target pose. Evaluating these decomposed components in Table 1 reveals clear deficiencies on both fronts: For directional choice, GPT-5.6-Sol selects effective rotation directions far less often than humans (58.75% vs. 83.91%) and performs only marginally better than uniform random sampling (53.78%), with other frontier models exhibiting

Table 2: Input ablation for GPT-5.6-Sol on 30 test tasks, with ablation conditions sorted by Final IoU. Columns report Task Success Rate (Succ.), final projection IoU, Effective Direction Rate (Eff. dir.), percentage distribution across the four legal rotation angles, and Spearman correlation (ρ) between pre-action IoU and rotation angle magnitude with 95% confidence intervals.

	Success	Final	Effective	Rotation angle (%)				IoU–angle correlation	
Condition	(%)	IoU	direction (%)	1°	5°	10°	30°	ρ	95% CI
A: Full input	**16.67**	**0.5656**	**56.52**	12.18	7.51	17.56	62.75	-0.472	[-0.626, -0.272]
B: No aux. view	3.33	0.5490	55.90	3.54	3.24	7.08	86.14	-0.463	[-0.605, -0.238]
D: Neither input	6.67	0.5087	55.02	1.29	1.29	0.81	96.60	-0.295	[-0.468, -0.066]
C: No history	0.00	0.4822	54.85	0.16	0.49	2.59	96.76	-0.242	[-0.331, -0.131]
Uniform random	0.00	0.4808	53.12	24.24	25.27	27.10	23.39	-0.027	[-0.073, 0.019]

similarly low rates (52.40%–61.87%). For angular precision, conditioned on choosing an effective direction, GPT-5.6-Sol exhibits a larger mean deviation between its selected angle and the best legal angle (6.49° vs. 2.99° for humans; best legal angle rate 61.80% vs. 71.82%), incurring an average geometric angle loss of 2.25° per step (more than double the 0.96° loss of humans). Appendix K.4 reports paired confidence intervals for the Human–Sol differences in Effective Direction Rate and Angle Gap. Together, these action metrics show that the human–model gap also appears in rotation direction and angle selection.

5.3 Ablation: Auxiliary Views and Interaction History

We evaluate the contribution of visual depth cues and multi-turn feedback by ablating the auxiliary depth view and interaction history for GPT-5.6-Sol across the first 30 tasks, whose distribution of reference solution steps strongly aligns with the 100-task test set (Spearman's $\rho = 0.9121$, $p = 3.56 \times 10^{-5}$; 10,000 bootstrap resamples yield a 95% CI lower bound of 0.7445). As reported in Table 2, removing the auxiliary view reduces task success from 16.67% (full input) to 3.33% and final IoU from 0.5656 to 0.5490. Removing interaction history reduces observed success from 5/30 tasks to 0/30 and final IoU from 0.5656 to 0.4822, close to uniform random performance (0.4808). The observed effect of removing history is larger than that of removing the auxiliary view.

Interestingly, across ablation settings, the effective-direction rate remains nearly unchanged (54.85%–56.52%), while angle selection changes markedly. Removing the auxiliary depth view increases 30° rotations to 86.14% but leaves the IoU–angle correlation close to that under full input ($\rho = -0.463$ vs. -0.472). Without interaction history, 30° rotations rise to 96.76% from 62.75%, 1° adjustments fall to 0.16% from 12.18%, and the correlation weakens to $\rho = -0.242$. In this ablation, history helps the model choose how far to rotate more than which way to rotate. Previous views and actions may help it recognize when to follow a coarse turn with a finer adjustment. Counts and paired intervals are in Appendix K.5. These results reveal that GPT-5.6-Sol's fine-grained control is sensitive to interaction history.

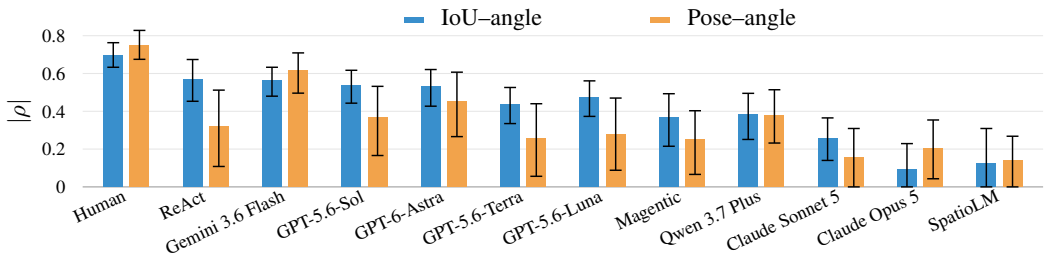
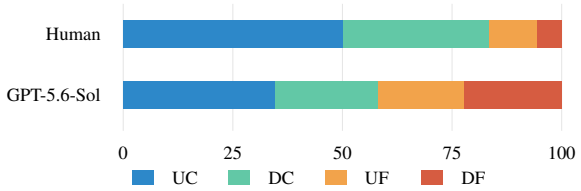


Figure 4: Absolute Spearman correlations between selected rotation angles and pre-action projection IoU or 3D pose error across the 100 test tasks. Bars show $|\rho|$; whiskers show the original 95% bootstrap intervals mapped to absolute values. An interval crossing zero has a lower bound of zero after this transformation. SpatioLM abbreviates SpatioLM-Understanding-InternVL3.5.



Measure	Human	Sol
Prior rotations	2.40	4.18
Prior coverage (%)	42.01	46.21
ΔIoU	0.0207	0.0071
$-\Delta$pose ($^\circ$)	10.0420	2.8740

Figure 5: Context and action outcomes across 211 matched states. Prior rotations and coverage are measured before each matched action. UC, DC, UF, and DF denote the four combinations of IoU increasing or decreasing and pose error decreasing or increasing, respectively.

5.4 Action Association with Projection and Pose Discrepancies

We next examine whether rotation-angle choices rely more on visible projection overlap or on the underlying 3D pose discrepancy. Figure 4 and Appendix Table 10 report their correlations with pre-action IoU and pose error. For GPT-5.6-Sol, the correlation is stronger with IoU than with pose error ($\rho = -0.536$, 95% CI $[-0.617, -0.443]$ vs. $\rho = 0.368$, 95% CI $[0.166, 0.532]$). Human angles are strongly associated with both ($\rho = -0.696$, 95% CI $[-0.763, -0.633]$, and $\rho = 0.751$, 95% CI $[0.675, 0.828]$). Most evaluated models show the same relative pattern as GPT-5.6-Sol. These results suggest that most models rely more on projection overlap when choosing rotation angles, whereas humans use both projection overlap and 3D pose discrepancy.

To conduct a comparative analysis under the same conditions as closely as possible, we examine action consequences across the 211 exact state pairs shared by Human and GPT-5.6-Sol, where direct visual coverage before the matched actions is close (42.01% vs. 46.21%, Figure 5 and Appendix Table 8). We find that image-space and pose-space progress can diverge even when actions start from the same pose. Humans execute substantially more DC actions (33.18% vs. 23.70%), showing a willingness to accept transient drops in 2D overlap to reduce underlying 3D pose error. Conversely, GPT-5.6-Sol executes nearly twice as many UF actions as humans (19.43% vs. 10.90%), showing a greater willingness to accept increases in 3D pose error to improve 2D overlap. For the remaining outcomes, humans achieve far higher UC actions (50.24% vs. 34.60%) and avoid DF moves (5.69% vs. 22.27%). Together, these results show that image-space and pose-space progress can diverge: at matched states, GPT-5.6-Sol more often improves silhouette overlap while worsening pose alignment than humans.

5.5 Limitations and Future Work

This work has several limitations. First, the 3D objects are extruded from a limited set of 2D icons, which constrains shape diversity and cannot fully represent complex 3D geometries. Second, each task features only a single target under discrete rotation actions, making the interaction dynamics relatively simplified compared to continuous physical environments. Future work can expand task and icon diversity by leveraging generative models to synthesize richer 2D icons or directly construct more expressive 3D meshes. In addition, the benchmark can be extended by introducing continuous rotation control, external occlusions, and multi-object alignment or assembly tasks to further heighten the spatial reasoning challenge. Finally, automatically generated tasks could provide scalable environments for reinforcement learning fine-tuning, offering a way to study whether interactive training improves the use of visual feedback for precise 3D control.

6 Conclusion

RoTA-Bench evaluates whether vision-language models can turn 2D observations into 3D understanding that supports precise action through interactive projection alignment. Its automated pipeline constructs silhouette-preserving 3D objects and uses reversible initialization to guarantee solvability, yielding 674 tasks and a stratified 100-task test set. Evaluations reveal a substantial human–model success gap, show that image-space and pose-space progress can diverge, and find that fine-grained control is sensitive to interaction history. These findings highlight the challenge of converting visual evidence into precise 3D control.

AI USE STATEMENT

Generative AI tools were used to aid and polish the writing and to retrieve and discover related work. The vision-language models evaluated in this paper served solely as experimental subjects and were not used to assist the authors' research process. The authors reviewed all AI-assisted materials, verified the reported claims and experimental results, and take full responsibility for the final content of this paper.

ETHICS STATEMENT

All human evaluation experiments were conducted in accordance with ethical research standards, with informed consent obtained from all 20 undergraduate participants prior to the study. Participants were briefed on the task and received fair financial compensation for their time. The evaluation platform displayed only task interactions and recorded only discrete action selections and interaction timestamps, without collecting any personally identifiable information (PII) or sensitive personal data. Furthermore, all 3D task geometries were procedurally constructed from open-source 2D vector icons under valid Creative Commons Attribution 3.0 (CC BY 3.0) and CC0 public domain licenses, with complete author attributions preserved.

REPRODUCIBILITY STATEMENT

The release provides the task construction code, evaluation pipeline, and the complete 674-task dataset. The appendix details full prompting protocols, 3D geometric construction, and validation pipelines.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *Advances in neural information processing systems*, 35:23716–23736, 2022.

Anthropic. Claude opus 5 system card, 2026a. URL https://www-cdn.anthropic.com/ceaf5c7ff2783855203fde8208ec311252dced5b/Claude%20Opus%205%20System%20Card.pdf.

Anthropic. Claude sonnet 5 system card, 2026b. URL https://www-cdn.anthropic.com/283ef97c476cf442c91d9a37d5b214242a55bb92/Claude%20Sonnet%205%20System%20Card.pdf.

John B. Carroll. *Human Cognitive Abilities: A Survey of Factor-Analytic Studies*. Cambridge University Press, 1993.

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

Google DeepMind. Gemini 3.6 flash model card, 2026. URL https://storage.googleapis.com/deepmind-media/Model-Cards/Gemini-3-6-Flash-Model-Card.pdf.

Ken Deng, Yifu Qiu, Yoni Kasten, Shay B Cohen, and Yftah Ziser. Lost in space? vision-language models struggle with relative camera pose estimation. *arXiv preprint arXiv:2601.22228*, 2026.

Danny Driess, Fei Xia, Mehdi SM Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, et al. Palm-e: An embodied multimodal language model. *arXiv preprint arXiv:2303.03378*, 2023.

Hafsteinn Einarsson. Mazeeval: A benchmark for testing sequential decision-making in language models. In *Proceedings of the Fifteenth Language Resources and Evaluation Conference (LREC 2026)*, volume 11, pp. 407–418, 2026.

Sagi Eppel, Mor Bismut, and Alona Strugatski. Shape and texture recognition in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1839–1849, 2026.

Adam Fourney, Gagan Bansal, Hussein Mozannar, Cheng Tan, Eduardo Salinas, Friederike Niedtner, Grace Proebsting, Griffin Bassman, Jack Gerrits, Jacob Alber, et al. Magentic-one: A generalist multi-agent system for solving complex tasks. *arXiv preprint arXiv:2411.04468*, 2024.

Peter E Hart, Nils J Nilsson, and Bertram Raphael. A formal basis for the heuristic determination of minimum cost paths. *IEEE transactions on Systems Science and Cybernetics*, 4(2):100–107, 1968.

Yining Hong, Haoyu Zhen, Peihao Chen, Shuhong Zheng, Yilun Du, Zhenfang Chen, and Chuang Gan. 3d-llm: Injecting the 3d world into large language models. *Advances in Neural Information Processing Systems*, 36:20482–20494, 2023.

Jiangyong Huang, Silong Yong, Xiaojian Ma, Xiongkun Linghu, Puhao Li, Yan Wang, Qing Li, Song-Chun Zhu, Baoxiong Jia, and Siyuan Huang. An embodied generalist agent in 3d world. *arXiv preprint arXiv:2311.12871*, 2023.

Drew A Hudson and Christopher D Manning. Gqa: A new dataset for real-world visual reasoning and compositional question answering. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6693–6702. IEEE, 2019.

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

Jing Yu Koh, Robert Lo, Lawrence Jang, Vikram Duvvur, Ming Lim, Po-Yu Huang, Graham Neubig, Shuyan Zhou, Russ Salakhutdinov, and Daniel Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 881–905, 2024.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International conference on machine learning*, pp. 19730–19742. PmLR, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. *Advances in neural information processing systems*, 36:34892–34916, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *International Conference on Learning Representations*, volume 2024, pp. 23439–23554, 2024.

Qingchuan Ma, Yuexiao Ma, Yongkang Xie, Tianyu Xie, Xiawu Zheng, and Rongrong Ji. A2rbench: An automatic paradigm for formally verifiable abstract reasoning benchmark generation. *arXiv preprint arXiv:2605.17278*, 2026.

Wufei Ma, Luoxin Ye, Celso M de Melo, Alan Yuille, and Jieneng Chen. Spatialllm: A compound 3d-informed design towards spatially-intelligent large multimodal models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 17249–17260. IEEE, 2025.

Julius Mayer, Mohamad Ballout, Serwan Jassim, Farbod Nosrat Nezami, and Elia Bruni. ivispar—an interactive visual-spatial reasoning benchmark for vlms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 26745–26769, 2025.

Ke Niu, Zhuofan Chen, Haiyang Yu, Yuwen Chen, Teng Fu, Mengyang Zhao, Bin Li, and Xiangyang Xue. Creft-cad: Boosting orthographic projection reasoning for cad via reinforcement fine-tuning. *Advances in Neural Information Processing Systems*, 38:82608–82630, 2026.

OpenAI. Gpt-5.6 system card, 2026a. URL `https://deploymentsafety.openai.com/gpt-5-6/gpt-5-6.pdf`.

OpenAI. Gpt-6 astra system card, 2026b. URL https://deploymentsafety.openai.com/gpt-6-astra/gpt-6-astra.pdf.

Jonathan Pappas and David Pappas. Maze Bench: Visual spatial reasoning in a 3d open world, 2026. URL https://mazebench.com. Project oversight by Florian Brand and Sebastian Müller.

Qwen Team. Qwen3.7: The agent frontier. Qwen Official Blog, May 2026. URL https://qwen.ai/blog?id=qwen3.7. Official model release; accessed 2026-09-03.

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

Krishan Rana, Jesse Haviland, Sourav Garg, Jad Abou-Chakra, Ian Reid, and Niko Suenderhauf. Sayplan: Grounding large language models using 3d scene graphs for scalable robot task planning. *arXiv preprint arXiv:2307.06135*, 2023.

WA Richards, Jan J Koenderink, and Donald D Hoffman. Inferring three-dimensional shapes from two-dimensional silhouettes. *Journal of the optical society of america a*, 4(7):1168–1175, 1987.

Corban Rivera, Grayson Byrd, William Paul, Tyler Feldman, Meghan Booker, Emma Holmes, David Handelman, Bethany Kemp, Andrew Badger, Aurora Schmidt, et al. Conceptagent: Llm-driven precondition grounding and tree search for robust task planning and execution. In *2025 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 8988–8995. IEEE, 2025.

Alberto Rodriguez Salgado. From pixels to bfs: High maze accuracy does not imply visual planning. *arXiv preprint arXiv:2603.26839*, 2026.

Manolis Savva, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, Vladlen Koltun, Jitendra Malik, et al. Habitat: A platform for embodied ai research. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9338–9346. IEEE, 2019.

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

Mohit Shridhar, Jesse Thomason, Daniel Gordon, Yonatan Bisk, Winson Han, Roozbeh Mottaghi, Luke Zettlemoyer, and Dieter Fox. Alfred: A benchmark for interpreting grounded instructions for everyday tasks. In *2020 IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 10737–10746. IEEE, 2020.

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 *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 15768–15780. IEEE, 2025.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*, 2023.

Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Pulsecheck457: A diagnostic benchmark for 6d spatial reasoning of large multimodal models. *arXiv e-prints, pp. arXiv–2502*, 2025.

Zihan Wang, Seungjun Lee, Guangzhao Dai, and Gim Hee Lee. D3d-vlp: Dynamic 3d vision-language-planning model for embodied grounding and navigation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 32463–32474, 2026.

Jing Wu, Jianhua Wu, Jiayi Guan, Jiahong Chen, Jinghui Lu, Hangjun Ye, Bingzhao Gao, and Long Chen. Spatiolm: Towards general physical spatial intelligence in vision-language models. In *International Conference on Machine Learning (ICML)*, 2026. URL https://arxiv.org/abs/2608.01899. To appear.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh J Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, et al. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Advances in Neural Information Processing Systems*, 37:52040–52094, 2024.

Yuechen Xie, Xiaoyan Zhang, Yicheng Shan, Zhu Hao, Rui Tang, Rong Wei, Mingli Song, Yuanyu Wan, and Jie Song. Spatialqa: A benchmark for evaluating spatial logical reasoning in vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 2645–2657, 2026.

Runsen Xu, Xiaolong Wang, Tai Wang, Yilun Chen, Jiangmiao Pang, and Dahua Lin. Pointllm: Empowering large language models to understand point clouds. In *European Conference on Computer Vision*, pp. 131–147. Springer, 2024.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*, 2022.

Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. Mmmu: A massive multi-discipline multimodal understanding and reasoning benchmark for expert agi. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9556–9567, 2024.

Qingqing Zhao, Yao Lu, Moo Jin Kim, Zipeng Fu, Zhuoyang Zhang, Yecheng Wu, Zhaoshuo Li, Qianli Ma, Song Han, Chelsea Finn, et al. Cot-vla: Visual chain-of-thought reasoning for vision-language-action models. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1702–1713. IEEE, 2025.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, et al. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024.

Ziyu Zhu, Xilin Wang, Yixuan Li, Zhuofan Zhang, Xiaojian Ma, Yixin Chen, Baoxiong Jia, Wei Liang, Qian Yu, Zhidong Deng, et al. Move to understand a 3d scene: Bridging visual grounding and exploration for efficient and versatile embodied navigation. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 8120–8132. IEEE, 2025.

Table 3: Test-task counts and GPT-5.6-Sol successes by Search-reference length, with success rates for the 4–6-step and 7–16-step groups.

Group	Search steps	Tasks	GPT-5.6-Sol successes	GPT-5.6-Sol success rate
4–6 steps	4	8	3	12/55 (21.8%)
	5	19	3	
	6	28	6	
7–16 steps	7	19	2	5/45 (11.1%)
	8	11	1	
	9	6	2	
	10	3	0	
	11	1	0	
	12	2	0	
	13	1	0	
	14	1	0	
	15	0	0	
	16	1	0	

A Appendix

The appendix covers icon sources and licensing, human evaluation, target orientation and icon details, icon processing and observation generation, search reference construction, dataset composition and test selection, implementation and evaluation settings, prompt examples, and success and failure examples.

B Icon Sources and Licensing

RoTA-Bench uses SVG shape assets from the official Game Icons repository ("https://github.com/game-icons/icons"), using the repository snapshot at commit "b659ba7773ed6ad1e1fee228b089f7104bea4984" (May 4, 2025) that was available during task construction. The repository's "license.txt" states that icons are provided under the Creative Commons Attribution 3.0 license, or under CC0 when explicitly indicated. We preserve the upstream attribution and license information with the source assets. The SVG files are rasterized, quality-filtered, converted to geometric point sets, and rendered as grayscale three-dimensional objects; the original source colors are not used.

The final canonical collection contains 674 tasks and 674 unique source icons. Of these, 641 icons are from contributor directories identified as CC BY 3.0, 33 are from the explicitly marked CC0 directories ("viscious-speed" and "zeromancer").

Icons made by Lorc, Delapouite, Aussiesim, Viscious Speed, Sbed, Skoll, DarkZaitzev, Caro Asercion, Cathelineau, Lucas, Andy Meneely, Carl Olsen, Zeromancer, Guard13007, John Redman, SpencerDub, Felbrigg, Sparker, Lord Berandas, GeneralAce135, and Faithtoken. The source license is available at `https://github.com/game-icons/icons/blob/b659ba7773ed6ad1e1fee228b089f7104bea4984/license.txt`; license terms are available at `https://creativecommons.org/licenses/by/3.0/` and `https://creativecommons.org/publicdomain/zero/1.0/`.

C Human Evaluation

Participants and formal evaluation. Twenty undergraduate participants each completed one consecutive 20-task block in the fixed order of the 100-task test set. The same test set was completed four times, yielding 400 successful human task runs; each participant achieved 100% success on their assigned tasks. Participation was voluntary, with informed consent and fair compensation. No personally identifiable information was collected. Future work will evaluate participants beyond the undergraduate population.

Interface and procedure. Participants received no practice trials or additional instruction before the formal evaluation. At every step, they received the same target projection, auxiliary view, current projection, and chronological history information as the evaluated models. The task environment, observation content, action vocabulary, allowed angles, success threshold, stopping rules, and pose-transition function were identical for humans and models. Humans selected the same discrete actions and angles through interface buttons, while models returned the corresponding actions in JSON. Thus, the interaction space and environment input–output mapping were identical; the buttons only provided a convenient human control interface. Human reasoning was not recorded.

Response validity and time. The model evaluator obtained valid JSON responses for all logged actions within at most six retries. This confirms that the structured action interface was available throughout evaluation; human button selections represented the same action tokens and angles. Across the 400 human task runs, 99% were completed within 278.12 seconds (4.64 minutes) per task. Task duration includes the time between task initialization and the final action, including pauses between actions.

Individual results and aggregation. Table 5 reports individual results. Task metrics average the four human task runs for each test task, then average across tasks. Action metrics combine all executed rotations; angle metrics use only effective directions.

Table 4: Shared human and model evaluation protocol. The interaction environment, visible observations, action space, and task rules are identical; humans use buttons to submit the same actions that models return in JSON.

Setting	Shared protocol
Tasks	Same 100-task test set; each participant completes one consecutive 20-task block
Current observations	Target projection, auxiliary view, current projection, and chronological history
Rotation actions	Six signed directions with 1°, 5°, 10°, or 30°
Action representation	Human button selection and model JSON both map to the same action token and angle
Success criterion	Projection IoU ≥ 0.90
Stopping rules	Success, STOP, 48 decisions, or 12 rotations without a new best IoU
Hidden information	Numerical IoU, internal poses, and reference paths remain hidden
Environment transition	The same pose update and observation regeneration after every action

Table 5: Human results for four passes over the 100-task test set. Each participant completed one consecutive 20-task block; participants with the same task-block index evaluated the same 20 tasks in different passes.

Participant	Pass	Task block	Successes/tasks	Success (%)	Final IoU	Rotations
1	1	1	20/20	100.00	0.9198	13.05
2	1	2	20/20	100.00	0.9190	11.90
3	1	3	20/20	100.00	0.9162	13.15
4	1	4	20/20	100.00	0.9172	12.05
5	1	5	20/20	100.00	0.9246	13.20
6	2	1	20/20	100.00	0.9283	18.85
7	2	2	20/20	100.00	0.9285	17.40
8	2	3	20/20	100.00	0.9266	14.95
9	2	4	20/20	100.00	0.9290	18.95
10	2	5	20/20	100.00	0.9265	16.65
11	3	1	20/20	100.00	0.9149	16.50
12	3	2	20/20	100.00	0.9197	14.25
13	3	3	20/20	100.00	0.9223	14.10
14	3	4	20/20	100.00	0.9233	16.55
15	3	5	20/20	100.00	0.9205	16.05
16	4	1	20/20	100.00	0.9179	23.15
17	4	2	20/20	100.00	0.9259	17.75
18	4	3	20/20	100.00	0.9283	16.80
19	4	4	20/20	100.00	0.9229	20.70
20	4	5	20/20	100.00	0.9252	18.35
Combined	–	–	400/400	100.00	0.9228	16.22

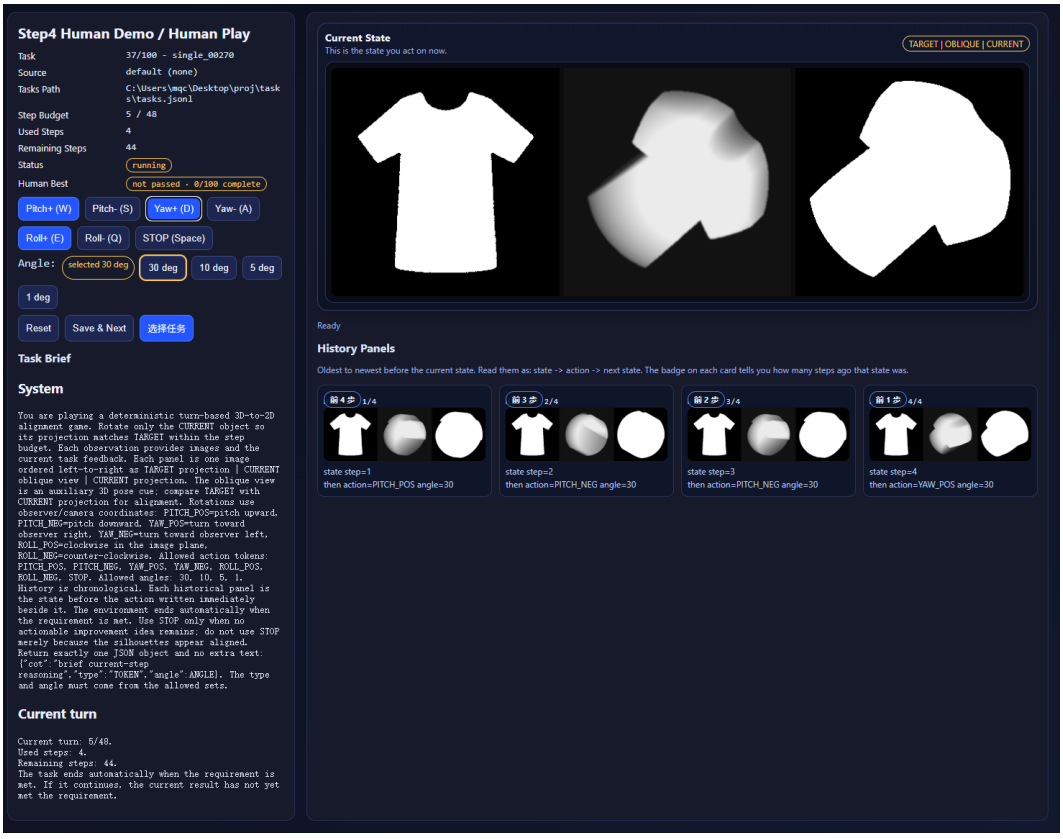


Figure 6: Human evaluation interface with the same prompt content, current observation, historical observations, and discrete action space used by the models. Humans submit actions through buttons that map to the model action tokens and angles.

D Computation of Object Symmetries

Symmetry allows different orientations to represent the same object pose. We estimate these symmetries from the normalized 2D points used to build each object, without using evaluation results.

Finding symmetries. We test rotational orders $n = 12, 11, \ldots, 2$. Order n requires every rotation by $360°k/n$, $k = 1, \ldots, n-1$, to pass the point-matching check below. We keep the first accepted order; otherwise, we keep only the identity rotation. For reflections, we test $Q(\alpha) = \mathrm{Rot}(\alpha)\,\mathrm{diag}(1, -1)$, where α is twice the mirror-axis angle. We search $0° \le \alpha < 360°$ in $2°$ steps and refine the four best candidates within $\pm 2°$. A reflection and all its combinations with the retained rotations must pass the same check.

Checking point matches. A symmetry should map the point set onto itself. We allow a small error because the points are sampled. Using up to 1,500 evenly indexed points, we estimate the median distance h to the nearest other point in the full set and set $\tau = 3h + 10^{-6}$. We measure nearest-neighbor distances in both directions between the original and transformed sets. In each direction, the 99th-percentile distance must be at most τ, and the maximum must be at most 2τ. We screen candidates with the sample, then check accepted candidates with all points.

Converting to 3D. The object's depth is symmetric about the 2D plane and depends only on distance from the center. Each planar symmetry Q therefore gives a 3D rotation

$$S(Q) = \begin{pmatrix} Q & 0 \\ 0 & \det Q \end{pmatrix}. \tag{7}$$

Planar rotations preserve depth; planar reflections also reverse depth. Both give determinant $+1$. These matrices, including the identity, form $\mathcal{G}$. We check that each matrix product matches a retained matrix within 10^{-6} in Frobenius norm.

Measuring pose error. Equation 6 takes the smallest rotation distance from the current pose to any equivalent target $R^\star S$. We clip the arccosine input to $[-1, 1]$ and report degrees. This is a numerical symmetry estimate limited by sampling tolerance and the search range. A matching silhouette need not imply an equivalent 3D pose, so pose error is a diagnostic and IoU determines success.

E Target Orientation and Icon Details

We set $R_{\mathrm{target}} = I_3$ because projection along the depth axis retains the source icon's details. For the continuous construction in Equation 5, projecting (x, y, z) to (x, y) recovers P: depth extension changes z but preserves the planar coordinates. Sampling and rasterization approximate this relation. At other orientations, extended depth can overlap in projection and obscure these details; the target need not have a unique matching orientation. Figure 10 illustrates this distinction: the initial projection is nearly circular, whereas the target preserves the shirt contour.

F Surface Coverage

We measure how much of an object becomes visible during an episode. First, we reconstruct the object's closed triangular surface with a grid spacing of 0.75 and assign each triangle its geometric area. We use the triangle centroid as its representative surface point. At each observed pose, a point is marked visible if its outward normal faces the camera and a ray from the point to the camera is not blocked by another triangle. Visibility is tested with exact ray–triangle intersections using a bounding-volume hierarchy.

Let V_t denote the visible surface points at step t, and let w_i be the area weight of point i. We accumulate visibility over the complete trajectory:

$$V_{\mathrm{episode}} = \bigcup_{t=0}^{T} V_t, \qquad \mathrm{Coverage} = \frac{\sum_{i \in V_{\mathrm{episode}}} w_i}{\sum_i w_i}. \tag{8}$$

Thus, a surface point is counted once even if it is observed at multiple steps. The reported coverage is the area-weighted fraction of the reconstructed surface observed at least once during the episode.

G Icon Processing and Observation Generation

Rasterization and foreground extraction. We rasterize each SVG to 512×512 RGBA pixels and estimate the background color c_{bg} from the four corners. For pixel (u, v), $C(u, v)$ is its RGB vector and $\alpha(u, v)$ its opacity. We retain a pixel when it is sufficiently opaque and differs from the background:

$$F(u, v) = \mathbf{1}[\alpha(u, v) > 8 \ \wedge \ \|C(u, v) - c_{\mathrm{bg}}\|_1 > 30]. \tag{9}$$

Here, $\mathbf{1}$ is an indicator, $\wedge$ requires both conditions, and $\|\cdot\|_1$ sums absolute RGB differences. This separates icon content from transparency and background before geometry construction.

Point extraction and filtering. We scan foreground pixels row by row and map (u, v) to $(24(u - 256)/256, 24(v - 256)/256)$, rounded to four decimals. If there are $n > 90{,}000$ points, we keep indices $\lfloor k(n-1)/(m-1) \rfloor$ for $m = 90{,}000$ and $k = 0, \ldots, m-1$; otherwise, all points are kept. This deterministic sampling bounds computation while preserving the icon outline. On a 112×112 occupancy grid, we retain icons only if the largest-component ratio is at least 0.985, circle occupancy at least 0.28, bounding-box occupancy at least 0.08, aspect ratio at most 2.2, and foreground count at least 2,000. These tests remove fragmented or degenerate shapes that would produce ambiguous tasks; circle occupancy is an area estimate, not mask IoU.

Geometry construction. We multiply points by 1.24, center them at their minimum enclosing circle, and fit them within $0.98r$ for $r = 28$. For centered (x, y), the sphere-bounded half-depth is $h(x, y) = \sqrt{\max(r^2 - x^2 - y^2, 0)}$ (Equation 5); the maximum prevents an invalid square root. Depth and complexity scales are 1. We use 31 samples per interval: endpoints $z_{i0} = -h(x_i, y_i)$ and $z_{i1} = h(x_i, y_i)$, plus 29 deterministic pseudorandom interior samples:

$$z_{ij} = (2q_{ij} - 1)h(x_i, y_i), \qquad j = 2, \ldots, 30. \tag{10}$$

Here, i and j are point and sample indices, $q_{ij} \in [0, 1)$ is a deterministic function of these indices and the task seed, and $2q_{ij} - 1$ maps it into the depth interval. This gives the 2D outline finite 3D thickness while keeping regeneration deterministic; the seed is stored with the task.

Projection. A rotated point (x_t, y_t, z_t) is mapped orthographically to $(c_x + \lambda x_t, c_y + \lambda y_t)$; z_t is depth, $(c_x, c_y) = (160, 160)$, and $\lambda = 5$ for 320×320 images. Coordinates are rounded, a 2-pixel disk is marked (255), and out-of-bounds pixels are discarded. Orthographic projection keeps alignment dependent on the visible outline rather than depth-dependent scale.

Auxiliary image. To add depth cues, the same point is mapped to $(c_x + 0.9\lambda x_t, c_y + 0.9\lambda y_t)$ with intensity

$$g_t = \min(235, \max(40, 110 + 6z_t + 1.8x_t)). \tag{11}$$

Here, 0.9 is the scale factor, 110 the baseline, 6 and 1.8 control depth and horizontal-position effects, and $\min(235, \max(40, \cdot))$ clips intensity to $[40, 235]$. The resulting gradient supplies depth cues that the binary mask does not contain.

Image smoothing. For Game-icons at 320×320, each point is rasterized with radius $\mathrm{round}[(320/64) \times 0.36] = 2$ pixels (the code computes this from image scale and source type), filling small gaps without strongly expanding the outline. The auxiliary background is 18; within each disk, intensity decreases by $\lfloor 2.2\sqrt{\Delta u^2 + \Delta v^2} \rfloor$, and overlaps keep the maximum. Both masks receive one morphological closing with a 3×3 elliptical kernel, and the auxiliary image receives two 3×3 Gaussian passes ($\sigma = 0$). These steps reduce raster gaps; both views use the same orientation.

Observation assembly. The target projection, current auxiliary image, and current projection form a 972×320 panel, separated by two 6-pixel strips of intensity 12. This fixed layout gives every method the same visual information. Formal evaluation uses the CUDA renderer with float32 geometry; software versions are listed in Appendix L. The complete implementation, including deterministic depth sampling, task seeds, and rendering parameters, is provided in the open-source code.

H Search Reference Construction

The inverse initialization sequence provides a feasible solution independently of search. Replaying it checks that the implemented transformations and projection scoring reproduce successful alignment. We combine pose-guided Beam-256 search with a breadth-first search (BFS) restricted to 30°

Table 6: Search-reference and inverse-initialization path lengths across 674 tasks. Counts and percentages compare the paired lengths; summary statistics report the mean, median, and range in rotation steps.

Comparison	Tasks	Percentage
Search reference shorter	585	86.8%
Equal length	89	13.2%
Search reference longer	0	0.0%
Inverse mean / median / range: 11.04 / 11 / 6–16		
Search mean / median / range: 6.54 / 6 / 2–16		

rotations, following the standard discrete-state search formulation (Hart et al., 1968). At depth d, Beam-256 computes the pose residual for each of the six directions, selects the closest angle from the four allowed angles for each direction, and therefore generates six candidate actions per retained path. It then keeps at most 256 candidates ranked by pose distance to the target and orientation diversity:

$$\mathcal{B}_{d+1} = \mathrm{Top}_{256}(\{p\,a \mid p \in \mathcal{B}_d,\ a \in \mathcal{A}\}), \qquad |\mathcal{A}| = 6. \tag{12}$$

The fixed-angle search instead performs BFS over rotation-matrix states, expanding every state at depth d with the same six actions and stopping at the first depth whose replayed projection reaches the success threshold:

$$\mathcal{S}_{d+1} = \{R(a)R \mid a \in \mathcal{A},\ R \in \mathcal{S}_d\}, \qquad \mathcal{S}_0 = \{R_{\mathrm{init}}\}. \tag{13}$$

Both searches replay candidate paths and verify them with the same projection IoU threshold used in evaluation. We retain the shorter verified candidate as the Search reference, although it is not necessarily optimal. Restricting the search keeps reference construction computationally feasible, while internal poses and reference paths remain hidden from evaluated models, agents, and humans. The complete implementation is provided in the open-source code.

Table 6 summarizes the comparison between the inverse initialization sequence and the resulting Search reference over all 674 canonical tasks.

I Dataset Composition and Test Selection

The canonical collection contains 674 tasks with distinct source-icon identifiers. The 642 tasks whose reference paths contain at least four rotations are eligible for test selection. We allocate 100 test places proportionally across reference-length groups using the Hamilton largest-remainder method: first assign the integer part of each proportional quota, then assign remaining places in descending order of fractional remainder. Within each group, tasks are ranked deterministically by a SHA-256 hash of the selection seed and task identifier. Table 7 gives the full dataset and test-set counts. The remaining 574 tasks are reserved for future training and research.

J Additional Results

Table 7: Task counts in the 674-task dataset and 100-task test set by Search-reference length. Tasks with at least four reference rotation steps are eligible for test selection.

Reference rotations	All tasks	Test tasks
< 4	32	0
4	53	8
5	122	19
6	181	28
7	124	19
8	69	11
9	38	6
10	21	3
11	9	1
12	10	2
13	5	1
14	5	1
15	1	0
16	4	1
Eligible (≥ 4)	642	100
Total	674	100

Table 8: Context before matched actions and their outcomes across 211 participant–task–state pairs. Prior rotations count completed rotations before each action. Prior coverage is cumulative direct surface coverage up to and including the current observation. All values are averaged over pairs. UC improves IoU and reduces pose error; DC reduces pose error only; UF improves IoU only; DF improves neither.

Method	Prior rotations	Prior coverage	ΔIoU	UC	DC	UF	DF	$-\Delta$pose (°)
Human	2.40	42.01%	0.0207	50.24%	33.18%	10.90%	5.69%	10.0420
GPT-5.6-Sol	4.18	46.21%	0.0071	34.60%	23.70%	19.43%	22.27%	2.8740

Table 9: Additional pose and trajectory statistics on the same 100 tasks. Final pose error is the symmetry-adjusted angular distance from the target at the end of a trajectory. Pose-improving steps are the percentage of all executed steps that reduce pose error. Mean ΔIoU per step and mean pose reduction per step are computed over complete trajectories; positive values indicate progress toward the target. Human combines 400 task runs from 20 undergraduate participants, and random policies average three seeds within each task. SpatioLM-InternVL3.5 abbreviates SpatioLM-Understanding-InternVL3.5.

Method	Final pose error (°)	Pose-impr. steps (%)	Mean ΔIoU / step	Mean pose red. / step (°)
Human	4.44	77.88	0.0273	5.4238
IoU oracle	79.06	54.82	0.0280	2.8687
Pose oracle	1.67	100.00	0.0535	10.9053
Uniform random	102.11	48.94	-0.0003	-0.4365
Random + IoU rollback	88.83	52.14	0.0018	0.1002
ReAct	82.69	52.84	0.0060	0.3944
Gemini 3.6 Flash	60.88	53.57	0.0052	1.2771
GPT-5.6-Sol	85.25	52.16	0.0050	0.3021
GPT-6-Astra	84.50	50.09	0.0043	0.3408
SpatioLM-InternVL3.5	111.35	40.72	-0.0009	-1.0074
GPT-5.6-Terra	92.48	50.28	0.0033	-0.0035
GPT-5.6-Luna	93.66	50.80	0.0030	-0.0518
Magentic	91.49	50.50	0.0031	0.0566
Qwen 3.7 Plus	93.32	47.84	0.0012	-0.0440
Claude Sonnet 5	106.91	46.88	0.0006	-0.7482
Claude Opus 5	102.13	47.62	0.0002	-0.4403

K Additional Evidence for the Three Findings

K.1 Success Counts and Confidence Intervals

Table 11 reports success counts and 95% Wilson confidence intervals for the nine models and two agents on the 100-task test set. These intervals summarize uncertainty under a binomial task model. GPT-5.6-Sol has the highest observed success rate, but this ranking alone does not establish a reliable difference between methods. Such comparisons require paired analysis on the same tasks.

K.2 Response Formats and Episode Outcomes

We audit the saved trajectories of all nine models and two agents on the 100 test tasks each. Table 12 reports episode outcomes and repeated actions. Stagnation denotes termination after consecutive rotations without sufficient improvement in the best IoU. STOP denotes an explicit stop below the success threshold, and Limit denotes termination at the step limit. Repeat is the percentage of consecutive rotation pairs with the same signed axis and angle within an episode. Repeated actions can be useful, and stopping reasons alone do not explain why a task failed.

We verified, on the saved trajectories, that every logged action was eventually returned as valid JSON, counting only responses that contain output tokens and excluding transmission failures. Ten of the eleven methods returned all actions in valid JSON within at most two retries. GPT-6-Astra had 2,318 rotation actions and one additional content-bearing STOP response; across these 2,319 returns, 90.86% were valid on the first response, 98.45% within two retries, and 100% within six retries.

K.3 Statistical Units and Resampling

We use existing trajectories only. Action rates combine all executed rotations; angle metrics use only effective directions. To estimate uncertainty, we repeatedly sample tasks with replacement and keep all steps from each sampled task. For each sampled task, we keep its four human task runs together. Paired comparisons use the same sampled tasks for both methods. For the additional bootstrap analyses in this section, we use 10,000 resamples and report the 2.5th and 97.5th percentiles as the 95% confidence interval.

Let Δs and Δe denote changes in IoU and pose error. UC increases IoU and reduces pose error; DC decreases both; UF increases both; DF decreases IoU and increases pose error. Changes within 10^{-6} for IoU or 10^{-4} degrees for pose error are treated as neutral.

K.4 Direction and Angle Diagnostics

Human exceeds Sol in Effective Direction Rate by 25.16 percentage points and has a 3.50° smaller Angle Gap (Table 14). Both paired 95% intervals exclude zero. This supports a difference in observed actions on the same 100 tasks, but does not directly measure the models' internal pose estimates.

The data contain 6,487 Human rotations and 2,366 Sol rotations. Longer episodes contribute more actions, so we resample whole tasks to keep related steps together.

K.5 Input Ablations and Fine-Grained Control

We compare four inputs on the same 30 tasks: full input (A), no auxiliary view (B), no history (C), and neither (D). History includes earlier observations, responses, and actions. Removing history keeps the current observation and instructions, but removes all prior context; it does not isolate a single reasoning mechanism.

Without history, 30° rotations rise from 443/706 (62.75%) to 598/618 (96.76%), while 1° rotations fall from 86/706 (12.18%) to 1/618 (0.16%). Thus, fine adjustments nearly disappear. Full input also improves success and Final IoU over no history, with paired intervals above zero (Table 16). The Final IoU interval for A minus B includes zero. Removing both inputs gives two successes, versus zero without history alone; these 30 runs do not explain this reversal.

Table 10: Spearman rank correlations (ρ) of selected rotation angles with pre-action projection IoU (IoU–angle) and 3D pose error (Pose–angle) across the 100 test tasks, sorted by Final IoU. Brackets report 95% bootstrap confidence intervals.

Method	IoU–angle ρ (95% CI)	Pose–angle ρ (95% CI)
Human	$\mathbf{-0.696}$ $[-0.763, -0.633]$	$\mathbf{0.751}$ $[0.675, 0.828]$
ReAct	-0.570 $[-0.674, -0.453]$	0.320 $[0.108, 0.512]$
Gemini 3.6 Flash	-0.562 $[-0.633, -0.480]$	0.618 $[0.496, 0.709]$
GPT-5.6-Sol	-0.536 $[-0.617, -0.443]$	0.368 $[0.166, 0.532]$
GPT-6-Astra	-0.533 $[-0.621, -0.427]$	0.452 $[0.266, 0.607]$
GPT-5.6-Terra	-0.438 $[-0.526, -0.335]$	0.254 $[0.056, 0.440]$
GPT-5.6-Luna	-0.471 $[-0.561, -0.373]$	0.280 $[0.088, 0.470]$
Magentic	-0.368 $[-0.493, -0.215]$	0.250 $[0.066, 0.403]$
Qwen 3.7 Plus	-0.383 $[-0.495, -0.251]$	0.380 $[0.232, 0.514]$
Claude Sonnet 5	-0.255 $[-0.365, -0.140]$	0.154 $[-0.021, 0.309]$
Claude Opus 5	-0.091 $[-0.229, 0.044]$	0.204 $[0.043, 0.354]$
SpatioLM	0.127 $[-0.063, 0.309]$	-0.142 $[-0.268, 0.005]$

Table 11: Success counts and 95% Wilson confidence intervals for models and agents on the 100 test tasks. Rows follow the order in Table 1.

Method	Successes/tasks	Success (%)	95% CI (%)
ReAct	16/100	16.00	[10.10, 24.42]
Gemini 3.6 Flash	10/100	10.00	[5.52, 17.44]
GPT-5.6-Sol	17/100	17.00	[10.89, 25.55]
GPT-6-Astra	10/100	10.00	[5.52, 17.44]
GPT-5.6-Terra	8/100	8.00	[4.11, 15.00]
GPT-5.6-Luna	4/100	4.00	[1.57, 9.84]
Magentic	2/100	2.00	[0.55, 7.00]
Qwen 3.7 Plus	0/100	0.00	[0.00, 3.70]
Claude Sonnet 5	0/100	0.00	[0.00, 3.70]
Claude Opus 5	0/100	0.00	[0.00, 3.70]
SpatioLM-InternVL3.5	0/100	0.00	[0.00, 3.70]

K.6 Comparing Associations and Shared-State Actions

To compare the two correlations in Figure 4, we compute the difference in their absolute values:

$$\Delta_\rho = |\rho(\theta, s_{\text{pre}})| - |\rho(\theta, e_{\text{pre}})|. \tag{14}$$

Positive values indicate a stronger association with IoU; negative values indicate a stronger association with pose error. We recompute both correlations on the same resampled tasks and obtain a confidence interval for their difference.

For GPT-5.6-Sol, $\Delta_\rho = 0.169$ (95% CI: $[0.020, 0.330]$). For Human, $\Delta_\rho = -0.055$ (95% CI: $[-0.084, -0.023]$). These differences use unrounded correlations. Both intervals exclude zero, supporting a stronger association with IoU for Sol and with pose error for Human.

Shared-state matching and weighting. Table 8 contains 211 participant–task–state pairs from 51 tasks. Within a task, two poses match if their rotation distance is below 10^{-4} degrees. We exclude initial states and repeated Human visits to a matched state, then pair the Human action with Sol's first visit. The four test-set passes contribute 52, 20, 64, and 75 pairs. A Sol action may appear in several participant pairs. Each pair has equal weight, and task resampling keeps related pairs together.

Matching controls the current pose, but earlier observations and actions may differ. Figure 5 reports completed rotations and cumulative direct coverage before each matched action, averaged over the 211 pairs. Human reaches these states after 2.40 rotations on average, compared with 4.18 for Sol, with coverage of 42.01% and 46.21%, respectively. These comparisons describe behavior at shared states; they do not establish a specific internal mechanism or a general limitation in 3D reasoning.

Table 12: Episode outcomes and repeated actions on 100 tasks per method. The four outcome columns sum to 100. Repeat is measured over consecutive rotation pairs.

Method	Success	Stagnation	STOP	Limit	Repeat (%)
Human-independent model methods					
GPT-5.6-Sol	17	78	3	2	53.35
ReAct	16	75	5	4	49.41
Gemini 3.6 Flash	10	83	2	5	23.90
GPT-6-Astra	10	85	1	4	53.65
GPT-5.6-Terra	8	84	2	6	53.90
GPT-5.6-Luna	4	91	3	2	52.13
Magentic	2	44	52	2	23.32
Qwen 3.7 Plus	0	99	1	0	38.34
Claude Sonnet 5	0	95	5	0	23.60
Claude Opus 5	0	95	3	2	19.16
SpatioLM	0	100	0	0	100.00

Table 13: Distribution of successfully returned actions by the number of retries needed to obtain valid JSON, over saved trajectories. Only responses that contain output tokens count as valid returns; transmission failures (no returned tokens) are excluded. "retries$\leq$ k" is the cumulative share of actions produced within k retries (i.e. on the $k{+}1$-th content-bearing response).

Method	Legal JSON	retries=0 (%)	$\leq$2 (%)	$\leq$6 (%)
GPT-5.6-Sol	2369	99.87	100.00	100.00
GPT-5.6-Luna	2432	92.27	100.00	100.00
GPT-5.6-Terra	2293	100.00	100.00	100.00
Gemini 3.6 Flash	2470	99.92	100.00	100.00
Claude Opus 5	2212	99.55	100.00	100.00
Claude Sonnet 5	1944	100.00	100.00	100.00
Qwen 3.7 Plus	2104	99.95	100.00	100.00
ReAct	2467	99.80	100.00	100.00
Magentic	1652	99.52	100.00	100.00
SpatioLM	1881	100.00	100.00	100.00
GPT-6-Astra	2319	90.86	98.45	100.00

K.7 Oracle Action Decomposition

To separate direction and angle selection, we evaluate the 2,366 states visited by GPT-5.6-Sol on the 100-task test set. For each state, we enumerate all 24 legal actions and compute the successor pose error of each action. Here, the best action is defined as the legal action that minimizes the symmetry-adjusted successor pose error. The direction oracle keeps Sol's selected angle and chooses the best rotation axis and sign; the angle oracle keeps Sol's selected direction and chooses the best legal angle; the joint oracle chooses the best action without either constraint. Uniform random averages successor pose error over all 24 actions. As shown in Table 17, correcting direction yields a larger one-step improvement than correcting angle, while correcting both yields the lowest expected error. These results indicate that GPT-5.6-Sol still has substantial room for improvement in both rotation-direction and rotation-angle selection.

Table 14: Action metrics on 100 tasks, with paired 95% bootstrap intervals. Angle Gap uses effective directions only. Differences are Human minus Sol; direction differences are in percentage points.

Method/comparison	Effective Direction Rate (%)	Angle Gap ($^\circ$)
Human	83.91 [82.63, 85.22]	2.99 [2.70, 3.30]
GPT-5.6-Sol	58.75 [56.62, 60.87]	6.49 [5.38, 7.64]
Human minus Sol	25.16 [22.79, 27.56]	-3.50 [-4.60, -2.42]

Table 15: Successes on 30 tasks, with Wilson 95% intervals. The last four columns count rotations at each angle.

Input	Successes	Success (%; 95% CI)	1°	5°	10°	30°
A: Full	5/30	16.67 [7.34, 33.56]	86	53	124	443
B: No auxiliary	1/30	3.33 [0.59, 16.67]	24	22	48	584
C: No history	0/30	0.00 [0.00, 11.35]	1	3	16	598
D: Neither	2/30	6.67 [1.85, 21.32]	8	8	5	597

Table 16: Full input minus each ablation on the same 30 tasks, with paired 95% bootstrap intervals. Success differences are in percentage points.

Comparison	Success difference	Mean Final IoU difference
A minus B	13.33 [0.00, 30.00]	0.0166 [−0.0598, 0.0961]
A minus C	16.67 [3.33, 30.00]	0.0834 [0.0255, 0.1477]
A minus D	10.00 [0.00, 20.00]	0.0568 [0.0138, 0.1107]

K.8 Cross-Geometry Comparison on Matched 30-Task Sets

To reduce reliance on semantic shortcuts and encourage visual-spatial reasoning, RoTA-Bench synthesizes 3D objects from 2D vector icons rather than drawing directly from common 3D CAD repositories, where familiar shapes may trigger pre-trained priors. To examine the effect of this design choice, we compare the original 30-task icon-derived ablation set with another 30-task set constructed from 30 distinct polygonal meshes from the Poly Haven CC0 asset collection. These meshes are based on real-world object categories and may therefore resemble objects encountered during model pretraining.

The two task sets have exactly matched reference-path length distributions. Both use the same point-splat renderer, auxiliary depth view, reversible 6–16-step initialization with 30° rotations, 24-action space, full interaction history, 48-decision budget, success threshold, and termination rules. As shown in Table 18, GPT-5.6-Sol achieves a substantially higher mean final IoU on the Poly Haven mesh set and a slightly higher success rate than on the icon-derived set. In this matched-distribution comparison, tasks based on real-world object geometries yield higher observed performance than tasks involving geometries generated from 2D vector icons. Although the two sets are not paired task-by-task, the comparison suggests that the familiarity of geometric appearance may affect model performance.

L Implementation and Evaluation Settings

Table 19 lists local software versions, hardware specifications, LLM hyperparameters, and evaluation settings.

SpatioLM-Understanding-InternVL3.5 (Wu et al., 2026) is evaluated using self-hosted local inference and incurs no commercial API token fees. Inference is hosted on an on-premise server equipped with NVIDIA A100-PCIE-40GB GPUs, dual Intel Xeon Silver 4314 processors (32 physical cores, 64 execution threads @ 2.40 GHz), and 256 GB of system RAM, running Ubuntu Linux with NVIDIA driver 615.71.09, CUDA 13.0, and PyTorch 2.10.0+cu130. The model is served through an internal OpenAI-compatible HTTP interface using the official `InternVL3RChatModel` architecture in `torch.bfloat16` precision. Generation uses deterministic greedy decoding (`do_sample=False`, corresponding to temperature 0), matching the zero-temperature setting of closed-source models, with a context window upper limit of 8,192 output tokens.

Table 20 estimates API costs from recorded token usage and published list prices below, including Magentic's internal workflow and the three input ablation conditions. We use published list rates

Table 17: Oracle decomposition on 2,366 GPT-5.6-Sol states. Successor pose errors and improvements are averaged over states. For Uniform random, the successor error is the average over all 24 legal actions in each state. Oracle actions are selected by minimizing the symmetry-adjusted successor pose error.

Policy	Successor error (°)	Improvement (°)
Sol actual action	91.99	0.00
Uniform random	93.60	-1.61
Oracle direction	76.38	15.61
Oracle angle	86.29	5.70
Joint oracle	72.00	19.99

Table 18: GPT-5.6-Sol on two 30-task sets with exactly matched reference-path length distributions. The icon-derived set is the original ablation subset; the comparison set uses 30 distinct Poly Haven CC0 meshes based on real-world object categories.

Task set	Success	Mean final IoU	Mean steps
Icon-derived tasks	5/30 (16.7%)	0.5656	23.53
Poly Haven meshes	6/30 (20.0%)	0.7179	22.17

Table 19: Software versions, hardware specifications, model hyperparameters, agent configurations, and interaction settings used for evaluation.

Component	Setting
Rendering	Python 3.11.14; NumPy 2.3.3; OpenCV 4.13.0; PyTorch 2.10.0+cu130; CUDA 13.0
Supporting packages	CairoSVG 2.8.2; cairocffi 1.7.1; Pillow 12.1.1; Numba 0.64.0
API and agent packages	OpenAI 2.48.0; HTTPX 0.28.1; Microsoft Agent Framework core 1.12.1, OpenAI 1.11.0, orchestrations 1.0.1
Local GPU server	8 × NVIDIA A100-PCIE-40GB; 2 × Intel Xeon Silver 4314 (32 cores / 64 threads); 256 GB RAM; Driver 615.71.09
Open-source model serving	SpatioLM-Understanding-InternVL3.5: `bfloat16`, greedy decoding (`do_sample=False`), official architecture; local HTTP server
Response format	JSON; no streaming
LLM hyperparameters	Temperature 0; reasoning effort medium and seed 42 where supported; maximum 8,192 output tokens
Request timeout	1,500 seconds
History and observations	Full interaction history; three 320 × 320 views in a 972 × 320 panel, separated by two 6-pixel gaps
Termination	Success at IoU ≥ 0.9, STOP, 48 decisions, or 12 consecutive rotations without a new best IoU
Agent tools	Python execution and image viewing; at most four tool calls in total per environment step
Magentic orchestration	One solver and one manager; at most four rounds and one reset
Request retries	Up to two outer retries; backoff parameter 1.6 seconds
Python tool limits	300 seconds per call; 4 GiB memory; 64 MiB workspace; up to four returned images
Agent workspace	Task images and derived files; no internal poses or reference paths

for GPT-6-Astra ($10.00 / $50.00 per million tokens) and GPT-5.6 ($5.00 / $30.00), base rates for Gemini and Qwen, and announced standard rates for Claude.[1]

[1]Official pricing sources: `https://developers.openai.com/api/docs/pricing/`, `https://ai.google.dev/gemini-api/docs/pricing`, `https://www.anthropic.com/news/claude-sonnet-5`, `https://www.anthropic.com/news/claude-opus-5`, and `https://www.alibabacloud.com/help/en/model-studio/model-pricing`.

Table 20: Published API prices, recorded token counts, and estimated costs. Main evaluations use 100 tasks; the three GPT-5.6-Sol input ablations use the same 30-task subset. Rates are USD per million tokens using uncached input pricing; costs are in USD.

Condition	Input/output rate	Input tokens	Output tokens	Total cost (USD)
GPT-5.6-Sol	5.00 / 30.00	28,883,151	853,753	170.0283
GPT-5.6-Terra	2.50 / 15.00	27,905,743	1,062,892	85.7077
GPT-5.6-Luna	1.00 / 6.00	22,630,532	1,087,401	29.1549
GPT-6-Astra	10.00 / 50.00	29,975,049	1,187,433	359.1221
Gemini 3.6 Flash	1.50 / 7.50	45,392,248	4,555,620	102.2555
Claude Opus 5	5.00 / 25.00	17,014,514	700,389	102.5823
Claude Sonnet 5	3.00 / 15.00	19,556,374	722,105	69.5007
Qwen 3.7 Plus	0.40 / 1.60	10,719,193	901,152	5.7295
GPT-5.6-Sol + ReAct	5.00 / 30.00	28,836,103	1,024,921	174.9281
GPT-5.6-Sol + Magentic	5.00 / 30.00	49,032,175	2,208,867	311.4269
Sol: no aux. view	5.00 / 30.00	6,621,163	208,670	39.3659
Sol: no history	5.00 / 30.00	2,065,970	305,183	19.4853
Sol: neither input	5.00 / 30.00	691,672	310,963	12.7873
Total	—	289,323,887	15,129,349	1,482.0747

M Prompt Examples

The following is a prompt example for single_00627 (Figures 7–9). We call the depth-enhanced auxiliary observation the "oblique view" because it exposes depth and occlusion cues that are absent from the binary projection, similar in purpose to an oblique depiction of a 3D shape. The name describes its visual role rather than a camera rotation: the view is rendered with the same camera direction and object orientation as the current projection, while depth-dependent shading provides the additional cues. Line breaks below are for readability.

System prompt.

> You are playing a deterministic turn-based 3D-to-2D alignment game. Rotate only the CURRENT object so its projection matches TARGET within the step budget. Each observation provides images and the current task feedback. Each panel is one image ordered left-to-right as TARGET projection | CURRENT oblique view | CURRENT projection. The oblique view is an auxiliary 3D pose cue; compare TARGET with CURRENT projection for alignment. Rotations use observer/camera coordinates: PITCH_POS=pitch upward, PITCH_NEG=pitch downward, YAW_POS=turn toward observer right, YAW_NEG=turn toward observer left, ROLL_POS=clockwise in the image plane, ROLL_NEG=counter-clockwise. Allowed action tokens: PITCH_POS, PITCH_NEG, YAW_POS, YAW_NEG, ROLL_POS, ROLL_NEG, STOP. Allowed angles: 30, 10, 5, 1. History is chronological. Each historical panel is the state before the action written immediately beside it. The environment ends automatically when the requirement is met. Use STOP only when no actionable improvement idea remains; do not use STOP merely because the silhouettes appear aligned. Return exactly one JSON object and no extra text: `{"cot":"brief current-step reasoning","type":"TOKEN","angle":ANGLE}`. The type and angle must come from the allowed sets.

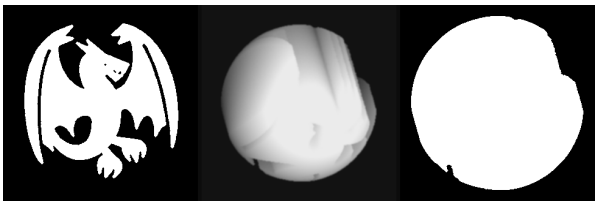
N Success and Failure Examples

We select the GPT-5.6-Sol run with the fewest executed rotations separately among successful and unsuccessful tasks, breaking ties by ascending task identifier. The selected runs are single_00270 (8 rotations) and single_00039 (12 rotations). These are short illustrative cases rather than representative samples of all model behavior.

In the successful case (Figure 10), the model reverses a yaw after it reduces alignment, then changes the roll and makes smaller pitch and yaw adjustments. IoU rises from 0.624 to 0.977. The sequence illustrates recovery from an unhelpful action followed by refinement across axes. In the failed case

User input.

```
Current turn: 1/48.
Used steps: 0.
Remaining steps: 48.
The task ends automatically when the requirement is met.
If it continues, the current result has not yet met the requirement.
Current state to act on now:
```



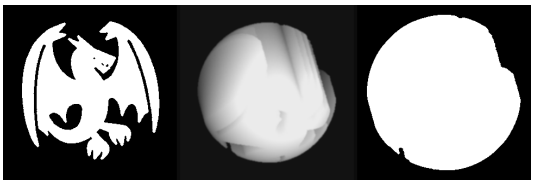
Example response.

```
{"cot":"...","type":"YAW_POS","angle":30}
```

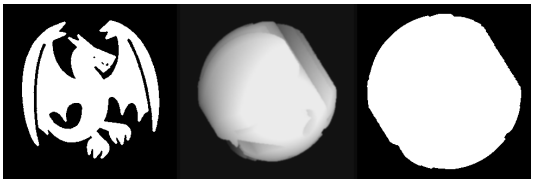
Figure 7: First-turn prompt for single_00627, showing the initial observation, user input, and an example JSON response.

User input.

```
Current turn: 2/48.
Used steps: 1.
Remaining steps: 47.
The task ends automatically when the requirement is met.
If it continues, the current result has not yet met the requirement.
Historical state 1; action after this state:
step=1 action=YAW_POS angle=30 cot=...
```



```
Current state to act on now:
```



Example response.

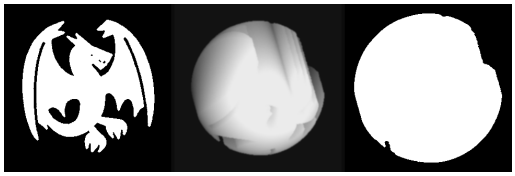
```
{"cot":"...","type":"PITCH_NEG","angle":10}
```

Figure 8: Second-turn prompt for single_00627, showing the first observation and action–COT record, current observation, user input, and an example JSON response.

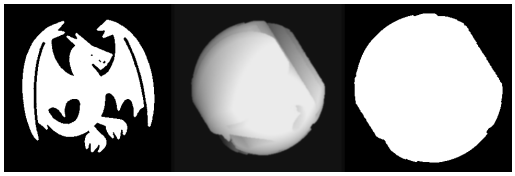
(Figure 11), all 12 actions are 30° rotations in the same yaw direction, completing a full turn and returning to the initial alignment (IoU 0.450). The model repeatedly proposes revealing a hidden face without changing its action strategy; no rotation improves on the initial IoU, so stagnation stopping ends the run.

User input.

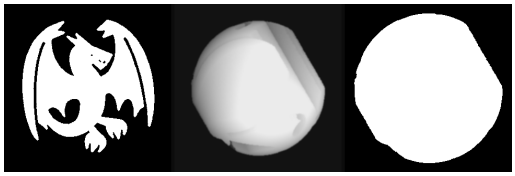
```
Current turn: 3/48.
Used steps: 2.
Remaining steps: 46.
The task ends automatically when the requirement is met.
If it continues, the current result has not yet met the requirement.
Historical state 1; action after this state:
step=1 action=YAW_POS angle=30 cot=...
```



```
Historical state 2; action after this state:
step=2 action=PITCH_NEG angle=10 cot=...
```



```
Current state to act on now:
```

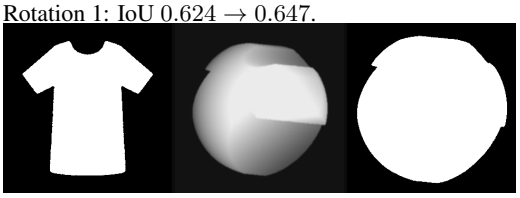


Example response.

```
{"cot":"...","type":"ROLL_POS","angle":5}
```

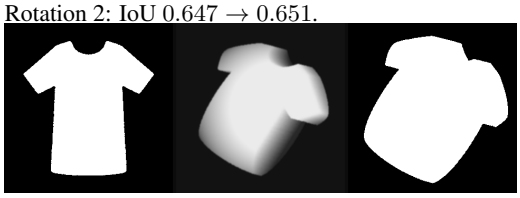
Figure 9: Third-turn prompt for single_00627, showing both earlier observations and action–COT records, current observation, user input, and an example JSON response.

Rotation 1: IoU 0.624 → 0.647.



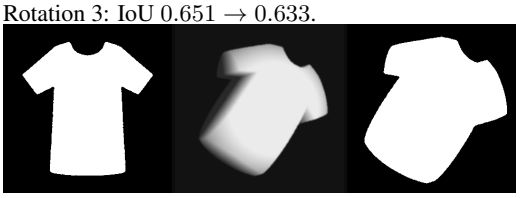
```
{
  "cot": "The target is a broad
front-facing shirt silhouette,
while the
current projection is nearly
circular.
The oblique cue suggests the
garment is
turned substantially away from the
target, so begin with a moderate
yaw
toward the observer's right.",
  "type": "YAW_POS",
  "angle": 30
}
```

Rotation 2: IoU 0.647 → 0.651.



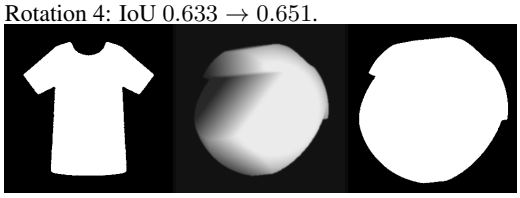
```
{
  "cot": "The first yaw improved
  the
silhouette from circular to a
recognizable garment, but it
remains
strongly oblique and narrow.
Continue
yawing toward the observer's right
to
bring the front projection closer
to the
broad target.",
  "type": "YAW_POS",
  "angle": 30
}
```

Rotation 3: IoU 0.651 → 0.633.



```
{
  "cot": "The projection is
  improving
with yaw but remains strongly
oblique
and diagonally oriented, so
continue
turning toward the observer's
right.",
  "type": "YAW_POS",
  "angle": 30
}
```

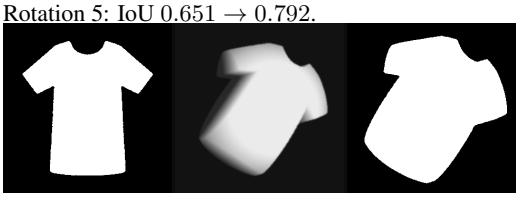
Rotation 4: IoU 0.633 → 0.651.



```
{
  "cot": "The projection has become
circular again after continued yaw,
suggesting the useful yaw range was
passed. Reverse yaw to recover a
broader
garment silhouette.",
  "type": "YAW_NEG",
  "angle": 30
}
```

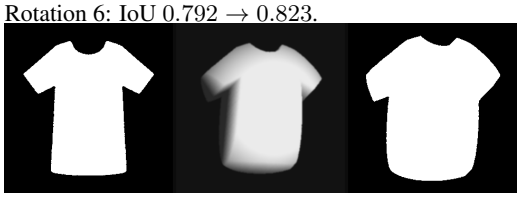
Figure 10: GPT-5.6-Sol's successful trajectory on single_00270: rotation steps 1–4 of 8, with pre-action observations and original JSON responses. Panels show the target projection, auxiliary view, and current projection from left to right; IoU annotations were not provided to the model.

Rotation 5: IoU 0.651 → 0.792.



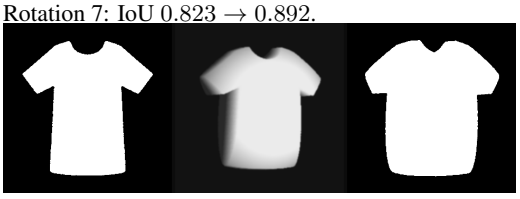

```
{
  "cot": "The garment is still
diagonally oriented, with its long
axis
slanting down-left instead of
vertically
like the target. Apply a moderate
counter-clockwise roll to correct
the
image-plane orientation.",
  "type": "ROLL_NEG",
  "angle": 30
}
```


Rotation 6: IoU 0.792 → 0.823.



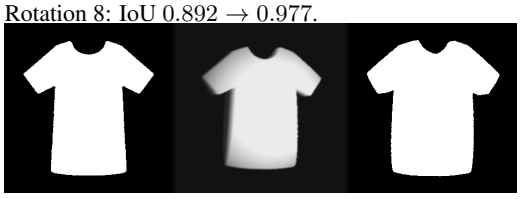

```
{
  "cot": "The projection is nearly
front-facing and correctly rolled.
Its
vertical extent is slightly too
tall,
with the shoulders too high and the
torso bottom too low, so apply a
small
downward pitch to compress the
vertical
silhouette toward the target.",
  "type": "PITCH_NEG",
  "angle": 10
}
```


Rotation 7: IoU 0.823 → 0.892.




```
{
  "cot": "The current silhouette is
  very
close, but the torso is slightly
too
wide and the sleeve undersides sit
lower
than the target. A small yaw
adjustment
is the most actionable refinement
while
preserving the nearly correct
vertical
scale.",
  "type": "YAW_NEG",
  "angle": 5
}
```


Rotation 8: IoU 0.892 → 0.977.




```
{
  "cot": "The projection is very
  close,
but the current torso remains
slightly
wider than the target and the
sleeve-to-torso transitions sit a
little
low. Continue the same small yaw
adjustment to narrow the torso
while
preserving the overall alignment.",
  "type": "YAW_NEG",
  "angle": 5
}
```


Final state after rotation 8: IoU 0.977.

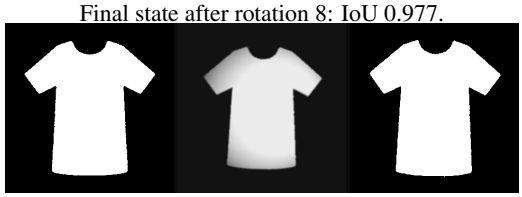
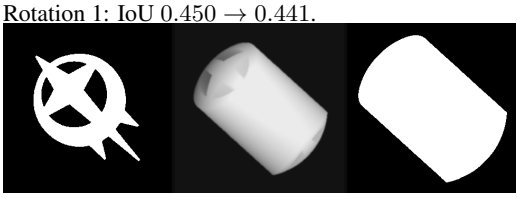


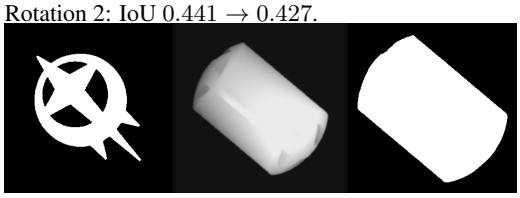
Figure 10: GPT-5.6-Sol's successful trajectory on single_00270: rotation steps 5–8 of 8 and the final state, with pre-action observations and original JSON responses. Panels show the target projection, auxiliary view, and current projection from left to right; IoU annotations were not provided to the model.

Rotation 1: IoU 0.450 → 0.441.



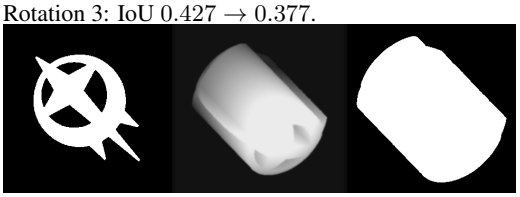

```
{
  "cot": "The target is an end-on
  view
while the current projection is a
side
view. The object's axis projects
down-right, so yawing left should
reduce
its rightward projection.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 2: IoU 0.441 → 0.427.



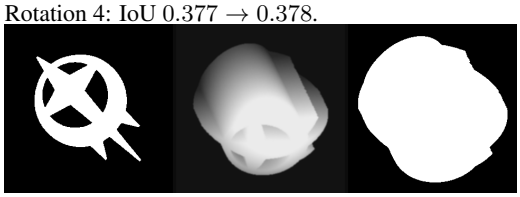

```
{
  "cot": "The current projection is
still a side view with the circular
face
hidden. Continue yawing left to
bring
the object's axis toward the viewer
and
expose the detailed end face.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 3: IoU 0.427 → 0.377.




```
{
  "cot": "The current projection is
still a side-on cylinder with the
detailed end face hidden. Continue
yawing left to bring the axis
toward the
viewer and reveal the target
end-face
pattern.",
  "type": "YAW_NEG",
  "angle": 30
}
```

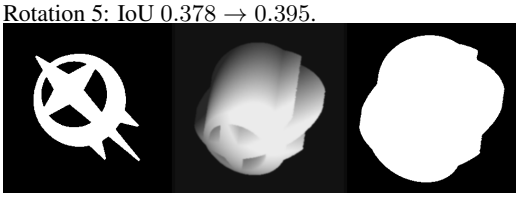

Rotation 4: IoU 0.377 → 0.378.




```
{
  "cot": "The projection is now
  nearly
end-on, but the target's internal
end-face pattern is still not
visible.
Continue yawing left to bring the
patterned face fully toward the
viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```

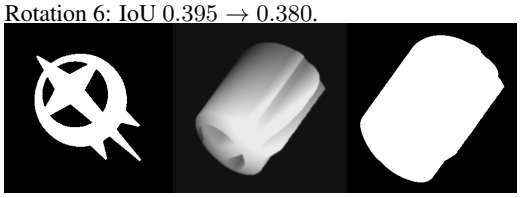

Figure 11: GPT-5.6-Sol's unsuccessful trajectory on single_00039: rotation steps 1–4 of 12, with pre-action observations and original JSON responses. Panels show the target projection, auxiliary view, and current projection from left to right; IoU annotations were not provided to the model.

Rotation 5: IoU 0.378 → 0.395.



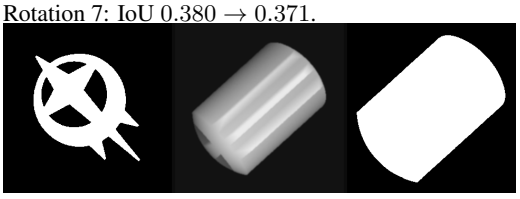

```
{
  "cot": "The projection is nearly
face-on but the patterned end face
is
still obscured. Continue yawing
left to
expose the target pattern.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 6: IoU 0.395 → 0.380.



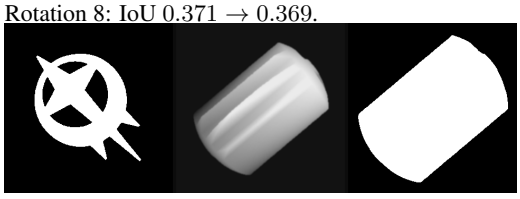

```
{
  "cot": "The patterned end face
  remains
hidden in the current projection.
Continue yawing left to bring that
face
toward the viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 7: IoU 0.380 → 0.371.




```
{
  "cot": "The patterned face is
  still
not visible and the current
projection
remains side-on. Continue yawing
left to
rotate the patterned end toward the
viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```

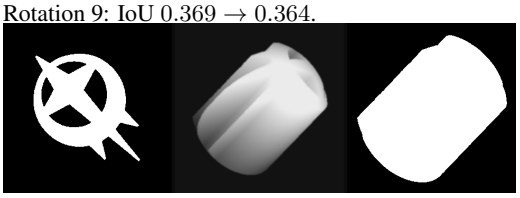

Rotation 8: IoU 0.371 → 0.369.




```
{
  "cot": "The patterned end face is
still not visible and the current
projection remains side-on.
Continue
yawing left to rotate the patterned
face
toward the viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```

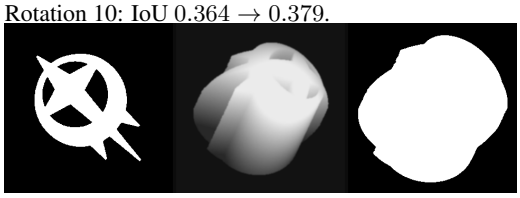

Figure 11: GPT-5.6-Sol's unsuccessful trajectory on single_00039: rotation steps 5–8 of 12, with pre-action observations and original JSON responses. Panels show the target projection, auxiliary view, and current projection from left to right; IoU annotations were not provided to the model.

Rotation 9: IoU 0.369 → 0.364.



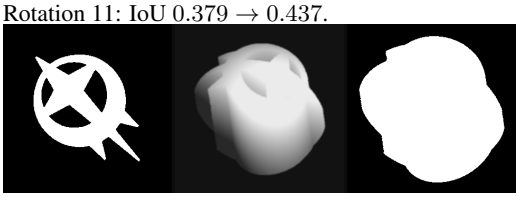

```
{
  "cot": "The current projection is
still side-on and lacks the
target's
patterned end face. Continue yawing
left
to bring the end face toward the
viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 10: IoU 0.364 → 0.379.



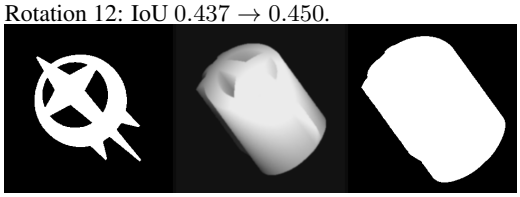

```
{
  "cot": "The patterned end face is
beginning to face the viewer, but
the
current projection still lacks the
target openings. Continue yawing
left to
expose and align the patterned
face.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 11: IoU 0.379 → 0.437.




```
{
  "cot": "The current silhouette is
close to the target, and the
patterned
end face is visible obliquely but
not
yet in the projection. Continue
yawing
left to bring that face fully
toward the
viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Rotation 12: IoU 0.437 → 0.450.




```
{
  "cot": "The patterned face is
  still
not visible in the current
projection,
so continue yawing left to bring it
fully toward the viewer.",
  "type": "YAW_NEG",
  "angle": 30
}
```


Final state after rotation 12: IoU 0.450.

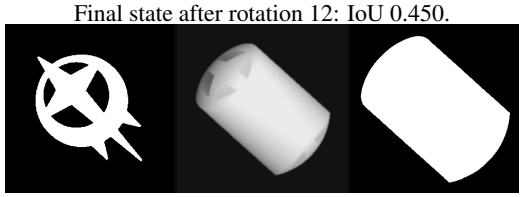


Figure 11: GPT-5.6-Sol's unsuccessful trajectory on single_00039: rotation steps 9–12 of 12 and the final state, with pre-action observations and original JSON responses. Panels show the target projection, auxiliary view, and current projection from left to right; IoU annotations were not provided to the model.