SCENEACTBENCH: CAN AGENTS ACT ON THE 3D SCENES THEY SEE?

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language model (VLM) agents increasingly use tools to act on 3D scenes rather than only describe them. Existing 3D benchmarks score textual responses or single-object operations, leaving agent action on complete multi-object 3D scenes under-evaluated. We present **SceneActBench**, a benchmark for visually conditioned action across five 3D tasks under a unified agent–environment loop. Given PNG images or sampled video frames and, where applicable, supplied 3D assets, an agent acts on a 3D environment. We evaluate each final output against hidden ground truth with task-specific geometric metrics. SceneActBench comprises five tasks built from 210 source instances, yielding 520 task cases including paired input conditions. Every task runs through one fixed agent loop to keep the comparison fair. Across eleven proprietary VLM configurations, Overall scores span 38.6–50.2, and none performs consistently well across tasks. We further analyse where and how failures manifest.

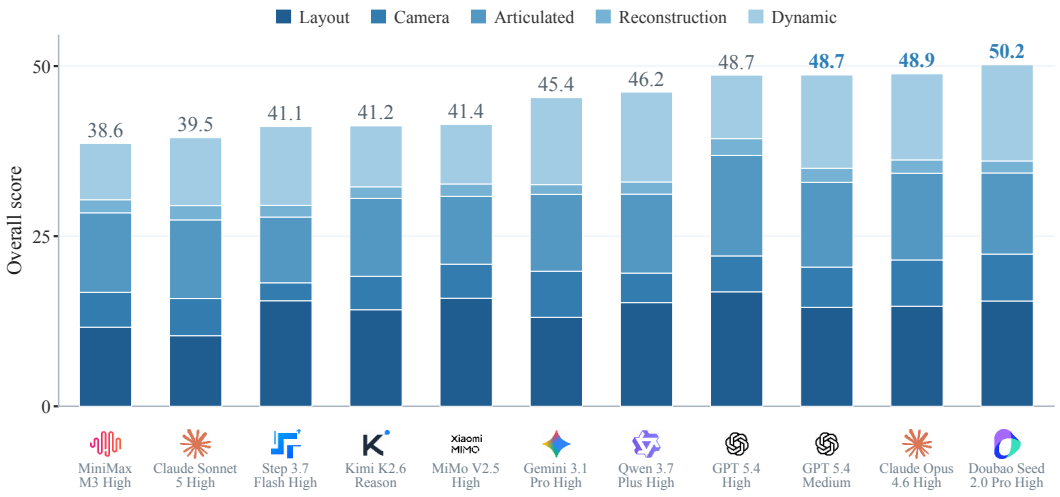


Figure 1: **SceneActBench leaderboard.** Overall scores for eleven configurations (38.6–50.2), decomposed into five equal-weight task contributions; vendor marks accompany configuration labels.

1 INTRODUCTION

Recent vision-language model (VLM) agents can act on 3D scenes through tools and code (Wang et al., 2024a; Hu et al., 2024; Sun et al., 2025; Doris et al., 2026; Huang et al., 2023). Acting on a scene, rather than describing it, is a stronger test of an agent's 3D understanding. Existing 3D benchmarks measure only part of this problem (Table 1); Appendix B provides the extended comparison. Many pose 3D visual question answering (VQA) (Azuma et al., 2022; Ma et al., 2023; Wang et al., 2025). They score text answers to scans or images without changing the scene. Others evaluate agents acting in 3D environments (Gu et al., 2025; Gao et al., 2026; Chi et al., 2026; Yang et al., 2025). These benchmarks typically cover a single object or static edit, and some report only task success. No benchmark asks one agent to act on a full scene of many objects. This matters in

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.

Table 1: **Benchmark coverage.** Coverage of five 3D capabilities and three evaluation properties (✓ full, ✓ partial, ✗ none).

Benchmark	3D capability					Evaluation		
	Spatial grounding	Egocentric spatial reasoning	Kinematic reasoning	Shape imagination	Dynamic reasoning	Geom. score	Multi-obj.	Inter-active
3D question answering								
ScanQA (Azuma et al., 2022)	✓	✗	✗	✗	✗	✗	✓	✗
SQA3D (Ma et al., 2023)	✓	✓	✗	✗	✗	✗	✓	✗
Spatial457 (Wang et al., 2025)	✓	✗	✗	✗	✓	✗	✓	✗
Code-driven 3D and embodied agents								
BlenderGym (Gu et al., 2025)	✓	✗	✗	✓	✗	✓	✓	✓
3DCodeBench (Gao et al., 2026)	✗	✗	✗	✓	✗	✓	✗	✓
GameDevBench (Chi et al., 2026)	✗	✗	✓	✗	✓	✗	✓	✓
EmbodiedBench (Yang et al., 2025)	✓	✗	✓	✗	✓	✗	✓	✓
SceneActBench	✓	✓	✓	✓	✓	✓	✓	✓

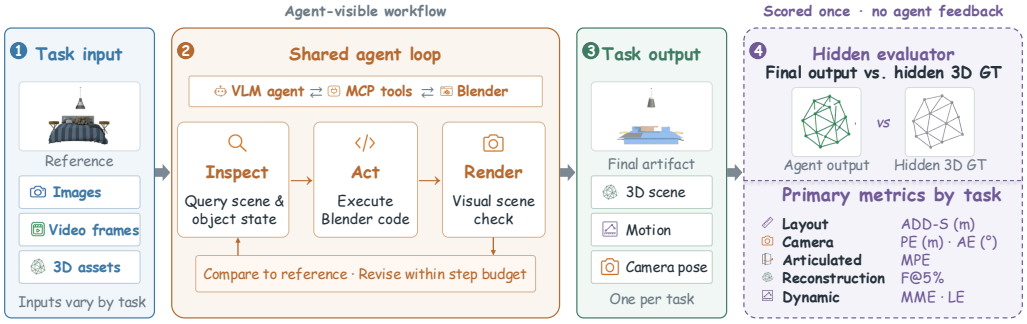


Figure 2: **Shared agent–environment loop.** The VLM receives standardised visual evidence and applicable GLB assets, then inspects, edits, renders, and revises through one MCP interface under a task budget. A hidden evaluator scores the final GLB or JSON output once without returning feedback.

practical 3D applications that require coordinated action across multiple objects. The question is therefore direct: *Can an agent that sees a scene act on a 3D environment to match it?*

We address this question with **SceneActBench**, an executable benchmark for visually conditioned 3D action. The agent observes images or sampled video frames and acts on a 3D scene to match the reference. SceneActBench draws on 210 source instances: 100 furnished rooms, 100 articulated objects, and 10 multi-object dynamic scenes. The rooms contain 3–7 objects from 27 furniture categories, and the dynamic set spans nine motion types (Table 2). Its five tasks are **Layout**, **Camera**, **Articulated**, **Reconstruction**, and **Dynamic**. To ensure a fair comparison, we developed a standardised evaluation harness with a fixed agent loop, as shown in Figure 2.

We evaluate eleven proprietary VLM configurations. Overall scores range from 38.6 to 50.2, and the stacked task contributions reveal different strengths among the leading configurations (Figure 1). This paper makes three contributions. First, we formulate visually conditioned 3D action as an executable evaluation problem and score final outputs against hidden 3D ground truth. Second, we introduce five tasks built from 210 source instances to test spatial grounding, egocentric spatial reasoning, kinematic reasoning, shape imagination, and dynamic reasoning under one fixed agent loop. Third, we benchmark eleven configurations and use case-level diagnostics to characterise where and how failures manifest beyond aggregate scores. The remainder presents the benchmark, evaluation, and failure analysis.

2 BENCHMARK

SceneActBench evaluates whether an agent can convert visual evidence into executable 3D outputs. The agent receives **PNG** images or sampled video frames and controls Blender in headless mode through the shared tool interface rather than a graphical interface. Depending on the task, the output

Table 2: **SceneActBench task specification.** Tasks share the loop in Figure 2; hidden ground truth is used only for scoring. Layout and Dynamic include paired input conditions. Budget denotes maximum agent steps and n source instances per condition and configuration.

Task	Source	Visual evidence	Initial state / assets	Agent output	Primary metric	Budget	n
Layout	3D-FRONT	1 or ∼11 calibrated views	N canonical GLBs at origin	Object poses in GLB	ADD-S ↓	30	100
Camera	3D-FRONT	1 view + known FOV	Fixed furnished scene	6-DoF pose (JSON)	PE / AE ↓	30	100
Articulated	S2O ACD	32 ordered frames	Closed, unlabelled GLB	32 GLB scene states	MPE ↓	60	100
Reconstruction	3D-FRONT	∼11 calibrated views	Empty Blender scene	Furnished-scene GLB	F@5% ↑	35	100
Dynamic	Kenney kits	Sampled 144-frame video + camera	Component GLB library	Animated GLB	MME / LE ↓	80	10

is stored in **JSON** or in one or more **GLB** files, which use the binary glTF format to package 3D geometry, materials, object transforms, and animation. Each output is evaluated against hidden 3D ground truth. Across the five tasks and paired input conditions, the 210 source instances yield 520 task cases. An *asset* is a supplied or importable 3D resource; an *object* is an item instantiated in a scene. Appendix C.1 defines the 3D-specific terms used throughout the benchmark, and Appendix C.3 gives the exact evaluator rules and all diagnostic metrics.

2.1 Data Collection

We assemble our benchmark from 3 public sources, each standardised into our own format.

- **Indoor Scenes Dataset.** We draw 100 furnished rooms from 3D-FRONT (Fu et al., 2021), keeping only the M3DLayout (Zhang et al., 2026) intersection where multi-view renders and pose annotations are reliable. Rooms contain 3 to 7 furniture objects across 27 categories, each provided as a **GLB** asset. We render 11 **PNG** views per scene from varied camera positions, recording the camera matrix and angle of view for each as **JSON**. These images serve as references in 3 of the 5 tasks.
- **Articulated Objects Dataset.** From the S2O Articulated Containers Dataset (ACD) (Iliash et al., 2026) we select 100 articulated objects, each provided as a **GLB** asset. Each object has a 32-frame open–close **MP4** video at 16 fps from a fixed viewpoint and one **GLB** mesh per frame as ground truth.
- **Dynamics Dataset.** 10 scenes are built from CC0-licensed low-poly asset kits (Kenney[1]), each containing several independently moving objects. Motion types vary deliberately. No single heuristic can cover all of these, which forces evaluation to be task-agnostic. Each scene provides a library of importable **GLB** assets and a 144-frame **MP4** reference video at 24 fps. The ground truth is an animated **GLB** scene, with per-frame movable-object coordinates, a static layout annotation, and a camera as **JSON**.

Post-processing. We remove answer leakage before evaluation. Indoor assets are centred, reset to neutral yaw, and scaled to their annotated metric size; articulated assets are reoriented to a common frame while joint metadata remain hidden. Anonymous identifiers replace file and node names. NVIDIA Cosmos (Agarwal et al., 2026) produces a paired photo-realistic Dynamic reference with unchanged layout and motion, and the shared interface exposes video inputs only as sampled PNG frames. Appendix C.2 gives the complete construction procedure.

2.2 Layout

Task. Given room images and standardised furniture objects at the origin, the agent sets each object's world position and rotation about the vertical axis (yaw), as shown in Figure 3. The ability under test is spatial grounding. The agent maps how a layout looks in an image to where each object sits and faces in 3D. We evaluate the same 100 scenes with one view and with all ∼11 views.

Metric. The primary score is **ADD-S** (Wen et al., 2024). For predicted object i and candidate ground-truth object j, let $\hat{V}_i \subset \mathbb{R}^3$ denote the sampled vertices of the predicted mesh after applying its final world transform, and let $V_j \subset \mathbb{R}^3$ denote the sampled vertices of that candidate ground-truth

[1] www.kenney.nl, released under Creative Commons Zero.

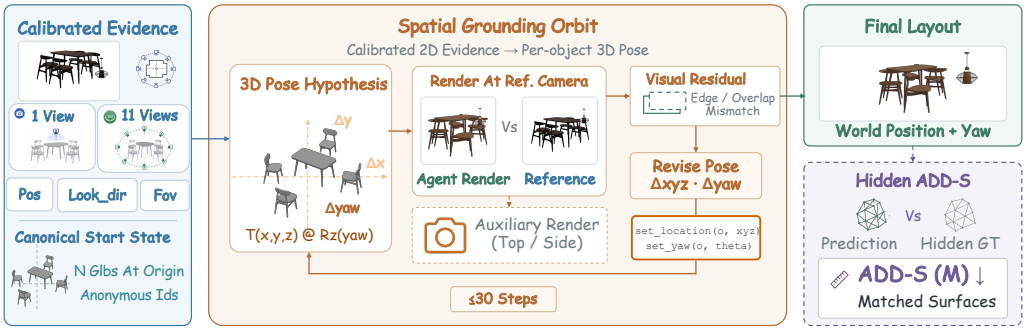


Figure 3: **Layout.** From one or ∼11 calibrated views and canonical furniture objects, the agent iteratively estimates metric position and yaw. Hidden ground truth scores the final layout using **ADD-S**.

mesh after applying its annotated pose. Their directed nearest-neighbour surface distance is

$$d_{\text{surf}}(\hat{V}_i, V_j) = \frac{1}{|\hat{V}_i|} \sum_{\hat{\mathbf{v}} \in \hat{V}_i} \min_{\mathbf{v} \in V_j} \|\hat{\mathbf{v}} - \mathbf{v}\|_2 .$$

Hungarian assignment using the pairwise costs $C_{ij} = d_{\text{surf}}(\hat{V}_i, V_j)$ produces a set $\mathcal{M}$ of matched predicted and ground-truth objects. The scene-level score is

$$\text{ADD-S}_{\text{scene}} = \frac{1}{|\mathcal{M}|} \sum_{(i,j) \in \mathcal{M}} d_{\text{surf}}(\hat{V}_i, V_j). \tag{1}$$

A lower score indicates that the predicted object surfaces are closer to their target positions and orientations. The primary score averages over matched objects, while separate count-sensitive audits account for missing objects. Appendix C.3 specifies the point-sampling procedure and the treatment of invalid outputs.

2.3 Camera

Task. The agent sees an arranged scene and the camera angle of view, but not the camera extrinsics, which specify the camera's 3D position and orientation. The ability under test is egocentric spatial reasoning. The agent infers where the observer stood from what the scene looks like. It places a virtual camera, renders, and refines the pose against the reference (Figure 4).

Metric. The primary metrics are **Position Error (PE)** and **Angular Error (AE)**. Let $\hat{\mathbf{c}}, \mathbf{c} \in \mathbb{R}^3$ denote the predicted and ground-truth camera centres, respectively, and let $\hat{\mathbf{d}}, \mathbf{d} \in \mathbb{R}^3$ denote their corresponding unit viewing directions, with $\|\hat{\mathbf{d}}\|_2 = \|\mathbf{d}\|_2 = 1$. We define

$$\text{PE} = \|\hat{\mathbf{c}} - \mathbf{c}\|_2 , \qquad \text{AE} = \frac{180^\circ}{\pi} \arccos\left(\hat{\mathbf{d}}^\top \mathbf{d}\right) . \tag{2}$$

PE measures the Euclidean distance between the two camera centres in metres, while AE measures the angle between their viewing directions in degrees and lies in $[0^\circ, 180^\circ]$. We report them separately because an estimated camera can be close to the correct position while facing the wrong direction. Because AE compares only the viewing directions, it does not penalise roll about the viewing axis.

2.4 Articulated

Task. The agent receives a closed object with movable parts linked by joints, together with 32 ordered frames from an open–close video. It finds movable parts, infers their joints, and reproduces the motion (Figure 5). The ability under test is kinematic reasoning. The agent recovers how each part moves from the sampled frames and reproduces that motion.

Metric. The primary metric is **Maximum Part Error (MPE)**. It is the largest opening-aligned geometry error or unreproduced motion range across all ground-truth movable parts. Let $\mathcal{P}_{\text{mov}}$ denote

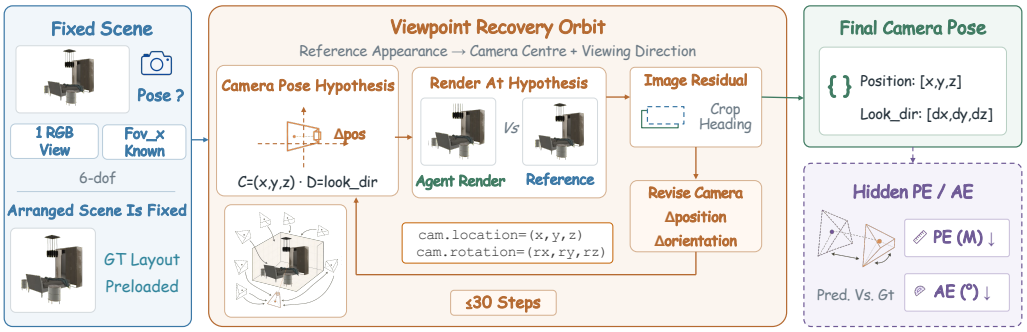


Figure 4: **Camera.** From one image and known field of view, the agent render-checks a 6-DoF camera hypothesis. Hidden ground truth scores the final JSON pose using PE and AE.

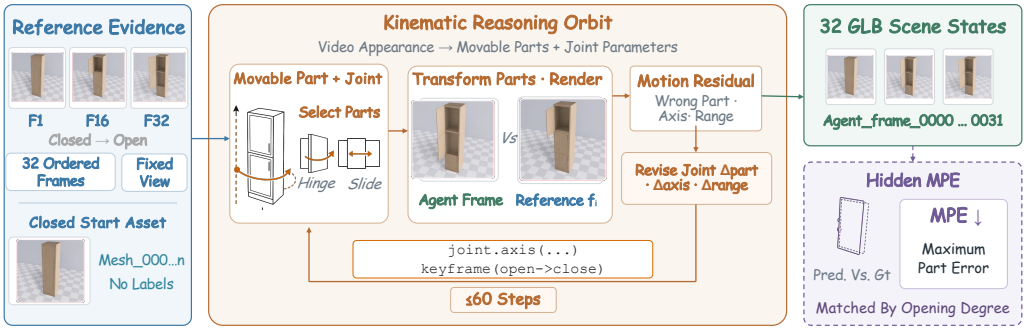


Figure 5: **Articulated.** From a closed GLB and 32 open–close frames, the agent infers movable parts and joints and exports 32 scene states. Opening-aligned hidden ground truth scores them using **MPE**.

the set of ground-truth movable parts. For each part $i \in \mathcal{P}_{\text{mov}}$, let ϵ_i be its largest geometry error after aligning the predicted and ground-truth frames by opening degree. Let $\kappa_i \in [0, 1]$ be the fraction of its full motion reproduced by the prediction, and let g_i be its full ground-truth travel:

$$\text{MPE} = \max\{\epsilon_i,\ (1 - \kappa_i)g_i \mid i \in \mathcal{P}_{\text{mov}}\}. \tag{3}$$

Here, both ϵ_i and g_i are measured in metres. The term $(1 - \kappa_i)g_i$ penalises incomplete motion; in particular, an unmoved part has $\kappa_i = 0$ and is charged its full ground-truth travel. Frames are aligned by opening degree rather than by frame index. MPE therefore reports the maximum error across movable parts directly in metres, without an object-diagonal term or size normalisation.

2.5 Reconstruction

Task. Given multiple room views and an empty scene, the agent builds, places, and textures each furniture object as a mesh, a 3D surface represented by vertices and faces (Figure 6). The ability under test is shape imagination. The agent infers complete 3D geometry from a few partial views.

Metric. The primary **F@5%** score (Tatarchenko et al., 2019) is computed after fixed-scale ICP alignment (Besl & McKay, 1992), DBSCAN clustering (Ester et al., 1996), and Hungarian matching (Kuhn, 1955). Let N be the number of ground-truth objects. For each ground-truth object j, let $V_j \subset \mathbb{R}^3$ denote its sampled surface points. If object j is matched, let $\hat{V}_j \subset \mathbb{R}^3$ denote the recentered surface points of its matched predicted cluster; an unmatched object has no assigned predicted cluster. We set the object-specific distance threshold to $\tau_j = 0.05\delta_j$, where δ_j is the bounding-box diagonal of object j. For a matched object j, let Prec_j be the fraction of points in $\hat{V}_j$ whose nearest point in V_j lies within τ_j, and let Rec_j be the fraction of points in V_j whose nearest point in $\hat{V}_j$ lies within τ_j.

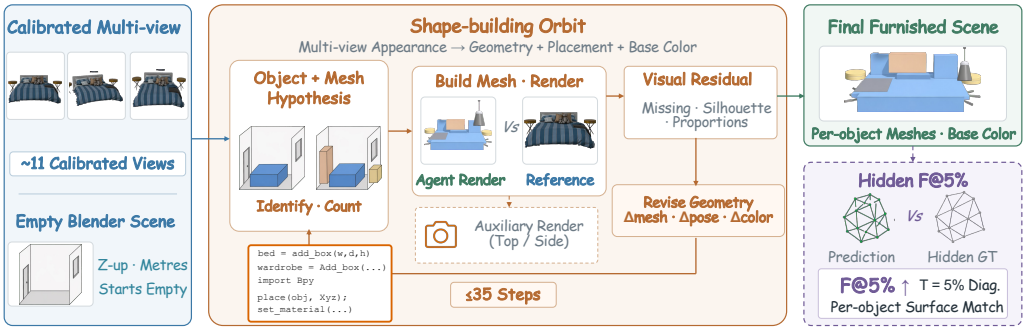


Figure 6: **Reconstruction.** From ∼11 calibrated views and an empty scene, the agent builds and render-checks furniture geometry, appearance, and pose. Hidden ground truth scores the final scene using object-level **F@5%**.

For an unmatched object j, we set $\mathrm{Prec}_j = \mathrm{Rec}_j = 0$. The scene-level score is

$$\mathrm{F@5\%} = \frac{1}{N}\sum_{j=1}^{N} \frac{2\,\mathrm{Prec}_j \cdot \mathrm{Rec}_j}{\mathrm{Prec}_j + \mathrm{Rec}_j}. \tag{4}$$

The summand is defined as zero whenever $\mathrm{Prec}_j + \mathrm{Rec}_j = 0$, including for every unmatched object. Thus, an object contributes a high score only when its predicted and ground-truth surfaces mutually cover each other. Appendix C.3 gives the full specification, including point sets, matching thresholds, and recentering.

2.6 Dynamic

Task. Given sampled video frames and an asset library, the agent rebuilds the static layout, start positions, and keyframed trajectories (Figure 7). The ability under test is dynamic reasoning. The agent recovers the simultaneous motion of several objects and replays it in 3D. We also test a photo-realistic reference variant on the same scenes and ground truth.

Metric. We call each movable object a mover, and let N_{mov} be the number of ground-truth movers. The evaluator matches mover-centroid tracks and corrects a single global translation. For a matched ground-truth mover $i \in \{1, \ldots, N_{\mathrm{mov}}\}$, let m_i be the shorter matched track length. We average the centroid error over these m_i frames and divide it by the scene scale S to obtain e_i. The scale S is the horizontal span of the ground-truth scene, floored at $5\,\mathrm{m}$. Each unmatched ground-truth mover is assigned $e_i = 1$.

After applying the same global translation, let $\hat{\mathcal{X}}_{\mathrm{stat}} \subset \mathbb{R}^3$ and $\mathcal{X}_{\mathrm{stat}} \subset \mathbb{R}^3$ denote the predicted and ground-truth sets of static-object centroids, respectively. We define $d_{\mathrm{stat}}(\hat{\mathcal{X}}_{\mathrm{stat}}, \mathcal{X}_{\mathrm{stat}})$ as the sum of two directed mean nearest-neighbour distances: the mean distance from each predicted centroid to its nearest ground-truth centroid, and the mean distance from each ground-truth centroid to its nearest predicted centroid. The two primary errors, **Maximum Mover Error (MME)** and **Layout Error (LE)**, are

$$\mathrm{MME} = \max_{1 \le i \le N_{\mathrm{mov}}} e_i, \qquad \mathrm{LE} = \frac{d_{\mathrm{stat}}\left(\hat{\mathcal{X}}_{\mathrm{stat}}, \mathcal{X}_{\mathrm{stat}}\right)}{2S}. \tag{5}$$

MME takes the maximum over all ground-truth movers, including every unmatched mover assigned $e_i = 1$. LE averages the two directed nearest-neighbour distances between the predicted and ground-truth static-object centroids and normalises the result by the scene scale. Both metrics are dimensionless, and lower values are better. Rotation and scale are not corrected, and extra predicted movers do not enter MME. Appendix C.3 specifies the exact track-length, matching, and empty-output rules.

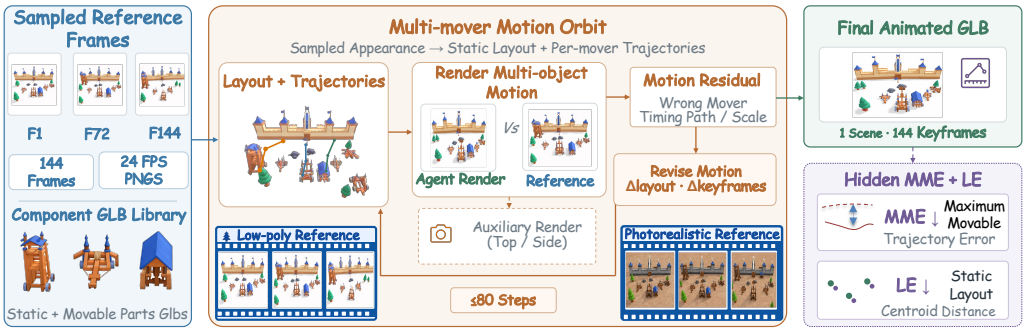


Figure 7: **Dynamic.** From sampled video frames, a GLB library, and a known camera, the agent reconstructs the layout and keyframes mover trajectories. Hidden ground truth scores the animated GLB using **MME** and **LE**.

3 Experiments

3.1 Setup

We evaluate eleven configurations from ten proprietary VLMs. They are Claude Opus 4.6 (Anthropic, 2026a), Claude Sonnet 5 (Anthropic, 2026b), GPT 5.4 (Singh et al., 2026), Gemini 3.1 Pro (Google DeepMind, 2026), Qwen 3.7 Plus (Qwen Team, Alibaba, 2026), MiniMax M3 (Lai et al., 2026), Doubao Seed 2.0 Pro (Seed, 2026), Step 3.7 Flash (StepFun, 2026), MiMo 2.5 (Xiaomi, 2026), and Kimi K2.6 (Kimi, 2026). Configurations labelled High use the provider's high setting. GPT 5.4 Medium is the lower-effort control. Kimi Reason enables reasoning.

To keep the comparison fair, we built one shared harness and ran every configuration through it. All configurations drive a headless Blender through the same Model Context Protocol (MCP) interface. The shared prompts name four core Blender tools: `get_scene_info` and `get_object_info` for inspection, `execute_blender_code` for Python, and `render_scene_view` for visual checks. Dynamic also provides `read_reference_frames` for retrieving video frames. The task budgets are 30 steps for Layout and Camera, 60 for Articulated, 35 for Reconstruction, and 80 for Dynamic. Appendix D.1 gives the shared interface and budget details, and Appendix E.5.3 compares Claude Sonnet 5 High under this harness and the official Claude Code CLI.

Overall score. The five tasks use different units, so we map each native metric to a score from 0 to 100 using $q^{\downarrow}$ for errors and $q^{\uparrow}$ for F@5% (Appendix D.2). Invalid outputs receive zero. Layout uses $q^{\downarrow}(\text{ADD-S}; 4\,\text{m})$. Camera averages $q^{\downarrow}(\text{PE}; 4\,\text{m})$ and $q^{\downarrow}(\text{AE}; 90°)$. Articulated uses $q^{\downarrow}(\text{MPE}; 1\,\text{m})$, Reconstruction uses $q^{\uparrow}(\text{F@5\%}; 1)$, and Dynamic averages $q^{\downarrow}(\text{MME}; 1)$ and $q^{\downarrow}(\text{LE}; 1)$. We average case scores within each task, then average the five task scores to obtain Overall. Overall uses single-view Layout and low-poly Dynamic; the paired multi-view Layout and photo-realistic Dynamic conditions stay outside Overall.

3.2 Main Results

Table 3 reports all eleven completed configurations. Under the fixed Overall summary, Doubao obtains the highest score at 50.2, followed closely by Claude Opus at 48.9 and GPT 5.4 Medium at 48.7. GPT 5.4 High trails Medium by only 0.02 points. Kimi reaches 41.2 and ranks eighth. We therefore use the exact point ordering and analyse the three highest configurations, while retaining every configuration in the leaderboard.

The three leaders differ by only 1.5 Overall points but exchange task leadership. Doubao leads the group on Layout (77.4), Camera (34.5), and Dynamic (70.7). Claude Opus leads Articulated (63.7), while GPT 5.4 Medium leads Reconstruction (10.4). These profiles motivate a case-level analysis rather than another comparison of aggregate means.

Table 3: **Main results.** Native metrics retain task units; shaded columns are fixed-reference task scores used in Overall. Camera reports PE (m)/AE (°) and Dynamic MME/LE. Higher scores are better; bold marks the best task score and Overall. Overall is a fixed normalised summary, not evidence of statistically distinct ranks.

	Layout		Camera		Articulated		Reconstruction		Dynamic		Overall
Configuration	ADD-S ↓	Score ↑	PE/AE ↓	Score ↑	MPE ↓	Score ↑	F@5% ↑	Score ↑	MME/LE ↓	Score ↑	Score ↑
Doubao Seed 2.0 Pro High	0.905	77.4	4.825/54.5	**34.5**	0.404	59.6	0.088	8.8	0.471/0.150	**70.7**	**50.2**
Claude Opus 4.6 High	1.059	73.5	4.980/51.6	34.0	0.375	63.7	0.098	9.8	0.583/0.152	63.2	48.9
GPT 5.4 Medium	1.120	72.7	5.404/54.5	29.6	0.381	62.3	0.104	10.4	0.738/0.235	68.5	48.7
GPT 5.4 High	0.766	**84.1**	5.748/51.1	26.4	0.275	**73.8**	0.123	**12.3**	0.746/0.320	46.7	48.7
Qwen 3.7 Plus High	0.954	76.2	6.489/70.9	21.7	0.603	58.0	0.090	9.0	0.441/0.238	66.0	46.2
Gemini 3.1 Pro High	6.953	65.4	5.272/53.8	33.9	0.452	56.5	0.071	7.1	0.527/0.335	63.9	45.4
MiMo 2.5 High	0.824	79.4	6.618/58.2	25.0	0.639	49.8	0.091	9.1	0.827/0.299	43.7	41.4
Kimi K2.6 Reason	2.547	70.9	6.396/57.2	24.6	0.442	57.3	0.085	8.5	0.855/0.281	44.8	41.2
Step 3.7 Flash High	0.898	77.5	7.191/78.8	13.2	0.541	48.3	0.086	8.6	0.712/0.164	57.9	41.1
Claude Sonnet 5 High	21.488	51.9	6.045/54.3	27.3	0.425	57.8	0.105	10.5	1.000/1.330	49.9	39.5
MiniMax M3 High	4.188	58.1	5.703/64.1	25.6	0.428	58.4	0.096	9.6	0.755/0.426	41.4	38.6

3.3 Analysis

3.3.1 What determines the ranking?

Overall rewards balance across tasks, not the number of task wins. GPT 5.4 High leads **Layout**, **Articulated**, and **Reconstruction**, yet ranks fourth because it trails GPT 5.4 Medium by 21.8 task-score points on **Dynamic**; the two configurations therefore both round to 48.7 Overall. Appendix E.2 and Figure 9 decompose the two gaps above Doubao. Against Claude Opus, Dynamic contributes $+1.49$ Overall points, while the other four tasks sum to -0.15, giving the $+1.34$ gap. That Dynamic term is concentrated: *raceloop* contributes $+0.86$ Overall points, *factory* contributes $+0.68$, and the other eight scenes together contribute -0.05. Removing the two large scenes only as an influence check changes the gap to -0.21; the published ranking still uses all scenes. In 50,000 paired case resamples, Doubao ranks first 64% of the time, compared with 19% for Claude Opus and 17% for GPT 5.4 Medium, and both 95% intervals for Doubao's gaps include zero. This bootstrap measures sensitivity to benchmark composition, not variation across repeated configuration runs.

Concentrated Deficits Shape the Ranking

Overall rank depends on the magnitude and concentration of task deficits, not the number of task wins. GPT 5.4 High wins three tasks but ranks fourth because of **Dynamic**, whereas Doubao's lead over Claude Opus is concentrated in two **Dynamic** scenes.

3.3.2 How do input conditions affect performance?

We changed only the visual input while keeping the scenes, target outputs, and evaluators fixed (Appendix E.3, Figure 10). Multi-view **Layout** improved nine of eleven configurations, including gains of 12.1 points for Sonnet, 9.4 for Gemini, and 8.5 for Claude Opus. Step and MiMo instead lost 1.4 and 0.6 points, showing that additional views were helpful but not uniformly so.

Photo-realistic **Dynamic** had a mixed effect: four configurations improved, six declined, and Claude Opus was unchanged at one-decimal precision. GPT 5.4 High gained 17.3 points, whereas Step and Sonnet lost 10.3 and 10.2 points. Because the target layout and motion were fixed, these differences measure sensitivity to reference appearance rather than changes in the required output. Each configuration–case pair was run once, so these differences do not estimate repeated-run variability.

More Visual Evidence Is Not Uniformly Beneficial

Additional views usually improve **Layout**, whereas photo-realistic appearance helps some configurations and hurts others on **Dynamic**. The value of richer visual input is therefore configuration- and condition-dependent.

3.3.3 Where do agents fail?

Similar primary scores hide different failure stages (Figure 8; Appendix E.4). In **Articulated**, the top-three **MPE** values are close at 0.375–0.404, but Doubao moves only 13/391 ground-truth parts,

Task	Failure stage	Doubao 2.0 H		Opus 4.6 H		GPT 5.4 M	
Articulated	Parts moved / GT	**3%**	13/391	**65%**	255/391	**34%**	132/391
	Type correct / eligible†	**67%**	8/12	**99%**	248/251	**98%**	124/127
	Direction correct / eligible†	**67%**	8/12	**95%**	238/251	**83%**	105/127
Recon.	Targets matched / GT	**84%**	432/515	**84%**	432/515	**83%**	425/515
	Targets covered / GT	**5%**	26/515	**5%**	24/515	**9%**	44/515
Camera	Avoids PE–AE cap / cases	**82%**	82/100	**84%**	84/100	**76%**	76/100
Dynamic	Movers matched / GT	**97%**	28/29	**79%**	23/29	**83%**	24/29
	Direction correct / eligible†	**44%**	7/16	**79%**	11/14	**73%**	11/15

Figure 8: **Denominator-aware failure stages.** Cells show success rates and exact success/eligible counts. Conditional rows (†) require measurable rigid motion for Articulated or matched non-loop movers for Dynamic; Reconstruction uses ground-truth targets, and Camera counts cases avoiding simultaneous PE ≥ 4 m and AE $\geq 90^\circ$ caps.

compared with 255/391 for Claude Opus and 132/391 for GPT 5.4 Medium. Joint type and direction are then evaluated only on parts with measurable rigid motion: Doubao is type-correct and direction-correct on 8/12 eligible parts, while Claude Opus reaches 248/251 and 238/251, and GPT 5.4 Medium reaches 124/127 and 105/127. In **Reconstruction**, the three configurations match 425–432 of 515 targets, yet cover only 24–44; object-level **F@5%** remains 0.088–0.104. In **Camera**, case-level **PE** and **AE** have Spearman correlations of 0.76–0.93, and both reach their zero-credit caps in 18 Doubao, 16 Claude Opus, and 24 GPT 5.4 Medium cases. In **Dynamic**, Doubao matches 28/29 movers but recovers direction on only 7/16 eligible tracks, compared with 11/14 for Claude Opus and 11/15 for GPT 5.4 Medium. Primary geometry, target selection, surface completion, and motion semantics therefore fail at different stages.

Diagnostics Locate the Failure Stage

The primary metrics provide the basis for ranking, while the broader diagnostic suite supplies fine-grained signals for locating failures and guiding model improvement. Primary scores alone conflate frozen outputs, wrong actions, incomplete surfaces, and direction failures.

3.3.4 How much interaction is used?

We define effective interaction as the selected trace prefix before scoring, compute steps divided by `MaxSteps` within each task, and then average tasks equally. Under this task-balanced summary, interaction volume did not positively track Overall across the eleven configurations (Spearman $\rho = -0.68$). Claude Opus used 82.9% of its budget and 39.1 calls per run, versus 32.8% and 19.9 calls for similarly scoring GPT 5.4 Medium; first-ranked Doubao used 34.3% and 11.4 calls. This descriptive association does not establish a causal effect; Appendix E.5 reports the process, budget-sensitivity, and stack-sensitivity analyses, including Figure 13.

More Interaction Does Not Imply Better Performance

Across the evaluated set, task-balanced interaction volume does not positively track **Overall** ($\rho = -0.68$). Similarly ranked configurations use different effective budgets; this descriptive association does not establish an effect of additional interaction.

4 Conclusion

SceneActBench makes 3D understanding executable by requiring agents to match hidden scene geometry. Across five tasks and eleven configurations, no configuration is consistently strong; the three leaders trade task wins despite lying within 1.5 Overall points. Rank gaps concentrate in a few Dynamic scenes, and additional views or photo-realism do not help every configuration. Failure-stage diagnostics separate missing actions, incorrect motion, and incomplete geometry. SceneActBench therefore supports both comparison and diagnosis; Appendix A details its boundaries.

AI USE STATEMENT

We used AI coding assistants to aid or polish writing and draft sections of the paper.

ETHICS STATEMENT

This work studies benchmark of VLM in Blender and uses no private human data.

REPRODUCIBILITY STATEMENT

We provide the evaluation settings, prompts, and experimental protocols throughout the main text and appendices. Appendix D gives implementation details.

REFERENCES

Niket Agarwal, Arslan Ali, Jon Allen, Martin Antolini, Adeline Aubame, Alisson Azzolini, Junjie Bai, Maciej Bala, Yogesh Balaji, Josh Bapst, et al. Cosmos 3: Omnimodal world models for physical ai. *arXiv preprint arXiv:2606.02800*, 2026.

Anthropic. Claude opus 4.6. Model card, 2026a. URL https://www-cdn.anthropic.com/14e4fb01875d2a69f646fa5e574dea2b1c0ff7b5.pdf. Released 2026-02-05.

Anthropic. Claude Sonnet 5. Model card, 2026b. URL https://www-cdn.anthropic.com/480e0bb54327b9622282e9c39a83a4f490ed377e/Claude%20Sonnet%205%20System%20Card.pdf. Released 2026-06-30.

Daichi Azuma, Taiki Miyanishi, Shuhei Kurita, and Motoaki Kawanabe. ScanQA: 3d question answering for spatial scene understanding. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19129–19139, 2022.

Paul J. Besl and Neil D. McKay. Method for Registration of 3-D Shapes. In Paul S. Schenker (ed.), *Sensor Fusion IV: Control Paradigms and Data Structures*, volume 1611, pp. 586–606. International Society for Optics and Photonics, SPIE, 1992. doi: 10.1117/12.57955. URL https://doi.org/10.1117/12.57955.

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

Wayne Chi, Yixiong Fang, Arnav Yayavaram, Siddharth Yayavaram, Seth Karten, Qiuhong Anna Wei, Runkun Chen, Alexander Wang, Valerie Chen, Ameet Talwalkar, et al. GameDevBench: Evaluating agentic capabilities through game development. *arXiv preprint arXiv:2602.11103*, 2026.

Anna C Doris, Ferdous Alam, Amin Heyrani Nobari, and Faez Ahmed. CAD-Coder: An open-source vision-language model for computer-aided design code generation. *Journal of Mechanical Design*, 148(7):071702, 2026.

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 226–231. AAAI Press, 1996.

Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. LayoutGPT: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems*, 36: 18225–18250, 2023.

Huan Fu, Bowen Cai, Lin Gao, Ling-Xiao Zhang, Jiaming Wang, Cao Li, Qixun Zeng, Chengyue Sun, Rongfei Jia, Binqiang Zhao, et al. 3D-FRONT: 3d furnished rooms with layouts and semantics. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 10933–10942, 2021.

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

Yipeng Gao, Lei Shu, Genzhi Ye, Xi Xiong, Ameesh Makadia, Meiqi Guo, Laurent Itti, and Jindong Chen. 3DCodeBench: Benchmarking agentic procedural 3d modeling via code. *arXiv preprint arXiv:2606.01057*, 2026.

Google DeepMind. Gemini 3.1 pro. Model card, 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/. Released 2026-02-19.

Yunqi Gu, Ian Huang, Jihyeon Je, Guandao Yang, and Leonidas Guibas. BlenderGym: Benchmarking foundational model systems for graphics editing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 18574–18583, 2025.

Yicong Hong, Kai Zhang, Jiuxiang Gu, Sai Bi, Yang Zhou, Difan Liu, Feng Liu, Kalyan Sunkavalli, Trung Bui, and Hao Tan. LRM: Large reconstruction model for single image to 3d. In *International Conference on Learning Representations*, volume 2024, pp. 50678–50702, 2024.

Ziniu Hu, Ahmet Iscen, Aashi Jain, Thomas Kipf, Yisong Yue, David A Ross, Cordelia Schmid, and Alireza Fathi. Scenecraft: An LLM agent for synthesizing 3d scenes as blender code. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=gAyzjHw2ml.

Wenlong Huang, Chen Wang, Ruohan Zhang, Yunzhu Li, Jiajun Wu, and Li Fei-Fei. Voxposer: Composable 3d value maps for robotic manipulation with language models. In *Towards Generalist Robots: Learning Paradigms for Scalable Skill Acquisition @ CoRL2023*, 2023. URL https://openreview.net/forum?id=4NnJ1iifoH.

Denys Iliash, Hanxiao Jiang, Yiming Zhang, Manolis Savva, and Angel X Chang. S2O: Static to openable enhancement for articulated 3d objects. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 6785–6795, 2026.

Kimi. Kimi K2.6. Model card, 2026. URL https://www.kimi.com/ai-models/kimi-k2-6. Released 2026-04-20.

Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955.

Xunhao Lai, Weiqi Xu, Yufeng Yang, Qiaorui Chen, Yang Xu, Lunbin Zeng, Xiaolong Li, Haohai Sun, Haichao Zhu, Vito Zhang, Jinkai Hu, Jiayao Li, Rui Gao, Zekun Li, Songquan Zhu, Jingkai Zhou, and Pengyu Zhao. Minimax sparse attention, 2026. URL https://arxiv.org/abs/2606.13392.

Minghua Liu, Ruoxi Shi, Kaiming Kuang, Yinhao Zhu, Xuanlin Li, Shizhong Han, Hong Cai, Fatih Porikli, and Hao Su. OpenShape: Scaling up 3d shape representation towards open-world understanding. *Advances in neural information processing systems*, 36:44860–44879, 2023a.

Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 9298–9309, 2023b.

Xiaojian Ma, Silong Yong, Zilong Zheng, Qing Li, Yitao Liang, Song-Chun Zhu, and Siyuan Huang. SQA3d: Situated question answering in 3d scenes. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=IDJx97BC38.

Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. ATISS: Autoregressive transformers for indoor scene synthesis. *Advances in neural information processing systems*, 34:12013–12026, 2021.

Qwen Team, Alibaba. Qwen3.7-plus. Qwen blog, 2026. URL https://qwen.ai/blog?id=qwen3.7-plus. Released 2026-06-02.

Johannes L Schonberger and Jan-Michael Frahm. Structure-from-motion revisited. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4104–4113, 2016.

Bytedance Seed. Seed2.0 model card: Towards intelligence frontier for real-world complexity. *arXiv preprint arXiv:2607.00248*, 2026.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, Akshay Nathan, Alan Luo, Alec Helyar, Aleksander Madry, Aleksandr Efremov, Aleksandra Spyra, Alex Baker-Whitcomb, Alex Beutel, Alex Karpenko, Alex Makelov, Alex Neitz, Alex Wei, Alexandra Barr, Alexandre Kirchmeyer, Alexey Ivanov, Alexi Christakis, Alistair Gillespie, Allison Tam, Ally Bennett, Alvin Wan, Alyssa Huang, Amy McDonald Sandjideh, Amy Yang, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrei Gheorghe, Andres Garcia Garcia, Andrew Braunstein, Andrew Liu, Andrew Schmidt, Andrey Mereskin, Andrey Mishchenko, Andy Applebaum, Andy Rogerson, Ann Rajan, Annie Wei, Anoop Kotha, Anubha Srivastava, Anushree Agrawal, Arun Vijayvergiya, Ashley Tyra, Ashvin Nair, Avi Nayak, Ben Eggers, Bessie Ji, Beth Hoover, Bill Chen, Blair Chen, Boaz Barak, Borys Minaiev, Botao Hao, Bowen Baker, Brad Lightcap, Brandon McKinzie, Brandon Wang, Brendan Quinn, Brian Fioca, Brian Hsu, Brian Yang, Brian Yu, Brian Zhang, Brittany Brenner, Callie Riggins Zetino, Cameron Raymond, Camillo Lugaresi, Carolina Paz, Cary Hudson, Cedric Whitney, Chak Li, Charles Chen, Charlotte Cole, Chelsea Voss, Chen Ding, Chen Shen, Chengdu Huang, Chris Colby, Chris Hallacy, Chris Koch, Chris Lu, Christina Kaplan, Christina Kim, CJ Minott-Henriques, Cliff Frey, Cody Yu, Coley Czarnecki, Colin Reid, Colin Wei, Cory Decareaux, Cristina Scheau, Cyril Zhang, Cyrus Forbes, Da Tang, Dakota Goldberg, Dan Roberts, Dana Palmie, Daniel Kappler, Daniel Levine, Daniel Wright, Dave Leo, David Lin, David Robinson, Declan Grabb, Derek Chen, Derek Lim, Derek Salama, Dibya Bhattacharjee, Dimitris Tsipras, Dinghua Li, Dingli Yu, DJ Strouse, Drew Williams, Dylan Hunn, Ed Bayes, Edwin Arbus, Ekin Akyurek, Elaine Ya Le, Elana Widmann, Eli Yani, Elizabeth Proehl, Enis Sert, Enoch Cheung, Eri Schwartz, Eric Han, Eric Jiang, Eric Mitchell, Eric Sigler, Eric Wallace, Erik Ritter, Erin Kavanaugh, Evan Mays, Evgenii Nikishin, Fangyuan Li, Felipe Petroski Such, Filipe de Avila Belbute Peres, Filippo Raso, Florent Bekerman, Foivos Tsimpourlas, Fotis Chantzis, Francis Song, Francis Zhang, Gaby Raila, Garrett McGrath, Gary Briggs, Gary Yang, Giambattista Parascandolo, Gildas Chabot, Grace Kim, Grace Zhao, Gregory Valiant, Guillaume Leclerc, Hadi Salman, Hanson Wang, Hao Sheng, Haoming Jiang, Haoyu Wang, Haozhun Jin, Harshit Sikchi, Heather Schmidt, Henry Aspegren, Honglin Chen, Huida Qiu, Hunter Lightman, Ian Covert, Ian Kivlichan, Ian Silber, Ian Sohl, Ibrahim Hammoud, Ignasi Clavera, Ikai Lan, Ilge Akkaya, Ilya Kostrikov, Irina Kofman, Isak Etinger, Ishaan Singal, Jackie Hehir, Jacob Huh, Jacqueline Pan, Jake Wilczynski, Jakub Pachocki, James Lee, James Quinn, Jamie Kiros, Janvi Kalra, Jasmyn Samaroo, Jason Wang, Jason Wolfe, Jay Chen, Jay Wang, Jean Harb, Jeffrey Han, Jeffrey Wang, Jennifer Zhao, Jeremy Chen, Jerene Yang, Jerry Tworek, Jesse Chand, Jessica Landon, Jessica Liang, Ji Lin, Jiancheng Liu, Jianfeng Wang, Jie Tang, Jihan Yin, Joanne Jang, Joel Morris, Joey Flynn, Johannes Ferstad, Johannes Heidecke, John Fishbein, John Hallman, Jonah Grant, Jonathan Chien, Jonathan Gordon, Jongsoo Park, Jordan Liss, Jos Kraaijeveld, Joseph Guay, Joseph Mo, Josh Lawson, Josh McGrath, Joshua Vendrow, Joy Jiao, Julian Lee, Julie Steele, Julie Wang, Junhua Mao, Kai Chen, Kai Hayashi, Kai Xiao, Kamyar Salahi, Kan Wu, Karan Sekhri, Karan Sharma, Karan Singhal, Karen Li, Kenny Nguyen, Keren Gu-Lemberg, Kevin King, Kevin Liu, Kevin Stone, Kevin Yu, Kristen Ying, Kristian Georgiev, Kristie Lim, Kushal Tirumala, Kyle Miller, Lama Ahmad, Larry Lv, Laura Clare, Laurance Fauconnet, Lauren Itow, Lauren Yang, Laurentia Romaniuk, Leah Anise, Lee Byron, Leher Pathak, Leon Maksin, Leyan Lo, Leyton Ho, Li Jing, Liang Wu, Liang Xiong, Lien Mamitsuka, Lin Yang, Lindsay McCallum, Lindsey Held, Liz Bourgeois, Logan Engstrom, Lorenz Kuhn, Louis Feuvrier, Lu Zhang, Lucas Switzer, Lukas Kondraciuk, Lukasz Kaiser, Manas Joglekar, Mandeep Singh, Mandip Shah, Manuka Stratta, Marcus Williams, Mark Chen, Mark Sun, Marselus Cayton, Martin Li, Marvin Zhang, Marwan Aljubeh, Matt Nichols, Matthew Haines, Max Schwarzer, Mayank Gupta, Meghan Shah, Melody Y. Guan, Melody Huang, Meng Dong, Mengqing Wang, Mia Glaese, Micah Carroll, Michael Lampe, Michael Malek, Michael Sharman, Michael Zhang, Michele Wang, Michelle Pokrass, Mihai Florian, Mikhail Pavlov, Miles Wang, Ming Chen, Mingxuan Wang, Minnia Feng, Mo Bavarian, Molly Lin, Moose Abdool, Mostafa Rohaninejad, Nacho Soto, Natalie Staudacher, Natan LaFontaine, Nathan Marwell, Nelson Liu, Nick Preston, Nick Turley, Nicklas Ansman, Nicole Blades, Nikil Pancha, Nikita Mikhaylin, Niko Felix, Nikunj Handa, Nishant Rai, Nitish Keskar, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Oona Gleeson, Pamela Mishkin, Patryk Lesiewicz, Paul Baltescu, Pavel

Belov, Peter Zhokhov, Philip Pronin, Phillip Guo, Phoebe Thacker, Qi Liu, Qiming Yuan, Qinghua Liu, Rachel Dias, Rachel Puckett, Rahul Arora, Ravi Teja Mullapudi, Raz Gaon, Reah Miyara, Rennie Song, Rishabh Aggarwal, RJ Marsan, Robel Yemiru, Robert Xiong, Rohan Kshirsagar, Rohan Nuttall, Roman Tsiupa, Ronen Eldan, Rose Wang, Roshan James, Roy Ziv, Rui Shu, Ruslan Nigmatullin, Saachi Jain, Saam Talaie, Sam Altman, Sam Arnesen, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Sarah Yoo, Savannah Heon, Scott Ethersmith, Sean Grove, Sean Taylor, Sebastien Bubeck, Sever Banesiu, Shaokyi Amdo, Shengjia Zhao, Sherwin Wu, Shibani Santurkar, Shiyu Zhao, Shraman Ray Chaudhuri, Shreyas Krishnaswamy, Shuaiqi, Xia, Shuyang Cheng, Shyamal Anadkat, Simón Posada Fishman, Simon Tobin, Siyuan Fu, Somay Jain, Song Mei, Sonya Egoian, Spencer Kim, Spug Golden, SQ Mah, Steph Lin, Stephen Imm, Steve Sharpe, Steve Yadlowsky, Sulman Choudhry, Sungwon Eum, Suvansh Sanjeev, Tabarak Khan, Tal Stramer, Tao Wang, Tao Xin, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Degry, Thomas Shadwell, Tianfu Fu, Tianshi Gao, Timur Garipov, Tina Sriskandarajah, Toki Sherbakov, Tomek Korbak, Tomer Kaftan, Tomo Hiratsuka, Tongzhou Wang, Tony Song, Tony Zhao, Troy Peterson, Val Kharitonov, Victoria Chernova, Vineet Kosaraju, Vishal Kuo, Vitchyr Pong, Vivek Verma, Vlad Petrov, Wanning Jiang, Weixing Zhang, Wenda Zhou, Wenlei Xie, Wenting Zhan, Wes McCabe, Will DePue, Will Ellsworth, Wulfie Bain, Wyatt Thompson, Xiangning Chen, Xiangyu Qi, Xin Xiang, Xinwei Shi, Yann Dubois, Yaodong Yu, Yara Khakbaz, Yifan Wu, Yilei Qian, Yin Tat Lee, Yinbo Chen, Yizhen Zhang, Yizhong Xiong, Yonglong Tian, Young Cha, Yu Bai, Yu Yang, Yuan Yuan, Yuanzhi Li, Yufeng Zhang, Yuguang Yang, Yujia Jin, Yun Jiang, Yunyun Wang, Yushi Wang, Yutian Liu, Zach Stubenvoll, Zehao Dou, Zheng Wu, and Zhigang Wang. Openai gpt-5 system card, 2026. URL https://arxiv.org/abs/2601.03267.

StepFun. Step-3.7 Flash. Model card, 2026. URL https://static.stepfun.com/blog/step-3.7-flash/. Released 2026-05-28.

Chunyi Sun, Junlin Han, Weijian Deng, Xinlong Wang, Zishan Qin, and Stephen Gould. 3D-GPT: Procedural 3d modeling with large language models. In *2025 International Conference on 3D Vision (3DV)*, pp. 1253–1263. IEEE, 2025.

Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. DiffuScene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20507–20518, 2024.

Maxim Tatarchenko, Stephan R Richter, René Ranftl, Zhuwen Li, Vladlen Koltun, and Thomas Brox. What do single-view 3d reconstruction networks learn? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 3405–3414, 2019.

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. *Transactions on Machine Learning Research*, 2024a. ISSN 2835-8856. URL https://openreview.net/forum?id=ehfRiF0R3a.

Shuzhe Wang, Vincent Leroy, Yohann Cabon, Boris Chidlovskii, and Jerome Revaud. DUSt3R: Geometric 3d vision made easy. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 20697–20709, 2024b.

Xingrui Wang, Wufei Ma, Tiezheng Zhang, Celso M de Melo, Jieneng Chen, and Alan Yuille. Spatial457: A diagnostic benchmark for 6d spatial reasoning of large multimodal models. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pp. 24669–24679, 2025.

Bowen Wen, Wei Yang, Jan Kautz, and Stan Birchfield. FoundationPose: Unified 6d pose estimation and tracking of novel objects. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 17868–17879, 2024.

Tong Wu, Liang Pan, Junzhe Zhang, Tai Wang, Ziwei Liu, and Dahua Lin. Balanced chamfer distance as a comprehensive metric for point cloud completion. *Advances in Neural Information Processing Systems*, 34:29088–29100, 2021.

Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. Structured 3d latents for scalable and versatile 3d generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 21469–21480, 2025.

Xiaomi. MiMo V2.5. Model card, 2026. URL https://mimo.xiaomi.com/mimo-v2-5. Released 2026-04-22.

Rui Yang, Hanyang Chen, Junyu Zhang, Mark Zhao, Cheng Qian, Kangrui Wang, Qineng Wang, Teja Venkat Koripella, Marziyeh Movahedi, Manling Li, et al. EmbodiedBench: Comprehensive benchmarking multimodal large language models for vision-driven embodied agents. In *International Conference on Machine Learning*, pp. 70576–70631. PMLR, 2025.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16227–16237, 2024.

Xumin Yu, Lulu Tang, Yongming Rao, Tiejun Huang, Jie Zhou, and Jiwen Lu. Point-BERT: Pre-training 3d point cloud transformers with masked point modeling. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 19313–19322, 2022.

Yiheng Zhang, Zhuojiang Cai, Mingdao Wang, Meitong Guo, Tianxiao Li, Li Lin, and Yuwang Wang. M3DLayout: A multi-source dataset of 3d indoor layouts and structured descriptions for 3d generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 34217–34226, 2026.

Appendix A states the study boundaries, and Appendix B provides the extended literature comparison. Appendix C documents benchmark construction and the task evaluators, while Appendix D specifies the experimental and scoring protocol. Appendix E follows the four questions in the main Analysis and provides their supporting evidence. Appendix F lists the exact prompts.

A Limitations and Scope

SceneActBench is designed as a geometric stress test for multi-object 3D action, but it has several limitations. The main study evaluates proprietary VLM configurations and uses one completed run per configuration–case pair, so the reported Overall scores should not be interpreted as repeated-run estimates. The **Dynamic** task contains ten scenes and is best viewed as a targeted stress test rather than a broad estimate of dynamic-scene performance. The Overall score depends on fixed normalisation constants, which are design choices; for this reason, we report native metrics and task-level scores alongside the aggregate. Comparisons with task-specialist 3D pipelines remain an important direction for future work.

B Related Work

3D understanding benchmarks. ScanQA, SQA3D, and Spatial457 score text answers to scans or images (Azuma et al., 2022; Ma et al., 2023; Wang et al., 2025); SpatialVLM and BLINK further show that fluent descriptions can miss spatial judgments (Chen et al., 2024; Fu et al., 2024). BlenderGym, 3DCodeBench, GameDevBench, and EmbodiedBench instead evaluate action, but focus on one object, a static edit, or task success (Gu et al., 2025; Gao et al., 2026; Chi et al., 2026; Yang et al., 2025). SceneActBench tests multi-object scene action with shared geometric evaluation (Table 1).

Agents and task-specific 3D systems. Voyager acts in an interactive world and VoxPoser grounds robot manipulation in 3D (Wang et al., 2024a; Huang et al., 2023); SceneCraft, 3D-GPT, and CAD-Coder generate scenes or shapes (Hu et al., 2024; Sun et al., 2025; Doris et al., 2026). Their outputs and evaluations are task-specific rather than one shared test across the five capabilities studied here.

Specialist models. Layout generators such as ATISS, DiffuScene, LayoutGPT, and Holodeck (Paschalidou et al., 2021; Tang et al., 2024; Feng et al., 2023; Yang et al., 2024), image-to-3D reconstructors such as Zero-1-to-3, LRM, and TRELLIS (Liu et al., 2023b; Hong et al., 2024; Xiang et al., 2025), and camera estimators such as COLMAP and DUSt3R (Schonberger & Frahm, 2016; Wang et al., 2024b) each solve one operation rather than acting across all five tasks.

C BENCHMARK CONSTRUCTION AND TASK EVALUATORS

C.1 3D TERMINOLOGY

Table 4 summarises the 3D-specific terms used throughout the benchmark.

Table 4: **3D terminology used in SceneActBench.** Definitions are limited to the benchmark interface and outputs.

Term	Meaning in SceneActBench
GLB	The binary form of glTF, used to package 3D geometry, materials, object transforms, and animation in one file.
Headless Blender	Blender running without a graphical interface and controlled through code and tools.
Mesh	A 3D surface represented by vertices and faces.
Object transform	An object's position, rotation, and scale in the scene.
Yaw	Rotation about the vertical axis.
Camera pose / extrinsics	The camera's 3D position and orientation.
Articulated object / joint	An articulated object has movable parts; a joint constrains how one part moves.
Joint axis / pivot	The axis specifies the direction of motion; for rotation, the pivot specifies its centre.
Render	A 2D image generated from a 3D scene and camera.

C.2 DATASET CONSTRUCTION

Indoor scenes. We use 100 3D-FRONT rooms that also appear in M3DLayout. Each room has reliable views and object poses. Rooms contain 3-7 furniture objects from 27 categories. We standardise each asset before use. We move its centroid to the origin and set a neutral yaw. We scale it to the annotated size in metres. We replace file and node names with anonymous identifiers. These steps hide the original placement. We render 11 views around each room at furniture height. For each view, we record the world-to-camera matrix and horizontal angle of view. Ground truth stores object location, yaw, box size, and the frame-to-world similarity transform.

Articulated objects. We select 100 objects from the ACD. We orient each object to a common frame: Z up, front toward $-Y$, and base on the ground. We replace mesh names with anonymous identifiers. The agent receives no joint metadata. The evaluator keeps each joint axis, type, range, and vertex-to-part map. We render a 32-frame open–close sequence from a fixed view. We also export the part meshes for every frame.

Dynamic scenes. We build 10 scenes from CC0 low-poly Kenney kits. Each scene has several independent movers. The scenes include road traffic, lane changes, racing loops, circular rail, and boats. They also include conveyors, spinning golf balls, platform jumps, and castle mechanisms. Each scene provides an asset library and a 144-frame video at 24 fps. It also provides animated ground truth, mover coordinates, a layout annotation, and a camera. NVIDIA Cosmos creates the photo-realistic reference from the low-poly render. Layout and motion stay fixed. Both conditions use the same ground truth.

C.3 TASK EVALUATORS

The main text introduces the primary metrics. This section uses the same notation to specify the complete per-case evaluators, including matching, thresholds, and missing-output rules. A hat denotes a prediction; the corresponding unmarked symbol denotes ground truth. For a point set V, let $b(V)$ and $\delta(V)$ denote its bounding-box centre and diagonal, respectively. For two sampled surface sets $A, B \subset \mathbb{R}^3$, we use

$$d^{\leftrightarrow}_{\mathrm{surf}}(A, B) = d_{\mathrm{surf}}(A, B) + d_{\mathrm{surf}}(B, A)$$

for the bidirectional surface distance used by audit metrics. The distinct symbol d_{stat} is reserved for static-object centroid sets in Dynamic. Point sampling uses deterministic seed 0. Configuration-level diagnostics are arithmetic means over cases with defined values. Required fields follow the invalid-case rule in Appendix D.2.

C.3.1 LAYOUT

Object matching and primary metric. The evaluator reads world-space vertices from predicted meshes named `object_*`. It transforms each ground-truth canonical mesh by its annotated pose. It retains at most 4,000 predicted vertices per mesh, then samples each pair to at most 2,000 points. For predicted object $\hat{V}_i$ and candidate ground-truth object V_j, the matching cost is $C_{ij} = d_{\text{surf}}(\hat{V}_i, V_j)$. Hungarian assignment under C returns the rectangular matching $\mathcal{M}$, and Equation 1 gives the scene score. Extra predictions and unmatched targets do not enter this mean. A scene with no predicted object has no valid ADD-S and receives zero task score under Appendix D.2.

Position and scene geometry. Mean position error isolates translation by comparing matched bounding-box centres:

$$\text{PosErr}_{\text{layout}} = \frac{1}{|\mathcal{M}|} \sum_{(i,j)\in\mathcal{M}} \|b(\hat{V}_i) - b(V_j)\|_2. \tag{6}$$

It is measured in metres and separates translation from facing. Let $\hat{V}_{\text{scene}}$ and V_{scene} be the merged predicted and ground-truth surface sets, each sampled to at most 20,000 points. The scene-level symmetric surface audit is $d^{\leftrightarrow}_{\text{surf}}(\hat{V}_{\text{scene}}, V_{\text{scene}})$ (Wu et al., 2021). It is an unnormalised error in metres and includes missing and extra geometry because it does not use object matching.

Count-sensitive audits. Let N be the number of ground-truth objects and $\delta_j = \delta(V_j)$. Let $\xi_j = C_{i(j)j}$ when object j is matched to prediction $i(j)$, and let $\xi_j = +\infty$ otherwise. Placement accuracy (PA) is the fraction placed within 10% of target size:

$$\text{PA} = \frac{1}{N} \sum_{j=1}^{N} \mathbf{1}[\xi_j < 0.1\delta_j]. \tag{7}$$

Binary scene success (SS) is 1 only when the predicted object count is correct and PA $= 1$; otherwise it is 0. The reported scene-success rate is the mean of SS over scenes. ADD-S is the Layout task-score component. $\text{PosErr}_{\text{layout}}$, $d^{\leftrightarrow}_{\text{surf}}$, PA, and SS are audit metrics and do not enter Overall.

C.3.2 CAMERA

Pose errors. The final active camera gives predicted centre $\hat{\mathbf{c}}$ and unit view direction $\hat{\mathbf{d}}$. The view direction is the negative local Z axis of its world matrix. Ground truth is $\mathbf{c}$, $\mathbf{d}$, and Equation 2 gives PE and AE. PE is measured in metres and AE in degrees. AE measures the optical-axis direction and does not penalise roll. These are the only Camera components used in the task score and Overall.

Field of view. The evaluator records the final horizontal field of view $\hat{f}_x$ from the active camera in degrees. The prompt supplies target field of view f_x, but the evaluator does not compute $|\hat{f}_x - f_x|$. The reported FOV is therefore an audit value, not an error metric. It does not affect PE, AE, the Camera score, or Overall.

Render-based audits. During final scoring, the evaluator renders the final scene twice at 384×384 pixels. One render uses the predicted camera matrix. The other uses the ground-truth matrix. Both use the predicted FOV, the same EEVEE renderer, and the same world settings. This pair isolates the effect of the camera extrinsics within the final scene.

The evaluator reports SSIM and PSNR after resizing both renders to 256×256. Both use RGB values in $[0, 1]$ and data range 1. It also reports the cosine similarity s_{CLIP} between OpenCLIP ViT-B/32 image embeddings, its distance $1 - s_{\text{CLIP}}$, and AlexNet LPIPS. LPIPS also uses 256×256 images. These appearance-dependent quantities are audit-only. A visual-render failure does not invalidate finite PE and AE. They compare the two evaluator renders, not the supplied reference image.

C.3.3 ARTICULATED

Frames and part labels. The evaluator uses 32 ground-truth frames in the standardised object frame. Frame 0 is closed and frame 16 is fully open. The predicted open and closed keyframes are the frames with the largest and smallest whole-object displacement from its first frame. At scoring time, each missing frame index is filled by exporting the current Blender scene state. This keeps static or partial outputs scoreable.

A private vertex map assigns each ground-truth vertex to a part. Exact vertex indices are used when the agent preserves topology. Otherwise, each agent vertex receives the label of its nearest ground-truth closed-state vertex. For topology-preserving outputs, the part error is the mean corresponding-vertex L2 distance. The fallback is $\max\{d_{\text{surf}}(\hat{V}, V), d_{\text{surf}}(V, \hat{V})\}$. The evaluator records whether this fallback was used.

Opening alignment and MPE. For movable part $i \in \mathcal{P}_{\text{mov}}$, let V_i^{cl} and V_i^{op} be its ground-truth closed- and open-state vertices. Its full travel is

$$g_i = \max\left\{ \operatorname*{mean}_k \|V_{i,k}^{\text{op}} - V_{i,k}^{\text{cl}}\|_2, 10^{-6}\,\text{m} \right\}. \tag{8}$$

Let Δ_i^t be the mean displacement from the agent's selected closed frame to frame t. When topology differs, the evaluator uses part-centre displacement instead. The predicted opening degree is

$$o_i^t = \frac{\Delta_i^t}{g_i}. \tag{9}$$

The evaluator searches ground-truth frames 0–16 and lets $\phi_i(t)$ be the frame whose opening degree is nearest to the clipped value of o_i^t.

Let err_i denote corresponding-vertex L2 or the fallback above. The two quantities used in Equation 3 are

$$\epsilon_i = \max_t \text{err}_i\left(\hat{V}_i^t, V_i^{\phi_i(t)}\right), \qquad \kappa_i = \min\left\{ \max_t o_i^t, 1 \right\}.$$

Thus ϵ_i is the largest opening-aligned geometry error and κ_i is the reproduced-motion fraction. Mean Part Error averages the aligned per-part errors instead of taking their maximum. The global open-state ADD-S compares the complete agent open keyframe with ground-truth frame 16. MPE is measured directly in metres and has no object-diagonal normalisation. It is the only Articulated component used in Overall.

Part-selection diagnostics. For movable part i, let a_i be the displacement of its predicted part centre between the selected closed and open keyframes. A part is recovered when $g_i > 10^{-3}\,\text{m}$ and $a_i > 0.2 g_i$; movable recall (MR) is the fraction of movable parts recovered. For static part s, let h_s be the mean of its largest 1% closed-to-open vertex displacements, or the maximum when the part has fewer than 100 vertices. The false-move rate (FMR) is the fraction of static parts with $h_s > 0.05\,\text{m}$. If topology differs, h_s uses bidirectional nearest-neighbour displacements. Empty denominator sets yield zero.

Joint diagnostics. Type and direction rates use only movable parts whose mean predicted vertex displacement exceeds $\max(0.05\,\text{m}, 0.25 g_i)$. A rigid rotation is inferred when the fitted rotation exceeds $15°$. Otherwise, translation is inferred when fitted translation exceeds 0.1 times the target range for a translational joint, or $0.05\,\text{m}$ for a rotational joint. Non-rigid motion remains visible in MPE and is not labelled a type mismatch. The type-mismatch rate is the fraction of eligible parts whose inferred rigid type differs from the target type. Direction is fitted from the predicted and ground-truth endpoint geometry. It is reversed when their directional cosine is below -0.7. Reverse-direction rate is the fraction of eligible parts that meet this condition. Cosines between -0.7 and 0.7 are recorded as off-axis rather than reversed. Frozen parts affect MR, not these conditional rates. Both rates are zero when no part is eligible.

C.3.4 Reconstruction

Point sets and global alignment. The evaluator merges all predicted mesh vertices, after retaining at most 4,000 vertices per mesh. Let $\hat{V}_{\text{scene}}$ be this merged predicted surface and V_{scene} the merged ground-truth furniture surface, excluding room structure. Both are sampled to at most 4,000 points for fixed-scale ICP. The fixed scale is the ratio of their axis-aligned bounding-box diagonals. ICP then optimises only rotation and translation. It tries initial yaw angles of $0°$, $90°$, $180°$, and $270°$. Each run uses at most 30 iterations and stops when the mean nearest-neighbour distance changes by less than 10^{-5}.

For the aligned scene surfaces, let $\text{Prec}_{\text{scene}}(\tau)$ and $\text{Rec}_{\text{scene}}(\tau)$ be point precision and recall at distance threshold τ. Their harmonic mean is

$$F_{\text{scene}}(\tau) = \frac{2\,\text{Prec}_{\text{scene}}(\tau)\,\text{Rec}_{\text{scene}}(\tau)}{\text{Prec}_{\text{scene}}(\tau) + \text{Rec}_{\text{scene}}(\tau)}. \tag{10}$$

$F_{\text{scene}}(\tau)$ is zero when its denominator is zero. Let $\delta_{\text{scene}} = \delta(V_{\text{scene}})$. The aligned scene diagnostics use $\tau \in \{0.02, 0.05, 0.10\}\delta_{\text{scene}}$, and the symmetric surface audit uses $d^{\leftrightarrow}_{\text{surf}}(\hat{V}_{\text{scene}}, V_{\text{scene}})$ on the same 4,000-point sets.

Object matching and primary F@5%. Let N be the number of ground-truth objects. DBSCAN separates the aligned prediction into candidates. It tests radii 0.10, 0.15, 0.20, 0.25, 0.30, and 0.40 times the mean ground-truth object diagonal, each with a $0.04\,\text{m}$ lower bound. DBSCAN uses `min_samples`=10 and removes clusters with fewer than 30 points. We choose the cluster count K closest to N, adding a 0.5 tie-break penalty when $K < N$ to discourage merged objects.

Hungarian assignment matches cluster means to ground-truth object means by Euclidean distance. Matches farther than $0.6\delta_{\text{scene}}$ are rejected. Each accepted cluster is translated to the matched ground-truth mean before shape scoring. For a matched ground-truth object j, let $\hat{V}_j$ denote the recentered predicted cluster and V_j its sampled ground-truth surface. Set $\delta_j = \delta(V_j)$ and $\tau_j = 0.05\delta_j$. Point precision and recall are

$$\begin{aligned} \text{Prec}_j &= \frac{1}{|\hat{V}_j|} \sum_{\hat{\mathbf{v}} \in \hat{V}_j} \mathbf{1}\left[\min_{\mathbf{v} \in V_j} \|\hat{\mathbf{v}} - \mathbf{v}\|_2 \le \tau_j\right], \\ \text{Rec}_j &= \frac{1}{|V_j|} \sum_{\mathbf{v} \in V_j} \mathbf{1}\left[\min_{\hat{\mathbf{v}} \in \hat{V}_j} \|\mathbf{v} - \hat{\mathbf{v}}\|_2 \le \tau_j\right]. \end{aligned} \tag{11}$$

For an unmatched object j, we set $\text{Prec}_j = \text{Rec}_j = 0$. Equation 4, with a zero contribution when $\text{Prec}_j + \text{Rec}_j = 0$, gives the primary F@5% score. It is the only Reconstruction component used in Overall. A scene with fewer than 10 points in either $\hat{V}_{\text{scene}}$ or V_{scene} is invalid.

Geometric diagnostics. Match rate is the accepted-match count divided by N. Mean centroid error averages pre-translation cluster-to-ground-truth mean distances over accepted matches. A second, region-based path expands each ground-truth box by $0.1\delta_{\text{scene}}$ and selects aligned predicted points inside it. A region with fewer than 10 predicted points receives local F@5% of zero. Region F@5% averages these values without cluster matching or recentering. Object coverage is the fraction of ground-truth objects whose region F@5% is at least 0.30. $F_{\text{scene}}(\tau)$, $d^{\leftrightarrow}_{\text{surf}}$, and region F@5% are audit metrics. They do not enter Overall.

Point-BERT and visual audits. Point-BERT similarity (Liu et al., 2023a; Yu et al., 2022) is the cosine similarity between OpenShape `openshape-pointbert-vitb32-rgb` embeddings. Each surface set is sampled or repeated to 10,000 points, centred, scaled to unit maximum radius, and assigned neutral-grey RGB features. Scene similarity uses the globally aligned surfaces. Object similarity uses the ground-truth box expanded by $0.1\delta_{\text{scene}}$. A region with fewer than 100 predicted points receives zero. Object Point-BERT is the mean over all ground-truth objects.

The final-run visual audit renders the prediction and ground truth from four orbit angles at 384×384. Each orbit is fitted to the corresponding scene bounds. The ground-truth pass keeps furniture meshes only. The predicted pass keeps all exported meshes. Both use the same EEVEE lighting recipe. The evaluator reports the same SSIM, PSNR, OpenCLIP, and LPIPS quantities as the Camera audit. These values do not affect Reconstruction scoring.

C.3.5 Dynamic

Tracks and scene scale. Ground-truth tracks are read in Blender's Z-up frame. Missing frames are forward-filled, and frames before the first observation use the first valid position. Predicted tracks are sampled from the exported GLB at 24 fps for 144 frames. All animation channels under the same top-level scene object are merged into one mover track. The mover position is the mean of the per-mesh vertex centroids in that subtree. GLB coordinates (x, y, z) map to Blender coordinates $(x, -z, y)$.

Let $\mathcal{X}_{\text{all}}$ contain every ground-truth static location and mover position. The scene scale is

$$S = \max\left\{\left\|\max_{\mathbf{x} \in \mathcal{X}_{\text{all}}} \mathbf{x}_{xy} - \min_{\mathbf{x} \in \mathcal{X}_{\text{all}}} \mathbf{x}_{xy}\right\|_2, 5\,\text{m}\right\}. \tag{12}$$

The extrema are taken coordinate-wise. If $\mathcal{X}_{\text{all}}$ is empty, the implementation uses $S = 40\,\text{m}$.

Global translation and trajectory matching. The evaluator first forms a Hungarian assignment on raw tracks. Its cost is mean per-frame Euclidean distance over the shared frame range. From these coarse matches, let $\{(\hat{\mathbf{y}}_k, \mathbf{y}_k)\}_{k=1}^{K_{\text{pool}}}$ be the pooled paired predicted and ground-truth track points. The applied global translation is

$$\mathbf{t}_{xy}^{\text{glob}} = \frac{1}{K_{\text{pool}}} \sum_{k=1}^{K_{\text{pool}}} (\mathbf{y}_{k,xy} - \hat{\mathbf{y}}_{k,xy}), \qquad t_z^{\text{glob}} = \underset{1 \le k \le K_{\text{pool}}}{\text{median}} (y_{k,z} - \hat{y}_{k,z}). \tag{13}$$

The same translation is applied to every predicted mover and static object. Rotation and scale are not corrected. The evaluator then recomputes Hungarian assignment on the translated tracks, giving the final set of predicted–ground-truth track pairs $\mathcal{M}_{\text{trk}}$, with $\sigma(i)$ the predicted mover matched to ground-truth mover i, and $\hat{N}_{\text{mov}}, N_{\text{mov}}$ the predicted and ground-truth mover counts.

For a matched ground-truth mover i, let m_i be the shorter track length. Its normalised trajectory error is

$$e_i = \frac{1}{m_i S} \sum_{r=1}^{m_i} \left\| \hat{\mathbf{p}}_{\sigma(i)}^r - \mathbf{p}_i^r \right\|_2. \tag{14}$$

Each unmatched ground-truth mover is assigned $e_i = 1$. Extra predicted movers do not enter $\{e_i\}_{i=1}^{N_{\text{mov}}}$. In Equation 5, the maximum is taken over all N_{mov} ground-truth movers. Average Mover Error (AME) is $\text{AME} = N_{\text{mov}}^{-1} \sum_{i=1}^{N_{\text{mov}}} e_i$. AME includes the 1.0 penalties. Dynamic movable recall is $|\mathcal{M}_{\text{trk}}|/N_{\text{mov}}$, and count error is $|\hat{N}_{\text{mov}} - N_{\text{mov}}|/N_{\text{mov}}$. If no predicted mover exists, MME and AME are 1 and movable recall is zero.

Motion-shape diagnostics. For track P, segments longer than $0.3S$ are treated as loop teleports and excluded from path length $L(P)$. A track is moving when $L(P) > 0.05S$. It is closed when it is moving and its net displacement is below $0.1L(P)$. The five-component descriptor $q(P)$ contains normalised endpoint displacement, path length, straightness, vertical range, and maximum deviation from the start-to-end chord. In order, these are $\|P_T - P_1\|_2/S$, $L(P)/S$, $\|P_T - P_1\|_2/(L(P) + 10^{-9})$, $(\max P_z - \min P_z)/S$, and $\ell_\perp(P)/S$. Path-shape error is

$$e_{\text{shape}} = \frac{1}{5|\mathcal{M}_{\text{trk}}|} \sum_{(k,i) \in \mathcal{M}_{\text{trk}}} \|q(\hat{P}_k) - q(P_i)\|_1. \tag{15}$$

It is set to 1 when no mover is matched.

Direction uses the first principal axis of each centred track. The axis sign follows net displacement. Direction-error rate is the fraction of matched, moving, non-closed ground-truth tracks whose predicted and ground-truth axes differ by at least 30°. Closed tracks are excluded because their principal-axis sign is unstable.

Heading uses the velocity direction $\theta_r = \text{atan2}(\Delta y_r, \Delta x_r)$. XY steps below $0.005S$ are removed. XY steps above eight times the median retained step are also removed as teleports. Total turning is the sum of absolute wrapped changes in consecutive headings. For mover i, heading error is the absolute difference between predicted and ground-truth total turning, divided by 2π and capped at 1. We average it over matched movers whose ground-truth track is moving. Closed loops are included. The error is zero when no matched ground-truth mover is moving.

The coarse alignment also estimates an isotropic XY scale by least squares on the centred pooled points. The estimate is bounded to $[0.1, 10]$ but is not applied. Scale error is the absolute log of this estimate and is zero when there is insufficient variation to estimate scale.

Static layout and size audits. Let $\hat{\mathcal{X}}_{\text{stat}}$ contain translated centroids of non-animated top-level predicted subtrees at frame 0, and let $\mathcal{X}_{\text{stat}}$ contain ground-truth static locations. Their bidirectional nearest-neighbour distance is

$$\begin{aligned} d_{\text{stat}}\left(\hat{\mathcal{X}}_{\text{stat}}, \mathcal{X}_{\text{stat}}\right) = &\underset{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_{\text{stat}}}{\text{mean}} \min_{\mathbf{x} \in \mathcal{X}_{\text{stat}}} \|\hat{\mathbf{x}} - \mathbf{x}\|_2 \\ &+ \underset{\mathbf{x} \in \mathcal{X}_{\text{stat}}}{\text{mean}} \min_{\hat{\mathbf{x}} \in \hat{\mathcal{X}}_{\text{stat}}} \|\mathbf{x} - \hat{\mathbf{x}}\|_2. \end{aligned} \tag{16}$$

Equation 5 then gives LE. If $\hat{\mathcal{X}}_{\text{stat}}$ is empty, LE is 1. If $\mathcal{X}_{\text{stat}}$ is empty, LE is undefined and the Dynamic scene is invalid. Layout-count error is

$$\frac{\left||\hat{\mathcal{X}}_{\text{stat}}| - |\mathcal{X}_{\text{stat}}|\right|}{|\mathcal{X}_{\text{stat}}|},$$

and remains diagnostic.

Static-scene size error compares predicted and ground-truth box extents. Ground slabs are removed when their XY span exceeds five times the median span and their height is below one eighth of that span. Axes shorter than $0.1\,\text{m}$ are omitted. Scene size error averages $|\log(\hat{s}_a/s_a)|$ over valid axes. Mover-size error compares matched mover boxes and omits axes shorter than $0.05\,\text{m}$. Skinned movers use Blender-evaluated bounding boxes. It first averages the log-ratio error over valid axes for each mover, then averages over movers. Both metrics remain outside Dynamic scoring.

MME and LE are the two required Dynamic task-score components. Both must be finite. A missing agent GLB or ground-truth trajectory file makes the case invalid. Undefined size audits do not invalidate finite MME and LE. Low-poly Dynamic enters Overall. Photo-realistic Dynamic uses the same evaluator and remains an input-condition ablation.

D Experimental Protocol and Score Aggregation

D.1 Shared Agent Interface and Budgets

Each configuration controls one headless Blender through the same Model Context Protocol interface. Shared prompts provide four core Blender tools. `get_scene_info` returns scene identifiers and transforms. `get_object_info` inspects one object. `execute_blender_code` runs Python inside Blender. `render_scene_view` returns an image for visual checks. Dynamic also provides `read_reference_frames`. Each configuration–case pair is run once. The task ceilings are 30 steps for Layout and Camera, 60 for Articulated, 35 for Reconstruction, and 80 for Dynamic. Appendix F gives the exact prompts.

D.2 Overall Score Normalisation

For native metric m and fixed reference u, we use

$$q^{\downarrow}(m;u) = 100\max(0, 1 - m/u), \qquad q^{\uparrow}(m;u) = 100\min(1, m/u). \tag{17}$$

Only Reconstruction F@5% uses $q^{\uparrow}$; all error metrics use $q^{\downarrow}$. A missing or non-finite required value receives zero. Camera averages PE and AE within each case. Dynamic averages MME and LE within each case. A task score is the mean of its case scores, and Overall is the mean of the five task scores. Layout uses $u = 4\,\text{m}$. Camera uses $u = 4\,\text{m}$ and $u = 90°$. The other three tasks use $u = 1$. Multi-view Layout and photo-realistic Dynamic stay outside Overall.

D.3 Task-Score Correlation Definition

The task-correlation matrix compares task-score associations across configurations. Each configuration contributes five fixed-reference scores. Camera uses the same PE/AE average as Overall. Dynamic uses the same MME/LE average. We correlate the normalised scores. Raw metric units do not affect the matrix.

Let $\mathbf{Z} \in \mathbb{R}^{11\times 5}$ contain one row of five task scores per configuration. We compute the standard Pearson correlation between each pair of columns in $\mathbf{Z}$. The resulting matrix is symmetric, has ones on the diagonal, and lies in $[-1, 1]$. Section E.7 reports the matrix. It describes only these eleven configurations; it is not a population or causal test, and it does not enter Overall or any main-text ranking claim.

E SUPPORTING EVIDENCE FOR THE MAIN ANALYSIS

E.1 GUIDE TO SUPPORTING EVIDENCE

The main Analysis asks what determines the ranking, how input conditions change performance, where agents fail, and how much interaction they use. Sections E.2–E.5 follow that order, while Section E.6 provides the complete numeric record and Section E.7 reports supplementary task-score correlations. Table 5 maps each main-text question to its supporting appendix evidence.

Table 5: **Guide to supporting evidence.** Complete audit values appear in Table 8.

Question	Main-text evidence	Appendix support
Ranking	Task-score balance and concentrated Dynamic gaps	Decomposition, influence check, and paired bootstrap in Section E.2
Input conditions	Paired multi-view Layout and photo-realistic Dynamic changes	All-configuration comparison in Section E.3
Failure stages	Conditional rates and top-three denominator counts	All-configuration counts, Camera diagnostics, and examples in Section E.4
Interaction	Realised budget, calls, and stack sensitivity	Budget curves, CLI comparison, and traces in Section E.5

E.2 RANKING ANALYSIS

Figure 9 visualises the exact linear task-score decomposition discussed in Section 3.3.1.

Scene concentration and influence check. For each Dynamic scene, we subtract Claude Opus's case score from Doubao's, divide by the 10 Dynamic scenes and five equally weighted tasks, and thereby obtain that scene's contribution to the Overall gap. *raceloop* and *factory* contribute $+0.86$ and $+0.68$ Overall points, while the other eight scenes sum to -0.05. As an influence check, we remove those two largest Doubao-favouring Dynamic differences, recompute the Dynamic mean over the remaining eight scenes, keep the other task means unchanged, and re-average the five tasks. The resulting Doubao–Claude Opus gap is -0.21 points. This check is not used for the published ranking.

Paired case-resampling analysis. Within each of the five tasks, we sample cases with replacement and apply the same sampled indices to all three leading configurations. We then recompute each task mean and their equal-weight Overall average. Across 50,000 replicates with random seed 20260715, Doubao ranks first in 64.44% of replicates, Claude Opus in 18.92%, and GPT 5.4 Medium in 16.64%. The 95% intervals for the Doubao–Claude Opus and Doubao–GPT 5.4 Medium Overall gaps are $[-1.89, 4.63]$ and $[-1.84, 4.88]$ points, respectively. This paired bootstrap measures sensitivity to benchmark composition rather than repeated-run uncertainty.

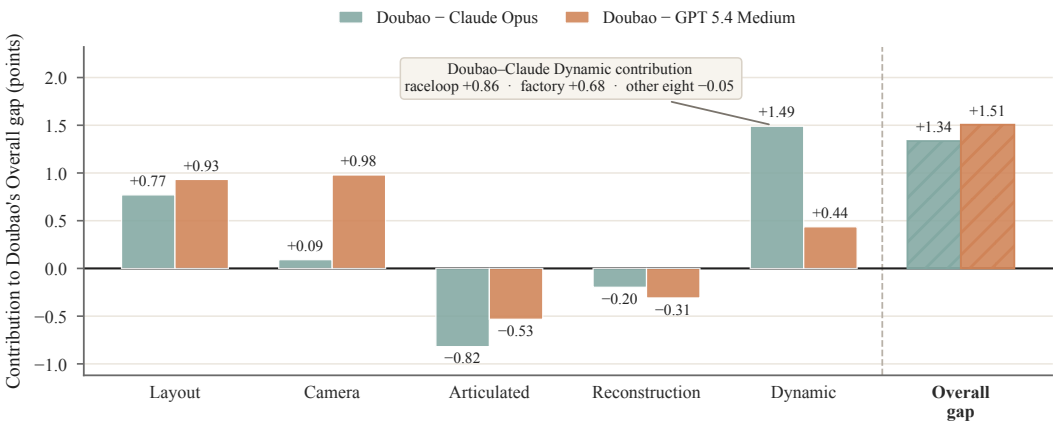


Figure 9: **Task contributions to Overall gaps.** The five task-score differences divided by five sum to each hatched Overall gap. The callout decomposes the Doubao–Claude Opus Dynamic term; this is an exact decomposition, not an independent experiment.

E.3 Input-Condition Sensitivity

Figure 10 provides the all-configuration paired comparison cited in the main Analysis.

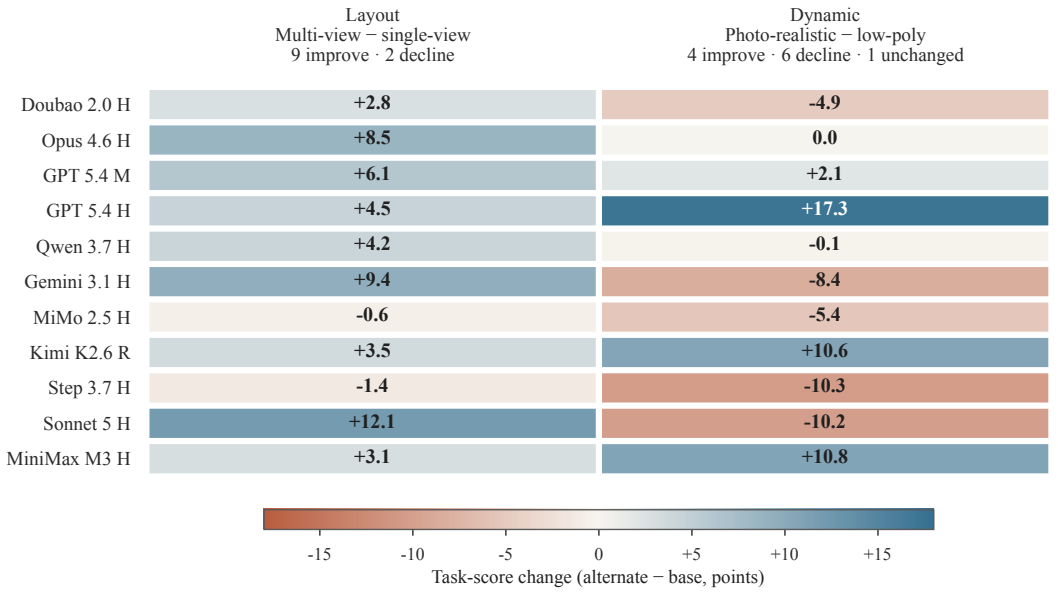


Figure 10: **Input-condition sensitivity.** Cells show paired task-score changes for multi-view versus single-view Layout (100 cases) and photo-realistic versus low-poly Dynamic (10 scenes). Positive values favour the alternate condition, which is excluded from Overall.

E.4 Failure-Stage Evidence

Table 6 extends the Articulated, Reconstruction, and Dynamic failure-stage comparison in the main text to all eleven configurations. Section E.4.2 gives the corresponding Camera diagnostics. Figures 11 and 12 show representative motion and static-geometry failures, and Section E.6 provides the complete diagnostic metrics.

E.4.1 Denominator-Aware Counts

Table 6: **Denominator-aware failure funnels.** Entries are pooled numerator/denominator counts. Articulated type and direction require measurable rigid motion; Dynamic direction requires matched non-loop movers with target motion. Thus, $0/0$ denotes no eligible output, not zero error.

	Articulated			Reconstruction		Dynamic	
Configuration	Moved parts	Type errors	Reverse direction	Matched objects	Covered objects	Matched movers	Direction errors
Doubao Seed 2.0 Pro High	13/391	4/12	4/12	432/515	26/515	28/29	9/16
Claude Opus 4.6 High	255/391	3/251	13/251	432/515	24/515	23/29	3/14
GPT 5.4 Medium	132/391	3/127	22/127	425/515	44/515	24/29	4/15
GPT 5.4 High	254/391	1/255	9/255	461/515	64/515	11/29	0/7
Qwen 3.7 Plus High	193/391	11/194	17/194	401/515	23/515	26/29	5/17
Gemini 3.1 Pro High	193/391	12/195	15/195	331/515	24/515	28/29	4/17
MiMo 2.5 High	50/391	3/45	3/45	442/515	27/515	12/29	0/7
Kimi K2.6 Reason	49/391	2/50	4/50	409/515	35/515	8/29	2/8
Step 3.7 Flash High	25/391	0/22	0/22	388/515	18/515	10/29	2/8
Claude Sonnet 5 High	50/391	3/46	2/46	439/515	26/515	0/29	0/0
MiniMax M3 High	124/391	8/115	9/115	428/515	33/515	8/29	1/8

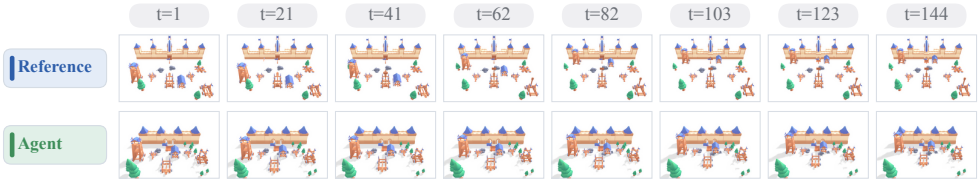
E.4.2 Camera Error Coupling

Across the 100 Camera cases, the Spearman correlation between case-level Position Error and Angular Error is 0.76 for Doubao, 0.93 for Claude Opus, and 0.81 for GPT 5.4 Medium. The two errors simultaneously reach their zero-credit caps in 18, 16, and 24 cases, respectively; equivalently, 82/100, 84/100, and 76/100 cases avoid the joint cap shown in Figure 8. These descriptive correlations show that translation and viewing-direction failures are coupled but do not identify a causal mechanism.

E.4.3 Qualitative Examples

The main Analysis reports the measured failures. Figures 11 and 12 show how those errors appear in rendered output. They are examples, not population statistics.

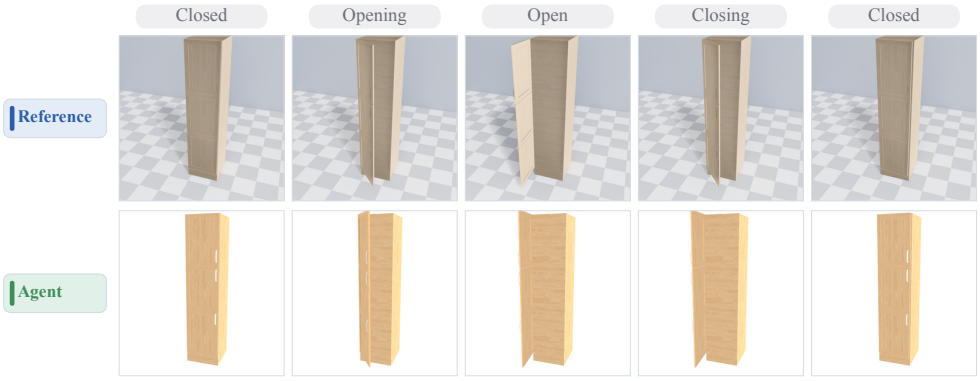
Motion tasks. Articulated outputs can miss a moving panel or move a static part. Dynamic outputs can recover the scene while missing trajectory scale or shape. Figure 11 shows both Dynamic reference styles and one Articulated sequence.



(a) Low-poly Dynamic at matched times.



(b) Photo-realistic Dynamic under the paired reference condition.



(c) Articulated motion from closed to open and back.

Figure 11: Motion examples for Claude Opus 4.6 High. Each panel compares reference frames with agent output at matched times.

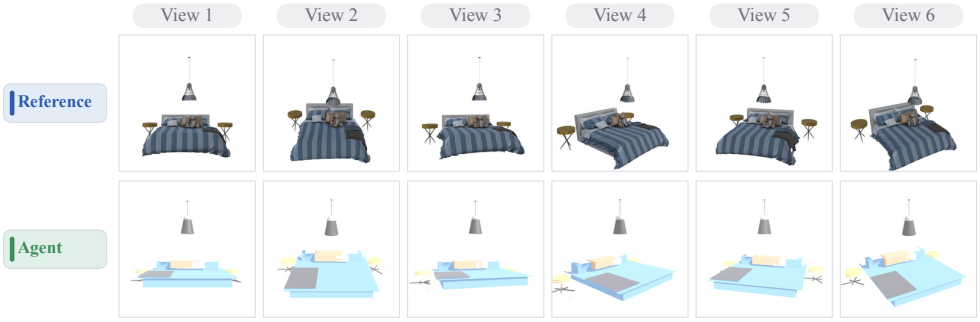
Static tasks. Layout failures include collapsed arrangements and object overlap. Camera errors appear as wrong crops or headings. Reconstruction often replaces detailed furniture with simple proxy shapes. Figure 12 shows these three cases.



(a) Layout on *DiningRoom-13034*. All outputs use the reference camera. Most recover the main arrangement. MiniMax M3 collapses it.



(b) Camera alignment on *Bedroom-6995*. Re-renders vary in crop and heading.



(c) Reconstruction by Doubao Seed 2.0 Pro High. Each output uses its reference camera.

Figure 12: **Static-geometry examples.** Each panel compares reference geometry with agent output.

E.5 Interaction, Budget, and Stack Sensitivity

The process analysis uses aggregate realised steps and tool calls, shared-budget curves, and a separate full Claude Sonnet 5 High comparison between two complete agent stacks.

E.5.1 Effective Interaction Budget

We define the *effective interaction budget* as the selected trace prefix before scoring. For fixed runs, realised steps are capped at the task's `MaxSteps`, and tool calls are counted from that same prefix, so every per-task budget fraction is at most one. To match Overall, we first average process values within each task and then give the five tasks equal weight. Under this task-balanced summary, interaction volume does not positively track Overall across the eleven configurations (Figure 13): the descriptive Spearman correlation is $\rho = -0.68$. Claude Opus uses 82.9% of the available budget and 39.1 task-balanced calls per run, while GPT 5.4 Medium uses 32.8% and 19.9 calls, yet they score 48.9 and 48.7. Doubao also uses only 34.3% and 11.4 calls while ranking first. These values describe different interaction regimes; they do not establish that additional interaction changes performance.

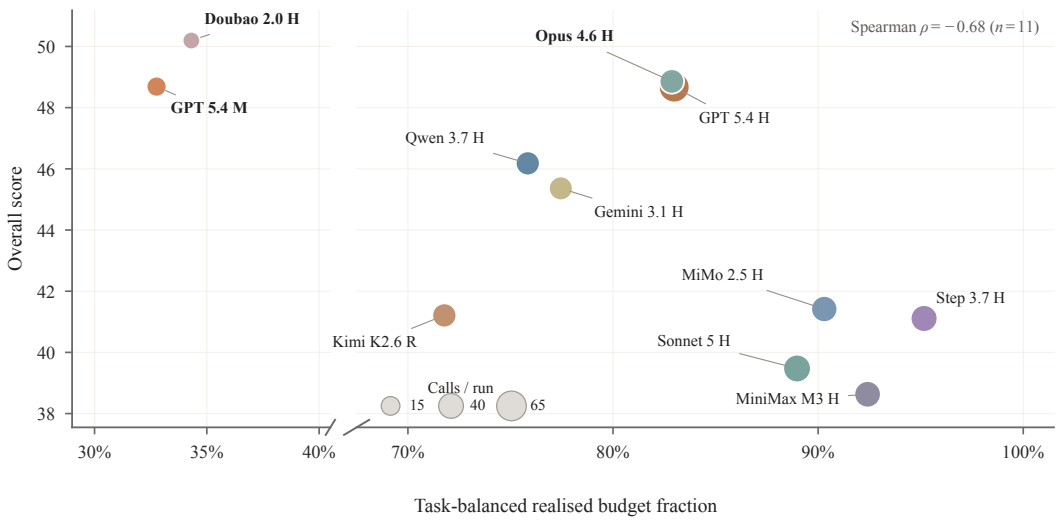


Figure 13: **Task-balanced effective interaction budget.** Each point is one configuration. The horizontal axis averages realised steps divided by `MaxSteps` within each task and then equally across the five tasks; the vertical axis is Overall. The broken horizontal axis omits 40–68%, an interval containing no evaluated configuration. Marker area encodes task-balanced tool calls per run, and labels identify configurations. The Spearman association is descriptive, not causal.

E.5.2 Step-Budget Sensitivity

The analysis retrospectively scores selected execution checkpoints under nominal budgets ranging from 10 to 150 steps. Every task uses the same MaxSteps value at each curve point.

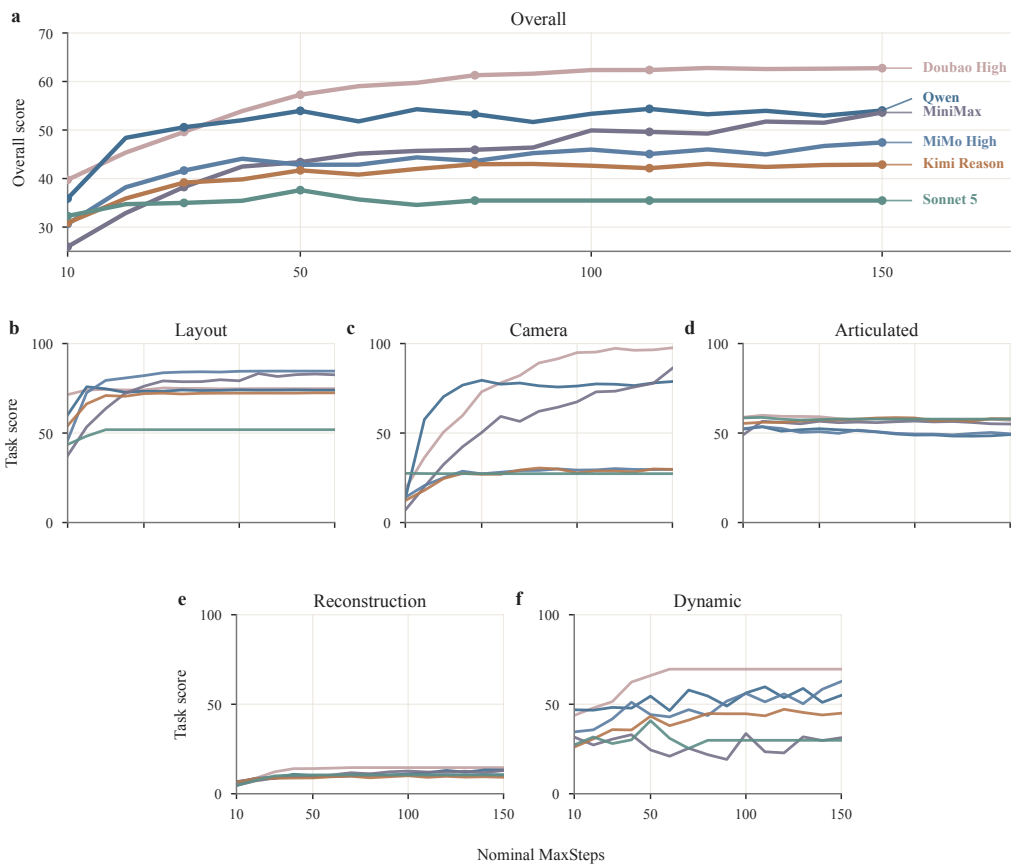


Figure 14: **Step-budget sensitivity.** (a) Overall and (b–f) task scores across MaxSteps values. Each checkpoint uses the same per-case scoring rule. Direct labels in panel (a) identify configurations, and colours are shared across panels.

Overall rises by 12.0–27.7 points between the two endpoints. Camera has the largest mean gain (51.3 points), but its change ranges from 15.7 to 79.2 across configurations. Articulated changes little (-3.0 to $+6.1$), while Reconstruction gains 2.7–8.8 points. Budget sensitivity is therefore both configuration- and task-dependent, and the configuration order changes as MaxSteps grows.

E.5.3 Official CLI Comparison

Configuration. We reran Claude Sonnet 5 High on all 520 cases with the official Claude Code CLI v2.1.181 at high reasoning effort. The run used Blender 5.1.2 and the same samples, task goals, step limits, and evaluator. One assistant response counted as one step. At the limit, we applied the final tool result before evaluation. The official CLI used the same Blender operations and its native image reader. We disabled unrelated shell, web, and file-editing tools.

Completion and scoring. All 520 cases completed and were scored. Five Dynamic cases produced a static scene but no object animation. Three used low-poly references, and two used photo-realistic references. We retained these failure scores.

Table 7: **Claude Code comparison.** Claude Sonnet 5 High under the shared harness and official Claude Code CLI. Higher fixed-reference task scores are better; Δ is CLI minus harness. Paired conditions remain outside Overall.

Task	n	**Harness**	**Claude Code**	Δ
Layout	100	51.9	46.8	-5.1
Layout (multi-view)	100	64.0	43.5	-20.4
Camera	100	27.3	30.7	$+3.4$
Articulated	100	57.8	60.3	$+2.5$
Reconstruction	100	10.5	8.3	-2.2
Dynamic	10	49.9	27.0	-22.9
Dynamic (photo-realistic)	10	39.7	26.7	-13.0
Overall	–	**39.5**	**34.6**	$\mathbf{-4.9}$

Result. The official CLI scores 34.6 Overall, compared with 39.5 under the shared harness (Table 7). The 4.9-point gap is substantially smaller than the task-level variation. The official stack improves Camera by 3.4 points and Articulated by 2.5, while the shared harness leads Reconstruction by 2.2 and Dynamic by 22.9.

Task dependence. The comparison does not show a uniform interface advantage. The official stack performs better on hidden camera state and articulated motion, but worse on multi-view Layout and multi-object Dynamic. Five missing official animations contribute to the Dynamic gap. The larger multi-view and Dynamic differences are consistent with different ways of allocating visual reads, edits, and checks, but they do not isolate one causal mechanism.

Effective interaction budget. The same MaxSteps value does not guarantee the same number or sequence of environment actions. The two stacks package image reading, Blender edits, rendering, context management, and stopping differently. Their scores therefore compare complete agent stacks, not an isolated interface component.

Scope. Both runs used Claude Sonnet 5 High. The official run used native image reading, while the main-table run used the SceneActBench interface. Their system prompts and context policies also differed. The experiment therefore measures stack sensitivity under shared tasks and scoring; it does not isolate one causal interface factor. This is why the main ranking uses one common stack.

E.5.4 Representative Interaction Traces

Aggregate step and tool-call counts do not show how an agent spent its interaction budget. Figure 15 therefore plots three scored episodes as tool-call timelines: input reads, code edits, render checks, tool errors, and the stopping event. Isolated calls are shown as dots, while consecutive calls of the same type are combined into rounded segments to reduce overlap; the exact totals remain listed for each episode. These are illustrative episodes rather than estimates of how often each pattern occurs.

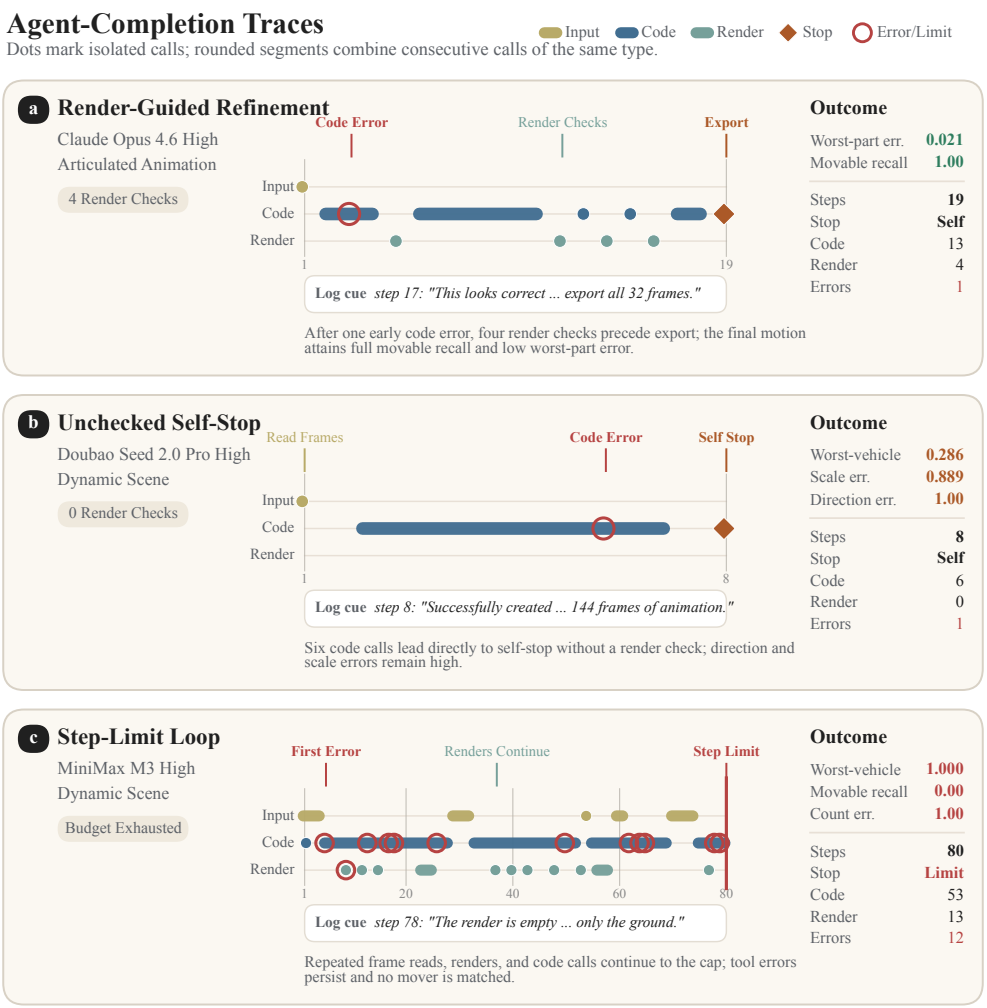


Figure 15: **Representative agent-completion traces.** Each row is one scored episode. Colours distinguish input reads, code edits, and render checks along the agent step axis. Dots denote isolated calls, rounded segments combine consecutive same-type calls, red rings indicate tool errors, and diamonds mark self-stopping. Selected annotations and log excerpts provide trace context; right-side values report exact event totals and task metrics.

Reading the traces. Panel 15a shows a short closed loop: after one early code error the agent makes four render checks and self-stops at step 19 with movable recall 1.00 and MPE 0.021. Panel 15b shows a completed but unchecked run: six code edits and a self-stop at step 8 with no render checks, yet direction error rate 1.00 and scale error 0.889. Panel 15c shows a long unresolved loop: twelve tool errors and thirteen render checks persist to the 80-step limit, with movable recall 0 and MME 1.000.

Implication and scope. These cases show why MaxSteps and raw call counts are insufficient process measures. Render checks help only when they inform later edits and errors are resolved; render activity alone does not guarantee convergence, as panel 15c illustrates. The episodes were selected for legibility and support process-level diagnosis rather than causal or prevalence claims.

E.6 Complete Numeric Record

Table 8 provides the full numeric record for all eleven configurations. It includes task scores and primary, diagnostic, and audit metrics. Each cell reports the mean and sample standard deviation across benchmark instances. Task-score standard deviations use the same per-instance normalised scores as the task means, while undefined diagnostic values are omitted from both statistics. These standard deviations describe variation across cases, not repeated executions. The table supports reproducibility; it does not introduce new claims.

Table 8: **Complete metrics.** Results for all eleven configurations. Cells report the mean with sample SD in grey; **bold** marks the best mean per row. Layout and Dynamic include paired input conditions. Undefined values are excluded; N/A denotes unavailable values. FOV and audit metrics are unscored.

Metric		Doubao H	Opus H	GPT M	GPT H	Qwen H	Gemini H	MiMo H	Kimi R	Step H	Sonnet H	M3 H
Layout Arrangement												
Single-view												
Task score	↑	77.4 ± 10.4	73.5 ± 16.5	72.7 ± 12.2	**84.1** ± 18.1	76.2 ± 11.6	65.4 ± 28.6	79.4 ± 17.0	70.9 ± 21.0	77.5 ± 12.7	51.9 ± 33.1	58.1 ± 27.6
ADD-S	↓	0.905 ± 0.417	1.059 ± 0.659	1.120 ± 0.695	**0.766** ± 1.559	0.954 ± 0.464	6.953 ± 20.03	0.824 ± 0.680	2.547 ± 13.68	0.898 ± 0.508	21.49 ± 115.7	4.188 ± 15.01
Chamfer	↓	1.081 ± 0.544	1.438 ± 0.938	1.314 ± 1.127	**0.929** ± 1.474	1.235 ± 0.694	7.083 ± 19.51	1.064 ± 0.912	3.009 ± 16.38	1.144 ± 0.654	22.75 ± 113.7	4.417 ± 14.28
Position error	↓	1.279 ± 0.516	1.437 ± 0.743	1.506 ± 0.748	**0.903** ± 1.281	1.338 ± 0.559	7.343 ± 20.06	1.132 ± 0.838	2.931 ± 13.71	1.260 ± 0.616	21.94 ± 115.8	4.627 ± 15.06
Placement acc.	↑	0.135 ± 0.134	0.093 ± 0.137	0.101 ± 0.114	**0.446** ± 0.407	0.098 ± 0.122	0.100 ± 0.172	0.274 ± 0.357	0.173 ± 0.294	0.206 ± 0.307	0.080 ± 0.151	0.065 ± 0.101
Multi-view												
Task score	↑	80.1 ± 12.8	82.0 ± 11.7	78.8 ± 9.0	**88.6** ± 11.3	80.3 ± 14.2	74.8 ± 23.0	78.8 ± 21.0	74.4 ± 16.9	76.1 ± 14.5	64.0 ± 29.5	61.2 ± 29.2
ADD-S	↓	0.794 ± 0.513	0.721 ± 0.468	0.847 ± 0.359	**0.456** ± 0.454	0.787 ± 0.569	1.069 ± 1.155	1.704 ± 9.071	1.030 ± 0.710	1.165 ± 2.462	11.07 ± 33.91	6.507 ± 20.34
Chamfer	↓	1.011 ± 0.748	0.930 ± 0.582	1.061 ± 0.413	**0.610** ± 0.587	1.088 ± 0.829	1.423 ± 1.522	1.848 ± 8.932	1.093 ± 0.802	1.418 ± 2.518	11.04 ± 32.92	6.942 ± 20.96
Position error	↓	1.113 ± 0.588	0.992 ± 0.547	1.153 ± 0.430	**0.644** ± 0.634	1.099 ± 0.665	1.389 ± 1.215	1.968 ± 9.116	1.383 ± 0.836	1.268 ± 0.572	11.45 ± 33.97	6.885 ± 20.38
Placement acc.	↑	0.143 ± 0.162	0.186 ± 0.183	0.155 ± 0.138	**0.509** ± 0.405	0.150 ± 0.202	0.147 ± 0.195	0.288 ± 0.352	0.220 ± 0.311	0.156 ± 0.214	0.140 ± 0.248	0.075 ± 0.119
Camera Alignment												
Task score	↑	**34.5** ± 28.7	34.0 ± 26.3	29.6 ± 25.0	26.4 ± 19.0	21.7 ± 24.8	33.9 ± 29.6	25.0 ± 24.5	24.6 ± 22.1	13.2 ± 19.6	27.3 ± 26.4	25.6 ± 27.1
Position error (m)	↓	**4.825** ± 3.119	4.980 ± 3.092	5.404 ± 2.787	5.748 ± 2.934	6.489 ± 3.404	5.272 ± 3.781	6.618 ± 3.823	6.396 ± 3.272	7.191 ± 3.155	6.045 ± 3.634	5.703 ± 2.907
Angular error (deg)	↓	54.49 ± 46.71	51.65 ± 37.67	54.48 ± 43.76	**51.12** ± 27.89	70.92 ± 44.28	53.81 ± 44.16	58.21 ± 34.46	57.23 ± 38.00	78.82 ± 38.78	54.34 ± 33.64	64.14 ± 45.83
Predicted FOV (deg)		39.60 ± 0.000	39.60 ± 0.000	39.78 ± 1.770	39.87 ± 1.878	39.75 ± 1.483	39.60 ± 0.000	39.64 ± 0.451	39.60 ± 0.000	39.94 ± 2.704	40.00 ± 4.040	40.36 ± 5.148
Articulated Animation												
Task score	↑	59.6 ± 17.1	63.7 ± 25.3	62.3 ± 20.9	**73.8** ± 24.8	58.0 ± 25.6	56.5 ± 27.2	49.8 ± 23.5	57.3 ± 19.0	48.3 ± 25.6	57.8 ± 21.2	58.4 ± 23.6
Maximum Part Error	↓	0.404 ± 0.173	0.375 ± 0.292	0.381 ± 0.223	**0.275** ± 0.292	0.603 ± 1.590	0.452 ± 0.314	0.636 ± 1.355	0.442 ± 0.246	0.541 ± 0.323	0.425 ± 0.220	0.428 ± 0.272
Mean Part Error	↓	0.336 ± 0.147	0.277 ± 0.215	0.303 ± 0.177	**0.183** ± 0.177	0.475 ± 1.141	0.360 ± 0.279	0.462 ± 0.704	0.371 ± 0.213	0.457 ± 0.273	0.352 ± 0.190	0.329 ± 0.177
Open-state ADD-S	↓	0.062 ± 0.082	0.057 ± 0.085	0.066 ± 0.098	**0.035** ± 0.056	0.137 ± 0.369	0.082 ± 0.122	0.223 ± 0.184	0.284 ± 1.163	0.278 ± 0.340	0.102 ± 0.138	0.076 ± 0.097
Movable recall	↑	0.051 ± 0.195	0.699 ± 0.348	0.433 ± 0.425	**0.754** ± 0.351	0.522 ± 0.427	0.513 ± 0.418	0.155 ± 0.306	0.169 ± 0.340	0.057 ± 0.193	0.208 ± 0.386	0.381 ± 0.409
False-move rate	↓	**0.200** ± 0.402	0.600 ± 0.492	0.540 ± 0.501	0.520 ± 0.502	0.760 ± 0.429	0.580 ± 0.496	0.263 ± 0.442	0.200 ± 0.402	0.210 ± 0.409	0.200 ± 0.402	0.530 ± 0.502
Type-mismatch rate	↓	0.008 ± 0.080	0.022 ± 0.143	0.017 ± 0.110	0.003 ± 0.033	0.061 ± 0.228	0.055 ± 0.213	0.010 ± 0.075	0.015 ± 0.111	**0.000** ± 0.000	0.008 ± 0.060	0.056 ± 0.222
Reverse-dir rate	↓	0.016 ± 0.116	0.073 ± 0.257	0.098 ± 0.268	0.041 ± 0.181	0.070 ± 0.225	0.053 ± 0.221	0.016 ± 0.108	0.023 ± 0.144	**0.000** ± 0.000	0.013 ± 0.103	0.056 ± 0.222
From-Scratch Reconstruction												
Task score	↑	8.8 ± 6.4	9.8 ± 6.4	10.4 ± 6.5	**12.3** ± 7.8	9.0 ± 6.0	7.1 ± 6.8	9.1 ± 7.0	8.5 ± 6.1	8.6 ± 6.4	10.5 ± 6.4	9.6 ± 6.4
Object F@5% (primary)	↑	0.088 ± 0.064	0.098 ± 0.064	0.104 ± 0.065	**0.123** ± 0.078	0.090 ± 0.060	0.071 ± 0.068	0.091 ± 0.070	0.085 ± 0.061	0.086 ± 0.064	0.105 ± 0.064	0.096 ± 0.064
Region object F@5%	↑	0.062 ± 0.042	0.071 ± 0.047	0.083 ± 0.038	**0.105** ± 0.056	0.063 ± 0.046	0.064 ± 0.046	0.065 ± 0.049	0.072 ± 0.046	0.059 ± 0.040	0.072 ± 0.041	0.076 ± 0.047
Merged-scene F@2%	↑	0.100 ± 0.085	0.122 ± 0.101	0.130 ± 0.077	**0.171** ± 0.120	0.107 ± 0.104	0.099 ± 0.089	0.105 ± 0.096	0.120 ± 0.099	0.098 ± 0.080	0.111 ± 0.087	0.122 ± 0.090
Merged-scene F@5%	↑	0.345 ± 0.149	0.401 ± 0.175	0.434 ± 0.157	**0.464** ± 0.164	0.356 ± 0.196	0.357 ± 0.167	0.373 ± 0.175	0.401 ± 0.184	0.350 ± 0.171	0.398 ± 0.175	0.418 ± 0.156
Merged-scene F@10%	↑	0.664 ± 0.162	0.699 ± 0.166	0.725 ± 0.163	**0.747** ± 0.132	0.650 ± 0.208	0.655 ± 0.182	0.685 ± 0.188	0.707 ± 0.183	0.653 ± 0.188	0.693 ± 0.179	0.718 ± 0.136
Object Point-BERT	↑	0.158 ± 0.092	0.201 ± 0.107	0.167 ± 0.079	**0.218** ± 0.094	0.153 ± 0.088	0.122 ± 0.097	0.173 ± 0.090	0.184 ± 0.104	0.134 ± 0.086	0.180 ± 0.087	0.173 ± 0.097

Continued on next page

Table 8 (continued)

Metric		Doubao H	Opus H	GPT M	GPT H	Qwen H	Gemini H	MiMo H	Kimi R	Step H	Sonnet H	M3 H
Scene Point-BERT	↑	0.351 ± 0.107	0.397 ± 0.121	0.350 ± 0.097	**0.426** ± 0.134	0.376 ± 0.122	0.345 ± 0.112	0.362 ± 0.122	0.356 ± 0.111	0.351 ± 0.119	0.385 ± 0.116	0.376 ± 0.115
Object coverage	↑	0.055 ± 0.102	0.052 ± 0.101	0.089 ± 0.103	**0.130** ± 0.132	0.049 ± 0.099	0.051 ± 0.099	0.057 ± 0.104	0.072 ± 0.110	0.037 ± 0.081	0.055 ± 0.095	0.072 ± 0.121
Match rate	↑	0.839 ± 0.232	0.846 ± 0.243	0.841 ± 0.217	**0.898** ± 0.162	0.775 ± 0.305	0.659 ± 0.383	0.857 ± 0.263	0.807 ± 0.289	0.769 ± 0.248	0.862 ± 0.209	0.835 ± 0.242
Mean centroid error	↓	0.591 ± 0.173	0.542 ± 0.173	**0.533** ± 0.152	0.586 ± 0.188	0.551 ± 0.164	0.553 ± 0.220	0.575 ± 0.208	0.553 ± 0.180	0.558 ± 0.205	0.581 ± 0.171	0.555 ± 0.182
Chamfer (aligned)	↓	0.416 ± 0.123	0.404 ± 0.143	0.381 ± 0.134	**0.369** ± 0.111	0.457 ± 0.202	0.436 ± 0.136	0.429 ± 0.193	0.393 ± 0.139	0.440 ± 0.153	0.409 ± 0.127	0.388 ± 0.111
Dynamic Scene												
Low-poly												
Task score	↑	**70.7** ± 18.8	63.2 ± 22.5	68.5 ± 31.0	46.7 ± 31.8	66.0 ± 19.4	63.9 ± 30.1	43.7 ± 29.6	44.8 ± 25.8	57.9 ± 23.5	49.9 ± 18.4	41.4 ± 35.0
Maximum Mover Error	↓	0.471 ± 0.412	0.583 ± 0.393	0.738 ± 1.267	0.746 ± 0.414	**0.441** ± 0.350	0.527 ± 0.496	0.827 ± 0.365	0.855 ± 0.401	0.712 ± 0.465	1.000 ± 0.000	0.755 ± 0.396
Average Mover Error	↓	0.322 ± 0.234	0.436 ± 0.343	0.585 ± 1.078	0.742 ± 0.421	**0.293** ± 0.182	0.371 ± 0.307	0.721 ± 0.386	0.765 ± 0.368	0.665 ± 0.424	1.000 ± 0.000	0.706 ± 0.399
Movable recall	↑	0.950 ± 0.158	0.710 ± 0.418	0.895 ± 0.257	0.300 ± 0.483	0.935 ± 0.142	**0.950** ± 0.158	0.325 ± 0.442	0.325 ± 0.472	0.500 ± 0.527	0.000 ± 0.000	0.350 ± 0.474
Mover-count error	↓	0.175 ± 0.334	0.340 ± 0.409	0.555 ± 0.808	0.700 ± 0.483	0.315 ± 0.780	**0.120** ± 0.210	0.675 ± 0.442	0.675 ± 0.472	0.500 ± 0.527	1.000 ± 0.000	1.150 ± 1.415
Direction-error rate	↓	0.367 ± 0.457	0.092 ± 0.149	0.192 ± 0.356	**0.000** ± 0.000	0.308 ± 0.405	0.125 ± 0.270	0.000 ± 0.000	0.067 ± 0.211	0.067 ± 0.211	0.000 ± 0.000	0.100 ± 0.316
Path-shape error	↓	**0.179** ± 0.157	0.311 ± 0.375	0.538 ± 1.018	0.770 ± 0.388	0.233 ± 0.185	0.294 ± 0.275	0.657 ± 0.447	0.689 ± 0.418	0.650 ± 0.467	1.000 ± 0.000	0.721 ± 0.372
Heading error	↓	0.098 ± 0.178	0.058 ± 0.156	0.223 ± 0.311	0.006 ± 0.014	0.164 ± 0.199	0.183 ± 0.315	0.004 ± 0.010	0.005 ± 0.012	0.139 ± 0.314	**0.000** ± 0.000	0.076 ± 0.143
Scale error	↓	0.890 ± 0.726	0.672 ± 0.678	0.559 ± 0.727	0.175 ± 0.401	0.834 ± 0.751	1.375 ± 0.928	0.140 ± 0.263	0.226 ± 0.630	0.289 ± 0.592	**0.000** ± 0.000	0.202 ± 0.390
Size error	↓	0.581 ± 0.792	**0.288** ± 0.273	0.727 ± 1.127	0.571 ± 0.398	0.461 ± 0.264	0.986 ± 0.900	0.576 ± 0.919	0.947 ± 0.772	0.536 ± 0.514	1.294 ± 1.140	1.029 ± 0.948
Mover-size error	↓	0.954 ± 0.881	0.536 ± 0.300	1.452 ± 1.472	**0.352** ± 0.384	0.667 ± 0.514	1.086 ± 0.835	0.509 ± 0.252	0.549 ± 0.182	0.639 ± 0.449	N/A	0.670 ± 0.145
Layout error	↓	**0.150** ± 0.089	0.152 ± 0.117	0.235 ± 0.357	0.320 ± 0.302	0.238 ± 0.234	0.335 ± 0.534	0.299 ± 0.373	0.281 ± 0.269	0.164 ± 0.094	1.330 ± 3.155	0.426 ± 0.428
Photo-realistic												
Task score	↑	65.8 ± 18.4	63.3 ± 22.7	**70.6** ± 20.3	64.0 ± 23.7	65.9 ± 21.7	55.5 ± 26.2	38.3 ± 26.5	55.4 ± 19.6	47.6 ± 17.3	39.7 ± 17.9	52.2 ± 22.3
Maximum Mover Error	↓	0.537 ± 0.516	0.616 ± 0.414	**0.501** ± 0.505	0.596 ± 0.439	0.570 ± 0.471	0.701 ± 0.475	0.904 ± 0.304	0.710 ± 0.375	0.852 ± 0.313	0.920 ± 0.253	0.765 ± 0.380
Average Mover Error	↓	0.367 ± 0.338	0.430 ± 0.349	**0.313** ± 0.260	0.566 ± 0.462	0.382 ± 0.283	0.502 ± 0.344	0.878 ± 0.306	0.690 ± 0.400	0.809 ± 0.305	0.920 ± 0.253	0.691 ± 0.405
Movable recall	↑	0.880 ± 0.316	0.677 ± 0.405	**0.930** ± 0.164	0.500 ± 0.527	0.910 ± 0.191	0.730 ± 0.416	0.133 ± 0.322	0.400 ± 0.516	0.250 ± 0.408	0.100 ± 0.316	0.380 ± 0.494
Mover-count error	↓	**0.120** ± 0.316	0.407 ± 0.369	0.220 ± 0.343	0.700 ± 0.675	0.140 ± 0.227	0.745 ± 1.204	0.867 ± 0.322	0.600 ± 0.516	0.750 ± 0.408	0.900 ± 0.316	0.653 ± 0.457
Direction-error rate	↓	0.158 ± 0.320	0.058 ± 0.124	0.108 ± 0.184	0.050 ± 0.158	0.100 ± 0.316	0.025 ± 0.079	0.000 ± 0.000	0.167 ± 0.360	**0.000** ± 0.000	0.000 ± 0.000	0.100 ± 0.316
Path-shape error	↓	0.279 ± 0.271	0.355 ± 0.363	0.229 ± 0.175	0.591 ± 0.454	**0.204** ± 0.191	0.416 ± 0.329	0.861 ± 0.313	0.708 ± 0.396	0.739 ± 0.351	0.956 ± 0.138	0.732 ± 0.365
Heading error	↓	0.081 ± 0.099	0.069 ± 0.155	0.378 ± 0.442	0.035 ± 0.066	0.204 ± 0.323	0.142 ± 0.187	0.011 ± 0.036	0.040 ± 0.121	0.144 ± 0.304	**0.000** ± 0.000	0.063 ± 0.190
Scale error	↓	0.938 ± 0.999	0.402 ± 0.513	1.019 ± 0.904	0.149 ± 0.262	1.120 ± 0.773	0.741 ± 0.893	0.249 ± 0.724	0.080 ± 0.162	0.157 ± 0.275	**0.000** ± 0.000	0.155 ± 0.395
Size error	↓	0.664 ± 0.829	**0.287** ± 0.119	0.915 ± 1.128	0.671 ± 0.668	0.631 ± 0.847	0.953 ± 0.872	1.077 ± 0.390	1.142 ± 1.108	0.611 ± 0.380	1.520 ± 0.826	0.882 ± 0.575
Mover-size error	↓	1.150 ± 1.104	0.696 ± 0.144	0.984 ± 0.942	0.524 ± 0.358	0.880 ± 0.897	0.772 ± 0.956	1.462 ± 0.437	0.503 ± 0.272	0.663 ± 0.157	**0.446** ± –	0.508 ± 0.273
Layout error	↓	0.213 ± 0.204	**0.118** ± 0.081	0.143 ± 0.056	0.125 ± 0.068	0.166 ± 0.093	0.245 ± 0.263	0.329 ± 0.357	0.182 ± 0.067	0.196 ± 0.078	0.286 ± 0.197	0.191 ± 0.088

E.7 Task-Score Correlations

Table 9 reports the Pearson correlations defined in Section D.3. The strongest off-diagonal association is between Articulated and Reconstruction ($r = 0.72$); Dynamic and Reconstruction are weakly negative ($r = -0.28$). With only eleven evaluated configurations, these values are descriptive and do not establish task dependence or a population-level relationship.

Table 9: **Task-score correlations.** Pearson r across the eleven configurations using the five fixed-reference task scores.

	Layout	Camera	Articulated	Reconstruction	Dynamic
Layout	1.00	−0.16	0.17	0.12	0.22
Camera	−0.16	1.00	0.46	0.00	0.36
Articulated	0.17	0.46	1.00	0.72	0.06
Reconstruction	0.12	0.00	0.72	1.00	−0.28
Dynamic	0.22	0.36	0.06	−0.28	1.00

F Prompt Templates

Each task sends two messages to the agent. The system prompt defines the role, world rules, and tool use. The task prompt gives scene inputs and instructions. Both prompts appear verbatim below. Braced placeholders are filled when each scene is built. Output paths are filled at run time.

The quoted prompts keep the harness wording. This includes *object*, *piece*, and *components*. The asset–object rule in Section 2 applies outside the quoted prompts. Legacy internal identifiers such as "TASK 6" and "T7" distinguish the low-poly and photo-realistic Dynamic conditions; they do not denote additional benchmark tasks. Some prompts name audit metrics beyond the primary scores, whose roles are defined in Appendix C.3. Blue cards show system prompts; amber cards show task prompts.

F.1 Layout

Both Layout conditions use one system prompt. The task prompt lists either one view or all views.

System prompt

```
You are a 3D scene-layout agent controlling Blender through MCP tools.

The Blender scene has been preloaded with several furniture objects named
object_00, object_01, ... Each object's geometry is CANONICAL: centered at the
world origin and reset to a neutral (un-rotated) orientation. They are all piled
near the origin and overlapping - none is in its correct place yet.

Your job: reconstruct the original room layout shown in the reference image(s).
For EACH object, decide its correct world position and its rotation about the
vertical (Z) axis, then move/rotate it there using execute_blender_code (bpy).

Conventions (IMPORTANT):
- World is Z-up. Objects rest on the floor at Z=0; +Z is height.
- Each object keeps its real-world metric size (do NOT scale objects).
- Rotation is yaw about the Z axis only (no tilting).
- A reference camera pose is given for each reference image (position + look
  direction in this same world frame). Use it to disambiguate
  left/right/front/back so your layout is in the correct global orientation.

How to place an object (use this exact method in execute_blender_code):
    import bpy, mathutils
    obj = bpy.data.objects["object_03"]
    T = mathutils.Matrix.Translation((x, y, z))     # target world location (Z-up)
    R = mathutils.Matrix.Rotation(yaw, 4, 'Z')      # yaw in radians about Z
    obj.matrix_world = T @ R
  Set the FULL matrix_world like this. Do NOT rely on obj.rotation_euler /
  obj.location piecemeal - in this headless context they may not take effect.
```

System prompt (continued)

```
Workflow:
1. Inspect objects with get_scene_info / get_object_info (geometry, size).
2. Study the reference image(s) to infer what each object is and where it goes.
3. Place every object with execute_blender_code, setting obj.matrix_world = T @ R.
4. Render with render_scene_view to compare your layout to the reference. Also
   create extra cameras (e.g. top-down and side views) and render from them -
   the reference is a 2D projection, so depth errors and overlapping objects
   only show from other angles. Verify the 3D layout, then refine.
5. When confident, stop with a brief summary. Do not ask questions.

You are graded on how closely your final layout matches the true scene
(per-object pose + overall arrangement). Only the real Blender transforms count.
```

Task prompt (single-view form)

```
Reconstruct this room. There are {N_OBJECTS} objects in the scene
(object_00..object_{N-1}), all canonical (centered, un-rotated) and piled at the
origin.
World frame: Z-up, meters.
Reference image: {REFERENCE_IMAGE_PATH}
Reference camera (world frame, Z-up): position={CAM_POSITION},
look_dir={CAM_LOOK_DIR}, fov_x={CAM_FOV_X_DEG} deg.
A camera object named '__ref__' is already created at this exact pose and set as
the active camera, so you can call render_scene_view(camera='__ref__').
Place every object at its correct world location and yaw to match the reference
image. Keep object sizes unchanged. Render to check; also create top-down and
side cameras to catch depth errors and overlaps. Refine, then stop.

--- multi-view task prompt lists all views instead of one ---
You are given {N_VIEWS} reference images from different camera views
(fov_x={CAM_FOV_X_DEG} deg), with poses:
  - {IMAGE_PATH}  cam pos={CAM_POSITION} look_dir={CAM_LOOK_DIR}
  ... (one line per view)
```

F.2 CAMERA

Reporting note. The historical prompt mentions LPIPS, CLIP, and PSNR. These values are audit checks. Only PE and AE enter the reported Camera score.

System prompt

```
You are a 3D camera-pose agent controlling Blender through MCP tools.

The scene is already correctly arranged (ground-truth furniture layout is
loaded). You are given ONE reference image taken from an unknown camera pose.
Your job: find a camera pose that reproduces the reference image as closely
as possible.

Workflow:
1. Inspect the scene (get_scene_info).
2. Create/position a camera via execute_blender_code, set its world matrix and
   lens. Render with render_scene_view and compare to the reference image.
3. Iterate: adjust camera position/orientation to match the reference view.
4. When the rendered view matches, stop and report the final camera pose
   (position + look direction or full matrix_world). Do not ask questions.

You are graded on camera-pose error vs the true camera, and image similarity
(LPIPS/CLIP/PSNR) between your render and the reference.
```

Task prompt

```
The scene is correctly arranged. Find a camera pose that reproduces this
reference image:
{REFERENCE_IMAGE_PATH}
World frame: Z-up, meters.
Create a camera, set its matrix_world and lens (fov_x={CAM_FOV_X_DEG} deg),
render with render_scene_view, and iterate until your render matches the
reference. Report the final camera position and look direction. Full 6-DOF
camera movement allowed.
```

F.3 ARTICULATED

System prompt

```
You are a 3D animation assistant controlling Blender through MCP tools. The
scene contains a static articulated object (parts named mesh_000, mesh_001, ...
with NO semantic labels). You are given a SEQUENCE OF 32 IMAGES that are the
consecutive frames of a short video (in order). The video shows the object's
movable part(s) opening and then closing. Infer which mesh object(s) move and
how, then reproduce the motion and export one GLB per frame. Use get_scene_info
/ get_object_info to inspect objects and their bounding boxes,
execute_blender_code to transform whole OBJECTS (object.location /
object.rotation_euler / object.matrix_world - never edit individual vertices)
and to export frames via bpy.ops.export_scene.gltf. You may call
render_scene_view to visually verify. Do not ask questions; when all frames are
exported, stop with a one-line summary.
```

Task prompt

```
The 32 reference images are the CONSECUTIVE FRAMES of a video, shown strictly in
order (image 1 = video frame 1, image 32 = video frame 32). Watch the whole
sequence: it shows this object's movable part(s) going from fully CLOSED to
fully OPEN and back to CLOSED. There are NO semantic part labels; you must decide
from the images which part(s) move.
Your job: REPRODUCE this motion on the imported GLB and export one GLB per frame.
Requirements:
1. Inspect the scene for the mesh objects and their bounding boxes.
2. From the image sequence, decide WHICH mesh object(s) move and HOW each moves
   (rotation about which hinge/axis, or translation along which direction, and
   how far). NOTE: more than one part may move - check every part.
3. Produce exactly 32 frames matching the video: frame i corresponds to
   reference image i+1. Motion goes CLOSED -> fully OPEN -> CLOSED.
4. For each frame i (0..31): pose the moving part(s), then export the WHOLE
   object to '{AGENT_FRAMES_DIR}/agent_frame_%04d.glb' % i via
   bpy.ops.export_scene.gltf(..., use_selection=False, export_format='GLB').
   Transform whole OBJECTS only; never edit/reorder vertices.
5. You may render_scene_view to compare against the reference frames.
Call os.makedirs('{AGENT_FRAMES_DIR}', exist_ok=True) first.
```

F.4 RECONSTRUCTION

System prompt

```
You are a procedural 3D modeling expert. Given multiple reference images of an
indoor scene, you reconstruct EVERY furniture piece in Blender by writing bpy
code, matching its geometry AND its visible color.

# Input
- N reference images of the SAME room from different camera viewpoints (each
  with a known camera pose). Cross-reference views to resolve depth, proportions,
  occluded structure, and hidden sides.
- The Blender scene starts EMPTY. The reference images are NOT available to your
  script at runtime.

# Reading the references - before you build
1. Identify each furniture piece visible in any view. Count them.
2. For each piece, infer category/shape, dimensions in METERS, position and yaw,
   symmetries, repeating sub-parts, distinctive ornament.
3. Reproduce quantitative cues (4 legs, 3 drawers, 5 cords) EXACTLY.
4. Read the dominant Base Color of every piece.

# Geometry quality bar
Compose primitives + bmesh edits + modifiers; use array/mirror for repeating
parts; exploit symmetry. Keep under a few hundred thousand vertices. Build at
real-world scale (meters), every piece on the floor (lowest Z = 0). Z-up.

# What NOT to build
- NO room shell (floor/walls/ceiling/windows), NO decorative props.
- DO NOT reproduce textures/patterns; only solid Base Color.

# Materials / placement
Attach a Principled BSDF with correct Base Color; set the FULL world matrix
```

System prompt (continued)

```
(T @ R) for placement.

# Workflow - iterate until satisfied
Build, then render_scene_view at reference camera poses and compare, walking a
priority list: missing pieces > silhouette > proportions > count of repeating
parts > misalignment > detail > color. Fix priority-1..6 issues; stop when only
minor color drift remains.

# Grading
Per-object f-score and Chamfer, Point-BERT cosine, position error, and multi-view
rendered fidelity. No points for rooms, walls, or decorations.
```

Task prompt

```
Reconstruct this indoor scene in 3D from {N_VIEWS} reference images of the SAME
ROOM taken from different camera viewpoints (Blender starts empty).
World frame: Z-up, meters. fov_x = {CAM_FOV_X_DEG} deg.
Reference images and their camera poses:
  - view {i}: {IMAGE_PATH}
      cam pos = {CAM_POSITION}, look_dir = {CAM_LOOK_DIR}, up = {CAM_UP}
  ... (one block per view)
Step 1 - IDENTIFY every furniture piece; count them; read Base Color.
Step 2 - infer size, world position and yaw; plan geometry.
Step 3 - BUILD with execute_blender_code.
Step 4 - RENDER at the reference poses; compare by priority; fix and re-render.
   Also create EXTRA cameras (top-down, side) to catch floating/sunk pieces.
Step 5 - Stop when satisfied; do not ask questions.
REMINDERS: Furniture only. Real-world meters; lowest Z = 0. Reproduce exact
counts. Use bmesh/mirror/array for detail.
```

F.5 DYNAMIC

Both Dynamic conditions use the same prompts. The photo-realistic condition adds one note about the reference style. The task remains unchanged.

System prompt

```
You are a 3D dynamic-scene assistant controlling Blender via MCP. Your job: from
reference frames showing a scene with moving vehicles, reproduce the scene in
Blender. (1) Import static objects (roads, buildings, lamps, trees) and vehicles
from a components folder and place them as in the reference. (2) Drive each
moving vehicle with an animation (keyframes on location/rotation) reproducing its
trajectory. (3) Export the entire animated scene to a single glb. To VIEW THE
REFERENCE, call the read_reference_frames tool (it returns the actual frame
images) - do not load reference PNGs onto planes. To SELF-CHECK, call
render_scene_view from cameras you create (the reference viewpoint plus extra
top-down/side angles). Use execute_blender_code for scene building; do not
hand-edit vertices. When done, stop with one short summary line.
```

Task prompt (photo-realistic note included)

```
TASK 6 - Dynamic Scene Reproduction.
HOW TO SEE THE REFERENCE: Call read_reference_frames with a list of frame numbers
(1..{N_FRAMES}, max 16 per call) to get the ACTUAL reference images. Start with
~8 evenly spaced frames to understand the motion, then zoom in. Video is {FPS}
fps, {N_FRAMES} frames total.
COMPONENTS LIBRARY (static + vehicle glb files) at: {COMPONENTS_DIR}/
Available glbs: {COMPONENTS_LIST}
WHAT TO DO:
  1. Call read_reference_frames to understand the static layout and how each
     object moves over the frames.
  2. Clear the scene and import the static components, placing each as in the
     reference. BASE CONVENTION: ground plane at Z=0; primary axis +X; Y lateral;
     Z height.
  REFERENCE CAMERA (world frame, Z-up): position={CAM_POSITION},
     look_dir={CAM_LOOK_DIR}, fov_x={CAM_FOV_X_DEG} deg. Use it to back-project
     image positions to world meters so absolute SIZE and placement match.
  OBJECT SIZE: scale each imported glb so its real-world size matches the image.
  3. Import each moving object's glb; place at its starting position.
  4. Animate each mover with {N_FRAMES} keyframes reproducing its trajectory
```

Task prompt (photo-realistic note included) (continued)

```
     (object.location / rotation_euler keyframes; do NOT bake to vertices).
  5. SELF-CHECK with render_scene_view at the reference camera pose and extra
     angles; fix and re-render until correct.
  6. Export the entire animated scene as a single glb via
     bpy.ops.export_scene.gltf(..., export_animations=True) to {AGENT_GLB_PATH}.
Your output is the single animated glb at {AGENT_GLB_PATH}.

--- photo-realistic variant prepends: ---
[T7 - REALISTIC-RENDER ABLATION] The reference frames are a PHOTO-REALISTIC
render (real lighting/shadows/materials) of the SAME scene. Task and output are
identical.
```