CODE4SCENE: BENCHMARKING CODING AGENTS FOR CONSTRUCTING AND EDITING 3D SCENES

Anonymous authors
Paper under double-blind review

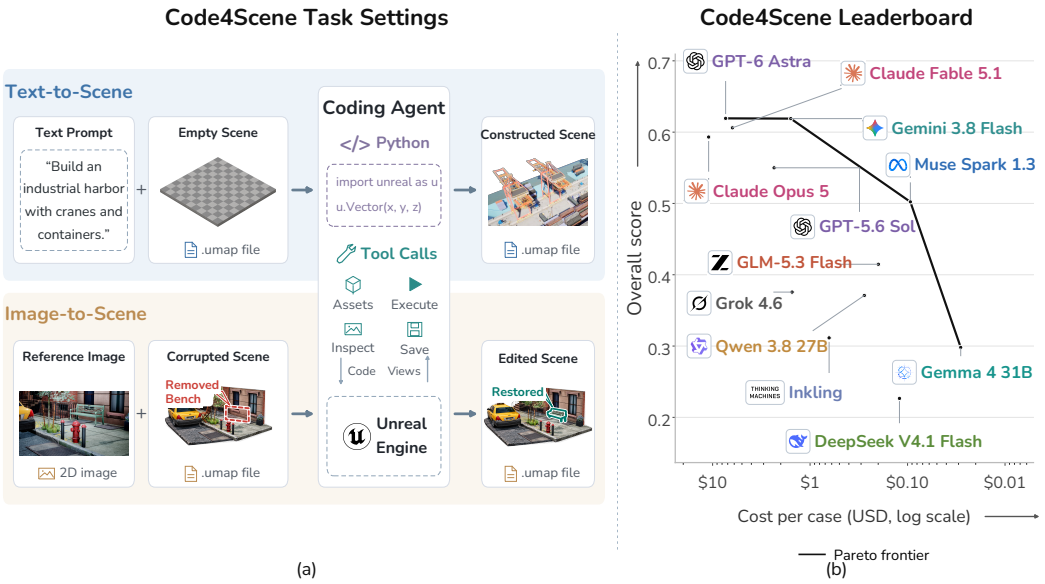


Figure 1: Code4Scene tasks and leaderboard. (a) Coding agents use Python and tool calls to construct or edit persistent Unreal Engine scenes based on text prompt or image reference. (b) Overall score (Table 2) for agent configurations against estimated cost per case on the public set (Appendix D.5). The solid line marks the Pareto frontier among the displayed agent configurations.

ABSTRACT

Frontier coding agents can now write and execute code that authors 3D environments, but whether they reliably understand 3D structure and precisely control scene state remains unclear. The resulting 3D scene is a persistent, executable artifact: a convincing render can hide incorrect spatial relations, intersecting objects, or unintended modifications. We introduce **Code4Scene**, a benchmark of 190 Unreal Engine cases built from human-assembled scenes that evaluates coding agents on two complementary settings under a shared execution interface. *Construction* tests scene-level spatial reasoning from open-ended language specifications, where many realizations are valid; *editing* tests precise control of scene state, where the agent must recover the target scene from reference images while preserving everything else. Rather than scoring code or rendered views, Code4Scene evaluates the engine-native scene produced by the agent for task fulfillment, artifact integrity, and static physical validity, with edits additionally compared against withheld ground truth. Across 14 coding-agent configurations on the 95-case public set, construction and editing performance are strongly correlated but not interchangeable (Spearman $\rho = 0.78$): Claude Fable 5.1 leads construction, Gemini 3.8 Flash leads editing, and GPT-6 Astra narrowly leads overall. Spatial Composition is the weakest construction category for every agent, while editing remains imprecise: the best Repair F1 is only 0.527, and 35.8% of edits that fully recover the target still introduce unintended changes elsewhere in the scene. These results expose a gap between plausible 3D generation and reliable spatial reasoning and state control.

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.

1 INTRODUCTION

Frontier coding agents are beginning to directly author 3D environments. Recent systems can write and execute scene-editing code, operate graphics engines, and iteratively refine scenes from visual feedback. Demonstrations such as GPT-6 Astra constructing a model in Blender and transferring it into a walkable Unreal Engine scene (OpenAI, 2026b) suggest rapid progress in spatially grounded generation. This capability could have broad value in game development, industrial design, and robotics simulation. But its reliability remains unclear: **can frontier coding agents reliably understand 3D structure and precisely control executable scene state?**

3D scene authoring makes this question difficult to answer because the output is not just code or an image, but a persistent artifact whose state must remain correct after execution. An agent must translate textual or visual intent into engine operations, inspect the resulting changes, and revise the scene without breaking what already works. A convincing render can still hide incorrect spatial relations, unsupported or intersecting objects, broken dependencies, or unintended modifications.

Recent work has begun to automate 3D scene creation with language models and coding agents, while new benchmarks evaluate scene generation, reconstruction, editing, and executable worlds (Kang et al., 2026a; Liu et al., 2026a; Gu et al., 2025; Xu et al., 2026; Lu et al., 2026). These evaluations, however, differ substantially in engine, scene representation, agent interface, and evaluation target, ranging from rendered outputs to structured scene states and engine-specific artifacts. It remains difficult to compare general-purpose coding agents in a common setting or distinguish visual plausibility from spatial correctness, physical validity, and precise state control.

To evaluate these capabilities systematically, we study two complementary aspects of 3D scene authoring: **construction** and **editing**. Construction probes scene-level spatial reasoning: given an open-ended language specification, an agent must compose a coherent scene, and many realizations may be valid. Editing probes precise control of scene state: given an existing scene and reference images of the desired result, the agent must infer the required changes and apply them while preserving everything else. The two settings therefore capture complementary aspects of 3D scene authoring and require different notions of correctness.

We introduce **Code4Scene**, a benchmark of 190 Unreal Engine cases for evaluating coding agents on 3D scene construction and editing under a shared execution interface. Rather than scoring only code or rendered views, Code4Scene evaluates the persistent engine-native scene produced by the agent, including task fulfillment, artifact integrity, and static physical validity. Construction is evaluated by semantic and spatial adherence to the specification, while editing is compared against withheld ground truth using actor-level Repair F1. This lets us distinguish scenes that merely look plausible from those that are correct, valid, and precisely controlled.

We evaluate 14 coding-agent configurations on the 95-case public set. Construction and editing performance are strongly correlated but not interchangeable (Spearman $\rho = 0.78$): Claude Fable 5.1 leads construction, Gemini 3.8 Flash leads editing, and GPT-6 Astra narrowly leads overall. **Spatial Composition is the weakest construction category for every agent**, and even the best agent's (GPT-6 Astra) Spatial Composition score reaches only 0.543. Importantly, 20.6% of requirements in the Spatial Composition family whose relevant entities are confirmed present are still judged mismatched, showing that generating the right elements does not imply arranging them correctly. Editing remains difficult as well, with the best Repair F1 reaching only 0.527; even among edits that fully recover the target, 35.8% still introduce unintended changes elsewhere in the scene. Together, these results expose a clear gap between generating plausible 3D content and reliably reasoning about its spatial structure and controlling its state.

2 CODE4SCENE BENCHMARK

2.1 TASK SETTINGS

Code4Scene formulates 3D scene authoring as *agentic code generation*. Rather than prescribing procedural rules (Deitke et al., 2022; Raistrick et al., 2023), layout synthesis (Feng et al., 2023; Yang et al., 2024b), or text-to-3D generation (as benchmarked by He et al., 2024), we evaluate a general-purpose coding agent that operates Unreal Engine, writes and executes code, inspects the resulting scene, and iteratively revises it. This formulation emphasizes closed-loop interaction with

Table 1: Comparison of related benchmarks and their reported coverage. Input references specify the desired outcome and exclude initial-scene observations. Open-ended settings admit multiple valid scene realizations; ground-truth-based (GT-based) settings use a hidden target state for evaluation. Full related work discussions appear in Appendix E.

Benchmark	Artifact	Input		Task Setting		Evaluation		
		Text	Images	Open-ended	GT-based	Semantic	Physical	Preservation
BlenderGym (2025)	Script		✓		✓	✓		
SceneActBench (2026)	GLB/pose		✓		✓	✓		
4DBuildBench (2026a)	.blend	✓		✓		✓	✓	
CutsceneBench (2026)	Sequence	✓		✓	✓	✓		
WorldCoder-Bench (2026)	HTML	✓		✓		✓	✓	
AuthorBench (2026)	JSON	✓		✓		✓	✓	✓
VWE-Bench (2026)	Asset map	✓		✓	✓	✓	✓	✓
Code4Scene (ours)	.umap	✓	✓	✓	✓	✓	✓	✓

the engine and produces an engine-native scene (.umap) that can be reopened, edited, and simulated, rather than only a rendered image, mesh, or scene specification. Our benchmark evaluates the final saved scene while recording the agent's interaction trajectory (Appendix A.1).

Within this formulation, we consider two complementary scene-authoring activities: creating a new scene from textual requirements and modifying an existing scene to match a reference. The first tests whether an agent can turn open-ended intent into a spatially coherent scene, while the second tests whether it can infer the required changes and apply them without disrupting the rest of the scene. Code4Scene instantiates these as **Text-to-Scene Construction** and **Image-to-Scene Editing** under the same engine and agent interface.

Formally, each case derives from a human-assembled scene $\mathcal{L}^\star$. From it we build an asset catalog $\mathcal{C}$, an initial scene $\mathcal{L}^0$, agent-visible evidence E, and, for editing, evaluator-only references H. Under a shared interface $\mathcal{A}$ and a resource budget B, an agent configuration produces the submitted scene through $f : (\mathcal{L}^0, \mathcal{C}, E, \mathcal{A}, B) \to \hat{\mathcal{L}}$. The two settings differ in $\mathcal{L}^0$, E, and H.

Text-to-Scene Construction. The agent starts from an empty scene and receives a scene description and the asset catalog ($\mathcal{L}^0$ is blank, E is the description, and $H = \varnothing$). The description admits many valid scenes, so $\mathcal{L}^\star$ guides prompt writing and the catalog but is not a reconstruction target. The agent must identify the requested entities, counts, attributes, and style, and match each request to assets in the catalog. It must place objects so that the stated spatial relations hold, keep them supported and free of interpenetration, and save a scene whose dependencies resolve. In our test set, a prompt contains 58 checkable requirements on average (up to 92). These demands motivate complementary checks of prompt fulfillment, physical plausibility, and artifact validity, captured by the *Semantic Verifier*, *Physical Safety*, and *Candidate Integrity*, respectively (Sections 2.3 and 2.4).

Image-to-Scene Editing. The agent receives a corrupted copy of a human-assembled scene ($\mathcal{L}^0$) and reference views of the original scene (E). H contains the original scene, held-out views, and its serialized state, accessible only to the evaluator (Section 2.3). The agent must compare the references with a scene containing approximately 95 to 6,700 actors and identify objects that were removed, inserted, duplicated, replaced, moved, rotated, rescaled, or mirrored. It must restore the intended scene state while preserving content that does not require repair. Restored objects must meet tight tolerances: 5 cm in position, 5° in rotation, and 5% in scale. Missed targets lower recall, while incorrect target states and collateral edits lower precision (Section 2.4). The target is specified through images, so the agent must infer the required changes from the reference views.

2.2 Data Curation

Code4Scene derives construction and editing tasks from human-assembled Unreal Engine scenes to reflect the complexity of scene authoring in practice (Figure 2). The benchmark contains 190 cases: 30 Text-to-Scene Construction tasks and 160 Image-to-Scene Editing tasks, comprising 80 indoor and 80 outdoor cases. Each case involves a long-horizon authoring task in a production-scale scene, with construction runs allowing up to 800 tool calls and 180 minutes. Every case is manually curated and reviewed for solvability. We therefore prioritize realistic, verifiable tasks over maximizing case count. The benchmark is comparable in size to other execution-based 3D benchmarks (Table 1), including

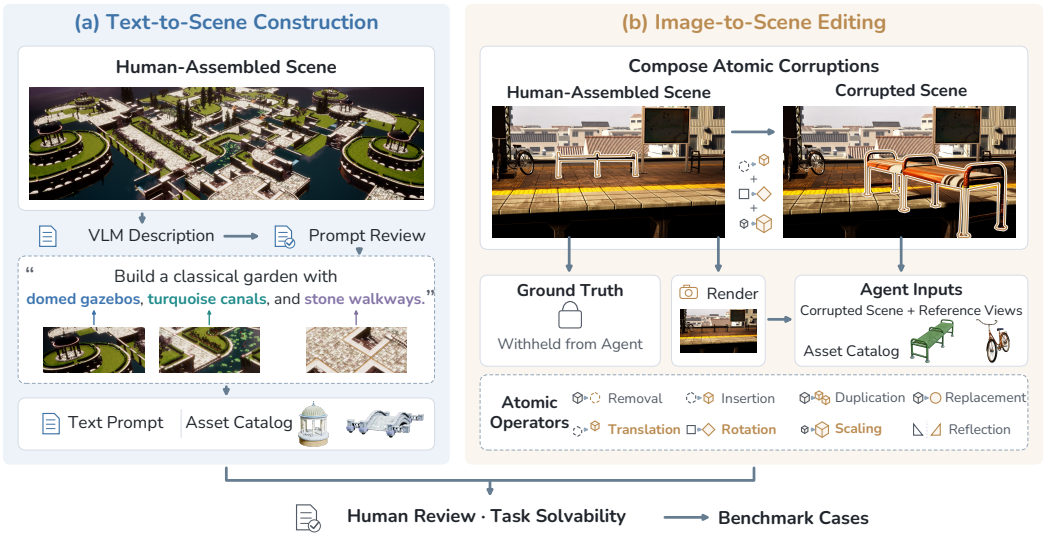


Figure 2: Task construction. Text-to-Scene prompts describe human-assembled scenes. Image-to-Scene inputs are corrupted copies, with the human-assembled scene retained as ground truth. Human review checks task solvability in both settings.

4DBuildBench (50 scenes), CutsceneBench (65 scenarios), and BlenderGym (245 start–goal pairs) (Liu et al., 2026a; He et al., 2026; Gu et al., 2025).

Human-assembled scenes. We source benchmark scenes from 522 Fab content packs marked *Allows usage with AI: Yes* (Epic Games, 2026). From these packs, we manually select Unreal Engine scenes created by human artists, requiring high visual quality, non-trivial compositional complexity, and assets that can be loaded and identified reproducibly. The selected scenes span historical, urban, industrial, domestic, and stylized environments, ranging from roughly 95 to 6,700 actors. Our 30 Text-to-Scene cases come from 30 distinct content packs, while the 160 Image-to-Scene cases comprise eight editing tasks for each of 20 source scenes (10 indoor and 10 outdoor). Human-assembled scenes expose agents to irregular layouts and intentional object arrangements found in production environments; for editing, they also provide exact ground truth for the intended repair.

Text-to-Scene task construction. We ground each text prompt in the assets and composition of a human-assembled scene. We first render screenshots and use a vision-language model (VLM) to generate an initial description. Two reviewers independently verify the description against the scene and consolidate their corrections into the final prompt. At test time, the agent starts from an empty scene and receives only the text prompt and the corresponding asset catalog. The human-assembled scene guides prompt and asset selection, while the prompt defines the construction task and allows multiple valid solutions. We automatically extract a fixed set of requirements from each prompt and reuse it across submissions, so all agents face the same checklist (Appendix C.4).

Image-to-Scene task construction. Image-to-Scene evaluates fine-grained editing against an exact ground-truth target. We retain the human-assembled scene as withheld ground truth and create each test input by applying one or more atomic corruptions to a copy, such as *removal*, *insertion*, *duplication*, or *replacement*. The original scene defines the intended repair, allowing us to measure both target recovery and preservation of the surrounding scene. Authors experienced with Unreal Engine design each corruption to ensure that the required edit is meaningful in context. At test time, the agent receives the corrupted scene, the asset catalog, and rendered reference views of the original scene: two views for each indoor case and one for each outdoor case. We select scenes whose reference views provide sufficient visual evidence for the repair and whose editable objects are stable, individually addressable actors (Appendix A.2).

Human review of solvability. Three independent reviewers who did not author the cases review every case for solvability from the resources available to the agent. For Text-to-Scene, they check

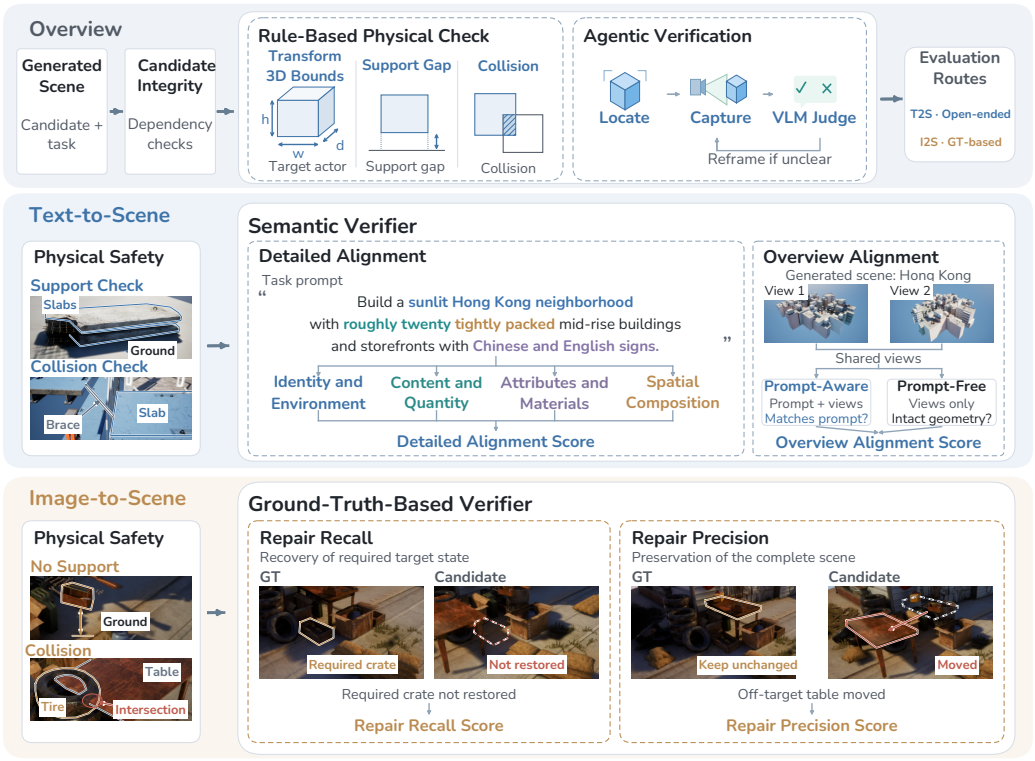


Figure 3: Evaluation pipeline. Both settings share Candidate Integrity, Physical Safety, and Agentic Verification. Text-to-Scene uses semantic alignment, whereas Image-to-Scene measures target recovery and scene preservation against withheld ground truth.

that the prompt can be satisfied using the provided asset catalog; for Image-to-Scene, they check that every required repair can be inferred from the reference images.

Public and private sets. Like ARC-AGI-3 (ARC Prize Foundation, 2026), Code4Scene separates public and private evaluation cases. The public set supports reproducible development and evaluation, while the private set retains unreleased cases to reduce direct tuning to benchmark tasks. Given the cost of long authoring runs, we select the public set to approximate full-suite performance at lower evaluation cost. We divide the 190 cases into a 95-case public set (20 Text-to-Scene, 25 indoor Image-to-Scene, and 50 outdoor Image-to-Scene) and a 95-case private set (10, 55, and 30, respectively). The sets contain distinct cases, although they can share source scenes. Across the ten agent configurations evaluated on both sets, public-set and full-suite overall scores correlate at $r = 0.998$, and no pair whose full-suite overall scores differ by at least 0.02 reverses order on the public set (Appendix A.3; Table 5).

2.3 Evaluation Pipeline

Because scene authoring leaves behind a persistent artifact, Code4Scene scores the scene the agent saves rather than its code or screenshots. For each case, an agent configuration retries only if its first attempt produces no valid submission, with at most two attempts in total. We retain the first valid submission without score-based selection; cases without one score zero (Appendix C.6). Reported token costs and tool counts cover only the retained attempt (Appendix D.1). The saved scene $\hat{\mathcal{L}}$ is scored after the run ends; the trajectory is recorded for analysis (Figure 4) but does not enter the score. During execution, only case-provided resources are accessible. Evaluator-only references H, e.g., the human-assembled scene, held-out views and ground-truth state, remain withheld.

Evaluation overview. Both settings need the same basic evidence about the saved scene, so the pipeline applies shared checks before branching into the task-specific routes shown in Figure 3 (top). *Candidate Integrity* verifies that the submitted Unreal Engine scene is a usable artifact, while the

Rule-Based Physical Check extracts deterministic scene-state measurements such as transforms, 3D bounds, support gaps, and collisions (Appendix C.3). Because small or occluded objects are hard to judge from a single global render, *Agentic Verification* locates the relevant actors, captures clear views of them, and applies a VLM judge under a fixed configuration whenever visual interpretation is required (Appendix C.1). Text-to-Scene then uses open-ended semantic verification against the prompt, whereas Image-to-Scene uses ground-truth-based verification against the withheld scene state. Separately, we evaluate *Downstream Scene Usability* with a player agent and report the resulting Embodied Utility Score (EUS) as an auxiliary metric (Section 2.4).

Text-to-Scene evaluation. Text-to-Scene follows an open-ended evaluation route because a prompt can admit multiple valid constructions. Physical Safety evaluates support and collision for the generated actors using the shared rule-based physical checks. Semantic alignment is evaluated by the Semantic Verifier at two complementary levels. Detailed Alignment checks the atomic requirements extracted from the prompt, organized into four families: Identity and Environment, Content and Quantity, Attributes and Materials, and Spatial Composition. Overview Alignment evaluates the generated scene holistically from shared overview views. Its prompt-aware component assesses whether the scene matches the requested content and overall intent, while its prompt-free component checks whether the scene's geometry is structurally intact. Together, Detailed Alignment and Overview Alignment measure prompt fulfillment at both the requirement and scene levels.

Image-to-Scene evaluation. Image-to-Scene follows a ground-truth-based evaluation route because each corruption has a precisely defined target state. Physical Safety evaluates the edited actors using support and collision checks analogous to those in Text-to-Scene. The Ground-Truth-Based Verifier then compares the submitted scene with the withheld ground-truth state. Repair Recall measures recovery of the required target state, while Repair Precision measures preservation of the complete scene by accounting for both incorrect target states and unintended changes to content that should remain unchanged. These two metrics characterize whether the agent performs the required repair while preserving the rest of the scene.

2.4 Evaluation Criteria and Metrics

Code4Scene evaluates coding agents that directly author engine-native 3D scenes, so success requires more than a plausible rendering. We therefore assess three properties of the submitted scene: artifact integrity, task fulfillment, and static physical validity. *Candidate Integrity* serves as a validity gate, checking that the saved scene can be reopened, contains the required scene content, and resolves its dependencies (Appendix C.2). Task fulfillment uses setting-specific criteria, because construction and editing require different notions of correctness, while physical validity uses shared checks.

Text-to-Scene. Text-to-Scene evaluates whether an agent can realize an open-ended description as a coherent 3D scene. Prompts combine requirements on content, appearance, and spatial organization (e.g., `"houses lining a canal"` or `"a dense town center surrounded by open farmland"`). Because many different scenes can satisfy the same prompt, evaluation measures semantic and spatial adherence to the prompt rather than similarity to a single reference. *Detailed Alignment* checks individual requirements using the strongest available evidence: engine state can resolve some properties, such as object presence and counts, but most, including appearance, style, and many spatial relations, require visual interpretation (e.g., `"a weathered facade"` or `"a Victorian-style building"`). A VLM therefore judges them from targeted rendered views. *Overview Alignment* complements these local checks by evaluating the scene as a whole, including global semantic fit, composition, style, and structural coherence. This captures cases where individual elements are present but do not collectively realize the intended scene. Together, these metrics assess individual requirements and the overall scene (Appendix C.4).

Image-to-Scene. Image-to-Scene tests precise control of scene state: the agent must infer a repair from visual evidence and apply it exactly in 3D. The required edit may affect only a small part of a large scene (e.g., *restore the missing globe*), while surrounding content should remain unchanged. Overall image or geometry similarity can therefore remain high even when the repair is missed or unrelated objects are modified. We instead compare the submitted scene with the withheld ground-truth state. *Repair Recall* measures how many required changes are recovered, while *Repair Precision*

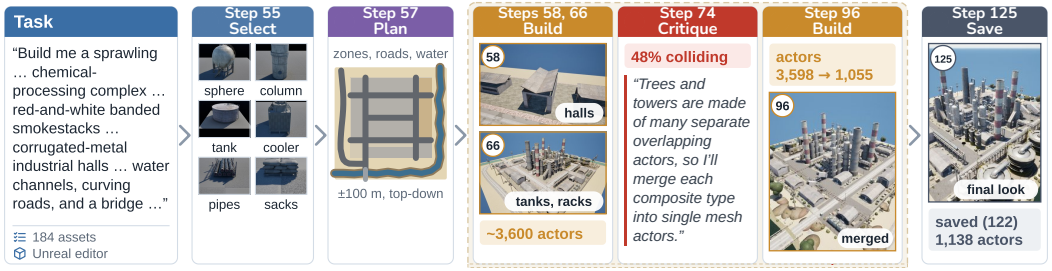


Figure 4: Selected key steps from a Text-to-Scene run by Claude Fable 5.1 (max) on the chemical-plant case. Step numbers index benchmark tool calls; intervening steps are omitted. The agent alternates building and self-critique, restructuring the scene after detecting extensive collisions. Quoted text shows the agent's messages. The final scene scores 0.877.

measures whether the resulting states are correct and unrelated scene content is preserved. Their harmonic mean, *Repair F1*, captures both repair completeness and control over the edit scope without being dominated by the many background actors that require no change (Appendix C.5).

Shared criteria. Task correctness alone does not ensure physical usability. An object may satisfy the semantic requirement while remaining physically invalid (e.g., `a chair floating above the floor` or `a cabinet intersecting a wall`). *Physical Safety* therefore measures static physical validity, namely support and interpenetration, directly from Unreal Engine state, applied to generated actors in Text-to-Scene and edited actors in Image-to-Scene (Appendix C.3). Because static checks cannot show whether a scene supports actual use, we additionally report the *Embodied Utility Score (EUS)* as a separate downstream measure. A fixed player agent navigates predefined routes in the submitted scene, and EUS combines object reachability and route completion (Yang et al., 2024a; Anderson et al., 2018). EUS is reported independently of the benchmark score because it evaluates subsequent embodied use of the authored scene (Appendix C.8).

3 Experiments

We use Code4Scene to measure how reliably current coding agents construct and edit 3D scenes (Section 3.2), to validate its evaluator (Section 3.3), and to locate where agents fail (Section 3.4).

3.1 Experimental Setup

We evaluate a diverse set of coding agents spanning frontier proprietary models and open-weight models under a common benchmark interface. Our evaluation includes 14 agent configurations on the public set and ten on the private set (Table 2). A configuration specifies both the underlying model and the coding-agent scaffold used to interact with Unreal Engine. Across configurations, we keep the benchmark environment fixed: agents receive the same case-specific asset catalog, Unreal Engine tools, skill instructions, and evaluation protocol. This design allows us to compare coding-agent systems as they are practically deployed, while controlling the task environment and evaluator.

Unreal Engine workers run on eight NVIDIA RTX A5000 GPUs, while model inference is served through native clients, OpenRouter, or self-hosted endpoints (Appendix A.1). Task-performance analyses use the public set unless otherwise specified. Auxiliary evaluator-validation and navigation studies use separate samples, detailed in Appendices D.7 and D.6.

3.2 Overall Performance

Construction and editing performance are strongly correlated but not interchangeable (Table 2): across the 14 public-set configurations, Text-to-Scene and Image-to-Scene scores have a Spearman correlation of $\rho = 0.78$, and different systems lead the two settings, with Claude Fable 5.1 leading construction (0.788) and Gemini 3.8 Flash leading editing (0.581). For example, GPT-6 Astra and Gemini 3.8 Flash obtain nearly identical overall scores (0.6193 and 0.6190), yet Astra performs better on construction (0.724 vs. 0.657), while Gemini performs better on editing (0.581 vs. 0.515). Precise editing remains challenging overall, with the best Repair F1 reaching only 0.527.

Table 2: Main results on the public and private sets. Scores lie in $[0, 1]$ (↑). Each setting averages its scheduled cases; S_{model} weights the two settings equally. Rows follow S_{model} within each set; dark/light shading marks the best/second-best values. Metrics and zero-score rules are in Appendix C.

A. Public set

Agent configuration	Text-to-Scene (20 cases)			Image-to-Scene (75 cases)			Overall
	Semantic ↑	Physical ↑	S_{T2S} ↑	Repair F1 ↑	Physical ↑	S_{I2S} ↑	S_{model} ↑
GPT-6 Astra (max)	0.727	0.712	0.724	0.445	0.796	0.515	**0.619**
Gemini 3.8 Flash (high)	0.657	0.655	0.657	**0.527**	0.796	**0.581**	**0.619**
Claude Fable 5.1 (max)	**0.760**	**0.898**	**0.788**	0.332	0.795	0.424	0.606
Claude Opus 5 (max)	0.695	0.811	0.718	0.389	0.786	0.468	0.593
GPT-5.6 Sol (high)	0.691	0.772	0.707	0.319	0.689	0.393	0.550
Muse Spark 1.3 (medium)	0.645	0.651	0.646	0.246	0.807	0.358	0.502
GLM-5.3 Flash (max)	0.504	0.533	0.509	0.188	**0.848**	0.320	0.415
Qwen 3.8 27B (thinking off)	0.524	0.688	0.557	0.083	0.667	0.200	0.378
Grok 4.6 (high)	0.563	0.585	0.567	0.044	0.744	0.184	0.376
Qwen 3.8 27B (thinking on)	0.471	0.694	0.516	0.118	0.660	0.226	0.371
Inkling (high)	0.351	0.722	0.425	0.095	0.609	0.198	0.312
Gemma 4 31B (thinking on)	0.415	0.694	0.471	0.003	0.615	0.126	0.298
Gemma 4 31B (thinking off)	0.387	0.725	0.455	0.001	0.572	0.115	0.285
DeepSeek V4.1 Flash (high)	0.175	0.513	0.243	0.066	0.787	0.210	0.227

B. Private set

Agent configuration	Text-to-Scene (10 cases)			Image-to-Scene (85 cases)			Overall
	Semantic ↑	Physical ↑	S_{T2S} ↑	Repair F1 ↑	Physical ↑	S_{I2S} ↑	S_{model} ↑
Gemini 3.8 Flash (high)	0.708	0.608	0.688	**0.553**	0.837	**0.610**	**0.649**
GPT-5.6 Sol (high)	**0.724**	**0.696**	**0.719**	0.375	0.685	0.437	0.578
Muse Spark 1.3 (medium)	0.582	0.560	0.577	0.278	0.844	0.391	0.484
GLM-5.3 Flash (max)	0.579	0.543	0.572	0.244	**0.878**	0.370	0.471
Qwen 3.8 27B (thinking off)	0.537	0.667	0.563	0.085	0.681	0.204	0.383
Qwen 3.8 27B (thinking on)	0.486	0.674	0.523	0.100	0.670	0.214	0.369
Gemma 4 31B (thinking on)	0.474	0.651	0.510	0.003	0.590	0.121	0.315
Inkling (high)	0.403	0.611	0.444	0.036	0.564	0.142	0.293
Gemma 4 31B (thinking off)	0.401	0.682	0.457	0.000	0.556	0.111	0.284
DeepSeek V4.1 Flash (high)	0.172	0.364	0.211	0.072	0.792	0.216	0.214

Takeaway 1. Construction and editing probe different capabilities: scene-level spatial reasoning and precise control of scene state. Despite nearly identical overall scores, Astra is stronger at construction and Gemini at editing (Table 2).

3.3 Evaluator Validation

Agreement with human judgments. Because Text-to-Scene scores rely on VLM judgments, we test whether they agree with human preferences on a dedicated prompt sample. On its retrospective holdout, the adopted policy agrees with 86.90% of Plackett-Luce consensus pairs and achieves Kendall's $\tau_b = 0.738$ (Figure 5). Individual-preference agreement is 79.37%, versus 80.95% between annotators on the same prompts. Earlier human-guided exploration used the full study sample, so this is internal validation, not a test on unseen prompts; it establishes neither above-human performance nor Image-to-Scene alignment (Appendix D.7).

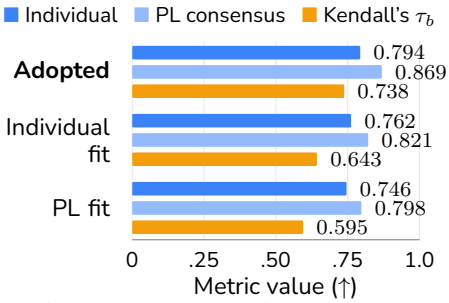


Figure 5: Text-to-Scene evaluator agreement on the retrospective human-study holdout.

VLM judge repeatability. We evaluate the repeatability of the VLM judge by rerunning Detailed Alignment scoring three times on six hand-selected Text-to-Scene cases while holding the scenes, evidence, prior decisions, and request schedules fixed. Across repeated evaluations, pairwise Kendall's τ_b ranges from 0.889 to 0.944, while the standard deviation of agent-configuration means ranges from 0.0010 to 0.0132 (Appendix D.7). These results characterize scoring stability under fixed inputs.

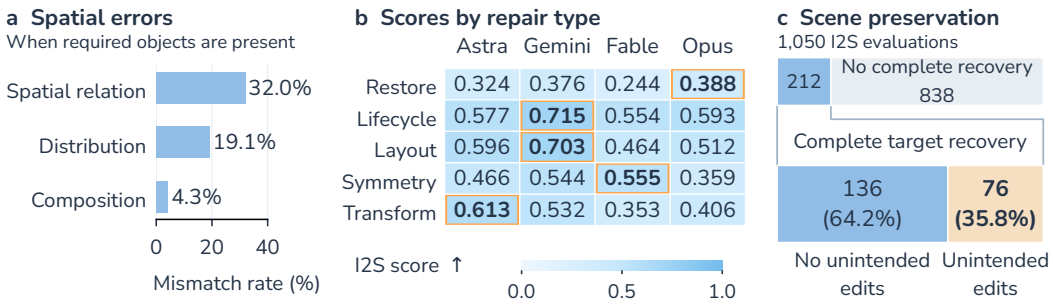


Figure 6: (a) Spatial errors with required objects present. (b) Mean I2S score by repair type for four agents; orange marks the best of the four. (c) Complete recoveries with or without unintended edits.

3.4 Analysis

We examine where agents fall short in spatial reasoning and in controlling scene state.

From scene content to spatial composition. Frontier agents' strengths in realizing scene content coexist with errors in spatial organization. Fable leads the Identity, Content and Attributes families, yet Spatial Composition is the lowest-scoring family for all 14 public-set configurations. Even Astra, which has the highest Spatial mean, reaches only 0.543 (Appendix D.3). The difficulty persists when content is present: 76 of 369 spatial requirements whose relevant entities pass separate existence checks still receive direct mismatch judgments (20.6%; Figure 6a). These failures show a gap between generating the requested elements and satisfying the relations that organize them (Appendix C.4).

> **Takeaway 2.** Spatial Composition remains the weakest requirement family for all 14 agents. Errors persist even when the required objects are present (Figure 6a).

Distinct strengths in visual scene repair. The repair types behind the overall editing score (Appendix A.2) reveal different frontier strengths (Figure 6b). Astra leads Transform with a category score of 0.613, ahead of Gemini's 0.532, whereas Gemini leads Lifecycle and Layout; on Layout, Gemini scores 0.703 versus Astra's 0.596. Fable narrowly leads Symmetry (0.555 vs. Gemini's 0.544 on eight cases; Appendix D.3). Thus, Gemini's overall editing lead does not extend to every repair type, and a single editing score hides which kinds of scene-state changes each agent controls well.

> **Takeaway 3.** Editing ability varies substantially across repair types. Astra performs best on Transform repairs, while Gemini is stronger on Lifecycle and Layout operations (Figure 6b).

Failure to preserve the scene after target recovery. Completing the required repair does not always keep changes within the intended scope. On Indoor editing tasks, Astra fully recovers all targets in 23 cases and Gemini in 19. Requiring complete recovery with no unintended changes reverses this ordering: Astra achieves 16 such repairs and Gemini 17 (Appendix D.5). All seven of Astra's complete Indoor recoveries with false positives include changes to non-target content. Across all 14 configurations, 76 of 212 complete recoveries (35.8%) still contain unintended additions or changes (Figure 6c). Recovering the desired target state therefore does not by itself demonstrate control over the rest of the scene. Error types and sensitivity checks are reported in Appendix D.5.

> **Takeaway 4.** Recovering the target does not guarantee precise editing. Across all 14 configurations, 35.8% of complete recoveries still contain unintended changes to the surrounding scene (Figure 6c).

Downstream usability for embodied agents. A repaired scene is useful only if agents can still move through it. The auxiliary navigation study compares edited Outdoor scenes with their matched inputs to assess changes in downstream usability (Section 2.4). Gemini shows the largest observed mean improvement: EUS increases from 0.264 to 0.323, a gain of 0.059. Other evaluated agent configurations show mean changes between -0.031 and $+0.014$. Study coverage, paired uncertainty and component scores are reported in Appendix D.6.

> **Takeaway 5.** Paired navigation tests separate editing gains from inherited usability. Gemini shows the largest observed EUS improvement ($+0.059$; Appendix Figure 17).

AI USE STATEMENT

Generative AI tools were used to assist with language editing, figure prototyping, analysis planning and interpretation, and the implementation and debugging of analysis code. The authors reviewed the resulting manuscript, figures, and analysis code, verified reported results against the underlying artifacts, and take responsibility for the final manuscript. AI also assisted task-prompt drafting and requirement-bundle construction. Task prompts were human-reviewed; frozen requirement bundles received automated checks but no human review (Section 2.2; Appendix C.4).

ETHICS STATEMENT

The human evaluation asks human annotators to rank rendered virtual scenes, with model identities and evaluator scores hidden (Appendix C.9). Access to and redistribution of third-party scene assets remain subject to their original licenses. Generated scenes may retain defects that affect downstream use, so benchmark scores should not be interpreted as general safety certification. Broader impacts and limitations are discussed in Appendix F.

REPRODUCIBILITY STATEMENT

Task setup, evaluator settings, and analysis protocols are documented in Appendices A, C, and D. We will release the public-set task code, evaluation and analysis scripts, agent configuration records, the shared skill specification and tool schemas, and judge instructions. Access to third-party scene assets will remain subject to their original licenses.

REFERENCES

Peter Anderson, Angel Chang, Devendra Singh Chaplot, Alexey Dosovitskiy, Saurabh Gupta, Vladlen Koltun, Jana Kosecka, Jitendra Malik, Roozbeh Mottaghi, Manolis Savva, and Amir R. Zamir. On evaluation of embodied navigation agents. *arXiv preprint arXiv:1807.06757*, 2018. URL https://arxiv.org/abs/1807.06757.

Anthropic. Introducing Claude Fable 5.1 and Claude Mythos 5.1, September 2026a. URL https://www.anthropic.com/claude-fable-and-mythos-5-1. Accessed: 2026-09-25.

Anthropic. Introducing Claude Opus 5, July 2026b. URL https://www.anthropic.com/news/claude-opus-5. Accessed: 2026-09-25.

ARC Prize Foundation. ARC-AGI-3: A new challenge for frontier agentic intelligence. *arXiv preprint arXiv:2603.24621*, 2026.

Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2CAD: Learning CAD model alignment in RGB-D scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2614–2623, 2019. URL https://openaccess.thecvf.com/content_CVPR_2019/html/Avetisyan_Scan2CAD_Learning_CAD_Model_Alignment_in_RGB-D_Scans_CVPR_2019_paper.html.

Matthew Chang, Gunjan Chhablani, Alexander Clegg, Mikael Dallaire Cote, Ruta Desai, Michal Hlavac, Vladimir Karashchuk, Jacob Krantz, Roozbeh Mottaghi, Priyam Parashar, Siddharth Patki, Ishita Prasad, Xavier Puig, Akshara Rai, Ram Ramrakhya, Daniel Tran, Joanne Truong, John M. Turner, Eric Undersander, and Tsung-Yen Yang. PARTNR: A benchmark for planning and reasoning in embodied multi-agent tasks. In *International Conference on Learning Representations*, pp. 65205–65268, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/a3cf318fbeec1126da21e9185ae9908c-Abstract-Conference.html.

DeepSeek-AI. DeepSeek-V4.1-Flash: Pushing the limits of KV cache compression. *arXiv preprint arXiv:2609.19969*, 2026. URL https://arxiv.org/abs/2609.19969.

Matt Deitke, Eli VanderBilt, Alvaro Herrasti, Luca Weihs, Kiana Ehsani, Jordi Salvador, Winson Han, Eric Kolve, Aniruddha Kembhavi, and Roozbeh Mottaghi. ProcTHOR: Large-scale embodied AI using procedural generation. In *Advances in Neural Information Processing Systems*, volume 35, pp. 5982–5994. Curran Associates, Inc., 2022.

doi: 10.52202/068431-0433. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/27c546ab1e4f1d7d638e6a8dfbad9a07-Abstract-Conference.html.

Epic Games. Licenses and pricing in Fab. https://dev.epicgames.com/documentation/en-us/fab/licenses-and-pricing-in-fab, 2026. Accessed September 2026.

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. In *Advances in Neural Information Processing Systems*, volume 36, pp. 18225–18250. Curran Associates, Inc., 2023. doi: 10.52202/075280-0802. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/3a7f9e485845dac27423375c934cb4db-Abstract-Conference.html.

Gemma Team. Gemma 4 technical report. *arXiv preprint arXiv:2607.02770*, 2026. URL https://arxiv.org/abs/2607.02770.

Dhruba Ghosh, Hannaneh Hajishirzi, and Ludwig Schmidt. GenEval: An object-focused framework for evaluating text-to-image alignment. In *Advances in Neural Information Processing Systems*, volume 36, pp. 52132–52152. Curran Associates, Inc., 2023. doi: 10.52202/075280-2270. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/a3bf71c7c63f0c3bcb7ff67c67b1e7b1-Abstract-Datasets_and_Benchmarks.html.

Google DeepMind. Gemini 3.8 Flash, 2026. URL https://deepmind.google/models/gemini/flash/. Accessed: 2026-09-25.

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 (CVPR)*, pp. 18574–18583, 2025. URL https://openaccess.thecvf.com/content/CVPR2025/html/Gu_BlenderGym_Benchmarking_Foundational_Model_Systems_for_Graphics_Editing_CVPR_2025_paper.html.

Chunchao Guo, Jinpeng Li, Yang Li, and Zilong Huang. WorldClaw: Agentic 3D open-world generation at scale. *arXiv preprint arXiv:2608.05248*, 2026. URL https://arxiv.org/abs/2608.05248.

Lanshan He, Haozhou Pang, Qi Gan, Xin Shen, Ziwei Zhang, Yibo Liu, Gang Fang, Bo Liu, Kai Sheng, Shengfeng Zeng, Chaofan Li, Zhen Hui, Keer Zhou, Lan Zhou, and Shujun Dai. Cutscene Agent: An LLM agent framework for automated 3D cutscene generation. *arXiv preprint arXiv:2604.25318*, 2026. URL https://arxiv.org/abs/2604.25318.

Yuze He, Yushi Bai, Matthieu Lin, Wang Zhao, Yubin Hu, Jenny Sheng, Ran Yi, Juanzi Li, and Yong-Jin Liu. T^3Bench: Benchmarking current progress in text-to-3D generation. *arXiv preprint arXiv:2310.02977*, 2024. doi: 10.48550/arXiv.2310.02977. URL https://arxiv.org/abs/2310.02977.

Tomáš Hodaň, Martin Sundermeyer, Bertram Drost, Yann Labbé, Eric Brachmann, Frank Michel, Carsten Rother, and Jiří Matas. BOP challenge 2020 on 6D object localization. In *Computer Vision – ECCV 2020 Workshops*, pp. 577–594. Springer International Publishing, 2020. doi: 10.1007/978-3-030-66096-3_39. URL https://link.springer.com/chapter/10.1007/978-3-030-66096-3_39.

Yushi Hu, Benlin Liu, Jungo Kasai, Yizhong Wang, Mari Ostendorf, Ranjay Krishna, and Noah A. Smith. TIFA: Accurate and interpretable text-to-image faithfulness evaluation with question answering. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 20406–20417, 2023. URL https://openaccess.thecvf.com/content/ICCV2023/html/Hu_TIFA_Accurate_and_Interpretable_Text-to-Image_Faithfulness_Evaluation_with_Question_Answering_ICCV_2023_paper.html.

Ziqi Huang, Yinan He, Jiashuo Yu, Fan Zhang, Chenyang Si, Yuming Jiang, Yuanhan Zhang, Tianxing Wu, Qingyang Jin, Nattapol Chanpaisit, Yaohui Wang, Xinyuan Chen, Limin Wang, Dahua Lin, Yu Qiao, and Ziwei Liu. VBench: Comprehensive benchmark suite for video generative models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21807–21818, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Huang_VBench_Comprehensive_Benchmark_Suite_for_Video_Generative_Models_CVPR_2024_paper.html.

Haoqiang Kang, Xiaokang Ye, Yuhan Liu, Siddhant Hitesh Mantri, Lingjun Mao, James Fleming, Drishti Regmi, and Lianhui Qin. SimWorld Studio: Automatic environment generation with evolving coding agent for embodied agent learning. *arXiv preprint arXiv:2605.09423*, 2026a. URL https://arxiv.org/abs/2605.09423.

Haoqiang Kang, Yiming Zhang, Yiyang Guo, Chuying Li, Jianzhi Shen, Tianruo Rose Xu, Xiaokang Ye, and Lianhui Qin. DeliveryGym: An RL environment for long-horizon embodied agent planning with adaptive curriculum. *arXiv preprint arXiv:2609.19801*, 2026b. URL https://arxiv.org/abs/2609.19801.

Chunjiang Liu, Xiaoyuan Wang, Haoyu Chen, Yizhou Zhao, Ming-Hsuan Yang, and László A. Jeni. SimWorlds: A multi-agent system for dynamic 3D scene creation. *arXiv preprint arXiv:2607.01766*, 2026a. URL https://arxiv.org/abs/2607.01766.

Junze Liu, Kun Qian, Florian Dubost, Kai Zhong, Arvind Srinivasan, Nan Chen, Anping Wang, Sam Zhang, Alejandro Mottini, Qingjun Cui, and Tian Wang. 3D primitives are a spatial language for VLMs. *arXiv preprint arXiv:2605.12586*, 2026b. URL https://arxiv.org/abs/2605.12586.

Shuo Lu, Yinuo Xu, Kecheng Yu, Siru Jiang, Yongcan Yu, Yubin Wang, Haitao Yang, Yuxiang Zhang, Bin Wang, Ran He, and Jian Liang. WorldCoder-Bench: Benchmarking physically grounded 3D world synthesis. *arXiv preprint arXiv:2606.01869*, 2026. URL https://arxiv.org/abs/2606.01869.

Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. tinyBenchmarks: Evaluating LLMs with fewer examples. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 34303–34326. PMLR, 2024. URL https://proceedings.mlr.press/v235/maia-polo24a.html.

Meta AI. Introducing Muse Spark 1.3, September 2026. URL https://research.meta.ai/blog/introducing-muse-spark-1-3. Accessed: 2026-09-25.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations. *arXiv preprint arXiv:2411.00640*, 2024. URL https://arxiv.org/abs/2411.00640.

Franck Ndzomga. Efficient benchmarking of AI agents. *arXiv preprint arXiv:2603.23749*, 2026. URL https://arxiv.org/abs/2603.23749.

Yansong Ning, Jingwen Ye, Zhongkai Wu, Yang Sun, Yiqin Zhu, Xingyi Li, Weidong Zhang, and Hao Liu. VibeWorlding: Can multimodal agents construct 3D open worlds end-to-end? *arXiv preprint arXiv:2608.15265*, 2026. URL https://arxiv.org/abs/2608.15265.

OpenAI. GPT-5.6 Sol. OpenAI model documentation, 2026a. URL https://developers.openai.com/api/docs/models/gpt-5.6-sol. Accessed: 2026-09-25.

OpenAI. GPT-6 Astra: A new generation of intelligence. https://openai.com/index/gpt-6-astra/, 2026b. Accessed: 2026-09-24.

Nicholas Pfaff, Thomas Cohn, Sergey Zakharov, Rick Cory, and Russ Tedrake. SceneSmith: Agentic generation of simulation-ready indoor scenes. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=WwS8CTpUA6.

Qwen Team. Qwen3.8-27B. Hugging Face model card, August 2026. URL https://huggingface.co/Qwen/Qwen3.8-27B. Accessed: 2026-09-25.

Alexander Raistrick, Lahav Lipson, Zeyu Ma, Lingjie Mei, Mingzhe Wang, Yiming Zuo, Karhan Kayan, Hongyu Wen, Beining Han, Yihan Wang, Alejandro Newell, Hei Law, Ankit Goyal, Kaiyu Yang, and Jia Deng. Infinite photorealistic worlds using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 12630–12641, 2023. URL https://openaccess.thecvf.com/content/CVPR2023/html/Raistrick_Infinite_Photorealistic_Worlds_Using_Procedural_Generation_CVPR_2023_paper.html.

Alexander Raistrick, Lingjie Mei, Karhan Kayan, David Yan, Yiming Zuo, Beining Han, Hongyu Wen, Meenal Parakh, Stamatis Alexandropoulos, Lahav Lipson, Zeyu Ma, and Jia Deng. Infinigen Indoors: Photorealistic indoor scenes using procedural generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 21783–21794, 2024. URL https://openaccess.thecvf.com/content/CVPR2024/html/Raistrick_Infinigen_Indoors_Photorealistic_Indoor_Scenes_using_Procedural_Generation_CVPR_2024_paper.html.

Kathakoli Sengupta, Kai Ao, and Paola Cascante-Bonilla. SceneCritic: A symbolic evaluator for 3D indoor scene synthesis. *arXiv preprint arXiv:2604.13035*, 2026. URL https://arxiv.org/abs/2604.13035.

SpaceXAI. Introducing Grok 4.6, August 2026. URL https://x.ai/news/grok-4-6. Accessed: 2026-09-25.

Hou In Ivan Tam, Hou In Derek Pun, Austin T. Wang, Angel X. Chang, and Manolis Savva. SceneEval: Evaluating semantic coherence in text-conditioned 3D indoor scene synthesis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pp. 7355–7365, 2026. URL https://openaccess.thecvf.com/content/WACV2026/html/Tam_SceneEval_Evaluating_Semantic_Coherence_in_Text-Conditioned_3D_Indoor_Scene_Synthesis_WACV_2026_paper.html.

Thinking Machines Lab. Inkling model card, July 2026. URL https://thinkingmachines.ai/model-card/inkling/. Accessed: 2026-09-25.

Heather L. Turner, Jacob van Etten, David Firth, and Ioannis Kosmidis. Modelling rankings in R: The PlackettLuce package. *Computational Statistics*, 35(3):1027–1057, 2020. doi: 10.1007/s00180-020-00959-3. URL https://doi.org/10.1007/s00180-020-00959-3.

Rajan Vivek, Kawin Ethayarajh, Diyi Yang, and Douwe Kiela. Anchor points: Benchmarking models with much fewer examples. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1576–1601. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.eacl-long.95. URL https://aclanthology.org/2024.eacl-long.95/.

Puyi Wang, Yuhao Wang, Linjie Li, Zhengyuan Yang, Kevin Qinghong Lin, Yangguang Li, and Yu Cheng. SceneCode: Executable world programs for editable indoor scenes with articulated objects. *arXiv preprint arXiv:2605.19587*, 2026. URL https://arxiv.org/abs/2605.19587.

Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson Env: Real-world perception for embodied agents. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9068–9079, 2018. doi: 10.1109/CVPR.2018.00945. URL https://openaccess.thecvf.com/content_cvpr_2018/html/Xia_Gibson_Env_Real-World_CVPR_2018_paper.html.

Hongchi Xia, Xuan Li, Zhaoshuo Li, Qianli Ma, Jiashu Xu, Ming-Yu Liu, Yin Cui, Tsung-Yi Lin, Wei-Chiu Ma, Shenlong Wang, Shuran Song, and Fangyin Wei. SAGE: Scalable agentic 3D scene generation for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22358–22368, 2026. URL https://openaccess.thecvf.com/content/CVPR2026/html/Xia_SAGE_Scalable_Agentic_3D_Scene_Generation_for_Embodied_AI_CVPR_2026_paper.html.

Ruijie Xu, Xinnan Zhu, Jiayu Ying, Daoguo Dong, Yuzhou Ji, and Xin Tan. MUSE: Agentic 3D scene authoring via memory-grounded incremental requirement satisfaction. *arXiv preprint arXiv:2606.14168*, 2026. URL https://arxiv.org/abs/2606.14168.

Yandan Yang, Baoxiong Jia, Peiyuan Zhi, and Siyuan Huang. PhyScene: Physically interactable 3D scene synthesis for embodied AI. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16262–16272, 2024a. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_PhyScene_Physically_Interactable_3D_Scene_Synthesis_for_Embodied_AI_CVPR_2024_paper.html.

Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, Chris Callison-Burch, Mark Yatskar, Aniruddha Kembhavi, and Christopher Clark. Holodeck: Language guided generation of 3D embodied AI environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 16227–16237, 2024b. URL https://openaccess.thecvf.com/content/CVPR2024/html/Yang_Holodeck_Language_Guided_Generation_of_3D_Embodied_AI_Environments_CVPR_2024_paper.html.

Yilun Yao, Xinyu Tan, Chao-Hsuan Liu, Yaoming Li, Zhengyang Wang, Wenhan Yu, Zhewen Tan, Yuxuan Tian, Guangxiang Zhao, Lin Sun, Xiangzheng Zhang, and Tong Yang. HarnessBench: Measuring harness effects across models in realistic agent workflows. *arXiv preprint arXiv:2605.27922*, 2026. URL https://arxiv.org/abs/2605.27922.

Xiaokang Ye, Jiawei Ren, Yan Zhuang, Xuhong He, Yiming Liang, Yiqing Yang, Mrinaal Dogra, Xianrui Zhong, Eric Liu, Kevin Benavente, Rajiv Mandya Nagaraju, Dhruv Sharma, Ziqiao Ma, Tianmin Shu, Zhiting Hu, and Lianhui Qin. SimWorld: An open-ended simulator for agents in physical and social worlds. In *Advances in Neural Information Processing Systems*, volume 38, pp. 165577–165628. Curran Associates, Inc., 2025. doi: 10.52202/085713-5519. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/f1eff44d59d9c66cc6e359d36d80b0b9-Abstract-Conference.html.

Z.ai. GLM-5.3-Flash. Hugging Face model card, 2026. URL https://huggingface.co/zai-org/GLM-5.3-Flash. Accessed: 2026-09-25.

Guanhua Zhang, Florian E. Dorner, and Moritz Hardt. How benchmark prediction from fewer data misses the mark. In *Advances in Neural Information Processing Systems*, volume 38, pp. 135638–135678. Curran Associates, Inc., 2025. doi: 10.52202/085713-4528. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/c57bacb5ea621582235c54ce2a83136d-Abstract-Conference.html.

Yifei Zhao, Xiangxin Zhou, Wenhao Yang, Jiaqi Tang, Pu Jian, Huanjin Yao, Jiarui Yao, Haowei Lin, Chunchao Guo, Zhuo Chen, Wenkai Lyu, Jianzhu Ma, Xueqian Wang, and Wenxi Zhu. SceneActBench: Can agents act on the 3D scenes they see? *arXiv preprint arXiv:2607.22393*, 2026. URL https://arxiv.org/abs/2607.22393.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems*, volume 36, pp. 46595–46623. Curran Associates, Inc., 2023. doi: 10.52202/075280-2020. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Yan Zhuang, Jiawei Ren, Xiaokang Ye, Jianzhi Shen, Ruixuan Zhang, Tianai Yue, Muhammad Faayez, Xuhong He, Xiyan Zhang, Ziqiao Ma, Lianhui Qin, Zhiting Hu, and Tianmin Shu. SimWorld-Robotics: Synthesizing photorealistic and dynamic urban environments for multimodal robot navigation and collaboration. In *Advances in Neural Information Processing Systems*, volume 38, pp. 51854–51895. Curran Associates, Inc., 2025. doi: 10.52202/085713-1729. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/4a81c1aad3064fb8243b8938a49ecca0-Abstract-Conference.html.

APPENDIX CONTENTS

A Benchmark Details **16**

A.1 Coding-Agent Interface 16

A.2 Case Construction 16

A.3 Public and Private Sets 18

B Running Cases **19**

C Evaluation Protocol **19**

C.1 Evaluation Outputs and Verifier Routing 19

C.2 Candidate Integrity Checks 22

C.3 Physical Safety 22

C.4 Semantic Verifier for Text-to-Scene 23

C.5 Ground-Truth-Based Verifier: Actor Repair F1 25

C.6 Score Aggregation and Reporting 26

C.7 Worked Scoring Examples 27

C.8 Embodied Utility Score Implementation 27

C.9 Human-Alignment Protocol and Weight Fitting 30

C.10 Fixed-Evidence Repeatability Protocol 32

C.11 Evaluator Integrity 32

D Additional Results and Analysis **32**

D.1 Analysis Cohorts and Measurements 32

D.2 Difficulty-Stratified Results 33

D.3 Task Requirement Analysis 33

D.4 Agent Behavior Analysis 34

D.5 Failure Analysis 35

D.6 Downstream Usability 37

D.7 Evaluator Validation 39

E Related Work **40**

F Limitations and Broader Impact **41**

A Benchmark Details

A.1 Coding-Agent Interface

All evaluated systems interact with Unreal Engine through the same tool interface and receive the same skill specification describing how to use that interface. Systems differ in the coding-agent client and the model behind it; the model-facing names in the results are therefore shorthand for complete agent configurations rather than estimates of model-only effects.

Tool interface. The 25 tools in Table 3 support asset discovery, scene editing, state inspection, Python execution, planning, verification, and persistence. Asset discovery provides semantic search over the case-specific catalog, metadata including physical dimensions and spawn paths, and previews where available. Agents can edit actors directly or use the optional deterministic planner to validate scene graphs, solve transforms, inspect plans, and build scenes in batches. Editor Python supports additional operations and asynchronous jobs. Inspection provides screenshots, scene measurements, and targeted geometric checks. The benchmark proxy blocks tools outside this interface.

Skill text. All 190 cases provide the same skill specification in the agent's initial prompt. It documents the available tools and their usage contracts, with general operating rules (for example, calling setup_environment first and last) and short code snippets, but no scene-specific construction plan. The agent must decide how to search, plan, construct, inspect, and revise the scene. The specification has four parts, covering asset retrieval, editor Python execution, scene-graph planning and scene building.

Execution environment. We use Unreal Engine 5.8.0 on an Ubuntu 24.04.3 LTS server with eight NVIDIA RTX A5000 GPUs (24 GB each) and two AMD EPYC 7513 CPUs. These specifications describe the Unreal execution host; model inference uses the services described below.

Coding-agent clients. GPT-5.6 Sol (high) (OpenAI, 2026a) uses Codex, Claude Fable 5.1 (max) (Anthropic, 2026a) and Claude Opus 5 (max) (Anthropic, 2026b) use Claude Code, and Gemini 3.8 Flash (high) (Google DeepMind, 2026) uses Antigravity. Inkling (high) (Thinking Machines Lab, 2026) uses Hermes for Text-to-Scene and indoor Image-to-Scene and Claude Code for outdoor Image-to-Scene, both through OpenRouter. Muse Spark 1.3 (medium) (Meta AI, 2026), GLM-5.3 Flash (max) (Z.ai, 2026), Grok 4.6 (high) (SpaceXAI, 2026), and DeepSeek V4.1 Flash (high) (DeepSeek-AI, 2026) use the benchmark's API-based agent loop with OpenRouter. Qwen 3.8 27B (Qwen Team, 2026) and Gemma 4 31B (Gemma Team, 2026) use the same loop with self-hosted OpenAI-compatible endpoints for both thinking settings. GPT-6 Astra (max) (OpenAI, 2026b) includes both Codex and API-based OpenRouter runs, with the client assignment recorded per case. All clients access tools through a shared metering proxy. Client versions are recorded where available and may vary across runs.

Resource limits. Text-to-Scene runs allow 800 tool calls and 180 minutes. Image-to-Scene runs allow 120 tool calls and 45 minutes for Indoor cases and 80 tool calls and 20 minutes for Outdoor cases.

Retries and submission selection. An agent configuration receives a second attempt only when the first produces no valid submission. The first valid submission is retained, with no score-based comparison between attempts. If neither attempt produces a valid submission, the case receives zero. Reported token costs and tool counts include only the retained attempt, not discarded attempts, and therefore do not represent cumulative expenditure across retries.

A.2 Case Construction

The source collection spans historical and fantasy settings, contemporary urban and transit scenes, industrial and working sites, domestic and office interiors, and stylized environments. This collection emphasizes the compositional irregularity, asset heterogeneity, and authored intent of human-assembled environments, at the cost of the scalability offered by procedural generation (Appendix E).

Table 3: The 25 shared agent tools. Arguments marked * are required; all others are optional.

Group	Tool (arguments)	Result
Asset discovery	search_assets(query*, category, k)	Ranked matching assets
	get_categories()	Asset categories with counts and descriptions
	get_asset_metadata(ids*)	Category, description, tags, dimensions in metres, spawn path
	get_asset_preview(id*)	Rendered preview image, if available
Scene editing	spawn_actor(name*, static_mesh*, location*, rotation, scale)	Spawns a static-mesh actor
	spawn_blueprint_actor(actor_name*, blueprint_id*, location*, rotation, scale)	Spawns a Blueprint actor
	set_actor_transform(name*, location, rotation, scale)	Updates an actor's transform
	delete_actor(name*)	Deletes one actor
	delete_all_spawned()	Deletes all actors spawned in the run
State inspection	get_actors_in_level()	All actors in the scene
	find_actors_by_name(pattern*)	Actors whose names match the pattern
Programmatic control	execute_python_script(script*, execution_mode)	Starts an asynchronous editor Python job; returns its job ID and log path
	read_job_log(job_id, log_path, max_bytes, wait_s)	Job status and output
Planning	validate_scene_graph(graph*)	Validity report, including entries that would be dropped
	solve_scene_graph(graph*, mode, half_extent_m)	Deterministic world transforms with de-overlap
	render_plan(solved)	Top-down image of the solved plan
	tier_derivation(solved)	Build tier for each placed entry
	ground_pass(scene_text*, half_extent_m)	Spawns ground tiles across the build area
	build_tier(tier, entries)	Spawns one solved tier or explicit entries in batches
Verification	take_screenshot(filename, filepath)	Viewport screenshot
	measure_scene()	Actor count, collision and floating rates, offending actors, extent
	check_floating(actor_names, threshold_cm)	Actors whose gap to the surface below exceeds the threshold
	check_collisions(actor_names, touch_tolerance_cm)	Overlapping actor pairs
Environment and persistence	setup_environment(ground_size, time_of_day)	Adds missing lighting, sky, fog and ground
	save_scene_as(name*, revert_to_original)	Saves the scene as a new .umap

Text-to-Scene cases use 130 m or 200 m square plates. The human-assembled scenes used for Image-to-Scene contain approximately 260 to 6,700 actors outdoors and 95 to 3,168 actors indoors. Each case specifies its source scene and required asset packages. The Image-to-Scene evidence contains two reference views per indoor case and one per outdoor case; both use the same editing formulation.

Image-to-Scene corruptions must be stable actor-level changes inferable from the reference evidence. Sources are excluded when procedural hierarchies do not expose stable targets, occlusion prevents reliable reference views, required content cannot be packaged reproducibly, or the scene does not support a clear actor-level edit.

Authored operation groups. Image-to-Scene cases retain the authors' operation-category assignments. The short labels in the analysis correspond to Repair/Completion (Restore), Object Lifecycle (Lifecycle), Layout/Rearrangement (Layout), Symmetry/Pattern (Symmetry), and Geometric Transformation (Transform). In that order, the full suite contains 20, 20, 10, 10, and 20 indoor cases and 25, 16, 19, 7, and 13 outdoor cases. These groups describe authored tasks; they are not labels inferred from agent scores. The selected-case mapping is retained with the Figure 14 data (Appendix D.3).

Outdoor label-chain lengths one through five occur in 25, 3, 26, 25, and 1 cases, respectively. Label chain length counts top-level authored operation entries, not atomic transformations or agent tool calls, because one entry may encode multiple atomic changes.

Difficulty labels. The initial difficulty prior considers the number and spatial extent of affected targets, the number and coupling of atomic operations, the relational constraints to recover, and ambiguity in the reference evidence. Results use the finalized difficulty assignments rather than this initial prior. Table 10 reports full-suite scores under the finalized Easy/Medium/Hard assignments; tier counts are given in Appendix D.2.

Indoor answer keys. Indoor answer keys provide the state of the human-assembled scene together with an author-provided aligned mapping between intended input and ground-truth actor states. They also include an author-provided task-to-operation grouping used for the descriptive operation-category analysis in Section 3.4. Unlike outdoor records, indoor category labels and actor-state mappings do not specify atomic edit sequences or chain lengths.

Table 4: Candidate public-set sizes at selection time. Errors are leave-one-model-out mean absolute errors of full-suite means on the $[0, 1]$ scale, averaged over setting totals or their components. Reversed pairs count agent configuration pairs whose full-suite scores differ by at least 0.02 but are ordered oppositely, over totals and components.

Cases	T2S/Indoor/ Outdoor	Kept	Total error	Component error	Reversed pairs
61	16/25/20	32.1%	0.016	0.025	8
75	20/25/30	39.5%	0.014	0.019	1
95	20/25/50	50.0%	0.011	0.017	1
128	28/50/50	67.4%	0.007	0.011	0

A.3 Public and Private Sets

The benchmark contains 190 cases, including 30 Text-to-Scene and 160 Image-to-Scene cases, divided into a 95-case public set and a 95-case private set. Task code will be released for the public set but not for the private set. The public set contains 20 Text-to-Scene, 25 Indoor and 50 Outdoor cases, and the private set contains the remaining 10, 55 and 30. All 14 agent configurations are evaluated on the public set; ten are also evaluated on the private set. Table 2 reports these cohorts in separate panels. Unless marked otherwise, analyses use the public set. Full-suite scores combine both sets for the ten agent configurations with all 190 cases, without changing submissions or scores.

Selection. The public set was selected so that its plain per-setting mean, without case weights or calibration, estimates the corresponding full-suite mean. Within each setting, greedy forward selection added the case that most reduced the largest estimation error over the available agent configurations and score columns, with each column's error normalized by the variation of random subsets of the same size. The selection pool contained six agent configurations evaluated on every setting, namely GPT-5.6 Sol (high), Gemini 3.8 Flash (high), Qwen 3.8 27B (thinking off), Qwen 3.8 27B (thinking on), Gemma 4 31B (thinking off) and Gemma 4 31B (thinking on), plus Gemini 3.7 Flash (high) for Text-to-Scene only. Selection used only case scores, not difficulty labels, content packages or categories. One selected Text-to-Scene case was later withdrawn from the benchmark and replaced by another Text-to-Scene case. The candidate sets are nested (61, 75, 95 and 128 cases), so the public set can be enlarged without rerunning its current cases.

Selection-time validation and size. Table 4 reports leave-one-model-out error at selection time. Each agent configuration is removed in turn, the cases are reselected from the remaining agent configurations, and the removed agent configuration's full-suite mean is predicted by its mean on the selected cases. The 95-case set has mean absolute error 0.011 on setting totals and 0.017 on their components, with no reversed pair of totals. Random subsets of the same size had 1.3–1.7 times the error of greedy selection. Greedy selection also had lower error than stratification by content package and category on both repair settings, and tied it on Text-to-Scene. Anchor-point selection by k-medoids (Vivek et al., 2024) was competitive on Indoor but requires per-case weights, so uniform weights were retained. Keeping only cases with mean score between 0.3 and 0.7 (Ndzomga, 2026) was two to four times worse than greedy selection and worse than random sampling. That filter targets model ranking on pass/fail tasks; discarding easy and hard cases biases continuous-score means.

For the size, we use two external reference points: an average estimation error near 0.02, as reported by tinyBenchmarks with 100 curated examples (Maia Polo et al., 2024), and the 44–70% task reduction that preserves agent rankings in Ndzomga (2026). The 61-case set was optimized for totals only, and its component error exceeds 0.02. The 128-case set reduces the suite by less than 44%. The 75- and 95-case sets meet both reference points on average. The additional 20 cases of the 95-case set are Outdoor cases; they reduce the Outdoor total error from 0.019 to 0.010 and the Outdoor Physical Safety error, the largest Outdoor column error at 75 cases, from 0.029 to 0.011. Both sets use the same Indoor cases; their Physical Safety and repair errors (0.022 and 0.029) still exceed 0.02.

Validation under current scoring. We compare public-set and full-suite scores under the current scoring for the ten agent configurations evaluated on both (Table 5). Six of them were in the selection pool; Muse Spark 1.3 (medium), GLM-5.3 Flash (max), Inkling (high) and DeepSeek V4.1 Flash (high) were not. The overall model score differs by 0.008 on average and at most 0.024, and the public-set ranking agrees with the full-suite ranking up to one pair separated by 0.004. Among

Table 5: Public-set versus full-suite scores under current scoring for the ten agent configurations evaluated on both. Δ is the public-set score minus the full-suite score. Kendall's τ compares rankings. Reversed pairs have full-suite gaps of at least 0.02 but the opposite public-set order.

Score	Mean $\lvert\Delta\rvert$	Max $\lvert\Delta\rvert$	Mean Δ	Kendall τ	Reversed pairs
S_{T2S}	0.009	0.023	-0.003	0.956	0
Indoor	0.034	0.115	$+0.028$	1.000	0
Outdoor	0.009	0.016	0.000	0.956	1
S_{I2S}	0.013	0.030	-0.005	1.000	0
S_{model}	0.008	0.024	-0.004	0.956	0

pairs whose full-suite scores differ by at least 0.02, one is reversed: Muse Spark 1.3 (medium) and DeepSeek V4.1 Flash (high) on Outdoor, with a full-suite gap of 0.029. The public-set S_{I2S} gives Indoor weight $1/3$ rather than $1/2$ (Appendix C.6). With this weighting, S_{I2S} differs by 0.013 on average and at most 0.030, whereas averaging the two public-set domain means equally would give 0.016 and 0.051. Errors are larger for the four agent configurations outside the selection pool than for the six inside it (0.010 versus 0.006 on average for the overall score), consistent with leave-one-model-out estimates describing agent configurations similar to the selection pool.

Use of public-set scores. We assess public-set representativeness through the score differences above (Miller, 2024). These validation errors describe the ten shared configurations; scores for the four public-only configurations refer to the selected 95 cases. The public set approximates setting means but is not proportional across content packages, categories or difficulty tiers; its category and tier results describe the public cases rather than estimating full-suite values. Sub-metrics outside the selection objective, such as individual Physical Safety leaves, deviated by up to 0.056 at selection time. Estimates may fail far outside the selection pool's score range (Zhang et al., 2025).

B Running Cases

Figures 7 and 8 show two Text-to-Scene cases. Each figure gives the prompt's scene description and the four overview views for a higher-scoring and a lower-scoring run, with scores and per-case ranks among the 14 agent configurations.

Figures 9 and 10 show two Image-to-Scene cases. Each figure summarizes the task instruction, then shows the withheld ground truth and the successful and unsuccessful submissions, all rendered from the evaluator's four repair-target cameras, which are identical across rows.

C Evaluation Protocol

This appendix defines the evidence and scoring protocols shared by the public and private sets. Validation and navigation results appear in Appendices D.7 and D.6.

C.1 Evaluation Outputs and Verifier Routing

The evaluator reports component scores, supporting measurements, applicability, and evaluation status. A MEASURED result has a finite score in $[0, 1]$; it denotes a completed measurement, not a passed quality threshold. NOT APPLICABLE, NOT EVALUATED, and ERROR have no numeric measurement. VALID and INVALID describe submission integrity. The conversion from component results to reported benchmark scores, including zero-score rules, is defined in Appendix C.6.

Every case undergoes Candidate Integrity and Physical Safety checks. Text-to-Scene additionally uses Detailed Alignment and Overview Alignment; Image-to-Scene uses actor-level Precision, Recall and Repair F1 (Appendix C.5). Evaluation runs in an isolated environment, and its reports are unavailable to agents during construction.

Judge configuration. Every VLM judgment made during evaluation uses Qwen 3.8 27B (Qwen/Qwen3.8-27B) with thinking disabled, temperature 0, seed 0, structured output, and a 16,000-token response limit. This configuration covers Detailed Alignment requirement judgments, both

Prompt: Nordic Harbour Build me a compact, sunlit Copenhagen-style canal district with narrow cobblestone streets and rectangular waterways enclosing dense urban blocks. Arrange roughly thirty attached four- to six-storey townhouses in continuous street and canal-front rows, using slender façades, repetitive white-framed windows, arched doors and carriage entrances, modest cornices, and steep tiled roofs packed with dormers and brick chimneys. Include occasional exposed party walls between staggered building rows. Finish the façades in weathered muted red, mustard yellow, dusty blue, warm grey, beige, and exposed brick, with red, orange, and charcoal roof tiles. Run a principal waterfront street along a stone quay, crossed by smaller streets and linked across the canals by short, gently arched stone-and-metal bridges, including one opening onto a broad paved promenade with rounded landings. Form the canal edges from stepped pale stone, dark timber retaining walls, and iron railings around reflective rippling water. Add regularly spaced leafy trees, ornate black lamps, stone bollards, benches, European road signs, zebra crossings, puddles, grime, and sparse weeds.
The text prompt also contains the ground instruction and the 79-asset catalog, omitted here.

Higher-scoring: Claude Fable 5.1 (max), rank 2 of 14 S_{T2S} 0.865 | Detailed 0.666 | Overview 0.894 | Physics 0.980



Lower-scoring: Inkling (high), rank 11 of 14 S_{T2S} 0.396 | Detailed 0.021 | Overview 0.373 | Physics 0.841

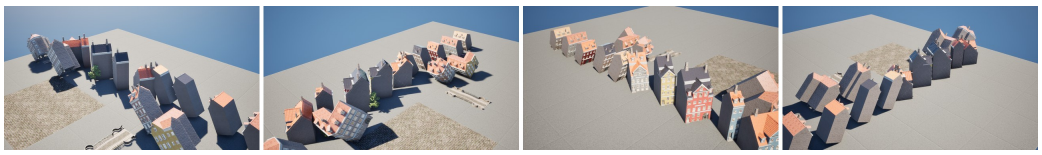
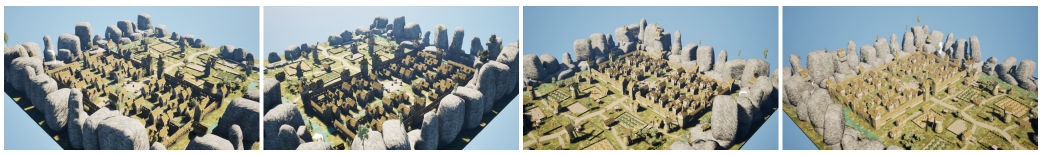


Figure 7: Text-to-Scene case: Nordic Harbour. Fable places 2,872 actors, arranging townhouses around rectangular canals with quays and bridges. Inkling places 24 actors, largely tilted townhouses on a bare plate without visible canals or quays.

Prompt: Medieval Big Farm Town Build me a sprawling medieval farm town set in a rugged, enclosed mountain basin, with a dense fortified settlement occupying the rocky northern half and broad, open farmland spreading southward. Ring the scene with massive irregular gray boulders, steep mossy outcrops, scattered grassy plateaus, and narrow turquoise streams that descend through gaps in the rocks, including several small white waterfalls. Place roughly sixty to eighty compact timber-and-stone buildings inside and around a rectangular walled town: mostly one- and two-story houses with dark wooden framing, pale infill or masonry walls, and steep orange-brown tiled roofs. Arrange them tightly and organically along narrow dirt lanes, small courtyards, canals, and bridges, with taller square gate towers and watchtowers punctuating the long timber-and-stone walls. Outside the walls, transition to a much sparser patchwork of fenced fields, orchards, livestock enclosures, winding dirt tracks, isolated cottages, sheds, tents, carts, stacked materials, and scattered rocks. Distribute several tall wooden windmills across the fields, some attached to heavy stone or timber tower bases, and place additional lookout buildings on remote rocky peaks and grassy ledges. Let streams skirt both sides of the settlement, feed small turquoise pools within the town, and pass beneath simple wooden footbridges. Use uneven grassy ground with worn pale paths, low fences, shrubs, tall grass, and thin, irregularly spaced trees, keeping the town center crowded and layered while the agricultural outskirts remain open, loosely ordered, and visibly bounded by cliffs and boulder formations.
The text prompt also contains the ground instruction and the 158-asset catalog, omitted here.

Higher-scoring: GPT-6 Astra (max), rank 3 of 14 S_{T2S} 0.767 | Detailed 0.747 | Overview 0.760 | Physics 0.809



Lower-scoring: DeepSeek V4.1 Flash (high), rank 14 of 14 S_{T2S} 0.258 | Detailed 0.142 | Overview 0.049 | Physics 1.000

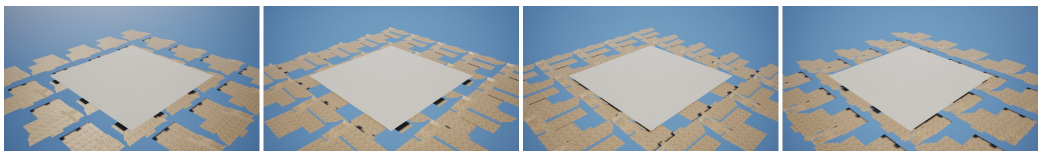
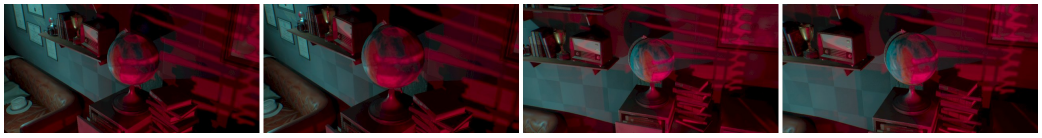


Figure 8: Text-to-Scene case: Medieval Big Farm Town. Astra places 1,518 actors in a walled settlement with timber houses, fenced fields and boulders. DeepSeek places 25 ground tiles on an otherwise empty plate. Its Physical Safety score is 1.000 with no detected support or penetration violations, while Overview Alignment is 0.049.

Task: Indoor Atomic v1 054 (Detective Office) Repair the current indoor scene so it matches the two provided reference images of the human-assembled scene. Make only the minimum changes needed, and preserve all unrelated actors and properties.
Corruption, not shown to the agent: the world globe's mesh is replaced with a fan.

Ground truth, withheld from the agent



Successful: Claude Opus 5 (max), tied 1st of 14 S_{I2S} 1.000 | F1 1.000 (TP/FP/FN 1/0/0) | Physics 1.000



Unsuccessful: Gemma 4 31B (thinking off), tied last of 14 S_{I2S} 0.100 | F1 0.000 (TP/FP/FN 0/1,310/1) | Physics 0.500

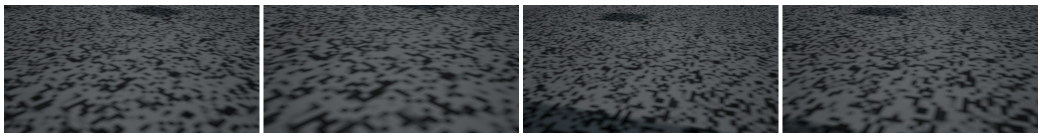


Figure 9: Image-to-Scene running case: Indoor Atomic v1 054. The successful run restores the world globe and changes nothing else. The unsuccessful run does not restore the globe and deletes nearly the whole room, keeping 31 of 1,325 actors. Rows are brightened identically.

Task: Comp 31 Old Industrial Pallet Bay Repair (Old Industrial Courtyard) Repair the current outdoor scene so it matches the provided reference image. Make only the minimum changes needed, and preserve all unrelated actors and properties.
Corruption, not shown to the agent: a duplicate of the stacked pallets is added $2.8\,m$ *away, and the original stack is moved by* $2\,m$ *and rotated by* 35°.

Ground truth, withheld from the agent



Successful: GPT-6 Astra (max), rank 1 of 14 S_{I2S} 1.000 | F1 1.000 (TP/FP/FN 2/0/0) | Physics 1.000



Unsuccessful: DeepSeek V4.1 Flash (high), tied 12th of 14 S_{I2S} 0.100 | F1 0.000 (TP/FP/FN 0/5/2) | Physics 0.500



Figure 10: Image-to-Scene running case: Comp 31 Old Industrial Pallet Bay Repair. The successful run removes the duplicate and returns the original stack, with no other change. The unsuccessful run keeps the duplicate, restores neither stack, and makes four unintended edits.

Overview Alignment passes, and the Image-to-Scene Agentic Verification comparisons of paired target views, which are reported but not scored. Actor Repair F1 itself uses structured scene state.

Table 6 summarizes the evaluation groups.

Table 6: Evaluation groups and their inputs. Ground truth is withheld from agents.

Group	Input	Criterion
Candidate Integrity all cases	Saved scene, execution record, and dependencies	Loadability, minimum content, scene-graph validity, dependency resolution, and asset-library manifest agreement
Physical Safety all eligible cases	Scene geometry and targeted engine measurements	Supported actors and collision-free actors
Semantic Verifier Text-to-Scene	Prompt, frozen requirements, scene state, and rendered evidence	Detailed Alignment and Overview Alignment
Ground-Truth-Based Verifier Image-to-Scene	Input scene, submitted scene, and withheld human-assembled scene	Actor repair TP/FP/FN, precision, recall and F1
Downstream EUS reported separately	Navigation mesh, eligible targets, and fixed-walker rollouts	Approachability and route completion

C.2 Candidate Integrity Checks

The evaluator independently exports the submitted scene and applies three integrity checks:

1. *Scene snapshot.* Validate the exported scene-graph schema and check the task's minimum actor count.
2. *Content and dependencies.* Verify that referenced packages resolve under the frozen dependency policy.
3. *Asset-library manifest.* Verify agreement among the submission, scorer and frozen content snapshot.

Ending at a resource limit does not itself invalidate a saved .umap; the saved scene remains eligible for these checks. Downstream verification requires all three checks to pass. A failed or errored gate leaves downstream measurements unevaluated; reporting follows Appendix C.6.

C.3 Physical Safety

Physical Safety combines support and solid-penetration checks. For a nonempty evaluated support population, $r_{\mathrm{unsup}} = N_{\mathrm{unsup}}/N_{\mathrm{tested}}$ is the fraction of actors for which the profile finds no support. This is a lower-is-better defect rate; its safety counterpart is $1-r_{\mathrm{unsup}}$. For $N_{\mathrm{eval}} > 0$, the penetration score $S_{\mathrm{pen}} = 1 - N_{\mathrm{viol}}/N_{\mathrm{eval}}$ is the collision-free fraction of eligible actors with trusted verdicts. For nonnegative weights $\mathbf{w}^P = (w^P_{\mathrm{supp}}, w^P_{\mathrm{pen}})$ summing to one,

$$S_{\mathrm{physics}}(\mathbf{w}^P) = w^P_{\mathrm{supp}}(1 - r_{\mathrm{unsup}}) + w^P_{\mathrm{pen}} S_{\mathrm{pen}}.$$

The adopted policy uses $\mathbf{w}^P = (0.5, 0.5)$ in both settings; alternative fitted weights are used only in the Text-to-Scene human-alignment comparison (Table 9). We omit the weight argument when referring to the adopted Physical Safety score. Required checks without numeric measurements, including explicitly inapplicable checks, contribute zero at their fixed weights. Case scores are averaged equally, rather than pooling actor counts across cases. An evaluated check with no eligible actors receives the protocol's neutral score of one; this differs from an unavailable measurement. An empty-scope score does not certify whole-scene safety; reports list each check's eligible population.

Support. Both task settings use a 5 cm support-gap tolerance. Text-to-Scene evaluates generated actors after excluding infrastructure and degenerate helpers. Its geometric support test uses world-space axis-aligned bounding boxes (AABBs): an actor is supported when its bottom is at most 5 cm above the world ground plane, or another actor lies beneath it with a vertical gap in $[0, 5]$ cm, including flush contact, and that actor's footprint, seen from above, contains its pivot within a small margin. Image-to-Scene evaluates edited actors using targeted Unreal support checks, with a 5 cm

ground-gap tolerance and, when applicable, a 5 cm lateral-support tolerance. These are static support checks, not tests of dynamic stability.

Solid penetration. The scene-wide Text-to-Scene population and the edited-actor Image-to-Scene population use deterministic scene-relative physics roles. Non-solid actors, support surfaces, and oversized environment proxies are excluded as scored targets. A trusted collision with an environment proxy can nevertheless count as a containment failure for a target actor. AABB overlap is only a broad phase; a violation requires confirmed Unreal collision-body overlap and native FHitResult minimum-translation depth exceeding the actor's tolerance. For Text-to-Scene, the tolerance is 5% of the shortest full world-AABB span, clamped to $[5, 50]$ cm; Image-to-Scene uses a fixed 5 cm tolerance. These are actor-contact thresholds, not scene-level pass/fail thresholds. Incomplete evidence can prove a violation but cannot establish a clean actor; a leaf with unresolved evidence and no confirmed violation remains NOT EVALUATED. Reports retain evaluated and unresolved actor counts, roles, contact depths, and effective tolerances. A reported fraction must therefore be interpreted together with its measurement coverage.

C.4 SEMANTIC VERIFIER FOR TEXT-TO-SCENE

Detailed Alignment requirements. Each Text-to-Scene prompt is parsed once into a graph of requirements with bindings to scene evidence. Validation checks prompt grounding, semantic coverage, reference resolution, scope, and graph consistency. The requirements are fixed before evaluation. They are generated automatically or with agent assistance and are not human-reviewed.

Evidence ladder. Stage 1 uses the trusted generated-scene inventory, identity bindings, and deterministic geometry for claims that can be resolved structurally. Stage 2 localizes the actors or collections needed by unresolved claims and captures targeted RGB evidence. Stage 3 sends only unresolved visual claims to the multimodal judge, with a bounded alternate view and a separate best-evidence binary arbitration when valid images remain. Camera selection controls observations but is not positive evidence. Missing targets or required images yield NOT EVALUATED; transport, parsing, contract, or capture failures yield ERROR. Reports retain all decisions and evidence.

Detailed Alignment aggregation. Each accepted atomic decision has score $s_j \in [0, 1]$. Binary matches and mismatches receive one and zero, deterministic checks may return intermediate scores, and an unresolved applicable requirement receives zero while retaining its weight,

$$q_j = \begin{cases} s_j, & \text{for an accepted decision,} \\ 0, & \text{for an unresolved requirement.} \end{cases}$$

Duplicate summaries already represented by scored descendants add no weight, and a retained conjunction is capped by its nested requirements. Known Coverage separately records the scoring weight for which the evidence supports a definite decision.

Atomic predicates are aggregated first within prompt clauses and then within four Detailed Alignment families. For family f, let $\mathcal{C}_f$ contain its nonempty clauses and $\mathcal{R}_{fc}$ the active predicates in clause c. For any requirement-level quantity x,

$$\mathcal{A}_f(x) = \frac{1}{|\mathcal{C}_f|} \sum_{c \in \mathcal{C}_f} \frac{1}{|\mathcal{R}_{fc}|} \sum_{j \in \mathcal{R}_{fc}} x_j.$$

Applying this operator to q_j gives the family score S_f. With nonnegative family weights $\mathbf{w}^D$,

$$S_{\text{detailed}} = \frac{\sum_{f \in \mathcal{F}} w_f^D S_f}{\sum_{f \in \mathcal{F}} w_f^D}, \tag{1}$$

where $\mathcal{F}$ contains the families applicable to the prompt. An absent family is removed and the remaining weights are renormalized; an unresolved requirement inside an applicable family is not removed. The published aggregate is a point score, accompanied by Known Coverage, family scores, and atomic decisions. If $k_j = 1$ when evidence determines a definite decision and $k_j = 0$ otherwise, then $K_f = \mathcal{A}_f(k)$; aggregate Known Coverage uses the same applicable-family weights as S_{detailed}. A conjunction with a known-false child is determined. Table 9 lists adopted and fitted weights.

Conditional Spatial Composition analysis. Figure 6a reports mismatch rates by predicate type within the Spatial Composition family: spatial relations between entities (e.g., along, between, inside), distributions of groups of entities (e.g., clusters or rows), and scene-level composition (e.g., overall density, openness, or a loose grid). A requirement is included only when every relevant entity passes a separate affirmative existence check without additional qualifiers. Missing, contradictory or unresolved existence evidence is excluded. Across 264 audited model–case pairs, these checks retain 369 of 4,455 Spatial Composition predicates (8.3%). Collection presence does not establish exact cardinality or member identity. The analysis uses direct predicate judgments before parent constraints are applied, separating spatial mismatches from inherited score reductions.

Overview Alignment. A scene-graph and clearance-aware camera plan produces exactly four generated-scene-only RGB overview views. A Prompt-Aware multimodal call scores global prompt alignment, composition and layout, style and atmosphere, and completeness and polish. With nonnegative weights $\mathbf{w}^O$ summing to one, their combination is

$$S_{\text{prompt}} = w^O_{\text{align}} S_{\text{align}} + w^O_{\text{layout}} S_{\text{layout}} + w^O_{\text{style}} S_{\text{style}} + w^O_{\text{complete}} S_{\text{complete}}. \tag{2}$$

An independent Prompt-Free call uses the same views to score intrinsic structural integrity, including stretching, shearing, tearing, collapse, and incoherent fragmentation. It does not score missing requested content, style, scene density, support, or inter-object collision. The resulting score is

$$\widetilde{S}_{\text{overview}} = S_{\text{prompt}} \left(0.75 + 0.25 S_{\text{struct}}\right).$$

If the evaluator identifies a specific severe defect with confidence at least 0.8 and corroborating evidence from at least two views, then

$$S_{\text{overview}} = \min\left(\widetilde{S}_{\text{overview}}, 0.40\right);$$

otherwise $S_{\text{overview}} = \widetilde{S}_{\text{overview}}$. The visible-scene summary is explanatory and is never used as a scoring intermediate. Table 9 reports the adopted and fitted Prompt-Aware weights.

Judging criteria and scoring rubrics. For Detailed Alignment, visual judgments use only evidence visible in rendered RGB images and cite the views used. For an unresolved visual requirement, the judge returns MATCH, MISMATCH, or UNKNOWN; insufficient visibility of the subject, relation scope, or counted collection yields UNKNOWN. A broader or functionally related category cannot substitute for a named object. Object identity is checked against the expected category and its aliases, separately from count, position, material, and style. When applied, final binary arbitration returns MATCH or MISMATCH and records residual uncertainty as confidence.

For Overview Alignment, Prompt-Aware scoring treats the task prompt as the evaluation target and uses anchors 0 (absent or contradictory), 0.25 (weak), 0.5 (partial), 0.75 (mostly satisfied), and 1 (clearly complete) for the dimensions in Eq. 2. A requested element counts if clearly visible in at least one view; capture quality and structural integrity are excluded from this score. Prompt-Free scoring assesses intrinsic geometry integrity using anchors 1 (no visible defect), 0.75 (minor localized defects), 0.5 (clear defects in several structures or one major structure), 0.25 (widespread severe deformation), and 0 (unusable). Floating, unsupported, or surreal layouts are assessed under prompt alignment or Physical Safety; severe geometric defects require evidence from at least two views.

Semantic Verifier aggregation. Detailed Alignment and Overview Alignment form one Semantic Verifier. For nonnegative final weights $\alpha + \beta + \gamma = 1$ with $\alpha + \beta > 0$,

$$S_{\text{semantic}} = \frac{\alpha S_{\text{detailed}} + \beta S_{\text{overview}}}{\alpha + \beta}. \tag{3}$$

For a case with fixed evidence $\mathbf{x}$, Physical Safety enters the Text-to-Scene case score as

$$\begin{aligned} S_{\text{T2S}}(\mathbf{x}; \theta) &= (\alpha + \beta) S_{\text{semantic}} + \gamma S_{\text{physics}} \\ &= \alpha S_{\text{detailed}} + \beta S_{\text{overview}} + \gamma S_{\text{physics}}, \end{aligned}$$

where $\mathbf{x}$ contains the frozen requirement decisions, Overview Alignment and Physical Safety measurements, and applicability information, and $\theta = (\mathbf{w}^D, \mathbf{w}^O, \mathbf{w}^P, \alpha, \beta, \gamma)$. The component scores on the right are computed from $\mathbf{x}$ using their respective weight blocks in θ; these arguments are suppressed for brevity. Unavailable required components retain their fixed weights and receive zero credit. Table 9 reports the adopted and individual-fit values.

Table 7: Nominal acceptance tests for desired-present repair targets. All applicable tests must pass; positions are in cm. These benchmark tolerances are not human-calibrated perceptual equivalences.

Quantity	Acceptance test
Asset and class	Exact normalized structural descriptor agreement
World position	$\|p_c - p_g\|_2 \leq 5$ cm
World rotation	$2\arccos(\min(1, \|q_c^\top q_g\|)) \leq 5^\circ$
Signed scale	$\max_j \|s_{c,j}/s_{g,j} - 1\| \leq 0.05$
Bounds centre, if recorded in GT	$\|b_c - b_g\|_2 \leq 5$ cm
Bounds extent, if recorded in GT	$\max_j \|e_{c,j}/e_{g,j} - 1\| \leq 0.05$
Attributes, if recorded in GT	Exact property-dictionary, normalized material-set and component/material-slot agreement

C.5 Ground-Truth-Based Verifier: Actor Repair F1

Actor Repair F1 measures target recovery and penalizes incorrect outputs and unintended changes to unrelated content. It uses independently exported input, ground-truth and candidate scenes, with each affected actor as one repair target.

Defining the repair targets. Comparing the input with the ground truth determines which actors must be added, removed or restored. These targets are fixed before inspecting the candidate, using the authored input–ground-truth correspondence and shared evidence fields. Let n_+ denote targets that should be present after repair (additions and state restorations), and n_- denote required removals. An actor remains one target even when several of its properties need repair. Every benchmark case has at least one target. Camera and scene-capture actors are excluded, and changes to runtime identifiers, labels or role tags alone are not semantic edits.

Determining repair success. A target that should be present is recovered only when a candidate actor has the correct asset and class and passes every applicable state test in Table 7. Asset and class agreement uses a normalized structural descriptor comprising actor class, mesh asset or Blueprint class, and component-asset signature. State tests cover position, rotation, signed scale, and any bounds or attributes recorded in the ground truth. The whole target state is checked, including properties that did not require repair. A one-to-one matching maximizes the number of recovered targets; one candidate actor cannot satisfy two targets, and partially restored targets receive no partial credit. Removal requires semantic absence of the original actor, not just renaming.

Counting correct and incorrect outcomes. Let m be the number of successfully matched targets and d the number of successful required removals. To count incorrect outputs, define C as the candidate actors that are newly added, semantically changed, or surviving targets that should be present. This includes a corrupted target left unchanged. An unchanged surviving actor that should have been removed instead counts as a missed removal. Let o count unintended losses of background actors, including actors removed or repurposed to satisfy a repair target; overlapping losses are counted once. The outcome counts are

$$\begin{aligned} \mathrm{TP} &= m + d, \\ \mathrm{FP} &= |C| - m + o, \\ \mathrm{FN} &= (n_+ - m) + (n_- - d). \end{aligned} \tag{4}$$

True positives (TP) count completed targets, false negatives (FN) count missed targets, and false positives (FP) count incorrect candidate outputs and unintended background losses. Every target contributes either a TP or an FN, so $\mathrm{TP} + \mathrm{FN} = n_+ + n_-$. This also covers pure deletions.

Computing Repair F1. Precision measures the share of counted repair outcomes that are correct, recall measures the fraction of targets recovered, and F1 combines them:

$$P = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}, \qquad F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP} + \mathrm{FP} + \mathrm{FN}}. \tag{5}$$

For illustration, suppose a task requires restoring a missing globe, repositioning a chair, and deleting an extra box. A candidate restores the globe and deletes the box, but leaves the chair misplaced and deletes an unrelated plant. The globe and box contribute two TP; the misplaced chair contributes

one FP and one FN; the lost plant adds one FP. Hence $(\mathrm{TP}, \mathrm{FP}, \mathrm{FN}) = (2, 2, 1)$, giving $P = 0.5$, $R = 2/3$ and $F_1 = 4/7$.

We set $P = 0$ when its denominator is zero. Nonempty target sets make the recall and F1 denominators positive. Unchanged background actors do not contribute true negatives: arbitrary new actors and edit states do not define a finite negative population. Cases assigned zero under Appendix C.6 receive zero Precision, Recall and F1; these reporting zeros are not measured TP/FP/FN counts.

Aggregating case scores. Precision, Recall and F1 are averaged over scheduled cases with equal case weights (Appendix C.6). Macro F1 is not the harmonic mean of macro Precision and Recall; pooling counts would give greater weight to cases with more targets and false positives. Repair F1 enters the I2S aggregate defined in Eq. 6.

Matching actors across input and candidate scenes. The input–candidate correspondence identifies which actors were preserved, changed, added or removed before the outcome counts above are computed. It does not use actor names or GUIDs. Matching first maximizes unchanged correspondences, preserving the multiplicity of unaffected background actors before matching target actors. Remaining actors with the same structural descriptor are assigned at minimum cost, $\delta/(1+\delta) + 0.2\varphi/180 + 0.05k$, where δ is position distance in cm, φ is quaternion angular distance in degrees, and k is the number of changed fields. Remaining same-class actors at the same pose can anchor an asset replacement or cleared mesh. Such correspondence identifies an edit; repair success still requires the ground-truth tests above. In an asset replacement, removing the old asset can satisfy a required removal, while the replacement is assessed separately as a candidate presence.

Evidence and numerical tolerances. Unchanged-state detection uses tolerances for editor save/export round trips: 0.1 cm per position axis, 0.01° quaternion rotation, and 10^{-4} absolute scale per axis, together with agreement of recorded attributes. These tolerances identify changes; the larger thresholds in Table 7 determine repair success. Equivalent Euler representations and renamed or recreated equivalent actors do not create spurious additions or deletions. Bounds-only changes of background actors are not independent edits, but target bounds remain subject to acceptance tests. Comparisons use absolute Unreal world coordinates, without fitting a global translation or yaw offset or allowing object-symmetry equivalences.

Rotations are compared as unit quaternions, and signed scale detects reflections. Missing, malformed or nonfinite required evidence fails. A zero ground-truth scale is invalid; a zero bounds-extent component passes only if the candidate component is also zero. Bounds and attribute channels absent from the ground truth are not required; the candidate cannot disable a required channel. Numeric comparisons allow a 10^{-9} boundary guard. Thresholded one-to-one pose evaluation has precedent in Scan2CAD and BOP (Avetisyan et al., 2019; Hodaň et al., 2020); their error functions, symmetry policies and thresholds are not adopted here. The nominal profile is $(5\text{ cm}, 5^\circ, 5\%)$; Appendix D.5 also tests strict $(2.5\text{ cm}, 2.5^\circ, 2.5\%)$ and relaxed $(10\text{ cm}, 10^\circ, 10\%)$ tolerances on public cases.

C.6 Score Aggregation and Reporting

The Text-to-Scene case score assigns 20% to Detailed Alignment, 60% to Overview Alignment, and 20% to Physical Safety. The Image-to-Scene case score assigns 80% to actor Repair F1 and 20% to Physical Safety:

$$S_{\mathrm{I2S}} = \lambda_{\mathrm{rep}} F_1 + (1 - \lambda_{\mathrm{rep}}) S_{\mathrm{physics}}, \qquad \lambda_{\mathrm{rep}} = 0.80. \tag{6}$$

The same weights apply to all configurations within each setting. T2S weights were informed by human judgments and are assessed in Section 3.3. Scores lie in $[0, 1]$; multiplying by 100 gives the percentage scale used in some figures.

Every scheduled case remains in its evaluation subset's denominator. Case i contributes

$$s_i = \begin{cases} S_i, & \text{for a valid submission with an authoritative primary score,} \\ 0, & \text{for an authoritative invalid verdict, a missing submission, or a missing result.} \end{cases}$$

All subscores, including repair Precision, Recall and F1, are zero for an invalid submission, including dependency-check failures. Missing submissions and missing results therefore also count as zero and remain in the denominator, as for the seven absent Muse Spark 1.3 (medium) submissions and the

Table 8: Public evaluations assigned zero, by configuration and reason (19 of 1,330 pairs). Configurations with no such cases are omitted.

Agent configuration	Track	Unresolved dependencies	Empty scene	No submission	Missing result	Total
Muse Spark 1.3 (medium)	Outdoor	–	–	7	–	7
DeepSeek V4.1 Flash (high)	T2S	4	1	–	–	5
GLM-5.3 Flash (max)	T2S	2	–	–	2	4
Claude Opus 5 (max)	T2S	1	–	–	–	1
GPT-6 Astra (max)	T2S	1	–	–	–	1
Grok 4.6 (high)	T2S	1	–	–	–	1
Total		9	1	7	2	19

two missing public T2S results for GLM-5.3 Flash (max), Bazaar and Paris. Within an otherwise valid case, unavailable required components retain their weights and receive zero credit; explicit inapplicability and optional visual fallbacks follow the metric-specific rules in Appendices C.3, C.4, and C.5.

Table 8 lists the 19 zero-score pairs among 1,330 public evaluations. Unresolved dependencies arise when the saved scene refers to assets created during the run that were not saved with it. The seven absent Muse submissions correspond to the Middle East Outdoor cases.

For an evaluation subset t with all N_t^{score} scheduled contributions resolved, the mean score is

$$\overline{S}_t = \frac{1}{N_t^{\text{score}}} \sum_{i=1}^{N_t^{\text{score}}} s_{ti}.$$

Otherwise the aggregate remains unreported, without dropping missing cases or publishing score bounds. The overall model score is

$$S_{\text{model}} = 0.50\overline{S}_{\text{T2S}} + 0.50\overline{S}_{\text{I2S}}. \tag{7}$$

Here the Image-to-Scene mean gives each scheduled case equal weight:

$$\overline{S}_{\text{I2S}} = \frac{N_{\text{in}}\overline{S}_{\text{I2S,indoor}} + N_{\text{out}}\overline{S}_{\text{I2S,outdoor}}}{N_{\text{in}} + N_{\text{out}}}.$$

The public set uses 25 Indoor and 50 Outdoor cases, giving domain weights $1/3$ and $2/3$; the private set uses 55 and 30, giving $11/17$ and $6/17$. The full suite uses 80 cases in each domain and therefore equal domain weights. Component scores are reported alongside the aggregate. Known Coverage, visual availability, resource usage, and embodied utility are auxiliary measurements and do not enter S_{model}. An overall score requires complete results for both settings. Case-level T2S scores are rounded to four decimal places and I2S scores to six before aggregation; tables use three decimals.

C.7 Worked Scoring Examples

Figures 11 and 12 trace two tasks from agent output through the evidence and decisions used by each verifier. Each pairs two valid submissions to the same task, selected for visible differences and available evaluation evidence. The examples illustrate the scoring procedure rather than estimate performance across tasks. Displayed component scores use a 0–100 scale; binary requirement verdicts and the intermediate calculations retain their native 0–1 values.

C.8 Embodied Utility Score Implementation

Scene-generation evaluations have measured reachability through layout connectivity or object clearance (Yang et al., 2024a; Tam et al., 2026). Embodied navigation evaluates whether routes are actually traversed (Anderson et al., 2018). EUS combines these complementary checks in reopened Unreal scenes (Ye et al., 2025). The fixed follower isolates scene usability from policy decisions (Chang et al., 2025); SceneSmith likewise separates reach and execution (Pfaff et al., 2026).

Definition and scope. EUS measures navigation by a player agent with a person-sized collision body and a fixed route-following controller in Unreal Engine. It evaluates scene usability rather than

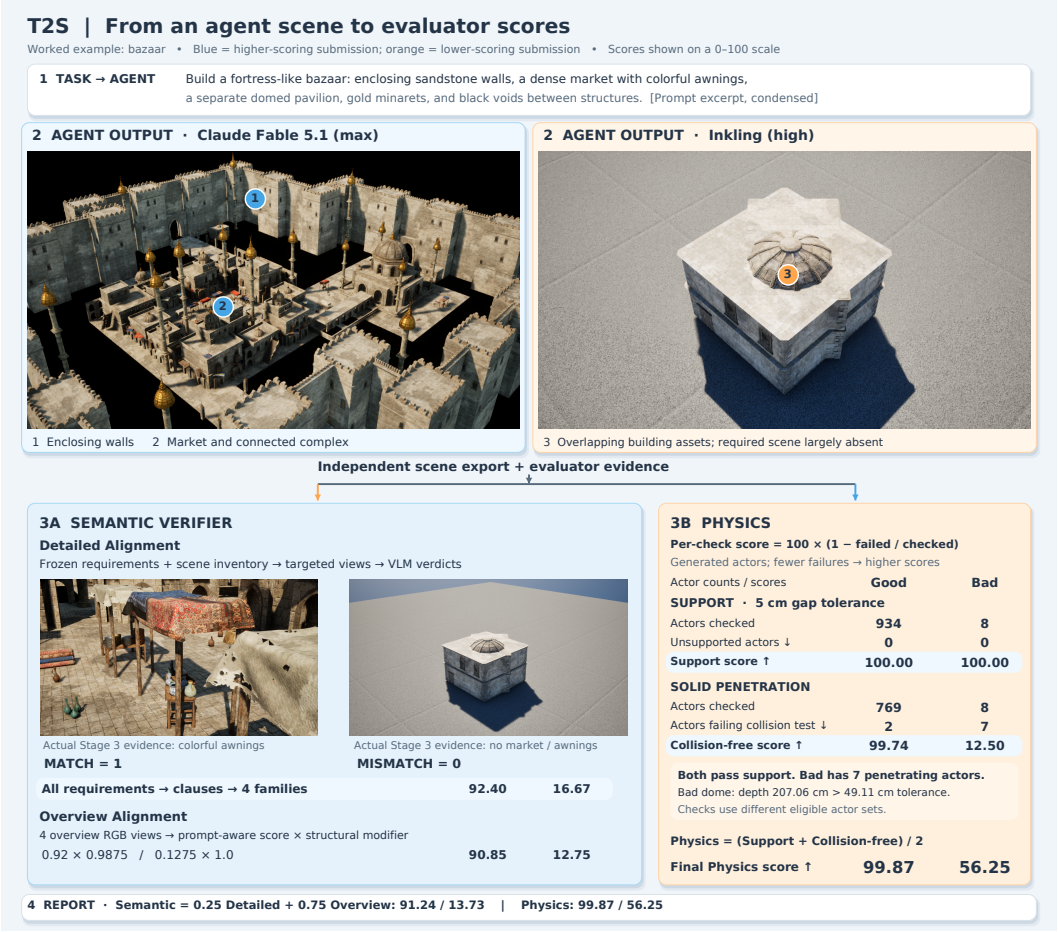


Figure 11: **Text-to-Scene scoring flow on Bazaar.** Claude Fable 5.1 (max) and Inkling (high) receive the same prompt. Targeted views in panel 3A resolve the colorful-awnings requirement; panel 3B shows support and penetration checks with their respective eligible actor populations (Appendix C.3). The dome marker denotes an engine-confirmed collision. Captures are annotated for clarity.

general interaction or learned-policy performance. The submitted scene is reopened in a fresh editor session. Let T be its pool of eligible floor-level, collidable actors. Approachability is the share of this content connected to the main navigable region,

$$A = \frac{\left|\{t \in T : \text{approachable}(t)\}\right|}{|T|}. \tag{8}$$

Walk validation is a simulated capsule's completion rate on selected non-trivial routes,

$$W = \frac{n_{\text{arrived}}}{n_{\text{walks}}}, \qquad r_{\text{arr}} = \min\big(300, \max(120, 0.25h)\big)\ \text{cm}, \tag{9}$$

where r_{arr} is the arrival radius and h is the smaller half-extent of the settled navigation mesh. The raw per-scene result is

$$\text{EUS} = \begin{cases} 0, & |T| < 15 \text{ or certified } A = 0, \\ A\,W, & A \text{ and } W \text{ are measured}, \\ \text{WITHHELD}, & \text{the navigation measurement is unreliable.} \end{cases} \tag{10}$$

A withheld result is not a certified zero and does not establish that the scene is unusable.

Eligible targets and the floor-level pool. The target pool uses live actors and their collision geometry from the reopened .umap. Lights, fog, reflection captures, sky and particle effects, cameras,



Figure 12: **Image-to-Scene scoring flow on roadside billboard reset.** GPT-6 Astra (max) restores the target and preserves unrelated actors. Inkling (high) leaves a 55° rotation error and enlarges two unrelated billboards; these off-target billboards also fail the physical checks and lie outside the displayed view. The reference image is the agent input; comparison images are evaluator base-color renders. Repair F1 uses structured scene states, while Physical Safety uses engine measurements.

engine bookkeeping actors, and surfaces such as floors, roads, and sidewalks are excluded. Semantic categories are used when available; otherwise documented name patterns are applied with near-miss checks. An actor whose base lies more than 150 cm above or 50 cm below the navigation point beneath it is treated as scenery and removed from both numerator and denominator. A floor-level actor that no navigation cell reaches remains in the denominator as unreachable.

Measuring A. Actor footprints come from colliding-component bounds, with the transform pivot used only when collision bounds are unavailable. A single component wider than 200 m in plan or 50 m in height is excluded from navigation bounds so that an environment proxy cannot dominate the build volume. The navigation mesh uses agent radius 50 cm, agent height 180 cm, maximum slope 35°, and minimum region area 150; the evaluator writes and reads back these values for every scene. A build that does not settle within 300 s is withheld.

The settled mesh is sampled by deterministic three-dimensional lattices with 200 cm pitch for connected components, 25 cm pitch for target projection, and 100 cm vertical strata. Each target footprint is projected through nine samples to cells within 100 cm in plan, preferring cells on the target's own storey. A target is approachable when its projected cell has a complete path to the largest connected component. At most three builds are attempted; certification requires two consecutive builds to agree on A within 0.05 and on the number of on-mesh cells within 0.5%.

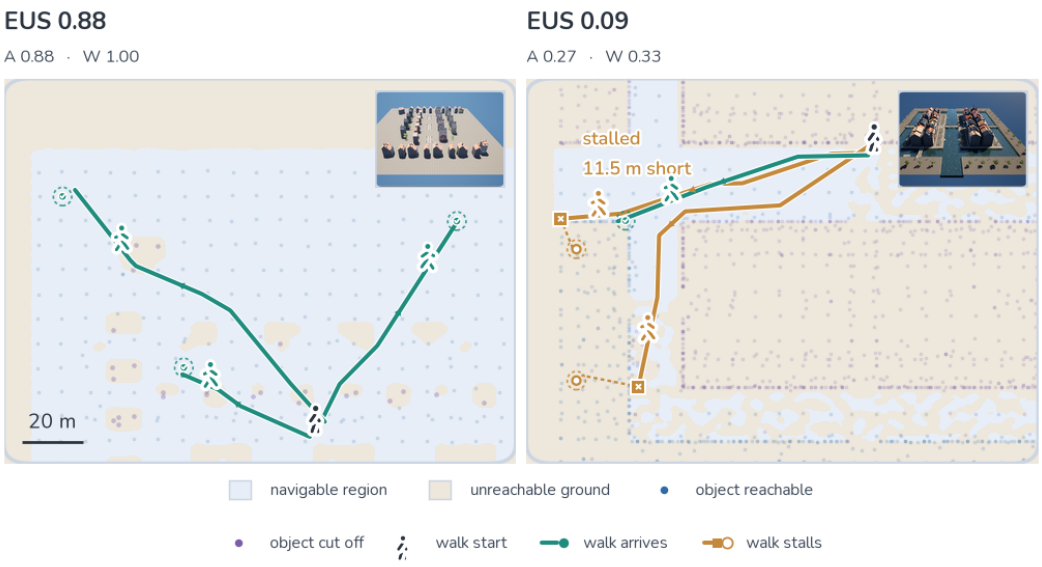


Figure 13: EUS scoring illustration on the same Nordic Harbour prompt. For approachability A, most floor-level objects in the left scene are reachable ($A = 0.88$), whereas most floor-level objects in the right scene are not ($A = 0.27$). For walk validation, three routes lead from a floor-level anchor to targets at the near, mid, and far route-distance quantiles; all three are completed in the left scene and one in the right. The overall EUS is therefore 0.88 for the left scene and 0.09 for the right.

Measuring W. The walk stage rebuilds the same mesh and settings. Its anchor is the floor-level member of the largest component nearest its centroid and within 150 cm of the ground median. Targets already within r_{arr} are excluded; the remaining targets are selected at the 0.20, 0.55, and 0.90 quantiles of route distance. The engine plans each route, and a capsule follows it at 200 cm/s. Two five-second intervals with less than 10 cm progress terminate a stalled walk. A walk succeeds only if the capsule ends within r_{arr} of its goal.

For a scene with half-extent below 1200 cm, the protocol instead selects six object-to-object routes of length at least $4r_{\text{arr}}$. If no non-trivial route exists, W is withheld. Only this oracle route follower contributes to EUS; more autonomous navigation agents are diagnostic and do not enter the score.

Withholding and aggregation. Failure to settle or certify the mesh, or to obtain a non-trivial route, produces a withheld result with a recorded reason. A scene blocked by an evaluation-environment content failure is excluded rather than attributed to the model. For the conservative reported aggregate, a withheld scene contributes zero while retaining its distinct withheld status. For task set t, define c_{ti} as the measured EUS (including certified zeros) or zero if withheld, and let N_t^{EUS} be the number of retained scene records. The aggregate is

$$\overline{\text{EUS}}_t = \frac{1}{N_t^{\text{EUS}}} \sum_{i=1}^{N_t^{\text{EUS}}} c_{ti}.$$

Table 15 reports measured-only means and coverage alongside $\overline{\text{EUS}}_t$. Standard errors summarize between-scene variation; fixed routes do not estimate variability across randomly sampled walks. Indoor results are grouped by base room because edits share its geometry. Figure 13 shows an example of how EUS is calculated for a scene.

C.9 Human-Alignment Protocol and Weight Fitting

The human-alignment study evaluates whether the composite Text-to-Scene score reproduces blinded human preferences, following evaluator-validation protocols for generative systems (Ghosh et al., 2023; Zheng et al., 2023; Ning et al., 2026). It contains 16 prompts and seven agent configurations per prompt, yielding 112 generated scenes. The human-review cohort comprises Qwen 3.8 27B

(thinking off), Qwen 3.8 27B (thinking on), Gemma 4 31B (thinking off), Gemma 4 31B (thinking on), Gemini 3.8 Flash (high), GPT-5.6 Sol (high), and GLM-5.3 Flash (max).

Each task presents three anonymous candidates and asks reviewers for a strict best-to-worst ranking, without ties or abstentions. A candidate is represented by four fixed overview images from its verifier run, with view switching and image enlargement. Reviewers see the scene description and the criterion being judged, while model identities, verifier scores, and other reviewers' answers remain hidden. Seven overlapping triplets form a balanced incomplete-block design over the seven candidates. Each candidate appears in three triplets, and each of the 21 candidate pairs appears in exactly one triplet. The tasks are distributed among seven graduate and undergraduate student annotators. For the primary Overall Task Success criterion, three different annotators rank each triplet, yielding $16 \times 7 \times 3 = 336$ triplet rankings and 1,008 pairwise preferences.

The interface separates Overall Task Success from six diagnostic criteria, namely Identity and Environment, Content and Quantity, Spatial Composition, Attributes and Materials, Holistic Overview Alignment, and Physical Plausibility. The four semantic criteria map to the Detailed Alignment families, Holistic Overview Alignment maps to Overview Alignment, and Physical Plausibility concerns visible unsupported placement and penetration. The primary reference is the independent Overall Task Success ranking, which is compared directly with S_{T2S}; it is not constructed by combining the diagnostic human rankings. Reviewers see the same four overview images used by Overview Alignment, but not other components' engine measurements or targeted views.

For each prompt and criterion, a Plackett-Luce model estimates a consensus ordering from the original triplet rankings (Turner et al., 2020). The implementation applies an L2 coefficient of 0.1 to the log-worths so that unanimous preferences yield finite estimates. Each triplet is also expanded into three pairwise preferences. Inter-annotator agreement is measured on the same candidate pair, while verifier agreement is reported against both individual preferences and the Plackett-Luce consensus. Kendall's τ_b is the primary rank association statistic, with Spearman's ρ, nominal Krippendorff's α_{K}, leave-one-reviewer-out agreement, and agreement between disjoint reviewer subsets reported as diagnostics. Agreement and rank correlation are computed within prompts and macro-averaged. Confidence intervals use whole-prompt bootstrap resampling with 5,000 draws for agreement and rank correlation and 2,000 draws for α_{K}, to account for shared candidates within prompts.

Aggregation weight fitting. We fit θ to maximize agreement with individual human Overall Task Success preferences. A deterministic, score-independent split of case identifiers (seed 20260915) assigns 12 prompts to development and four to retrospective holdout evaluation. Let $\mathcal{P}_{\mathrm{dev}}$ denote the development prompts, $\mathcal{Q}_p$ the 21 candidate pairs for prompt p, and $\mathcal{R}_{pij}$ the three reviewers of pair (i, j). Let $\mathbf{x}_{pi}$ denote the fixed evidence for candidate i on prompt p, as defined for $S_{\mathrm{T2S}}(\mathbf{x}; \theta)$ above. Write $y_{pijr} = +1$ if reviewer r prefers candidate i and -1 otherwise, and let $\Delta_{pij}(\theta) = S_{\mathrm{T2S}}(\mathbf{x}_{pi}; \theta) - S_{\mathrm{T2S}}(\mathbf{x}_{pj}; \theta)$. The primary objective is

$$\begin{aligned} &\underset{\theta \in \Theta}{\text{maximize}}\ \mathcal{A}_{\mathrm{dev}}(\theta), \\ &\mathcal{A}_{\mathrm{dev}}(\theta) = \frac{1}{|\mathcal{P}_{\mathrm{dev}}|} \sum_{p \in \mathcal{P}_{\mathrm{dev}}} \frac{1}{|\mathcal{Q}_p|} \sum_{(i,j) \in \mathcal{Q}_p} \frac{1}{|\mathcal{R}_{pij}|} \sum_{r \in \mathcal{R}_{pij}} \mathbf{1}[y_{pijr} \Delta_{pij}(\theta) > \varepsilon]\,, \end{aligned} \tag{11}$$

where $\varepsilon = 10^{-8}$ identifies numerical ties, which receive no credit against strict preferences. Prompts contribute equally. The feasible set Θ uses nonnegative normalized weight blocks, a fixed final Physical Safety share $\gamma = 0.20$, $\alpha \geq 0.20$, and positive total weight for each applicable-family set.

The search combines a 0.05 simplex grid, 8,192 seeded joint proposals, and up to five coordinate-refinement passes with 0.01 pairwise weight transfers. Equal-agreement candidates are ordered by pairwise logistic loss $\log(1 + \exp[-10y\Delta])$, averaged with the same prompt and pair weights, and then by squared distance to the comparison policy. This is a bounded numerical search rather than exhaustive continuous optimization. Scores are reconstructed from saved component scores, preserving rounding, applicability rules, and the fixed Overview Alignment modifier and cap; neither VLM judgments nor engine measurements are refitted. The secondary fit instead uses within-prompt Plackett-Luce consensus labels. All fitted parameters are frozen before holdout evaluation.

The adopted weights came from earlier human-guided exploration, which used all 16 prompts. The split therefore provides retrospective internal validation of the fitted alternatives. Component measurements remain fixed throughout.

Table 9: Aggregation weights for the adopted score and two preference fits. Each vector follows the displayed subscore order and sums to one; both fits fix the final Physics share at $\gamma = 0.20$. Fitting and the retrospective holdout protocol are defined in Appendix C.9.

Weights and order	Adopted	Individual fit	PL fit
$\mathbf{w}^D$: identity/environment, content/quantity, attributes/materials, spatial composition	$(.25, .40, .15, .20)$	$(.18, .30, .13, .39)$	$(.08, .20, .17, .55)$
$\mathbf{w}^O$: prompt alignment, layout, style/atmosphere, completeness/polish	$(.40, .25, .20, .15)$	$(0, 0, 1, 0)$	$(.10, 0, .90, 0)$
$\mathbf{w}^P$: supported fraction, collision-free fraction	$(.50, .50)$	$(0, 1)$	$(0, 1)$
(α, β, γ): Detailed, Overview, Physics	$(.20, .60, .20)$	$(.25, .55, .20)$	$(.30, .50, .20)$

Empirical comparisons appear in Appendix D.7.

C.10 Fixed-Evidence Repeatability Protocol

Scenes and repeated judgments. The diagnostic cohort contains nine agent configurations on six prompts, namely *Battleground*, *Classical Modular Gardens*, *Cyberpunk*, *Industrial Harbor*, *Medieval Buildings Volume 1*, and *Street New York*. The prompts were selected for scene-type coverage and reusable evidence, without optimizing model rankings or score differences. They form a purposive subset, not a random sample of the full benchmark. The cohort includes Gemini 3.7 Flash (high) but excludes six main-leaderboard configurations: GPT-6 Astra (max), Claude Fable 5.1 (max), Claude Opus 5 (max), Grok 4.6 (high), Muse Spark 1.3 (medium), and DeepSeek V4.1 Flash (high). Each of the 54 submitted scenes was scored three times. Scenes, requirement bundles, image order, earlier-stage decisions, and the recorded visual-request schedule were fixed; each repeat made fresh VLM requests. No scene generation, image capture, or Overview Alignment scoring was repeated.

Judge and aggregation. The study used the fixed judge configuration in Appendix C.1. The saved request schedule contained 8,764 logical requests per repeat, excluding transport retries. The 54 scenes and three repeats yield 162 scene-level scores. Requests were served by one to three replicas with matching model metadata and varying concurrency; the repeats include this serving variation. Detailed Alignment used the same family weights in every repeat, $(0.25, 0.40, 0.20, 0.15)$ for identity/environment, content/quantity, spatial composition, and attributes/materials, respectively.

The repeated-score results appear in Appendix D.7.

C.11 Evaluator Integrity

The packaging audit scans task packages for source maps, external actors, Level Instances, packed actors, prefabs that encode the target assembly, cached layouts, screenshots, and revealing metadata. Mutation tests cover proxy geometry, off-camera objects, disabled collision, excessive supports, camera changes, and protected-region edits to verify that these shortcuts are not rewarded.

D Additional Results and Analysis

This appendix supplements Section 3.4 with complete task profiles, repair diagnostics and navigation results. It also covers analysis cohorts, difficulty strata, agent behavior and evaluator validation.

D.1 Analysis Cohorts and Measurements

The public set contains 14 configurations on 20 T2S, 25 Indoor and 50 Outdoor cases (1,330 configuration–case pairs). The private set contains ten configurations on 10 T2S, 55 Indoor and 30 Outdoor cases (950 pairs), selected by removing public task IDs from the full suite within each domain. Together, public and private cover 30 T2S and 80 cases per repair domain for the ten common configurations. GPT-6 Astra (max), Grok 4.6 (high), Claude Opus 5 (max) and Claude Fable 5.1 (max) have only public coverage. Comparisons across task sets use the same ten configurations. Public and private tasks are disjoint but can share source scenes.

Table 10: Difficulty-stratified scores for ten agent configurations on all 190 public and private cases.

Agent configuration	T2S Easy ↑	T2S Medium ↑	T2S Hard ↑	I2S Easy ↑	I2S Medium ↑	I2S Hard ↑	S_{model} ↑
Gemini 3.8 Flash (high)	0.724	0.618	0.660	**0.627**	**0.610**	**0.553**	**0.632**
GPT-5.6 Sol (high)	**0.733**	**0.713**	**0.687**	0.451	0.427	0.372	0.564
Muse Spark 1.3 (medium)	0.662	0.605	0.602	0.485	0.330	0.311	0.500
GLM-5.3 Flash (max)	0.630	0.462	0.499	0.427	0.338	0.276	0.438
Qwen 3.8 27B (thinking off)	0.598	0.566	0.512	0.258	0.199	0.149	0.380
Qwen 3.8 27B (thinking on)	0.572	0.540	0.442	0.291	0.222	0.146	0.369
Gemma 4 31B (thinking on)	0.497	0.468	0.485	0.125	0.126	0.117	0.303
Inkling (high)	0.503	0.405	0.386	0.227	0.158	0.119	0.300
Gemma 4 31B (thinking off)	0.510	0.452	0.405	0.116	0.111	0.112	0.284
DeepSeek V4.1 Flash (high)	0.311	0.172	0.213	0.244	0.188	0.208	0.223

Score means include all scheduled cases under Appendix C.6; trajectory statistics use recorded runs. Scores use $[0, 1]$ unless a 0–100 scale is stated.

Trajectory measurements. Each configuration–case pair contributes one retained attempt under Section 2.3. Tool counts and token-cost estimates exclude discarded attempts, rather than summing across retries. Tool counts include requests with a metered start, including failed and unfinished requests, with duplicate request IDs counted once. Non-polling counts exclude asynchronous status checks. Mean calls use recorded trajectories, including zero-call runs.

A fixed API-basename dictionary classifies parseable Python submissions into eight operation families. Families may co-occur; percentages use parseable scripts and describe attempted operations. Execution feedback links synchronous responses and asynchronous logs to their originating submissions; repeated error reports count once per submission. A startup acknowledgement alone does not establish execution feedback. Missing code or feedback cannot rule out an operation or error.

Uncertainty. Confidence intervals use a source-cluster bootstrap. Each cluster groups cases derived from the same source environment and their results across configurations. Clusters are resampled with replacement within task domains, retaining all cases and configurations in each sampled cluster. Trajectory comparisons use 10,000 draws with seed 23917 (20/10 public/private T2S sources; each task set has ten Indoor rooms (from nine source levels, which form the Indoor bootstrap clusters) and ten Outdoor sources). Spatial and selective-repair diagnostics use 10,000 draws with seed 20260925. Repair resampling is stratified by domain, retaining observed domain weights. Reported intervals are pointwise percentile 95% intervals over retained sources; each configuration–case pair has one selected run. They describe source heterogeneity, not generation variability, and do not use multiplicity-adjusted significance thresholds.

D.2 Difficulty-Stratified Results

Difficulty-stratified scores. Table 10 reports scores for ten configurations on the full 190-case suite using the authored Easy/Medium/Hard mapping. Tier counts are 10/10/10 for T2S and 27/26/27 for each repair domain. Cases have equal weight within each tier; scores follow Appendix C.6.

Across configurations, I2S means decrease from Easy to Medium to Hard: $0.325/0.271/0.236$; the corresponding T2S means are $0.574/0.500/0.489$. These tiers are descriptive strata; individual configurations can reverse the ordering.

D.3 Task Requirement Analysis

Figure 14 gives the complete repair-category and construction-requirement profiles for all 14 public-set configurations, extending the frontier comparison in Section 3.4. Beyond the contrasts discussed there, Sol leads Restore (0.402) across all 14 configurations. Spatial Composition is the weakest construction family for every configuration.

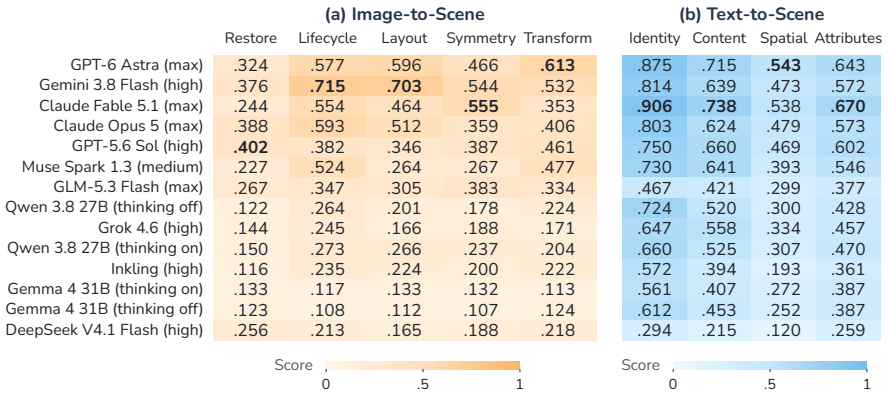


Figure 14: Task-category performance of 14 agent configurations on the public set: (a) Image-to-Scene scores by operation category and (b) Text-to-Scene Detailed Alignment.

Table 11: Paired differences between GPT-6 Astra (max) and Gemini 3.8 Flash (high) on public tasks, computed as Astra minus Gemini, with 95% confidence intervals. Scores use 0–100; calls exclude polling. Interval estimation follows Appendix D.1.

Task	Δ score [95% CI]	Δ calls [95% CI]
Text-to-Scene	6.69 [−3.23, 13.46]	−6.60 [−19.55, 6.75]
Indoor	6.89 [−4.62, 17.22]	8.44 [5.64, 11.65]
Outdoor	−13.38 [−25.78, −1.55]	1.88 [0.02, 3.54]

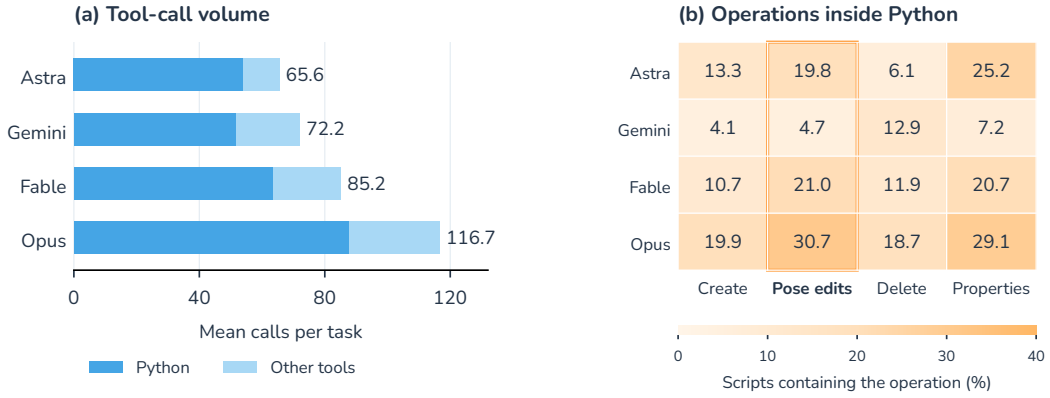


Figure 15: Construction behavior of Astra, Gemini, Fable and Opus, the four highest-ranked public configurations overall, on the same 20 T2S tasks. (a) Mean non-polling calls per task, split into Python and other tools. (b) Percentage of parseable Python scripts containing each operation family shown; families can co-occur. The outlined column shows pose-editing operations. All four configurations make zero named transform calls. Counts and definitions are in Appendices D.4 and D.1.

D.4 Agent Behavior Analysis

Interaction volume and scene quality. Table 11 compares Astra and Gemini on the same public tasks. Astra uses 6.60 fewer calls per construction case and averages 6.69 points higher. On Indoor, it uses 8.44 more calls and averages 6.89 points higher. On Outdoor, however, Astra uses 1.88 more calls but scores 13.38 points lower. Only the Outdoor score-difference interval excludes zero. The paired results show that more interaction need not yield better scenes.

Operation content behind tool calls. Figure 15 distinguishes named-tool calls from operations inside Python. On the 20 public construction tasks, Astra, Gemini, Fable and Opus submit 1,078, 1,037, 1,274 and 1,757 Python requests; 1,077, 488, 1,274 and 1,756 scripts, respectively, are parseable. Pose-editing APIs occur in 213, 23, 268 and 539 of those scripts, despite zero named transform calls for all four configurations. These counts describe submitted operations and use the available code as their denominator.

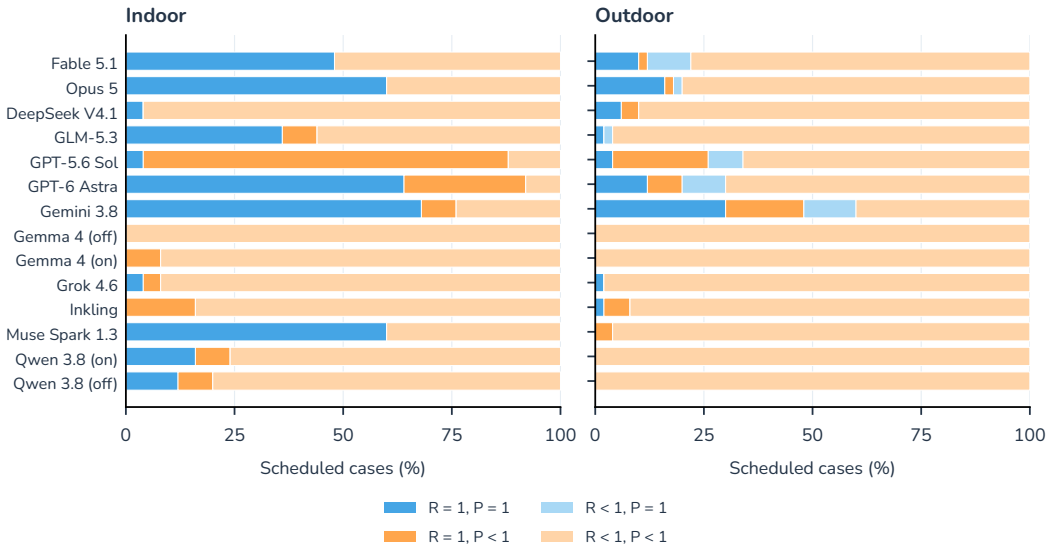


Figure 16: Public repair outcomes by configuration (25 Indoor and 50 Outdoor scheduled cases). All scheduled cases enter the denominators. R denotes Recall and P denotes Precision; R=1 means complete target recovery, and P=1 means at least one recovered target and no false positives; the renamed-material export artifact described below is excluded.

Opus averages 116.7 calls per construction task, versus 85.2 for Fable, while their mean T2S scores are 0.718 and 0.788. Gemini makes no named Indoor asset searches, yet Python catalog APIs occur in 4/25 public and 14/55 private cases. Code submissions therefore expose editing and retrieval attempts that named-tool counts alone omit.

D.5 Failure Analysis

Execution errors. Object-access and type-related errors recur in frontier configurations: AttributeError or TypeError occurs in 26/34 of Astra's error-bearing public I2S submissions and 11/12 of Gemini's. Across all public configurations and both settings, these families occur in 83.8% of error-bearing submissions. These are shares of submissions with observed errors, not error rates over all requests; missing feedback does not establish error-free execution (Appendix D.1).

Final repair outcomes. Astra leads Indoor target recovery, but requiring complete recovery with zero false positives changes the comparison. Among submissions that recover every target, false positives (FP) denote unmatched added actors or unintended modification, deletion or repurposing of actors that should be preserved. On the 25 public Indoor cases, Astra recovers every target in 23 cases and Gemini in 19; adding the requirement of zero false positives changes these counts to 16 and 17, respectively. On Outdoor, Gemini produces 24 complete recoveries and 15 without false positives, compared with Astra's ten and six. Fable produces fewer complete recoveries overall (18 across the 75 public repair cases), of which 17 have no false positives. Figure 16 separates complete recovery and false positives for all configurations; Table 12 reports their domain-specific P/R/F1.

These analyses exclude one export artifact. In the eight Indoor tasks built from the laboratory-loft scene, re-saving the level renames a lighting blueprint's runtime material instance, which the scene comparison records as a modified non-target actor although no agent edited it. We remove this single false positive from the 88 affected submissions (41 public); official Precision, Recall and F1 (Table 2; Table 12) retain it. Across all 14 public configurations, 76 of 212 submissions that recover all targets also contain false positives (35.8%; source-cluster 95% CI, 27.6–43.2%). The 76 submissions span 43 tasks and 16 source clusters. Among them, 63 include changes to non-target actors and 27 include unmatched added actors; Table 13A separates their overlapping FP types. FP counts have median three, interquartile range two–four, and maximum 3,341. Case prevalence complements actor counts so that large scenes do not dominate the interpretation.

Table 12: Actor Precision / Recall / F1 (0–100) on public and private Image-to-Scene tasks. Values are means over cases, so F1 need not equal the harmonic mean of the displayed Precision and Recall.

Agent configuration	Indoor Pub.	Indoor Priv.	Outdoor Pub.	Outdoor Priv.
GPT-6 Astra (max)	71.7 / 92.0 / 78.1	—	31.0 / 31.5 / 27.6	—
Gemini 3.8 Flash (high)	66.7 / 76.0 / 69.1	53.4 / 72.7 / 58.1	48.7 / 55.5 / 44.6	51.8 / 56.2 / 50.2
Claude Fable 5.1 (max)	46.0 / 48.0 / 46.7	—	31.4 / 26.1 / 26.4	—
Claude Opus 5 (max)	54.7 / 60.0 / 56.5	—	31.3 / 30.7 / 30.0	—
GPT-5.6 Sol (high)	31.4 / 88.0 / 44.2	30.8 / 78.2 / 41.6	26.0 / 41.8 / 25.8	23.5 / 52.8 / 30.1
Muse Spark 1.3 (medium)	56.7 / 60.0 / 57.9	35.2 / 40.0 / 36.7	7.2 / 10.7 / 8.0	10.7 / 14.4 / 11.6
GLM-5.3 Flash (max)	36.0 / 44.0 / 37.9	32.2 / 39.1 / 33.7	9.3 / 11.4 / 9.3	5.9 / 11.3 / 7.2
Qwen 3.8 27B (thinking off)	13.6 / 20.0 / 14.7	8.4 / 11.8 / 9.0	3.8 / 9.7 / 5.1	7.3 / 8.3 / 7.6
Grok 4.6 (high)	5.0 / 8.0 / 5.6	—	3.3 / 5.0 / 3.8	—
Qwen 3.8 27B (thinking on)	17.5 / 24.0 / 18.5	11.0 / 19.1 / 12.7	6.4 / 14.9 / 8.4	4.0 / 8.0 / 5.1
Inkling (high)	4.0 / 16.0 / 6.4	1.2 / 4.5 / 1.8	10.0 / 16.3 / 11.1	5.8 / 10.0 / 6.9
Gemma 4 31B (thinking on)	0.4 / 8.0 / 0.7	0.2 / 1.8 / 0.4	0.1 / 6.0 / 0.1	0.1 / 3.3 / 0.2
Gemma 4 31B (thinking off)	0.0 / 0.0 / 0.0	0.0 / 1.8 / 0.0	0.1 / 6.0 / 0.1	0.0 / 1.1 / 0.0
DeepSeek V4.1 Flash (high)	2.0 / 4.0 / 2.7	7.1 / 15.5 / 9.0	7.7 / 12.0 / 8.6	2.8 / 7.8 / 4.0

Repair tolerances and asset/material changes. Table 13B first halves or doubles the default error tolerances used to accept a repaired target. Target definitions, actor correspondence and non-target comparison tolerances remain fixed. The share of complete recoveries with FP stays near 36%, indicating that the recovery–preservation distinction persists across these target acceptance thresholds. No non-target FP in a complete recovery is solely a within-tolerance pose change (Appendix C.5).

The final row excludes non-target modifications confined to asset/material properties. These properties record asset and material references; dynamic material-instance identities can change between saved packages without establishing a visible change. The exclusion also removes actual asset or material substitutions, so it is a diagnostic rather than a correction to official scores. After exclusion, 75 public complete recoveries retain FP across 13 configurations, including 62 with remaining non-target changes. Only one of the 76 affected submissions contains only excluded modifications. For Astra, all seven affected Indoor submissions and all four affected Outdoor submissions retain FP. Thus, the distinction extends beyond asset/material differences.

Private-task check. Table 13C compares public and private tasks for the same ten configurations under the default target tolerances. Each entry reports submissions with FP out of those recovering all targets. Separately excluding non-target modifications confined to asset/material properties leaves the private count unchanged at 92 of 191.

Observed tool use. Table 14 reports actual tool use by task setting and public/private set. Counting conventions follow Appendix D.1.

Estimated cost. Figure 1(b) prices recorded public-set tokens from retained attempts at list rates, excluding discarded attempts rather than reporting total billed spend. Cache reads, cache writes, fresh input and output are priced separately. Proprietary models use developer rates; Muse uses Meta's contributor tier. Open-weight models use OpenRouter rates, which are counterfactual for self-hosted Gemma and Qwen runs. Rates were read between 7 and 23 September 2026, using standard tiers without batch/volume discounts and base rates where prices vary by context length or time.

Track means cover cases with usage records and are weighted 20/25/50 over the 95 public tasks. Coverage ranges from 95 cases for seven configurations to 62 for Astra, whose Outdoor usage covers 21/50 cases. The estimate assumes the observed cache reuse and excludes non-token charges. Token counts alone are not cost ratios: clients differ in accounting, prices and scene quality.

Table 13: **Unintended changes despite complete target recovery.** Complete recovery means all required repairs succeed (Recall=1). Here, false positives (FP) are added actors without a target match or unintended modifications, deletions or repurposing of actors that should be preserved. Counts refer to submissions (one configuration on one task), not actors. A: submissions containing each change type; categories overlap and n is the number of complete recoveries. B: all 14 public configurations; threshold triplets give distance (cm), rotation (degrees) and relative scale/bounds-extent (%) tolerances. The last row excludes non-target changes confined to asset/material properties. C: submissions with FP / complete recoveries (percentage), for ten matched public/private configurations. All counts exclude the renamed-material export artifact (Appendix D.5).

A. Types of unintended changes

Change type	Public, 14 configs. $n = 212$	Public, 10 configs. $n = 134$	Private, 10 configs. $n = 191$
Added actors without a target match	27	23	40
Modified non-target actors	62	52	73
Deleted non-target actors	3	3	3
Non-target actors repurposed for a repair	2	2	0
Unrecovered target state	0	0	0

B. Effect of repair tolerances and asset/material exclusions

Analysis setting	All targets recovered	Of these, with FP	Share with FP
Strict (2.5 cm, 2.5°, 2.5%)	209	76	36.4%
Default (5 cm, 5°, 5%)	212	76	35.8%
Relaxed (10 cm, 10°, 10%)	216	77	35.6%
Default; exclude changes confined to asset/material properties	212	75	35.4%

C. Public/private comparison for the same ten configurations

Domain	Public	Private
Indoor	35/85 (41.2%)	74/154 (48.1%)
Outdoor	27/49 (55.1%)	18/37 (48.6%)

Table 14: Mean tool calls per recorded run, excluding asynchronous status polling. Public and private tasks are reported separately; a dash marks configurations without private-task evaluation.

Agent configuration	Text-to-Scene		Indoor		Outdoor	
	Public	Private	Public	Private	Public	Private
GPT-6 Astra (max)	65.6	—	19.2	—	14.2	—
Gemini 3.8 Flash (high)	72.2	73.0	10.8	10.7	12.4	11.9
Claude Fable 5.1 (max)	85.2	—	12.4	—	14.2	—
Claude Opus 5 (max)	116.7	—	18.9	—	20.5	—
GPT-5.6 Sol (high)	55.4	56.4	23.9	28.2	22.3	23.3
Muse Spark 1.3 (medium)	53.5	74.4	56.1	55.0	17.7	21.6
GLM-5.3 Flash (max)	114.2	115.2	23.9	22.5	13.3	15.2
Qwen 3.8 27B (thinking off)	75.8	70.4	62.0	58.9	54.2	56.1
Grok 4.6 (high)	109.8	—	19.4	—	27.7	—
Qwen 3.8 27B (thinking on)	40.7	57.4	56.6	61.5	56.6	53.8
Inkling (high)	27.8	28.0	11.4	14.1	22.4	21.5
Gemma 4 31B (thinking on)	27.1	33.3	88.2	86.5	23.0	23.1
Gemma 4 31B (thinking off)	10.1	18.5	32.3	32.4	25.5	23.2
DeepSeek V4.1 Flash (high)	78.9	72.8	15.8	15.8	24.9	23.0

D.6 Downstream Usability

The navigation study uses cohorts distinct from the main score and Physical Safety comparisons. Table 15 reports construction results for nine configurations (30 scheduled prompts; 29 retained GLM records), Outdoor repair on the same 71 tasks for eight agent configurations, and Indoor base-room coverage. These are auxiliary study cohorts, not additional public/private splits. Methods and withholding rules are defined in Appendix C.8.

Outdoor changes relative to the input. The matched task inputs were probed with the same navigation settings and conservative withholding rule as the candidates. Figure 17 centres the comparison on the paired candidate-minus-input change, separating inherited navigability from the

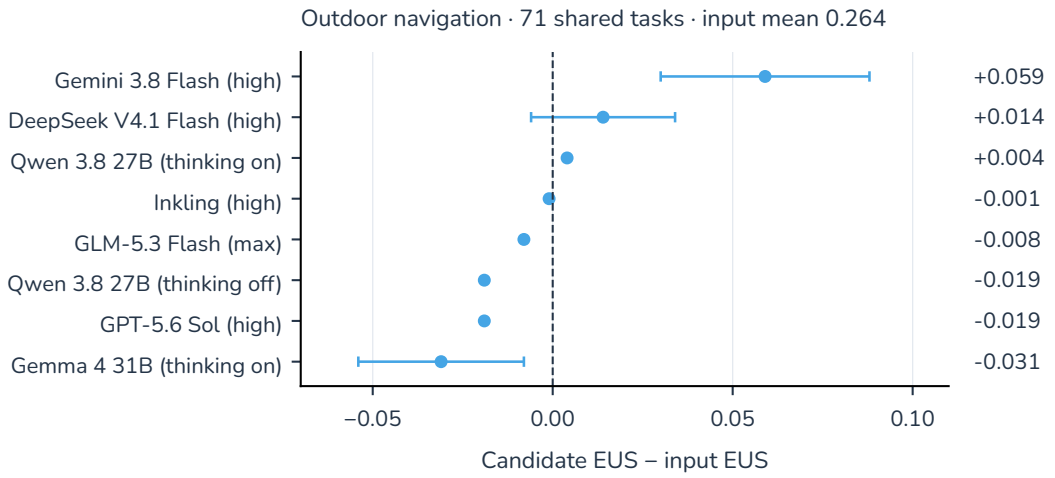


Figure 17: Outdoor navigation changes relative to the matched input. Dots show candidate-minus-input means on matched navigation tasks, using the conservative withholding rule. Whiskers show paired standard errors for Gemini, DeepSeek and Gemma; the other five marks show means only. Differences use the reported three-decimal means. Candidate means, measured-only means, components and coverage appear in Table 15.

change after editing. Seven mean changes lie between -0.031 and $+0.014$; Gemini's is $+0.059$ with paired SE 0.029. Gemini and Sol have similar Outdoor walkability ($W = 0.793$ and 0.799), but Gemini has higher approachability ($A = 0.411$ versus 0.342) and candidate EUS (0.323 versus 0.245). Gemini's higher EUS thus accompanies broader target approachability, with comparable route-completion rates (Table 15).

Construction results. Gemini and Sol have nearly identical mean construction EUS (0.479 and 0.477), with a paired difference of 0.002 ± 0.061 on shared prompts. Their Outdoor repair means diverge: Gemini gains 0.059 over the shared input mean, while Sol changes by -0.019. Similar usability after construction therefore does not imply similar navigation gains after editing; the construction and repair cohorts are assessed separately.

Alignment with benchmark scores. Eight configurations are shared by the construction EUS summary and the benchmark-score summary available at the time of the EUS study. Both studies used 30 scheduled prompts; GLM has 29 retained EUS records. The association concerns configuration means, not paired individual scenes. The alignment uses the eight configurations matched when the EUS summary was produced; DeepSeek has no construction EUS result. Adding Gemma 4 31B (thinking off), whose reasoning mode was confirmed afterwards, gives $r = 0.752$ and $\rho = 0.667$. From the reported three-decimal configuration means, Pearson correlation is $r = 0.781$ and Spearman rank correlation is $\rho = 0.714$. Using measured-only EUS gives $r = 0.731$ and $\rho = 0.643$. Construction and repair are analyzed separately because their navigation cohorts differ.

Indoor repair. The indoor cohort contains eight edits of each of ten base rooms; rooms, not edits, are the source-level units. The number of rooms with a numeric result ranges from six to nine across systems, and the corresponding room means range from 0.158 to 0.279. An unmeasured room has no approachable target beyond the grading radius for any of its edits. Because room coverage differs across systems, Panel C uses alphabetical agent order.

Navigation measurement coverage. Text-to-Scene has two to four withheld scenes per configuration, due to mesh settling, route availability, or certification failures. Outdoor repair has five to thirteen withheld scenes per configuration, including 58 configuration–task measurements with mesh-settling failures. These results follow the withholding and exclusion rules in Appendix C.8.

Table 15: Embodied Utility Score (EUS); uncertainties are standard errors. Conservative means assign zero to withheld records; measured means omit them. Coverage is the measured share; Scored counts measured records; Status lists zero / withheld counts. Panel A has 29 GLM-5.3 Flash (max) records and 30 per other agent; Panel B has 71 per agent. Panel C summarizes indoor rooms.

A. Text-to-Scene Construction (30 scheduled prompts)

Agent configuration	EUS ↑	Measured ↑	Coverage	A ↑	W ↑	Scored	Status
Gemini 3.8 Flash (high)	**0.479 ± 0.050**	**0.513**	93%	**0.533**	0.905	28	0 / 2
GPT-5.6 Sol (high)	0.477 ± 0.068	0.511	93%	0.519	**0.926**	27	1 / 2
GLM-5.3 Flash (max)	0.387 ± 0.066	0.432	90%	0.478	0.778	24	2 / 3
Qwen 3.8 27B (thinking off)	0.373 ± 0.061	0.399	93%	0.420	0.810	28	0 / 2
Gemma 4 31B (thinking on)	0.369 ± 0.069	0.410	90%	0.419	0.792	24	3 / 3
Gemma 4 31B (thinking off)	0.347 ± 0.064	0.372	93%	0.389	0.736	24	4 / 2
Muse Spark 1.3 (medium)	0.344 ± 0.051	0.368	93%	0.380	0.852	27	1 / 2
Inkling (high)	0.268 ± 0.060	0.309	87%	0.328	0.907	18	8 / 4
Qwen 3.8 27B (thinking on)	0.247 ± 0.058	0.274	90%	0.285	0.710	23	4 / 3

B. Image-to-Scene Editing, Outdoor

Agent configuration	EUS ↑	Measured ↑	Coverage	A ↑	W ↑	Scored	Status
Gemini 3.8 Flash (high)	**0.323 ± 0.044**	**0.396**	82%	**0.411**	0.793	58	0 / 13
DeepSeek V4.1 Flash (high)	0.278 ± 0.037	0.319	87%	0.340	0.788	62	0 / 9
Qwen 3.8 27B (thinking on)	0.268 ± 0.037	0.312	86%	0.344	0.775	60	1 / 10
Inkling (high)	0.263 ± 0.036	0.307	86%	0.323	0.839	60	1 / 10
GLM-5.3 Flash (max)	0.256 ± 0.034	0.313	82%	0.330	**0.854**	58	0 / 13
Qwen 3.8 27B (thinking off)	0.245 ± 0.036	0.286	86%	0.323	0.783	60	1 / 10
GPT-5.6 Sol (high)	0.245 ± 0.035	0.295	83%	0.342	0.799	59	0 / 12
Gemma 4 31B (thinking on)	0.233 ± 0.035	0.250	93%	0.273	0.793	58	8 / 5

C. Image-to-Scene Editing, Indoor (base rooms, not ranked)

Agent configuration	Rooms measured	Mean EUS	Median EUS
DeepSeek V4.1 Flash (high)	6/10	0.194	0.192
Gemini 3.8 Flash (high)	6/10	0.207	0.189
Gemma 4 31B (thinking on)	8/10	0.220	0.248
Gemma 4 31B (thinking off)	9/10	0.279	0.242
GLM-5.3 Flash (max)	6/10	0.169	0.157
GPT-5.6 Sol (high)	7/10	0.227	0.289
Inkling (high)	6/10	0.194	0.187
Qwen 3.8 27B (thinking off)	8/10	0.158	0.139
Qwen 3.8 27B (thinking on)	8/10	0.184	0.203

D.7 Evaluator Validation

Table 16 summarizes human agreement, judge repeatability and repair-weight sensitivity. Human-review and repeatability protocols appear in Appendices C.9 and C.10.

Agreement with human judgments. The preference fits improve development agreement but do not improve the retrospective holdout results. Across all 16 prompts, pairwise inter-annotator agreement is 80.75% and nominal Krippendorff's $\alpha_{\mathrm{K}} = 0.613$. On the four holdout prompts, individual-preference agreement is 79.37% for the adopted policy and 80.95% between annotators. Separate reporting distinguishes consensus from individual-preference labels.

Detailed Alignment repeatability. All 162 evaluations completed. Response validation produced 111 fallbacks recorded as UNKNOWN and handled by the aggregation policy. Figure 18 shows that the four highest configurations retain their ordering, with reversals confined to ranks 5–7. Pairwise rank agreements are 0.889, 0.944 and 0.944. Inkling has the largest variation in repeat means.

Repair-weight sensitivity. Table 17 varies the Repair F1 weight in the existing I2S aggregate. Gemini remains the leader at tested weights $\lambda_{\mathrm{rep}} \in \{0.25, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0\}$; physics-only scoring selects GLM. This checks sensitivity on fixed public cases, not whether 0.8 is optimal.

Table 16: Evaluator validation and its scope. Panel A separates three validation questions; Panel B reports agreement fractions for the retrospective 12/4-prompt split. Fitted weights are in Table 9.

A. Validation questions

Question	Unit	Result	Scope
Human agreement	Four holdout prompts, 84 consensus pairs	73/84 agreement; 95% CI 76.19–94.05%; $\tau_b = 0.738$	Internal validation: earlier exploration used all 16 prompts.
Judge repeatability	Nine configurations, six cases, three repeats	Rank $\tau_b = 0.889$–0.944; model-mean SD 0.0010–0.0132	Fixed evidence and routing; not complete adaptive-verifier variability.
Repair-weight sensitivity	Fourteen configurations, 75 public repairs	For tested $\lambda_{\text{rep}} \geq 0.5$, rank $\rho \geq 0.952$; at most three places shifted	Fixed scenes, thresholds and cases; no human labels fitted.

B. Policy agreement

Policy	Dev. individual	Holdout individual	Holdout PL
Adopted	0.7672	**0.7937**	**0.8690**
Individual fit	**0.8082**	0.7619	0.8214
PL fit	**0.8082**	0.7460	0.7976

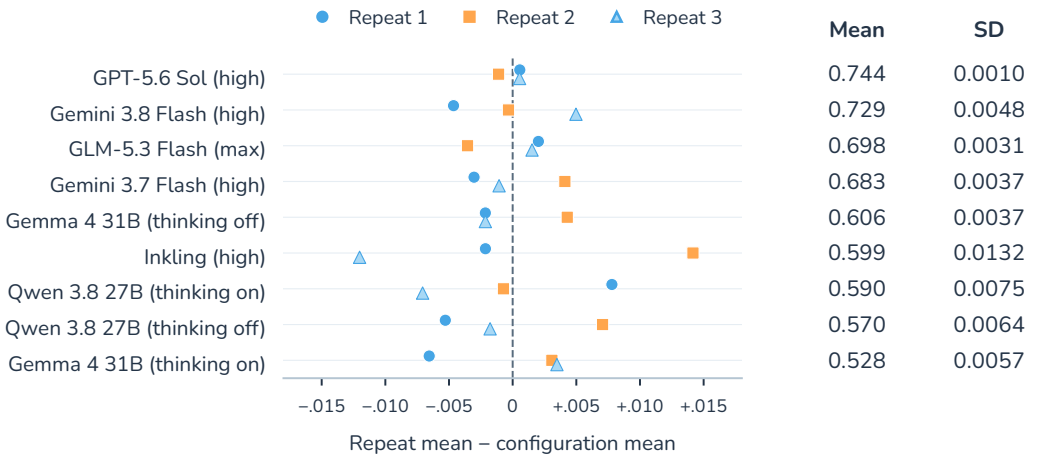


Figure 18: Detailed Alignment repeatability on six selected T2S cases. Marks show each repeat's six-case mean minus the configuration's three-repeat mean; colors and shapes distinguish repeats. The right columns report the absolute mean and sample standard deviation across the three repeats. Evidence and routing are fixed.

Table 17: I2S ranking sensitivity to the Repair F1 weight for 14 configurations on 75 public repairs. Correlations and maximum rank shifts are relative to $\lambda_{\text{rep}} = 0.8$ in $\lambda_{\text{rep}} F_1 + (1 - \lambda_{\text{rep}}) S_{\text{physics}}$. Scenes, thresholds and case selection remain fixed.

λ_{rep}	Spearman ρ	Kendall τ_b	Max. rank shift
0.00	0.705	0.516	6
0.25	0.916	0.802	3
0.50	0.952	0.868	3
0.60	0.978	0.912	2
0.70	0.991	0.956	1
0.80	1.000	1.000	0
0.90	0.982	0.934	2
1.00	0.982	0.934	2

E Related Work

Reading the benchmark comparison. Table 1 distinguishes task evidence, artifacts and evaluation coverage. Input references specify the desired outcome and exclude initial-scene observations, code and execution feedback. Open-ended tasks allow multiple valid realizations; ground-truth-based tasks provide a target scene or state, including reference sequences. Executable assertions alone

do not constitute a target scene. Checkmarks indicate reported coverage, not equivalent scope or accuracy; blank cells indicate no corresponding component reported. Physical validity may concern static geometry or dynamics, whereas preservation requires explicit checks of non-target content.

SceneActBench compares outputs with hidden geometry, pose and motion targets without a separate physical-validity score (Zhao et al., 2026). 4DBuildBench audits physics mechanisms and spatial relations alongside visual prompt fulfillment (Liu et al., 2026a). CutsceneBench combines cinematic judgments with reference-sequence checks; its agent-side collision-avoidance tools are not a benchmark physical metric (He et al., 2026). Code4Scene measures static support and penetration.

Environment generation and executable representations. Captured spaces preserve acquired geometry, while procedural systems generate controlled variation through rules (Xia et al., 2018; Deitke et al., 2022; Raistrick et al., 2023; 2024). Language-conditioned systems add asset retrieval and layout synthesis; recent coding-agent systems construct editable environments through planners, engine tools and executable scene programs (Yang et al., 2024b; Pfaff et al., 2026; Xia et al., 2026; Kang et al., 2026a; Wang et al., 2026; Guo et al., 2026). Executable scenes also support embodied-agent training and evaluation, e.g., urban navigation and multi-robot collaboration in SimWorld-Robotics (Zhuang et al., 2025) and long-horizon delivery planning in DeliveryGym (Kang et al., 2026b). Code4Scene evaluates complete agent configurations using a shared Unreal interface, task assets and accounting rules. Representation and harness choices can themselves change performance (Liu et al., 2026b; Yao et al., 2026); the benchmark does not isolate spatial reasoning from competence with the execution interface.

Construction, editing and preservation. BlenderGym evaluates edits from start scenes toward goal renders (Gu et al., 2025). AuthorBench combines construction and language-guided editing with requirement, preservation and physical checks on scene JSON (Xu et al., 2026). VWE-Bench uses language to specify changes to placed-asset worlds, with current-scene images as context; its verified queries check required edits and reject unauthorized changes (Ning et al., 2026). Code4Scene's Image-to-Scene references instead depict the desired human-assembled scene. Agents infer the repair from visual differences, then submit a persistent Unreal scene that is reopened for integrity, physical and task checks. These differences concern evidence, interfaces and artifacts; preservation and physical validity are also evaluated in prior work.

Evaluation beyond rendered appearance. Requirement-based and multidimensional evaluation separate object, attribute, relation and visual-quality judgments (Hu et al., 2023; Huang et al., 2024; He et al., 2024). Symbolic scene evaluation and engine measurements provide complementary evidence to visual judgments (Sengupta et al., 2026; Tam et al., 2026). Code4Scene combines structured scene-state checks with bounded visual judgments and separately measures navigation with a fixed route follower (Appendix C.8). This complements navigation proxies in MUSE (Xu et al., 2026) without measuring general embodied interaction.

F Limitations and Broader Impact

Scope and transfer. Code4Scene evaluates static scene construction and editing in Unreal Engine. It does not evaluate authored dynamics, e.g., cloth, fluids, or articulated mechanisms (Liu et al., 2026a). The Text-to-Scene support score is a bounding-box contact proxy with a world-ground assumption, not a test of mesh contact, load-bearing stability, or whether a chain of supports reaches the ground. The auxiliary fixed-walker analysis (EUS) asks whether a person-sized body can reach and traverse the scene, but does not test object interaction, manipulation, learned-policy task completion, or multi-agent behavior (Chang et al., 2025; Pfaff et al., 2026). Performance can also vary substantially with the scene representation and action interface (Liu et al., 2026b; Yao et al., 2026). The execution logs expose issued calls and returned failures, but they do not fully disentangle competence with the Unreal interface from scene-construction quality. Comparisons characterize complete agent configurations, including their clients (Appendix A.1).

Statistical and evaluator validity. Each score cohort retains one scene per agent configuration and case, without estimating variability across repeated agent trajectories. The paired source-cluster bootstrap intervals describe variation across sampled cases with those scenes fixed (Appendix D.1).

Leaderboard stability under repeated scene generation remains unmeasured. Agreement with human preferences was assessed for Text-to-Scene on 16 prompts and seven agent configurations, with only four prompts in the retrospective holdout. Reviewers saw four overview images, whereas the automatic score also uses engine measurements and targeted views. Image-to-Scene has not been validated against human judgments, and the fixed-evidence VLM study does not measure repeatability of the complete adaptive evaluator. The actor-repair tolerances and $0.8/0.2$ repair/physics weighting are design choices, not human-calibrated optima. Table 17 retrospectively tests ranking sensitivity to weights on fixed scenes, not to repair thresholds or generation randomness.

Dataset and annotation coverage. Public-set agreement with full-suite scores is weaker for agent configurations outside the selection pool; full-suite error remains unmeasured for the four agent configurations evaluated only on the public set (Appendix A.3). The public set is not proportional across categories, content packages or difficulty tiers. The cases derive from human-assembled Unreal scenes, so their coverage reflects the available content packages rather than a sample from a defined population. The authored difficulty labels are not empirically calibrated and do not guarantee decreasing scores for every model or task setting (Table 10). Indoor cases do not record the same operation-level corruption contracts as outdoor cases. Their author-provided operation groups support descriptive category analysis but not operation-chain conclusions (Section 3.4). Requirement graphs are produced automatically or with agent assistance rather than independently hand-annotated; an error in a graph can therefore propagate into the score.

Broader impact. Reliable construction of executable 3D scenes could increase the availability of environments for embodied and physical AI. It could also reduce demand for some manual authoring work or shift creative labor toward supervision and revision. Failures such as hidden collisions, missing dependencies, or unintended edits may propagate into downstream experiments if generated scenes are accepted without independent checks; a high Code4Scene score is not a general safety certification. Because the cases use existing content packages, access and redistribution remain subject to the packages' original licenses.